Development of the Control System of a Low-Cost Robot for Upper Limb Rehabilitation of Stroke Patients

Adam Giles Metcalf

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

The University of Leeds
School of Mechanical Engineering

April 2023

Declaration

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others. In the paper listed below contributing to this thesis the primary author completed all experimental studies, evaluation of data and preparation of publication. All authors contributed to proof reading of the articles prior to publication.

- Metcalf, Adam G., Justin F. Gallagher, Andrew E. Jackson, and Martin C. Levesley. 2021. ”Multi-Domain Dynamic Modelling of a Low-Cost Upper Limb Rehabilitation Robot” *Robotics* 10, no. 4: 134. https://doi.org/10.3390/robotics10040134

Adam Metcalf.
2022.

Acknowledgments

I want to thank everyone who has supported me through this difficult process. To my Dad, who gave me the support and courage to leave the army and go to university. To my lovely fiancée Becky, who endures with such grace my idiosyncrasies and office space lebensraum antics and is a thoroughly good egg. Finally to my supervisors who bent over backwards to help me access the necessary space and equipment even when the whole world was locked in at home.

Development of the Control System of a Low-Cost Robot for Upper Limb Rehabilitation of Stroke Patients

Adam Giles Metcalf

Abstract

Stroke is the leading cause of disability in the UK and is expected to increase in prevalence due to an aging population. Stroke patient outcome is improved by early and intense physiotherapy after Stroke, but NHS services are increasingly under strain, particularly in the wake of the COVID 19 pandemic. Over the last thirty years there has been a great deal of development in the use of robotic devices to provide rehabilitation to Stroke patients. The system architecture of a rehabilitation robot is well defined, with a High-Level Controller generating rehabilitation tasks, a trajectory generation procedure to encode the tasks and a Low-Level Controller to implement the tasks. All commercially available rehabilitation robots are high cost, however, and a recent study has found that the cost-benefit ratio is too poor to be viable [1]. Further to this, in the wake of the pandemic there has been increased interest in home-based devices. There has been research into low cost devices, but this area of research has not been sufficiently explored.

MyPAM is low-cost upper-limb rehabilitation robot designed for home use. There have been two previous iterations, and the first version of MyPAM was proven to improve the outcome of Stroke patients in trials. The current system has been rearchitected, with much focus on the Low Level controller which has been implemented on lower cost hardware than previous iterations. Responsibility for trajectory generation has been moved from the High-Level controller to the Low level controller, and a novel method for affecting the trajectory with Attractors and Repulsors has been designed and validated, which has important implications on patient motivation. A multidomain dynamic model is presented, which is necessary for creating a baseline against which to compare patient performance. A novel integrated end-effector/2-axis force sensor and a novel end-effector/Tristate grip sensor are presented, both based on the MagOne sensing methodology. An Admittance controller with instability protection is presented, and the system integration is discussed.

Contents

List of Figures

List of Tables

List of Abbreviations

API	Application Programming Interface.
AR	Altered Reality.
CTE	Cognitive Therapeutic Exercise.
DAQ	Data Acquisition.
DoF	Degrees of Freedom.
FPGA	Field Programmable Gate Array.
HLC	High-Level Controller.
I/O	Input/Output.
I^{2}C	Inter-Integrated Circuit (a data communications protocol).
JSON	JavaScript Object Notation.
LLC	Low-Level Controller.
LSB	Least Significant Bit.
LUT	Look Up Table.
MIMO	Multiple Input Multiple Output.
MLC	Mid Level Controller.
MSB	Most Significant Bit.
MSE	Mean Squared Error.
NN	Neural Network.
OS	Operating System.
PCB	Printed Circuit Board.
PID	Proportional, Integral and Derivative control.
PNF	Proprioceptive Neuromuscular Facilitation.
PWM	Pulse Width Modulation.
RATULS	Robot Assisted Training for the Upper Limb after Stroke.
RTOS	Real Time Operating System.
SCARA	Selective Compliance Assembly Robot Arm.
sEMG	Surface Electromyography.
SISO	Single Input Single Output.
SPI	Serial Peripheral Interface (a data communications protocol).
SVE	Shared Variable Engine.
TCP	Transmission Control Protocol.
TDD	Test Driven Development.
UML	Unified Modelling Language.
UDP	User Datagram Protocol.
VR	Virtual reality.

Chapter 1

Introduction

Stroke is a condition which leads to cell damage or cell death in the brain due to a lack of Oxygen and is the leading cause of disability in the UK [2]. The symptoms of Stroke are wide ranging, and a large proportion of Stroke patients experience some form of upper-limb impairment. Physiotherapy is a necessary element for the rehabilitation of Stroke patients, and recovery is largely a neurological process. There exists little agreement on the effectiveness of different rehabilitation strategies and the mechanisms of recovery are not well understood. It is widely agreed, however, that physical rehabilitation is more effective when performed intensively and early after Stroke [3, 4], though there is little agreement on the ideal rehabilitation pathway which is likely to be required to be defined individually for each patient according to their injury and needs.

There are current systemic limiting factors in Stroke rehabilitation. Stroke rehabilitation is resource intensive, particularly in the early phases. Rehabilitation providers have reported that resourcing issues including inadequate staffing and insufficient time have been contributing barriers to providing the necessary amount of rehabilitation [5]. An additional limiting factor is that it has been found that the active participation and high levels of motivation required by patients for good rehabilitation outcomes commonly drop off when they are sent home with further exercises [6], which are often repetitive and mundane.

There has been great interest in the use of robots to assist in the rehabilitation of Stroke patients. In general these rehabilitation robots are active-powered devices which assist the user to perform tasks such as reaching. Commercial products already exist, with the InMOTION Arm as a notable example. The InMOTION Arm has undergone extensive medical trials, most recently in 2019 in the Robot Assisted Training for the Upper Limb after Stroke (RATULS) study [1]. As another example of commercial rehabilitation robotics, Tyromotion produce a full suite of rehabilitation technology designed to equip comprehensive rehabilitation facilities [7]. Commercial solutions are high cost, however. The RATULS study found that the use of commercial rehabilitation robotic devices in an NHS setting was not cost effective compared with traditional face to face physiotherapy. Coupled with this is the ongoing effect of the COVID-19 pandemic. The need for social distancing has affected rehabilitation services [8, 9] and has highlighted a need for low-cost robotics which may be used unassisted in the home [9, 10].

There have been two previous iterations of MyPAM, this thesis documents the third iteration with focus on the Low Level Control system. Each successive iteration has solved problems of the previous system. This aim of this version of MyPAM is a modular system hierarchy, allowing any future improvements to be more easily integrated because to date each new version has required a large amount of re-engineering work. The overarching goal of the MyPAM project is a fleet of robots in patients homes, all linked back to a central hub where a physiotherapist will use an interface to monitor patient progress and set training parameters. The hierarchy of MyPAM has been reorganised into a series of High level controllers, which provide the the interfaces for a library of engaging games, a Mid level controller, which launches the games and provides and interface for the physio therapist, and a Low Level controller which performs trajectory generation and robot control.

1.1 Project Aim

The aim of the work is to design, implement and evaluate Low Level Control software for the latest iteration of the MyPAM, which is a low-cost robot designed for autonomous upper limb rehabilitation of Stroke patients.

1.1.1 Research Objectives

The objectives required to meet to complete this aim are:

Objective 1 To review current research on upper limb rehabilitation robotic technology and identify opportunities for the development of MyPAM.

Objective 2 To develop and validate an appropriate trajectory generation strategy for MyPAM to optimise the rehabilitation outcomes of patients.

Objective 3 To develop and validate a detailed dynamic model for MyPAM.

Objective 4 To develop, validate and deploy a sensor embedded end-effector/handle for MyPAM which may be used as a control input and a user input for rehabilitation games.

Objective 5 To develop and validate an appropriate low-level control scheme for MyPAM.

Objective 6 To finalise, document, test and deploy a completed low-level control infrastructure for MyPAM.

1.2 Introduction to MyPAM

The current (3^{rd}) iteration of MyPAM is a 2 Degree of Freedom (DoF) planar robot for upper limb rehabilitation after neurological trauma such as Stroke. The robot is a powered arm which assists the user to reach targets, with visual feedback is provided by a screen. Each of the two joints is powered by a Maxon brushed DC motor with an ENX encoder to provide position feedback. High-level control is achieved through gamification, where a series of computer games programmed using the Unity development environment generate the final target of a reaching task. Trajectory generation and low-level control are implemented on a low cost National Instruments myRIO and programmed with LabVIEW. The MyPAM is shown by Figure 1.1.

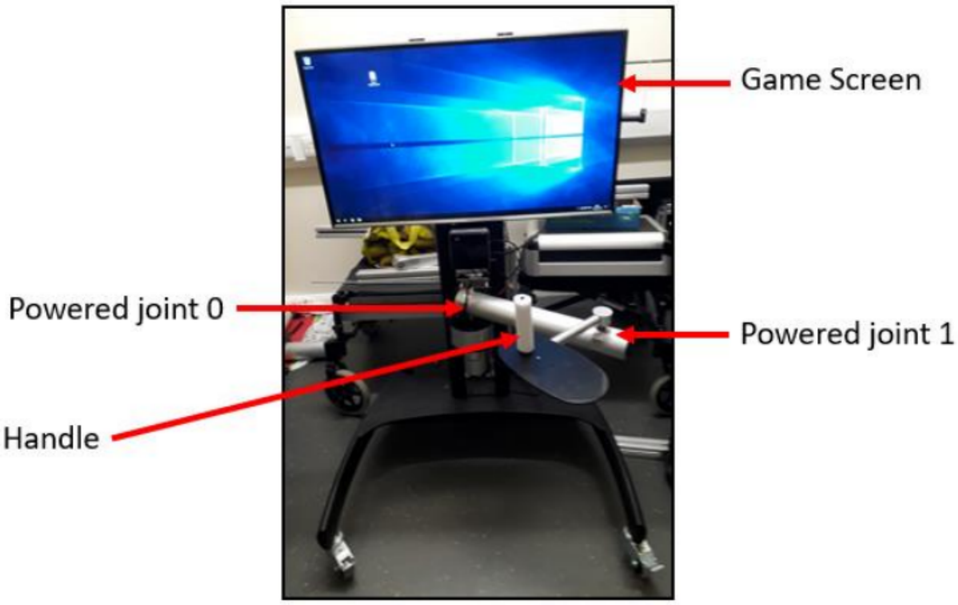

Figure 1.1: *The Current Iteration of the MyPAM Rehabilitation Robot, Showing the Powered Joints, Game Screen and Handle.*

1.2.1 The First Iteration of MyPAM (hCAAR)

Initial development of hCAAR focused on upper-limb rehabilitation of children with Cerebral Palsy in a community setting, but focus was shifted to upper-limb rehabilitation of Stroke patients in the home. hCAAR differed from the current version of MyPAM in a few important ways:

1. Joint 1 was actuated by a belt drive from the main housing, as opposed to the directed drive implemented in the current version.
2. Low-level control was implemented using an expensive National Instruments compactRIO system as opposed to the low cost National Instruments myRIO implemented in the current version.
3. The game software used to present targets were written from scratch using C++ whereas the current version utilises the Unity development environment for speed of development and programming accessibility.

The hCAAR is shown by Figure 1.2.

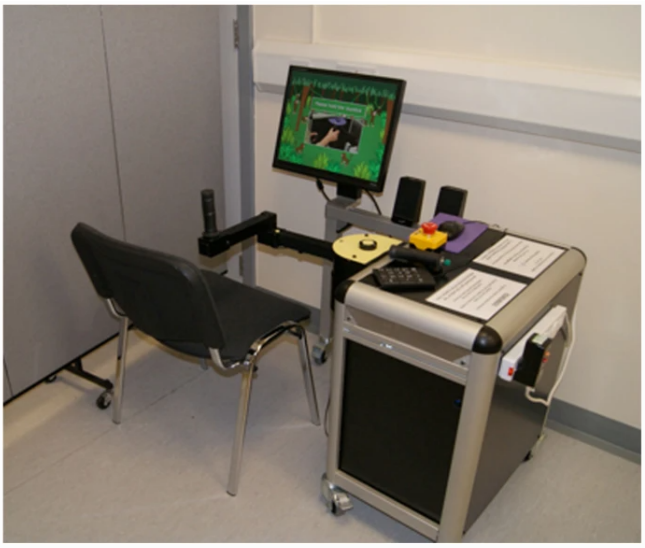

Figure 1.2: *The hCAAR Rehabilitation Robot [11], Which Had a Much Larger Footprint Than the Current Iteration of MyPAM.*

The hCAAR was initially subject to a study where the feasibility of rehabilitation robotics for upper limb rehabilitation of children with Cerebral Palsy was assessed [12].The hCAAR was installed in three schools in two phases of four weeks each separated by a three week maintenance phase. Eleven child participants were presented with reaching tasks and schools were requested to allow each child to use the device for 30 minutes each day, though only an average of 12 minutes per day were achieved. It was reported that 12 minutes of activity per day was insufficient to benefit functional performance, though there were improvements in the kinematics of the upper limb of participants.

The hCAAR was subject to a second study where the feasibility of rehabilitation robotics for post-Stroke upper limb exercise in the home was assessed [11]. Nineteen participants were selected and seventeen completed the study. The hCAAR was set up in the home of each participant for eight weeks, with rehabilitation outcome measures performed before first use, at the end of the eight weeks, and one month after the end of use. Importantly, it was reported that statistically significant improvements were observed in both clinical difference and in kinematic analysis.

Whilst the efficacy of hCAAR was determined, its large footprint made it largely unsuitable for home installation and it was much more appropriate for use in a community setting. Further to this, the time required to write each game from scratch using C++ meant that there was a limited library of rehabilitation games available. This was an issue because patient engagement is an important factor for successful rehabilitation outcomes, and it is likely that a patient will become bored of playing a limited range of games leading to decreased motivation. A final point of note is that the cost of the system was much higher than the current iteration.

1.2.2 The Second Iteration of MyPAM

The 2^{nd} version of MyPAM introduced the physical structure still present in the current version, with each of the two joints powered by a Maxon motor and position feedback provided by ENX encoders. The 2^{nd} version of MyPAM implemented a control hierarchy which is well established in rehabilitation robotics. Reaching tasks were generated by the HLC (high-level Controller), which in this case was a series of games programmed using the Unity development platform. Low-level control was achieved using a National Instruments myRIO and programmed with LabVIEW, which generated the motor demands and performed the data acquisition required for feedback control via a Field Programmable Gate Array (FPGA) chip. A major difference between the 2^{nd} version of MyPAM and the current version is the method of task encoding, where in the 2^{nd} version the trajectory is generated by the games and in the current version the trajectory is generated by the LLC (low-level Controller). The system architecture of the 2^{nd} iteration of MyPAM is shown by Figure 1.3.

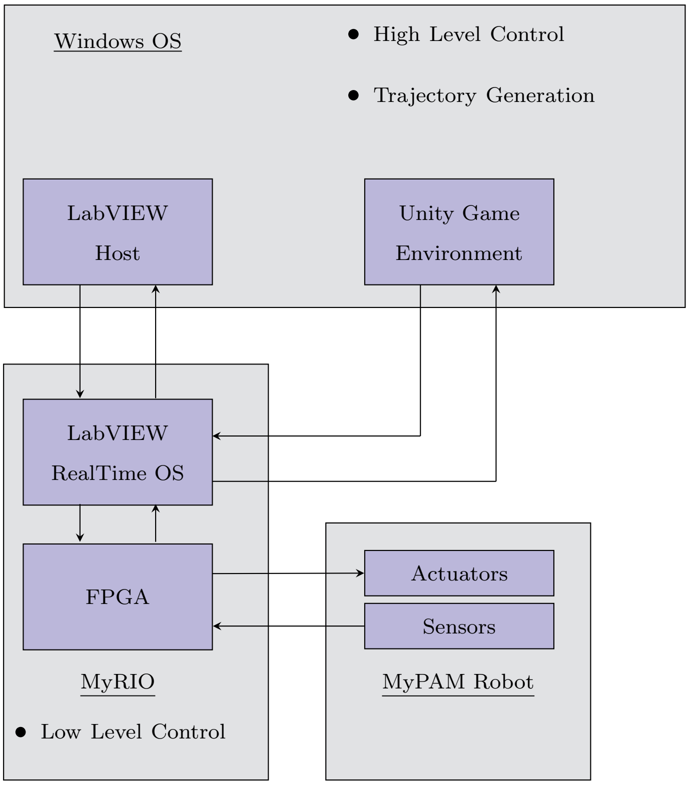

Figure 1.3: *System Architecture of the 2^{nd} Iteration of MyPAM, Showing High Coupling Between the Real Time Operating System and the Windows Operating System.*

In the 2^{nd} iteration of MyPAM the high level controller game was responsible for generating the trajectory for each reaching movement, running at 30Hz on the host PC. The game created the final target position according to a high-level control strategy and generated equidistant intermediate positions between the start position and the final position as a linear path, as shown by Figure 1.4.

These intermediate positions were passed one at a time through a Transmission Control Protocol (TCP) connection to the myRIO running LabVIEW, which acted as the low-level controller. The myRIO, operating at 500Hz on a Real Time Operating System (RTOS), was only aware of the current position and the next intermediate position. Using position PID control, the myRIO generated the motor demands. This is considered to be a highly coupled arrangement, since there was a dependence on reliable and timely communication between the game and the low-level controller to ensure correct and safe operation of the MyPAM. Position PID was used, rather than a control

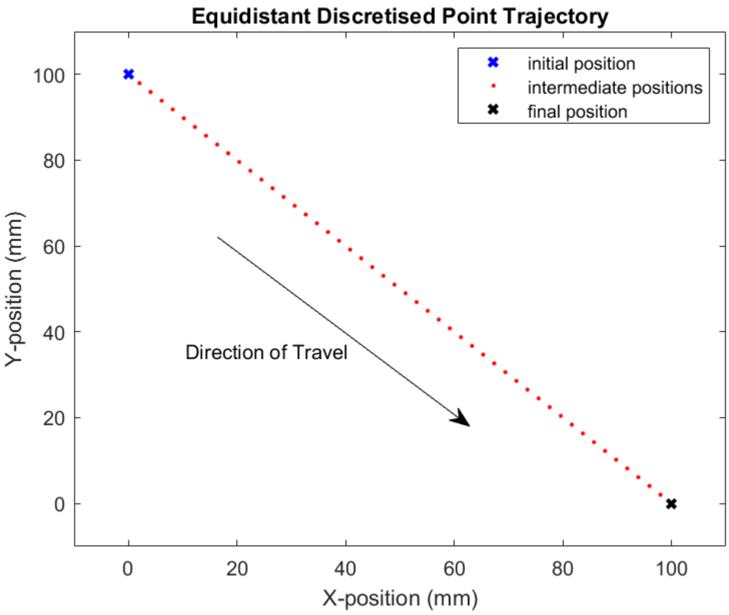

Figure 1.4: *An Example of a Linear Trajectory Generated by the Game.*

scheme which modulates interaction forces between the robot and the patient as a cost saving effort, since reliable industrial force sensors are expensive.

The 2nd iteration of MyPAM improved on the hCAAR in three important ways. First, the footprint of the 2nd iteration of MyPAM is much smaller, meaning that the device is much more appropriate for home use. Secondly, the 2nd iteration of MyPAM uses lower cost hardware, replacing the cRIO system with a myRIO (which costs circa £300 compared with circa £7000). Finally, and most importantly, the game software was developed using a readily available and widely used game engine (Unity). This was an important improvement because it meant that game development was much quicker and the programming skill level required was lower. This allowed a larger suite of rehabilitation games to be developed by multiple people, which is likely to lead to improved rehabilitation outcomes because there is a lower barrier to patient motivation. The low level control scheme generated a simple linear trajectory however, since the setpoints were determined by the games. The preferred trajectory generation strategies for rehabilitation robotics more closely mimic human movement. Finally, the high level of coupling between the games and the hardware responsible for control was not satisfactory for three reasons:

1. The reliability of the control system is reliant upon the non-deterministic Windows OS.

2. It makes the game designers responsible for control decisions.

3. The requirement for background processing in the games makes game development more complicated than necessary, and must be development afresh for every game, rather than just once for the control hardware.

1.2.3 The Current Iteration of MyPAM

This thesis documents the work done for the latest iteration of MyPAM, which has focused on the development of the LLC and decoupling the communication between the HLC and the LLC to ensure that the operating speed of the control scheme is not affected by the graphics processing or non deterministic OS (Operating System) of the HLC. In this latest version of MyPAM reaching tasks are generated by the high-level Controller and passed to the low-level Controller in the form of a single coordinate as a final target, alongside any points of Attraction or Repulsion. Points of Attraction or Repulsion may be used by the game designers to create more engaging games to increase patient motivation.

Trajectory generation in this latest iteration of MyPAM has been improved. The LLC creates a discretised Minimum Jerk trajectory in real time, which is adjusted by the points of Attraction or Repulsion. Work was done to develop a low cost 2-axis force sensor and an Admittance Control Scheme was developed and implemented.

1.2.4 Summary of Versions of MyPAM

Table 1.2 provides a summary of the previous iterations of myPAM.

Table 1.2: *A Summary of the System Architecture of the Three Iterations of MyPAM*

	Iteration 1	Iteration 2	Current Iteration
Low Level Control Hardware	cRIO	myRIO	myRIO
Low Level Control Software	LabVIEW	LabVIEW	LabVIEW
Joint 0 Actuation	Direct Drive	Direct Drive	Direct Drive
Joint 1 Actuation	Belt Drive	Direct Drive	Direct Drive
High Level Control Game Software	C++	Unity (C#)	Unity (C#)
Trajectory Generation	HLC	HLC	LLC
Validation Trials	Yes	No	No
HLC/LLC Communication	TCP/IP	TCP/IP	UDP
Low-level Control Strategy	Position Control	Position Control	Admittance Control

1.2.5 The High Level Controller and Middleware

Much of the effort in the work presented in this Thesis has been made possible by the introduction of software that has been termed 'Middleware' or Mid Level Controller, which has enabled decoupling of the High level controller (ie the rehabilitation games) from the Low level controller. This has allowed a clearer distinction of responsibilities for each level of control. Modularising the system in this was has also made it possible to more easily change components at the system level in the future. The Middleware was created by a separate team, with system design input from the author of this thesis. The introduction of Middleware to the system architecture of MyPAM has allowed the possibility of the device to be linked to the internet, allowing remote monitoring of patient progress and updating of rehabilitation setpoints such as game difficulty and level of assistance. The end goal of this is to allow a physiotherapist to access a hub whereby they may periodically monitor any number of MyPAM devices embedded in multiple patients' homes. In the current iteration of MyPAM the High level controller is responsible for:

1. Presenting the reaching task as a game.
2. Generating the final target of each reaching movement when the previous target has been met, and sending this final target to the Middleware.

3. Generating points of Attraction and/or Repulsion sending these to the Middleware.
4. Displaying gamified elements to the user (Points/Scores, Badges, trophies etc).
5. Displaying the current position of the end effector to the user.

The Middleware is responsible for:

1. Transmitting final targets, points of Attraction and Repulsion, and assistance level to the Low Level controller.
2. Transmitting current position of the end effector to the High Level Controller.
3. Selecting the game (High Level controller instance) to be played and launching the software.
4. Logging all movement data and session data in a database.
5. Providing an interface for the physiotherapist to monitor performance.

The Low Level controller is responsible for:

1. Generating each setpoint of a trajectory.
2. Tracking the position of the end effector.
3. Determining instantaneous motor demands.
4. Applying motor demands to the robot.
5. Transmitting the position of the end effector to the Middleware.
6. Monitoring system health and performing safety monitoring.

1.3 Covid 19 Impact Statement

The Covid 19 pandemic has had a significant impact on the direction of the work presented in this thesis. The original planned outcome was to further develop and finalise the control system for MyPAM, before a successful completion of medical trials and detailed analysis of the trial data. As a result of the pandemic and successive lockdowns it was not possible to arrange access to participants. Further to this, 9 months of lab access was lost. This significantly changed the direction of this work, which focused more heavily on dynamic modelling, trajectory planning and development of a low cost force sensor.

1.4 Novel Contributions

The key novel contributions described in this thesis are:

1. A methodology for affecting a live trajectory with points of attraction and repulsion, described in Chapter 3.
2. A multidomain dynamic model of the MyPAM, documented in Chapter 4.
3. The new application of the MagOne methodology to design a Tristate grip sensor, documented in Chapter 5, which was adapted into a 2-axis force sensor, documented in Chapter 6.

1.5 Thesis Overview

This thesis consists of nine chapters. Each chapter covers a specific aspect of the Low-Level Control scheme developed for this latest iteration of MyPAM and lays out specific chapter objectives at the beginning of each chapter.

Chapter 2: Literature Review
The current state of upper-limb rehabilitation robotics is reviewed to assess opportunities for My-PAM. A disconnect between the cost of robotics and patient benefit is identified. Patient motivation is identified as a key component of successful rehabilitation outcomes.

Chapter 3: Trajectory Generation
This chapter presents two different trajectory generation strategies and justifies the selection of the selection of the Minimum Jerk trajectory as most appropriate for MyPAM. A novel method for adjusting a live discretised trajectory is presented and validated, which allows the creation of more engaging rehabilitation games in the High-level Controller.

Chapter 4: Dynamic Modelling and Analysis
This chapter derives kinematic models for MyPAM, which are necessary for low-level control. A preliminary dynamic model and joint friction models are derived. Finally a novel multi-domain dynamic model encompassing the control domain, electro-mechanical and mechanical domains is presented and validated.

Chapter 5: Grip Sensor
This chapter presents the development and validation of a sensor embedded end-effector/handle for MyPAM, used to detect the presence of a user. The grip sensor has two operating modes. The first is grip mode, where the sensor detects whether the handle is grasped by a user or not. This mode is useful as a control input to the low-level controller to determine an operating mode and appropriate control gains for MyPAM. The second is squeeze mode, where the sensor detects whether the handle is grasped by a user, squeezed by a user, or detects that there is no user present. This mode is similarly useful as an input to the low-level controller, but is also useful as an input to the high-level controller where a squeeze may be used as an input to a game.

Chapter 6: Force Sensor
This chapter presents the further development of the grip sensor developed in Chapter 5 into a two-axis force sensor, intended for use as a control input to an Admittance Control scheme for MyPAM.

Chapter 7: Control
This chapter documents the Position Control scheme for MyPAM. The Position Control scheme has two sets of control gains depending on the presence of a user, since the dynamic conditions are different when MyPAM is connected to a user or not. The Admittance filter used as for the Admittance Control scheme is derived, and the effect of the Admittance filter is validated. Finally, an instability protection strategy is derived and validated.

Chapter 8: System Integration
This chapter presents the implementation of the Low-Level Controller. The software architecture is presented and design choices are justified.

Chapter 9: Discussion and Conclusion

The concluding chapter presents a general discussion of the outputs and findings from this work. The research objectives are then reviewed, and future research possibilities are identified.

Chapter 2

Literature Review

This review presents an analysis of the relevant literature related to robotic devices used to assist with the rehabilitation of Stroke patients. A particular problem identified is patient access to sufficient rehabilitation because the expense of rehabilitation robotics places them solely in the domain of a clinical setting. The mechanisms and effects of Stroke are initially presented, before the traditional approach to physiotherapy is described. The main body of the review explores the control methodology of rehabilitation robotics and some specific examples of current robotic solutions. Finally, the effects of Covid 19 on patient rehabilitation were examined. The aim of this chapter is to highlight the significant need for low cost and accessible robotic tele-rehabilitation solutions to meet the increasing rehabilitation demand.

2.1 Stroke

2.1.1 Stroke Mechanisms and Effects

Stroke, also known as Cerebrovascular Accident, is the leading cause of disability in the UK [2]. Stroke is a neurological lesion caused by a abrupt interruption of blood flow to the brain [13]. Stroke is classified by two mechanisms: Haemorrhagic Stroke and Ischaemic Stroke. Haemorrhagic Stroke occurs when an artery in the brain ruptures, often as a result of high blood pressure. Ischaemic Stroke occurs due to the blockage of an artery in the brain, usually caused by a blood clot or fatty deposits. Both mechanisms lead to cell damage or cell death in the affected region of the brain because of a lack of Oxygen [14].

The symptoms of Stroke are wide ranging and dependant on which region of the brain has been affected and the severity of the Stroke. Different regions of the brain control different behaviour, as shown by Figure 2.1.

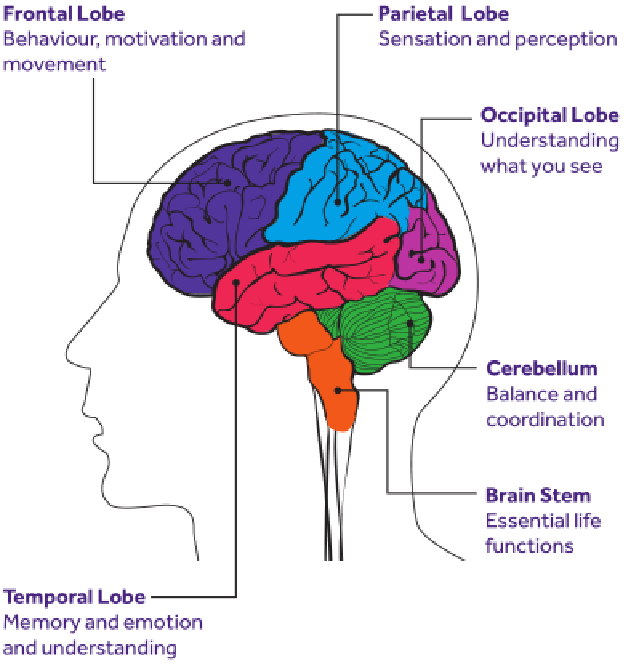

Figure 2.1: *Different Regions of the Brain Associated with Different Behaviours [2].*

Common symptoms include motor impairment along one side of the body (known as hemiparesis), paralysis along one side of the body (hemiplegia), impairment to speech (aphasia), difficulties swallowing, muscle spasticity (hypertonia) and impairment to memory. It was found that up to 80% of Stroke patients initially experience motor difficulties [15]. A community-based study on first-time Stroke patients found that 77.4% of the Stroke patients suffered from some form of upper limb impairment [16].

Stroke has a significant negative impact on a patient's quality of life. Regular activities such as walking, eating, and manipulating objects become difficult or impossible. This often leads to dependency on care and assistance from others. Aside from the personal impact on the patient, Stroke has financial implications for society. It is estimated the mean cost of health and social care per Stroke patient to be £46,039 [17]. This figure is in close agreement with the estimate in 2017 that the mean cost of health and social care per Stroke patient was £45,409 [18].

2.1.2 Stroke Prevalence

A study found that 1.75% of a sample population of 2000 had suffered from Stroke [19]. Stroke can occur in people of any age, but it has been shown that the likelihood of an individual having a Stroke increases with age [2]. The population of the UK is aging, with 26.5% of the population projected to be aged 65 or older by 2041 [20]. This 'greying' of the population is common across most Western societies due to falling birth-rates and an increased life expectancy, a trend which is projected to become an issue globally. The number of adults living with the effects of Stroke is projected to increase by 27% in the European Union by 2047 [21]. Figure 2.2 shows an age group distribution of the population using data gathered from 195 United Nations countries from 1950 onward and projected to 2050 alongside a projection of the population aged 65 or older in the UK.

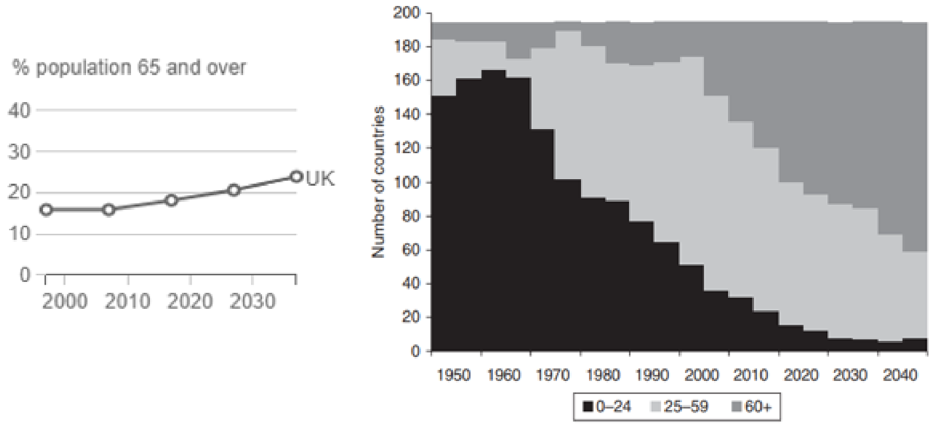

Figure 2.2: ***Left:*** *A Chart showing a Projection of Aging of the Population in the UK [20].* ***Right:*** *A Chart showing a Projection of Aging of the Population in the World [22].*

Observing the projected trend, it is reasonable to expect that the total number of Strokes will increase. This will increase the demand and financial costs upon the NHS and rehabilitation services, especially when considering that the research shows that early and intensive physical rehabilitation is an important factor in recovery [3, 4, 23].

2.1.3 Neurological Recovery After Stroke

Since Stroke is a neurological issue, it follows that Stroke recovery must exploit neurological mechanisms. Cerebral plasticity (otherwise known as neurofunctional plasticity) is the ability of the brain to "reorganise during ontogeny, learning or following damage" [24]. It is this ability of the brain to reorganise that provides the mechanism for Stroke recovery, though this mechanism is not yet fully understood [25].

Without the intervention of rehabilitation, there does remain some natural motor recovery after Stroke, though this varies considerably from patient to patient. The timeline for natural motor recovery after Stroke is summarised in Table 2.1.

It can be seen from Table 2.1 that the neurofunctional plasticity of the brain is most dynamic after the Hyperacute phase, but then the dynamism slows. Once the patient has reached the Chronic stage, the plastic processes become static and motor deficits remain unchanged after this point without the intervention of therapy [25].

Table 2.1: *A Timeline for Natural Motor Recovery After Stroke, Showing the Decline of Neoplasticity Over Time.*

Stage		**Timeline**	**Stroke Events**	**Plasticity**
Onset			Stroke Occurs	
Acute	Hyperacute	Up to 48 hours from onset	Consequences most prominent.	
	Acute	Up to 4 days from onset	Secondary events reach full force.	Most Dynamic
	Subacute	Anywhere from 48 hrs after onset, lasting 2-3 weeks	Secondary events begin to subside.	Most Dynamic
Period of Consolidation		From end of Subacute phase, lasting no more than several months after onset.		Dynamism Slows
Chronic		Defined as the time after which the direct and secondary consequences have subsided and plastic process become static.	All primary and secondary events subsided	Dynamism Ends

2.1.4 Physiotherapy After Stroke

The use of physiotherapy is an accepted element for the rehabilitation of Stroke patients. Physiotherapy is applied by trained physiotherapists, though there has been a rise in the use of robots for post-Stroke physiotherapy in recent years. There is little agreement on the effectiveness of different rehabilitation strategies, though the two most common strategies are Proprioceptive Neuromuscular Facilitation (PNF) and Cognitive Therapeutic Exercise (CTE) [4, 26]. PNF involves stretching and contracting a targeted muscle group, as shown by Figure 2.3.

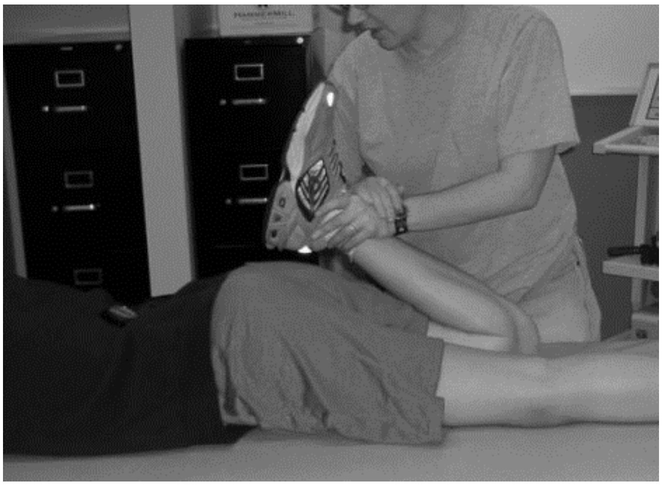

Figure 2.3: *The Application of the Proprioceptive Neuromuscular Facilitation Technique [27].*

More advanced PNF involves resisting the movement of the patient, although this relies on the patient having enough motor control to move the exercised limb.

CTE involves high level cognitive training through task-based activity [28]. Robotic rehabilitation devices use the CTE strategy due to the ease of the gamification of tasks using computer game or virtual reality technologies.

It has been shown that physical rehabilitation is more effective when performed intensively and early after Stroke [3, 4]. This make sense when considering the neurofunctional plasticity of the brain is most dynamic early after onset, as shown in Table 2.1. However, it is acknowledged that "the optimal schedule and content of rehabilitation in the acute phase of care is still undefined" [4]. It is generally agreed that early intervention of physical rehabilitation is important for recovery, but there is little evidence to support the existence of an optimal rehabilitation strategy since the recovery mechanisms are not understood [25]. It is the opinion of the author that the use of robotics in Stroke rehabilitation is much more suited to CTE base therapy than PNF or Stretching based therapy for reasons of patient safety and comfort.

2.2 Stroke Rehabilitation Techniques

The necessary rehabilitation techniques depend on both the affected part of the brain and the corresponding symptoms. In this thesis the focus is upon upper limb rehabilitation, though the following concepts apply for many other aspects of Stroke rehabilitation. Commonly applied techniques include the Bobath concept, Muscle strengthening exercises, Stretching, Bilateral training, Forced use, Constrain-Induced Movement therapy (alongside complementary techniques such as Music Therapy and Electrical Stimulation of muscles) [29].

The Bobath treatment involves guiding patients through the initialisation and completion of intended tasks and is a specific application of CTE, with the main goal of the treatment to normalise muscle tone and facilitate volitional movement through assisted postural control of the patient [30]. There is evidence that the technique promotes desirable rehabilitation outcomes [31], though a lot of research suggests that the technique is no more effective than other forms of rehabilitation [32][33][34]. In terms of robotic rehabilitation, the control of a system using the Bobath would be more complex then other forms of CTE, and since there is no evidence that it promotes better outcomes than other forms of CTE there is little reason to pursue it as a rehabilitation strategy for MyPAM.

Muscle strengthening exercises involve active exercise of the affected limb against a resistance, which may be applied by the physiotherapist or by weighted apparatus (or indeed a robot). It was once considered that muscle strengthening exercises promoted spasticity in the affected limb, though it is now considered that Muscle strengthening exercises add value to the rehabilitation pathway as a result of further evidence [35]. It is apparent that this particular technique is one which is easily replicable through the use of robotics, and is thus a prime candidate for further consideration for MyPAM. This technique is applicable only to patients who have regained some use of the affected limb, however, and is not appropriate for stronger limbs requiring a degree of assistance to even begin an exercise.

Stretching (part of the PNF technique) is a technique intended to combat spasticity and is commonly applied directly by a physiotherapist. Stretching alone is not considered to be sufficient for regaining upper limb mobility, and must be used in conjunction with CTE techniques, though there is recognisable value in combating spasticity. In the opinion of the author, unlike CTE techniques discussed in this section stretching is a less appropriate domain for home based rehabilitation robotics such as MyPAM because it is difficult to apply appropriate force levels to ensure patient safety and comfort.

Bilateral training is based on the movement of the unaffected limb simultaneously with the affected limb to exploit human tendencies towards symmetric or asymmetric motions, with evidence of error correction in the affected limb [36] . For upper limb rehabilitation, Bilateral training consists of repeated movements of both arms in symmetric or asymmetric movements. There is good evidence of the efficacy of Bilateral training [37][38]. Bilateral training has been used in rehabilitation robotic devices, with a notable example being the MIME [39] (documented in detail further in Section 2.5.3) which uses Bilateral training as one of its rehabilitation strategies.

Forced use is a method where unimanual use of the affected limb is forced by restraining the unaffected limb for extended periods of time during normal daily living. There is evidence that the rehabilitation outcomes of forced use is comparable to standard task based physiotherapy [29]. This technique is not applicable to robotic rehabilitation, since forced use refers to constraining the unaffected limb for extended periods of time rather than just for a physiotherapy session. An extension of forced use is Constrain-Induced Movement therapy, whereby the unaffected limb is constrained as before and the unaffected limb is exercised during intensive blocks of training (up to 6 hours per day of repetitive CTE task-based rehabilitation for 10 consecutive days). The repetitive nature of these task based training is well suited to robotic devices, and the gamification of these task is likely to increase patient motivation over such a long time period compared with non-gamified tasks.

A final approach worth mentioning is that promoted by Carr and Shepphard, which proposes that rehabilitation must be task oriented and use cognitive and perceptual functions [40]. Active participation from the patient is required (as is the case in all task based rehabilitation). The approach is based around the idea of neuroplasticity and the brain's ability to reorganise. Such task based activity is again ideally suited to robotic applications, with the benefit of this being that gamifying the tasks allows greater stimulation of cognitive and perceptual functions because of the ability to create interesting environments with exciting visual feedback to disguise mundane and repetitive tasks.

2.3 Limiting factors in Stroke Rehabilitation

There exist numerous challenges in the rehabilitation of Stroke patients. A major challenge is the lack of understanding of the mechanism of recovery [25] and that the ideal rehabilitation journey is yet to be defined [4], and is likely to be required to be defined individually for each patient according to their injury and needs in the opinion of the author. Despite this, the efficacy of several techniques has been proven.

Stroke rehabilitation is resource intensive, particularly in the early stages of rehabilitation since it is widely understood that rehabilitation is more effective when applied intensively and early after Stroke [3, 4]. Coupled with this is the ageing population and corresponding rise in Stroke patients leading to greater strain on resources making it difficult to fully serve all Stroke patients sufficiently, particularly considering that the most effective rehabilitation relies on one-to-one sessions with highly skilled physiotherapists. Rehabilitation providers have reported that resourcing issues including inadequate staffing and insufficient time have been contributing barriers to providing the necessary amount of rehabilitation [5]. Compounding this issue is the location of the population. In cities it is likely that there is better access to rehabilitation services, but one study in the USA identified that individuals in more remote rural settings have difficulty accessing rehabilitation [41]. In short, Stroke rehabilitation relies on many intensive sessions with a highly trained practitioner, which is both expensive and a logistical challenge making it difficult to provide sufficient access.

Additional challenges remain for patient motivation. High levels of motivation are important to achieve good rehabilitation outcomes [42][43], with the requirement of active participation from the patient. It has been found that in general adherence to prescribed exercises (which are repetitive and often mundane) decreases once patients are sent home, often due to a lack of motivation [6]. This clearly has a negative impact on patient outcomes, but certainly highlights an opportunity for home-based robotic systems such as MyPAM. Home based robotic systems are able to gamify rehabilitation tasks, providing much more engaging rehabilitation and encouraging higher levels of motivation. Indeed, robotics as a field is well suited to repetitive tasks of this nature.

2.4 Robotics in Stroke Therapy

In recent years there has been increasing interest and research into the use of robots for rehabilitation of Stroke patients. Rehabilitation robots are categorised by their mechanical structure as either an end-effector based device or an exo-skeleton based device [44–46]. These can be further categorised as Class 1 or Class 2 devices [11, 47]. Class 1 devices are of high cost and intended for lab or hospital use, whereas Class 2 devices are low cost and intended for home use. Most of the research in robotic rehabilitation devices has focused on Class 1 devices, since it was necessary to produce evidence that robotic rehabilitation was a valid rehabilitation strategy. However, it has been identified that there is a need to improve the cost-to-benefit ratio of rehabilitation robotics, focusing on home use and low supervision [48], which is a niche filled by Class 2 robots.

2.4.1 Control Hierarchy

An established control hierarchy exists for rehabilitation robotics, which is necessary because a high-level rehabilitation strategy must be encoded as low-level control strategy [49]. The high-level control strategy is responsible for generating tasks to fulfil the rehabilitation aims. A trajectory must be generated from these tasks, and finally the low-level controller must use these trajectories to generate actuator demands. The low-level control strategies usually run in real time, since these control specific implementations of force, position or other types of interaction control. This hierarchy is shown by Figure 2.4.

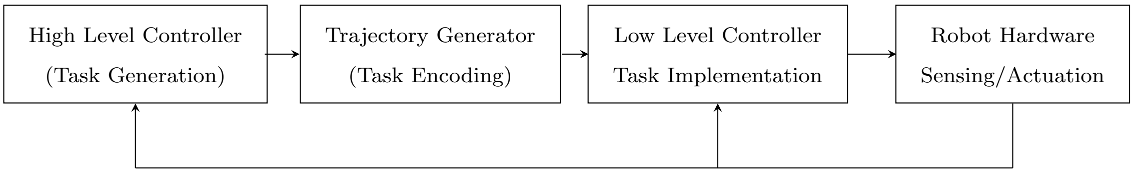

Figure 2.4: *One Accepted Rehabilitation Robotics Control Hierarchy [49], Implemented on the My-PAM System Architecture.*

2.4.2 High Level Control Strategies

All rehabilitation robotic devices must consider and implement both high-level control strategies and low-level control algorithms [44, 49]. The high-level control strategy describes the movement strategy of the robot and is designed to promote neurofunctional plasticity of the damaged motor control areas of the brain. It is suggested that the role of the high-level controller is equivalent to the role of the physiotherapist [50], in that it monitors the status of the task, monitors the safety of the patient and informs the low-level controller about the task. High-level control strategies can be broadly split into four categories: Assistive control, Challenge based control, Haptic stimulation, and Non-contacting coaching [44, 49].

Assistive control is a strategy whereby the patient is aided to complete the task. Usually, measures are put into place to allow the patient to move unrestricted as long as the correct trajectory is being followed. If there is deviation from the desired trajectory a restoring force proportional to the level of deviation is applied, as seen with the MIT-MANUS [51]. An Assistive control strategy is commonly implemented with Impedance or Admittance control as the low-level control algorithm. Another type of Assistive control uses a counterbalance to make a task easier for the patient, the Wilmington Robotic Exoskeleton (WREX) [52] being a good example. A further method of implementing Assistive control is to use Surface Electromyography (sEMG) sensors to measure signals in the nerves which are used to trigger assistance according to the patient's movement intention. This is difficult, however, since the noise to signal ratio is very high.

Challenge based control methods are designed to make the task more difficult for the patient, and are categorised as resistive, error amplifying or constraint induced. Resistive strategies resist the movement of the patient, simulating the more advanced techniques of Proprioceptive Neuromuscular Facilitation (PNF). Error amplifying strategies amplify movement errors rather than decrease them [49]. Error amplification strategies have been shown to increase motor learning compared with assistive strategies in a study which tested 18 hemiparetic Stroke patients [53].

Constraint induced strategies involve constraining the unimpaired limb, so that the impaired limb must perform the task. This particular strategy is particularly suited to exercises involving 2 limbs, for example reaching for a large object. Constraint induced strategies are not relevant, however, for end effector type devices such as the MyPAM. In general, challenge-based control methods are not useful for severely impaired patients with little or no motor control, since the patient does not have sufficient control to begin the required movement.

Haptic strategies involve the use of Virtual Reality (VR) or Altered Reality (AR), where the user must where a headpiece which provides visual feedback in a 3-Dimensional environment. It was found that the use of an engaging VR environment for visual feedback coupled with an exoskeleton robotic rehabilitation device (L-Exos) significantly increased patient motivation [54]. A clinical trial of the L-Exos device showed increased motor control after 6 weeks of use [54], though only 3 chronic patients were tested and there is no evidence to show that the implementation of VR provides a greater clinical benefit than simply displaying visual feedback via a computer screen, as implemented by many other robotic rehabilitation devices.

Non-contact coaching devices do not interact with the patient, and simply provide instructions to the patient [44] and [49]. This strategy may be useful for patients with high amounts of motor control but is not useful for patients with higher levels of disability who require assistance to complete exercises.

2.4.3 Trajectory Generation

As with any robot designed to move an end-effector from a starting position to a target position, a trajectory must be generated. The simplest solution is to generate a simple linear trajectory which covers the shortest distance between the current position and the desired position, which is the trajectory generation method used the 2nd iteration of MyPAM as shown in section 2.5.7 of this report. This method, however, potentially means that unacceptable changes in acceleration may be planned.

A commonly used trajectory generation strategy solution produces a Minimum Jerk Trajectory [55–58]. A Minimum Jerk Trajectory minimises the third time derivative of position (known as jerk), which prevents unacceptable changes in acceleration. Minimum Jerk Trajectories are an example of trajectories based on normative mathematics, which cover the most common trajectories used for rehabilitation robotics [49]. There is no evidence that using normative trajectories in rehabilitation robotics promote motor plasticity, though there has been a link shown between the smoothness of a movement and the brain activation [59]. Further to this there has not been any work done to compare the efficacy of different trajectory generation strategies or to ascertain user preference.

Another common approach is to pre-record a trajectory [60], whilst a further less common method is to generate the trajectory based on the movement of the non-affected limb [61], but this is only suitable for bilateral tasks [49] and is therefor not applicable to MyPAM.

2.4.4 Low Level Control

It is the responsibility of the low-level controller to generate actuator demands according to the provided trajectory. This may be achieved simply by position control, as with MyPAM, or Force control. More complex low-level controllers employ interaction control schemes. It is argued that the case of a robotic physiotherapy device interacting with a human patient should be considered as a coupled mechanical system [44]. This means that the use of a force control strategy or a position control strategy alone is insufficient, since interaction forces with the patient are not accounted for and are thus inherently unsafe. Further to this, failure to account for interaction forces raises the possibility of controller instability. This instability was demonstrated by showing that the Rough-Hurwitz stability criterion were met when considering an example system in isolation but were not met when considering the same system in a coupled mechanism [62].

In order to account for interaction forces, the majority of rehabilitation robotic devices employ Impedance Control or Admittance Control as the low-level control strategy. Impedance Control and Admittance Control involve modulating the dynamic behaviour of the robot alongside position or force control [63], by specifying the robot's position and force relationship using virtual mass, spring and damping characteristics. Essentially, the desired position changes due to the application of an external force in a predictable manner defined by heuristically determined mass, spring and damping characteristics [64]. This is shown by Figure 2.5.

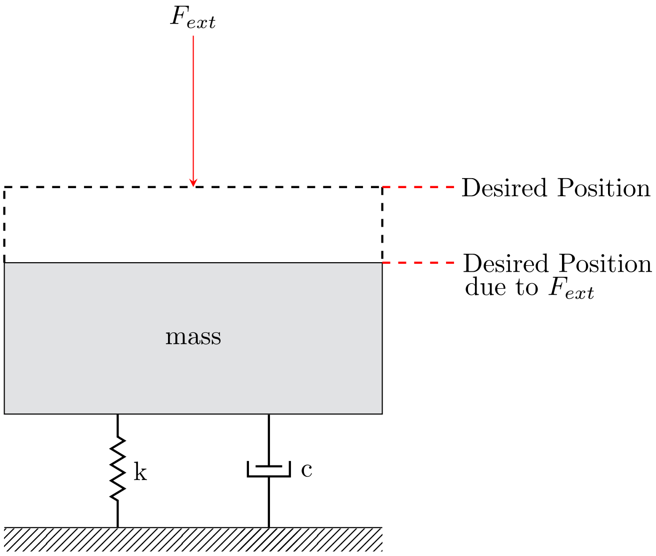

Figure 2.5: *Application of an External Force Changing the Desired Position [64].*

A physical system which accepts force inputs and produces position outputs is defined as an Admittance. A physical system which accepts position inputs and produces force outputs is defined as an Impedance [63, 65]. The end effector of a mechanically coupled robot is subject to physical constraints, in such a way that it may act as either an Admittance or an Impedance. If the environment acts as an Admittance, the end effector must act as an Impedance. Conversely, if the environment acts as an Impedance, the end effector must act as an Admittance. The practicalities of implementing this is discussed in the next three sections.

2.4.5 Admittance Control

Admittance control is a strategy where the force exerted on the end effector is measured, and the robot provides the corresponding displacement [44]. This means that the controller is acting as an Admittance and the environment is acting as an Impedance. As such, an Admittance control strategy is based around an inner loop position controller, as shown by the block diagram in Figure 2.6.

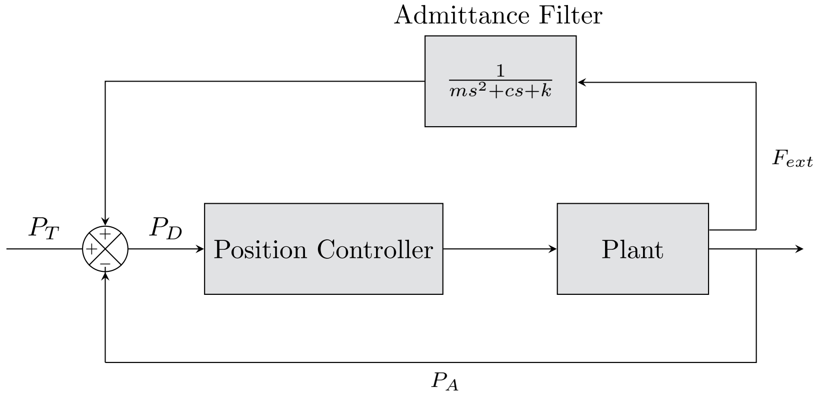

Figure 2.6: *A Block Diagram for a Generic Admittance Controller [64].*

The control signal can be simply defined as shown by Equation 2.1 [66].

$$P_D = P_t + \frac{F_{ext}}{ms^2 + cs + k} \tag{2.1}$$

Where:

P_D is the position demand
$P_t = P_T - P_A$ is the initial trajectory
F_{ext} is the interface force error

This means that the interaction force error is used to adjust the position demand of the robot, where there are two control goals applied in order of importance:

1. Minimise the interaction force error between the environment and the end effector. In the context of rehabilitation robotics, the patient can be considered as the environment and the desired interaction force is 0N.

2. Minimise the distance between the Desired Position and the Actual Position.

2.4.6 Impedance Control

Impedance control is a strategy whereby the motion of the end effector is measured, and the robot provides the corresponding force-feedback [44]. This means that the controller is acting as an impedance and the environment is acting as an admittance. An Impedance control strategy is based around an inner loop force controller, as shown by the block diagram in Figure 2.7.

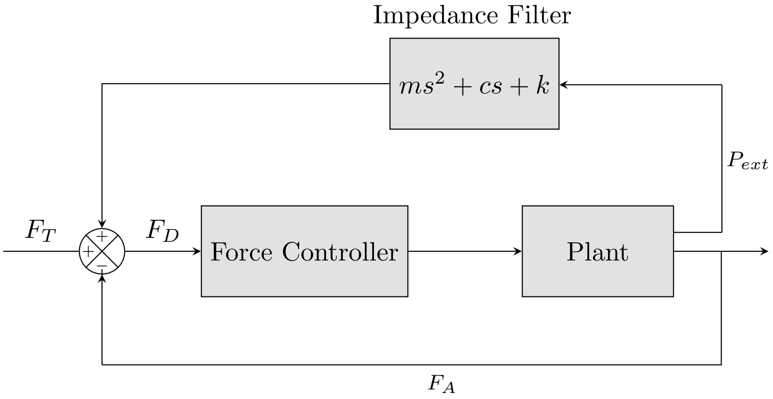

Figure 2.7: *A Block Diagram for a Generic Impedance Controller [64]*

The control signal can be simply defined as shown by Equation 2.2 [66].

$$F_D = F_t + P_{ext}(ms^2 + cs + k) \tag{2.2}$$

Where:

F_D is the force demand
$F_t = F_T - F_A$ is the initial force trajectory
P_{ext} is the position error

This means that the position error is used to adjust the Force demand of the robot, where there are two control goals applied in order of importance:

1. Minimise the interaction position error of the robot, where an error in position is caused by the interaction force between the environment and the end effector.

2. Minimise the difference between the Desired Force and the Actual Force.

2.4.7 Selecting Impedance Control or Admittance Control

The advantages and disadvantages of Impedance and Admittance Control systems are opposite [44] [65]. This makes intuitive sense considering that the definition of a mechanical Impedance is inverse, or dual [45], to the definition of a mechanical Admittance. Devices using Impedance control are stable in stiff environments but can be inaccurate in free environments due to unmodelled dynamics such as friction [65]. Devices using Admittance control provide "high level of accuracy in non-contact tasks but can result in instability during dynamic interaction with stiff environments" [65], which is corroborated by [44].

It is suggested that the implementation of Admittance Control and Impedance Control follow duality theory, which is that "the manipulator should be controlled to respond as the dual of the environment" [67]. Simply put, this means that inertial environments, which act as an Impedance, require position based low-level control and the optimal interaction control strategy is Admittance control. In contrast, environments characterised by a mass, spring and damper relationship, which act as an Admittance, require force based low-level control and the optimal interaction control strategy is Impedance control.

However, these criteria are not the sole determining factor on the most appropriate low-level control strategy for a device. For example, whilst a robot actuated by direct drive electric motor systems may lend itself to force control (and thus Impedance control as an interaction control strategy), in reality this relies on an accurate dynamic model which can be difficult to obtain. Further to this, the robot would require force sensors at each joint, whereas a robot using Admittance control would require a force sensor only at the end effector, thus proving to be a more cost-effective solution. A comparison of Admittance control and Impedance control are summarised in Table 2.2.

Table 2.2: *Comparison of the Impedance and Admittance Interaction Control Techniques [68].*

	Admittance Control	Impedance Control
Advantages	Only requires a force sensor at the end effector.	Lends itself to direct drive electric motors, which are easy to move by torque control.
	Does not require an accurate dynamic model.	
Disadvantages	Requires a robust position control inner loop.	Requires an accurate dynamic model to ensure controller stability.
		Requires a force/torque sensor at each joint.

2.5 Rehabilitation Robots

Over the last 30 years, much work has been done in the area of rehabilitation robotics. In this section of the Literature Review there follows a brief overview of a selection of devices designed for upper limb rehabilitation of Stroke patients.

2.5.1 MIT-MANUS

MIT-MANUS was the first robotic device designed for the rehabilitation of upper limbs of Stroke patients. The device consists of a direct-drive five bar-linkage SCARA (Selective Compliance Assembly Robot Arm) which provides 2 DoF movement for the elbow and forearm in the horizontal plane [51]. MIT-MANUS guides the patient's arm through a series of predetermined exercises, with visual feedback provided on a computer screen.

A series of extension devices were designed to aid in rehabilitation, since trials of MIT-MANUS found that positive motor learning effects on the exercised muscle groups did not have any effect on unexercised muscle groups. The first module extends the operating range of the MIT-MANUS by adding a third degree of freedom, which allow exercises to be performed in 3D space [51]. The second module was designed to rehabilitate the muscle groups in the hand. The MIT-MANUS and the hand module were successful enough that commercial products were released as the InMOTION Arm and the InMOTION Hand. Figure 2.8 shows the InMOTION Arm.

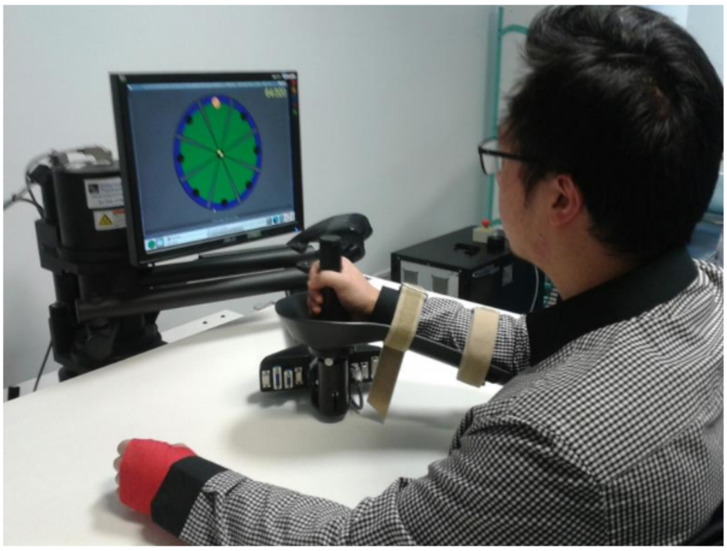

Figure 2.8: *The InMOTION Arm Assisting a User to Reach Targets Presented on a Screen [69].*

One of the driving design features for the MIT-MANUS is that it is "configured for safe, stable, and compliant operation in close physical contact with humans" [51]. This was achieved using Impedance Control as the low-level control strategy and ensuring that the hardware was backdrivable enough that frail patients could easily move the device. The Control hierarchy, which is similar that that seen across all robotic rehabilitation devices and defined in Section 2.4.1, is shown by Figure 2.9.

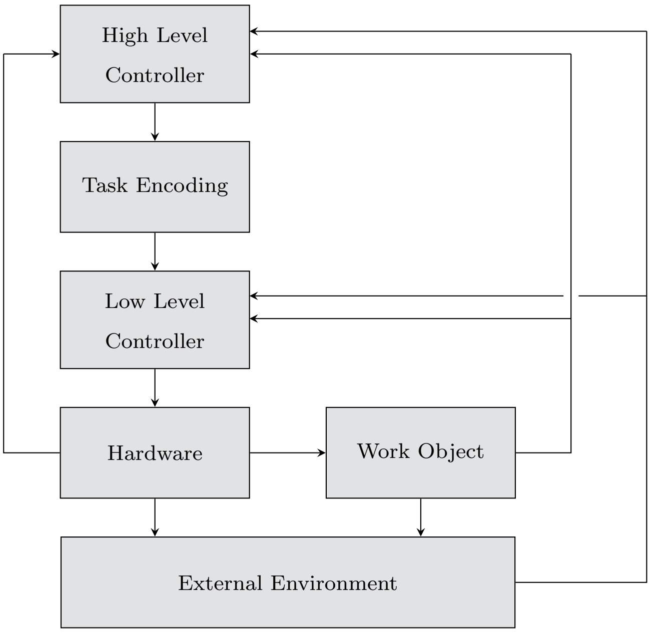

Figure 2.9: *The InMOTION Arm Control Hierarchy [56], Similar to the Heirarchy Presented by Figure 2.4.*

The hierarchy in Figure 2.9 shows the following process:

1. A high-level controller sets the sequence of targets for the therapy session.

2. The Task encoder translates the sequence of targets into sets of Minimum Jerk Trajectories.

3. The low-level controller uses an Impedance control strategy which uses the trajectories to provide varying assistance levels to the patient and control the interaction forces.

4. Force and position feedback from the hardware and environment are used as feedback parameters.

MIT-MANUS has been subject to extensive clinical tests. An initial pilot study was performed in which half of a cohort of 20 patients, as the control group, received only physiotherapist guided therapy and the second half of the cohort, as the experimental group, received physiotherapist guided therapy alongside robot guided therapy. The results of the pilot trial showed that the patients who received robot led therapy alongside physiotherapist led therapy gained significant motor control in the targeted muscle groups [70]. Importantly, it was also found that patients in the experimental group improved “further and faster” than those in the control group [70]. A further study retested a subset of the patients 3 years after the initial therapy, and it was found that the experimental group “showed further significant decreases in impairment measures of the affected limb” [71].

The InMOTION suite (the commercialised version of MIT-MANUS) has recently undergone extensive trials, known as the Robot Assisted Training for the Upper Limb after Stroke (RATULS) trial, to determine whether robot-assisted training improved upper limb function compared with both normal post-Stroke care and a program of enhanced upper limb training [1]. The study consisted of three groups. The first test group were assigned to robot assisted training using the InMOTION suite and 223 people in this group completed the study. The second test group were assigned enhanced upper limb therapy and 222 people in this group completed the study. The control group were assigned to usual care and 190 people in this group completed the study. The two test groups received 45 mins of therapy 3 times a week for a period of 12 weeks alongside the usual care provided by the NHS. The control group received the no more than the usual care provided by the NHS.

Progress assessments were performed by each participant after the trial was complete, and then 3 months after and 6 months after the trial.

It was found that the use of robot-assisted training did not improve upper limb function after Stroke compared with enhanced upper limb training or normal care. Robot assisted training led some improvement in secondary outcomes compared with normal care but was not cost effective. Robot-assisted training had a mean cost of £5387 per patient compared with the normal care cost of £3785 per patient [1]. Essentially, it was determined that using robots in a clinical setting proved too expensive for too little gain in comparison to usual care.

2.5.2 MEMOS (Mechatronic System for Motor Recovery After Stroke)

The Mechatronic System for Motor Recovery After Stroke (MEMOS) is a 2DoF planar robotic rehabilitation system designed to be as low cost as possible. This was achieved by building the device using as many 'off the shelf' parts as possible and ensuring that any part which could not be simply bought was able to be manufactured as simply as possible [72], similar to the approach taken during the development of the MyPAM. Much like the MIT-MANUS, the MEMOS system guides the patient's arm through a series of exercises with visual feedback provided on a computer screen.

The result of these cost saving measures is that the device costs only €4,450. This is considerably more cost effective compared with the estimated $100,000 for the InMOTION Arm, which is also a 2DoF planar robot. The MEMOS system consists of a handle connected to a trolley which runs on rails in a Cartesian configuration, shown by Figure 2.10.

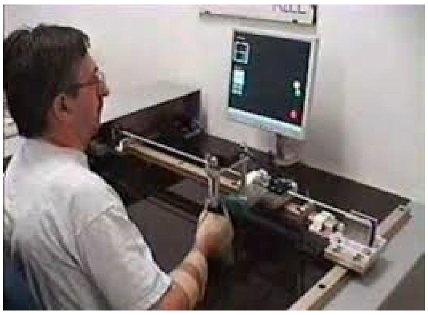

Figure 2.10: *The MEMOS System Assiting a User to Reach Targets Presented on a Screen [72].*

The MEMOS system may use one of three high-level control strategies:

1. Completely assisted movement, where the patient provides no input.
2. Assisted movement, where the patient provides some input.
3. Unassisted movement, where the patient provides total input.

If the patient fails to produce a minimum force after a certain amount of time, the robot moves the handle to the target with a predefined velocity. This is clearly seen in the control signal shown by Equation 2.3 [72].

$$s(t) = k_p F_p + V_R t \delta(F_{min}, T_D) \tag{2.3}$$

Where:

k_p is the Admittance Filter
F_p is the interaction force
V_R is the maximum assistance velocity
F_{min} is the minimum force threshold
T_D is the time threshold
$\delta(F_{min}, T_D)$ is the function used to enable assistance

It can be seen from the control signal that the low-level control strategy implemented is Admittance Control using a simple virtual spring model only, although this choice is not explained. It is possible that the Admittance filter was implemented in this manner to save of computational resources, but a detailed analysis of a Admittance filter suggests that the virtual spring constant k and input force F are the important components of the filter for adjusting the position demand, whereas the constant c and time input t merely regulate the effect of the Admittance, which may be achieved more simply in other parts of a control algorithm.

MEMOS was subjected to a preliminary clinical trial containing 8 patients suffering from chronic hemiparesis after Stroke. The hemiparesis in all 8 subjects was considered to be only slightly to moderately impairing. Testing consisted of a 40-minute session with the device twice a day for 3 weeks, where the exercises consisted of reaching tasks [72]. The results showed that 7 of the 8 subjects improved in motor control of the targeted muscles groups. Importantly, the results of the clinical trial showed that it is possible for rehabilitation to have an effect even in the chronic phase, although it should be noted that none of the tested subjects had severe impairment, so it is not possible to extrapolate the findings to severely disabled chronic Stroke patients. Indeed, it was noted that the MEMOS device would be best suited for use further down the rehabilitation timeline for home use and was no substitute for the more complex Class 1 devices available.

2.5.3 Mirror Image Motor Enabler (MIME)

After an initial 2-DoF prototype was built, the second iteration of the Mirror Image Motor Enabler (MIME) system used an industrial PUMA 6-DoF robot to move the impaired limb of a Stroke patient. The MIME device moves the patient's arm in a planar motion, with the weight of the arm borne by a separate support containing position sensors. The unaffected arm is connected to a separate support, also containing position sensors. A third iteration of the device used a larger PUMA 6 DoF robot which could support the full weight of the impaired limb and allow the support for the unaffected to be removed. The benefit of this was that a 3D workspace could be utilised. This arrangement is shown by Figure 2.11.

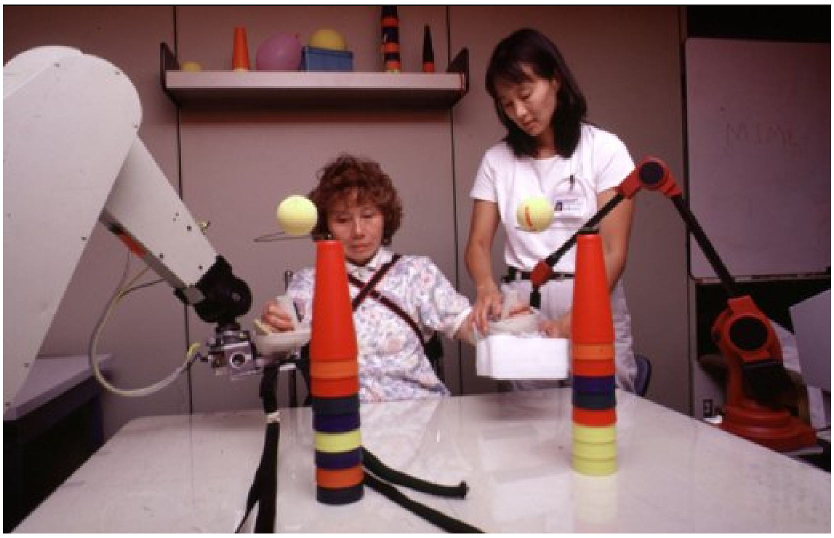

Figure 2.11: *The 3rd Iteration of the MIME System Assisting a User to Perform Reaching exercises [39].*

The MIME system has 4 high-level control strategies. The first is a completely assisted mode (in the literature called 'passive-guided mode'), where the robot moves the impaired limb along a predefined trajectory and the patient is required to input no effort. The second is an assisted mode, where the patient initiates movement of the impaired limb and the MIME robot provides assistance to complete the exercise. The third is a resistive mode, where the patient moves the impaired limb and the MIME robot resists the motion. The fourth is a novel bilateral mode, where the robot moves the impaired limb as a mirror image to the movement of the unimpaired arm, in a master/slave relationship [73]. There is no explicit description of the low-level control system, but the literature states that both joint position and patient-handle interaction force were measured [74], which suggests the use of an Admittance control scheme.

The MIME system has undergone extensive clinical trials. In an initial trial of the 3rd iteration of the device 11 chronic subjects were exposed to robot training therapy as the test group and 10 chronic subjects received traditional physiotherapy as the control group. Therapy sessions lasted for 1 hour, and this occurred for 24 sessions over a 2-month period [39]. It was found that the robot test group experienced increased motor control in the targeted muscles groups to a greater extent than the control group, though at a 6 month follow up it was found that the gains were equivalent in both the test group and the control group.

In a further study subacute subjects were split into 4 test groups. The first of the groups were exposed to robot therapy that started as completely assisted and progressed into resistive therapy. The second of the groups were exposed to bilateral robot therapy. The third of the groups were exposed to robot therapy that was split between bilateral training and unilateral training. The fourth group were exposed to no robot therapy, but instead received an equal amount of traditional physiotherapy [74]. The therapy sessions lasted for 1 hour, and this occurred 15 times over a 4-week period. The robot test groups demonstrated significantly increased motor control in the targeted muscle groups at the end of the testing, to a much greater extent than the control group. This is consistent with the previous study. However, at a 6 month follow up it was found that "gains in robot and control groups were equivalent" [74], similar to the chronic test group from the previous study. This suggest that in the long term robot assisted therapy has equivalent outcomes to traditional rehabilitation, possibly triggering long term neurofunctional plasticity in a similar way.

In a follow up it is suggested that from a pragmatic point of view, robotic therapy is useful for patient motivation when access to a physiotherapist may be limited [73], even if the long-term gains from robotic therapy are equivalent to that of traditional physiotherapy. To this end, it is suggested that research efforts should be directed towards producing low-cost versions of clinically tested robots. Indeed, devices which are simple to use and do not require a multi-disciplinary team but which are shown to have positive effects on rehabilitation outcomes are likely to be adopted by the clinical community [75]. This is an especially important comment in the current climate of social distancing due to Covid 19, where access to rehabilitation services are limited and it has been shown that increased motivation leads to improved rehabilitation outcomes as discussed by [42] and [43].

2.5.4 Assisted Rehabilitation and Measurement (ARM) Guide

The Assisted Rehabilitation and Measurement (ARM) Guide is a 2 DoF device designed to rehabilitate and measure upper limb reaching movements of Stroke patients. The device is mechanically simple and consists of a splint connected to a linear slide rail. The splint is driven along the rail using an electric motor. The slide mechanism can be adjusted in the horizontal and vertical planes, allowing a variety of reaching exercises to be performed [76]. The interaction force between the patient and the ARM Guide is measured using a 6 DoF force sensor. The ARM Guide is shown by Figure 2.12.

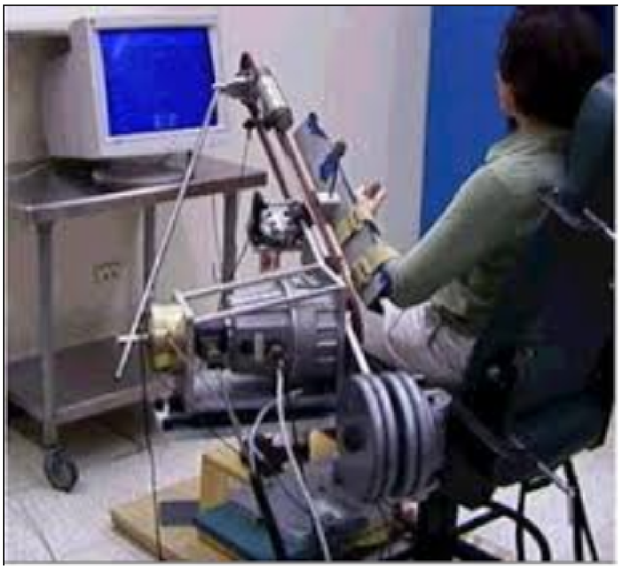

Figure 2.12: *The ARM Guide System Assisting a User to Reach Targets presented on a Screen [76].*

The ARM Guide system uses 2 distinct high-level control strategies. Due to the linear mechanical design of the ARM Guide system, all exercise trajectories are linear. The first high-level strategy is termed ‘Counterpoise Assistance’, which is based on traditional physiotherapy techniques. Counterpoise Assistance provides enough assistance to overcome passive forces resisting the desired motion, such as gravity and arm tone [76]. The low-level control scheme implemented to achieve this involves measuring the resistive forces and counteracting them by applying an opposite force with the motor. The second high-level strategy is Triggered Assistance, where full assistance is given to complete the reaching exercises as soon as the patient initiates the movement. The low-level control scheme supporting this uses a PD position control loop.

The ARM Guide was subjected to trials containing 19 Chronic Stroke patients suffering from hemiparesis, who were split randomly into 2 groups. The test group engaged in robot assisted reaching exercises and the control group engaged in unconstrained and unassisted reaching exercises. The study consisted of 24 separate sessions lasting 45 minutes over a period of 8 weeks. It was found that both groups had a significantly improved range of movement and velocity control at the end of the study, and that the robot assisted test group did not experience statistically significant detectable improvements in motor control beyond that experienced by the unassisted control group [77].

2.5.5 End Effector Upper Limb Rehabilitation Robot (EEULRebot)

The End Effector Upper Limb Rehabilitation Robot (EEULRebot) is a planar system designed to assist in the rehabilitation of Stroke patients with upper limb motor control deficiencies. The planar workspace is adjustable in the vertical direction by adjusting the inclination angle, making the device quasi-3 DoF. The 2 joints are powered using Maxon motors. The end effector contains a force sensor to measure the interaction force between the patient and the EEULRebot device. A SolidWorks model of this arrangement is shown by Figure 2.13.

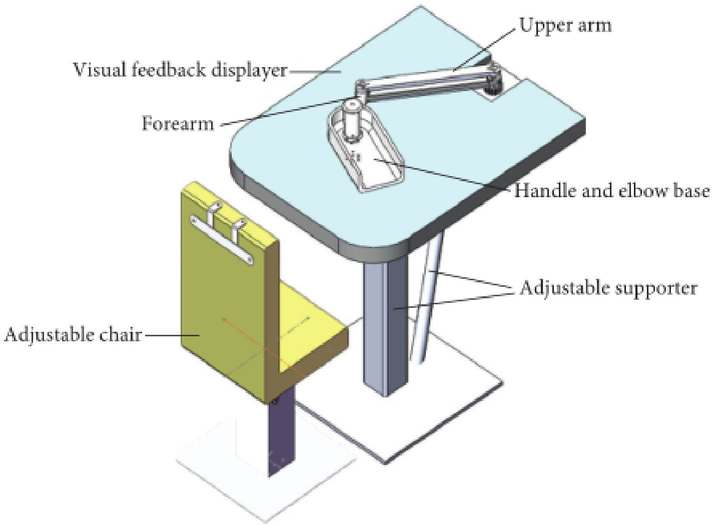

Figure 2.13: *A SolidWorks Model of EEULRebot System [78], Which has a Similar Structure to MyPAM.*

The EEULRebot System has three distinct high-level control strategies [78]. Similar to the MIME System in Section 2.5.3, The first is a completely assisted mode (in the literature called 'passive-guided mode'), where the robot moves the impaired limb along a predefined trajectory and the patient is required to input no effort. The second is an Assistive mode (in the literature called 'Active-Constrained mode') where the robot provides assistance to complete the exercise, and in particular provides a restoring force to ensure that any deviation from the desired trajectory is corrected. The third is a Resistive mode (in the literature called Active Assistant or Resistant Mode') where the robot resists the movement of the end effector if the user exceeds a velocity threshold in the direction of the desired trajectory, thus making the exercise more difficult.

Impedance control was used as the low-level control strategy, with a restoring force normal to the trajectory designed to align the current position with the current point on the desired trajectory [78]. A force parallel to the direction of the desired trajectory was also defined. In the assistive mode this force was positive, helping the patient to move the impaired limb along the desired trajectory. In the resistive mode this force was negative, making it more difficult to complete the exercise.

The EEULRebot device underwent trials with 11 healthy subjects and 3 hemiplegic subjects who had suffered from a Stroke. The healthy subjects tested all 3 of the high-level control strategies, but only 1 of the hemiplegic subjects was physically able to do so. The other 2 hemiplegic patients had insufficient motor control to test the Assistive mode or the Resistive mode, and so only tested the Passive-guided mode. The trial was designed to test the robustness of the device rather than to validate it in as a useful clinical tool. The trials demonstrated that the EEULRebot was robust in that the control strategies worked as intended, though it was noted that further extensive trials were required with a greater number of impaired test subjects [78].

2.5.6 intelligent Pneumatic Arm Movement (iPAM)

The intelligent Pneumatic Arm Movement (iPAM) is a cooperative dual robot system designed for upper limb rehabilitation of Stroke patients. Each of the dual iPAM robots uses pneumatic actuators to power the movement of its three joints [79]. Similar to other rehabilitation robotic devices, iPAM guides the patient's arm through a series of exercises with visual feedback provided on a computer screen. The iPAM dual robot system is shown by Figure 2.14.

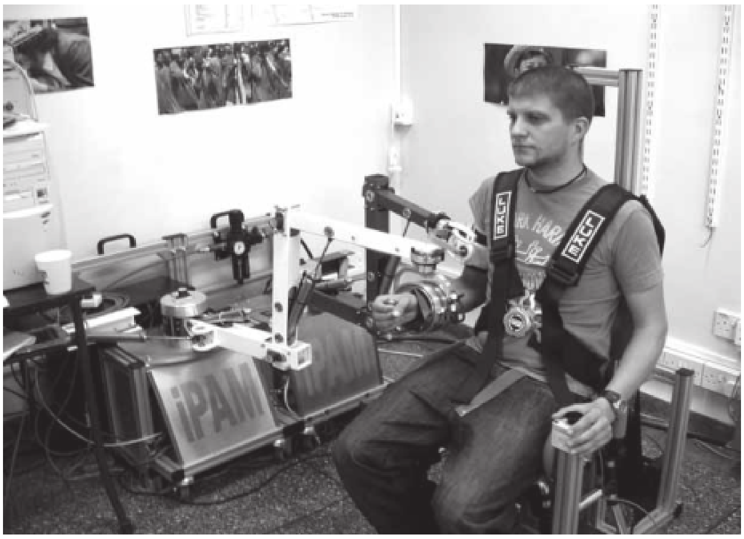

Figure 2.14: *The iPAM System [80], Showing the Cooperative Dual Robots Guiding a User's Arm.*

The iPAM uses Assistive control as the high-level control strategy, and so it assists the patient to complete the exercises. The input trajectory fed to the low-level control scheme is based on the kinematics of the human arm. The high-level control strategy, task encoding and low-level control implementation procedure is consistent with the control hierarchy defined in Section 2.4.1.

The iPAM uses Admittance control for the low-level control strategy. The control scheme is cooperative, since both robots must act in unison [80]. Admittance control was chosen because it favours pneumatic actuation [66, 81]. This is because it is difficult to model the non-linear dynamic effects, such as stiction, of pneumatic actuators accurately enough to ensure the accuracy of the force control inner loop required for Impedance control. The level of assistance provided to the patient is changed by altering the stiffness coefficient c in the Admittance filter.

Initial trials demonstrate that the iPAM is capable of providing assistance to the upper limb with similar trajectories and patterns of movement to a subject's unconstrained motion [82]. This was considered important because the system was designed not to exert unwanted and uncontrolled forces on the limb, which would encourage unnatural motions. It was noted, however, that the device was unsuitable for use with patients who had "little to no voluntary movement" [82]. It was necessary for the patient to have some amount of motor control because the device was not capable of providing total assistance

2.5.7 hCAAR (home-based Computer Aided Arm Rehabilitation)

The hCAAR (home-based Computer Aided Arm Rehabilitation) system, the predecessor of the MyPAM, is a 2DoF planar device developed to be installed in the houses of Stroke patients for upper limb rehabilitation, as well as in community locations to aid in the rehabilitation of children with Cerebral Palsy. The aim of this Class 2 rehabilitation robot was to increase patient therapy hours, since literature suggests that the more access to therapy a patient has the greater the potential for neurofunctional motor recovery. The hCAAR system guides the patient's arm through a series of games, with visual feedback provided on a computer screen, as shown by 2.15.

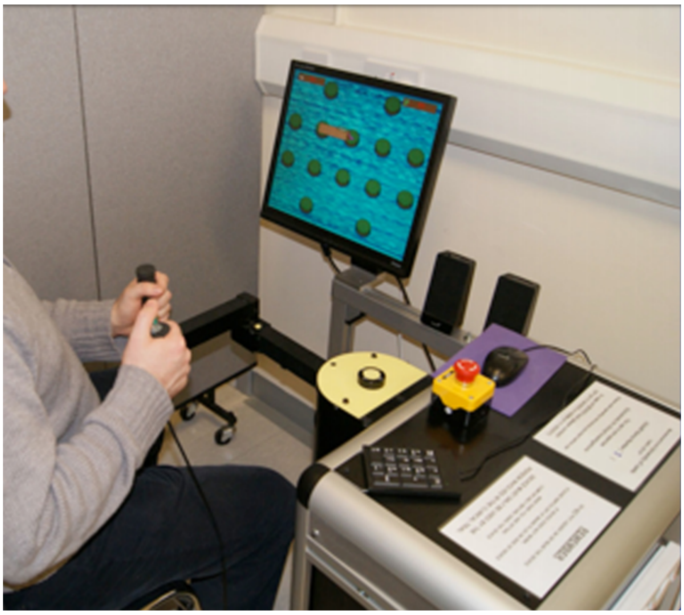

Figure 2.15: *The hCAAR System Assisting a User to Reach Targets Presented by a Screen [11].*

Since the hCAAR was intended for home use, it was designed to be as cost effective as possible. To this end, the first iteration of hCAAR used PID position control as the low-level control strategy to limit the use of expensive force sensors. An experimental version of the hCAAR system tested a novel form of Impedance Control as the low-level control strategy whereby the motor current draw at each joint motor were estimated from a system model, allowing for an inner control loop which controls motor current draw rather than directly controlling torque [83]. This meant that expensive torque sensors were not required for the force feedback necessary for an inner force control loop. It was determined, however, that current measurements were too noisy and thus the reliability of the control system was questionable.

The hCAAR has 2 distinct operation modes, which can be considered as the high-level control strategies. The first mode is assistive, where a variable level of assistance aids the patient to complete the tasks. The second mode is entirely passive and is used to collect for monitoring about patient progress.

The hCAAR system was subjected to clinical trials with 19 patients, 17 of which completed the trial. Each patient had a hCAAR device installed in their home for a period of 8 weeks in order to undertake home exercises alongside their usual rehabilitation. A baseline assessment was carried out just before home installation, an assessment was carried out after the 8-week trial period and a further assessment was carried out after another 4 weeks (at the 12-week period). Statistically significant improvements were observed, though it was noted that a study "comparing the combination of conventional therapy and hCAAR with conventional therapy alone needs to be explored" [11]. It should be noted also that with position control only there is no mechanism for measuring or controlling the interaction forces with the patient, and thus dangerous torques or forces could occur.

2.5.8 RUPERT (Robotic assisted Upper Extremity Repetitive Therapy)

The Robotic assisted Upper Extremity Repetitive Therapy (RUPERT) device is an exoskeleton robot designed to rehabilitate Stroke patients suffering from upper limb mobility problems. The RUPERT device is aimed specifically at training 3DoF reaching tasks critical for daily living [84]. Assistance is provided to the patient through the use of pneumatic 'air muscles', much like the iPAM in Section 2.5.6. The RUPERT system is shown in Figure 2.16.

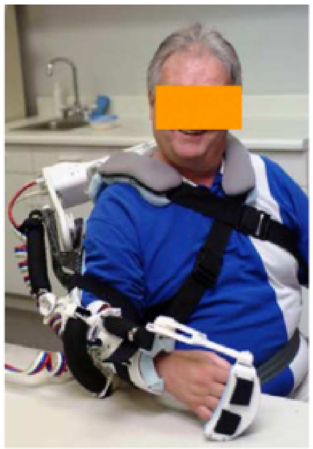

Figure 2.16: *The RUPERT System [84]. Note That the System Uses an Exoskeleton Design, in Contrast to the MyPAM.*

Common to many robotic rehabilitation devices, RUPERT uses Assistive control as the high-level control strategy. The patient is requested to make a movement. If after a certain amount of time the patient has been unable to do so, RUPERT provides assistance. Interestingly, the low-level control of the RUPERT device relies on open loop feedforward position control. This means that the physiotherapist must set and monitor speed and position parameters in order to ensure patient safety, along with a set limit on the maximum angular speed of the joints. The limitations of this low-level control algorithm are acknowledged, and it is stated that closed loop control would be required when dealing with more severely impaired patients [84] .

The second iteration of the RUPERT device was subjected to limited clinical trials, which ran for 3 weeks, containing 10 Stroke patients who had moderate or mild upper limb motor control difficulties. Though the testing description is limited, 6 of the 10 test subjects completed the trials and 5 of these showed improvements in motor control in the targeted muscle groups [84].

2.6 Low Cost Rehabilitation Robotics

One of the main limiting factors of the adoption of robotics in post-Stroke rehabilitation has been the cost of the devices. There has, however, been some development towards more affordable devices which make home deployment feasible. It is noted that there has been very little clinical testing of low-cost devices (particularly in comparison to higher cost robots), which makes it difficult to prove their efficacy. It is apparent that visual feedback on a screen is an important aspect of rehabilitation robotics, but it is also apparent that for these low cost devices there is scope for further gamifaction and an increase in the number of games playable for each device which would serve to promote patient motivation. This is certainly an area of the field which is deserving of more attention.

2.6.1 Upper Limb Devices

The LINarm is a low cost device, designed specifically for home use, featuring force-feedback akin to the MIT-Manus [85]. The LINarm operates in single degree of freedom, and uses the 'assist as needed' paradigm to assist a user to achieve a reaching task. The physical design of the LINarm consist of a motorised handle assembly attached to a linear bearing able to slide along rails. The mechanical design is interesting, incorporating variable stiffness actuators to acheive mechanical compliance. LINarm++ is an extension of LINarm, incorporating sensors into the handle to measure physiological parameters (electrocardiography, skin conductance and peripheral skin temperature) [86], the purpose of which is unclear. The most important improvement in LINarm++, in the opinion of the author, is the addition of simple games which visually display the movement and also add a cognitive element. The LIN arm has not yet been subjected to medical trials.

The DUALarm is an interesting concept based on open source 3D printed hardware and open source software which uses a bimanual rehabilitation paradigm. The DUALarm is a passive device which presents bimanual reaching exercise to users on a screen, with position feedback measured by sensors in the device and processed using an Arduino [87]. The device has undergone preliminary testing with healthy participants to fine tune some of the parameters, followed by initial testing with four hemiparetic Stroke patients. The trial data were not recorded, since the aim of the trial was to assess usability of the device. The device was reported as usable, and suffered no mechanical failure [87]. Whilst the exercises have been displayed on a screen, there was no gamifaction to encourage patient motivation, which could be considered a weakness.

2.6.2 Hand and Wrist Devies

OpenWrist is an exoskeleton robot designed for 3DoF actuation of a patient's hand and wrist, and is designed to be compatible with another exoskeleton system (Masetro) [88]. The work builds upon previous success, namely the RiceWrist-S. The RiceWrist-S was tested on a single participant with a spinal injury who participated in ten sessions of robotic-assisted arm training over 20 days of training. The participant showed a small increase in hand grip, pinch grip and lifting heavy objects after testing, and the system was validated as mechanically sound [89]. There is little description of the visual feedback or gamification beyond visually-guided target hitting movements. It is apparent that a great deal of effort has been spent of mechanical design, control and modelling but not much attention paid to patient motivation. Exercise modes include backdriving the robot whilst unpowered, and challenge based resitive exercises.

The ReHapticKnob is a 2 DOF hand rehabilitation robot which supports hand opening and closing with a translational actuator and wrist rotation with a rotational actuator, with an impedance controller as the low level control scheme [90]. The device presents three different cognitive exercises, with feedback provided to the patient by use of a screen. In the first exercise the hand of the patient is moved one of five positions, which are represented by bars of different length on the screen.

The patient must determine which bar they are supposed to be grasping. The second exercise involves identifying sponges shown on a screen, by grasping them and evaluating the simulated compliance. The third involves rotating a picture on screen by moving their wrist. The device has undergone initial trials with 5 participants, who performed exercises for 45min for between three and six sessions over a two week period. No clinical assessment was made, rather the patients completed questionnaires, with the results showing that the robot assisted therapy was well accepted by the 5 participants [90]. The strength of the ReHapticKnob methodology is the element of cognition required in the exercises, the the limited selection of games could be considered a weakness since it is preferable to provide engaging tasks to increase patient motivation.

Closed-chain Robot for Assisting in Manual Exercise and Rehabilitation (CRAMER) is a robot designed to assist impaired persons in making 3 DoF movements of the forearm and wrist. Actuation is achieved using two servomotors [91]. CRAMER uses a Nintendo Wii remote and the Nintendo Wii games system, since it was recognised that by encouraging patients to play already existing games, which are well designed and intended to be stimulating, high levels of patient motivation could be achieved. Unfortunately, the games required operation of small buttons on the remote and also the highly sophisticated games were too difficult to play for moderately and severely disabled patients. This is an encouraging attempt at gamification, but unfortunately no trials have been performed because of the technical difficulties of interfacing a feedback from commercial video game system into a rehabilitation robotic framework [91] and the ability of patients to push the buttons on the remote.

2.7 Gamification in Rehabilitation

It is understood that increased patient motivation leads to increased rehabilitation outcomes [42][43][92]. A key aspect of rehabilitation robotics is gamification, whereby rehabilitation tasks are presented in the form of a game with the aim of increasing patient motivation by holding their attention on the task in hand.

A psychological concept termed 'flow' is recognised as a state of mind where one is highly focused on an activity, first researched by Getzels and Csikszentmihalyi in the 1960s [93]. This state, which is the gold standard for motivation, is much more achievable with tasks which are engaging compared to repetitive tasks which are boring. Indeed, whether a neurologically diseased patient achieves flow during rehabilitation using gamified technology (such as rehabilitation robotics, Virtual Reality etc) is increasingly measured [94], though it is important to note that the methodologies for measuring flow have not traditionally been rigorously applied, nor is it a state of mind with an obvious metric.

It is apparent to the author that there exist two levels of gamification with respect to rehabilitation tasks. The first level is simply presenting rehabilitation tasks in a gamified format, a good example of which is presenting reaching tasks as a computer game displayed on a screen [51][11][72][76][80] or in a Virtual Reality/Augmented Reality environment [95][96]. The benefit of this is that similar repetitive tasks may be presented in a series of dis-similar game formats, increasing patient motivation because while a task may be mechanically the same, the format in which it is performed changes frequently. A good example of this may be found in the second iteration of MyPAM, where the patient was guided though an immersive game world playing games such as fishing or flying an aeroplane. All the games in this immersive world simply presented reaching tasks to the patient.

The second level is the addition of typical game features to further encourage a patient to progress through the game through positive feedback and appropriate levels of challenge. These features include awarding points, awarding digital badges, awarding digital trophies, incrementing through game levels of increasing difficulty, and ranking players against each other to promote competitiveness [97] (though in the opinion of the author this may have detrimental effects on patients who are progressing less quickly). An example of challenge based gamification may be found in the first iteration of MyPAM (known as hCAAR), where two devices were linked together to allow children with Cerebral Palsy to compete with their friends while playing the same game[12]. Points and trophies were implemented in the aforementioned immersive world in the second iteration of MyPAM.

The use of robotics and other technology in rehabilitation provide a clear opportunity to gamify the rehabilitation tasks, with the benefit of promoting attention to the task and increasing patient motivation. The desired outcome of this is that by promoting patient motivation, rehabilitation outcomes may also improve because there is a link between increased motivation and increased outcome. To this end, efforts must be taken during the development of the third iteration of MyPAM to promote ease of gamification, such that the game developers can concentrate of gamifying rehabilitation tasks without having to consider the low-level control implications of their design choices.

2.8 Advantages and Disadvantages of Robotics in Stroke Therapy

There are several clear advantages to the use of robotics in stroke therapy. Rehabilitation excises are often repetitive, indeed one of the core principles of stroke rehabilitation is repetitive task-based activity [98]. Tireless, repetitive, and repeatable motion is certainly a paradigm well suited to the field of robotics. Additionally, from the perspective of a patient repetitive movement can be mundane. It has been found that in general adherence to prescribed exercises (which are repetitive and often mundane) decreases once patients are sent home, often due to a lack of motivation [6]. This has implications on rehabilitation outcomes, since it is required that patients display motivated active participation. Using rehabilitation robotics for these repetitive motions provides an opportunity to gamify the tasks, providing a much more stimulating and motivating experience [99].

Another advantage is the potential to collect large amounts of user movement data for analysis. This data can be analysed for movement quality criteria, allowing physiotherapists to easily track the progress of a patient [100]. Indeed, kinematic movement data were analysed alongside traditional rehabilitation metrics during the trials of the first iteration of MyPAM [11]. An advantage highlighted by the trials of the first iteration of MyPAM but not highlighted anywhere in the literature is that robots are deployable. This provides much more access to rehabilitation, particularly should a robot be deployed in the home. This is of huge benefit, because rehabilitation is more effective when performed intensively and early after Stroke [3].

There are, however, some disadvantages of the use of robotics in the field of rehabilitation. The key disadvantage is that there is limited definitive evidence that the use of robot therapy is more beneficial than traditional therapy[101]. It is the opinion of the author, however, that there is a niche for home based robotics which hasn't yet been fully explored, which much of the current evidence focused on clinic based robotics.

High cost and staffing resources required to set up and monitor robotic rehabilitation sessions have been reported as another disadvantage (from the perspective of physiotherapists) [102]. This too should be considered also, that staffing resource is already considered a problem in Stroke rehabilitation [5], as discussed in section 2.3. The paradigm of home based robotic systems could be the solution to this, since they would require only initial setup and remote monitoring. Indeed, the therapist in charge of monitoring could monitor multiple systems simultaneously.

One final disadvantage lies in the 'assist as needed' technique employed by many robotic systems. It has been reported that users become reliant on the robot feedback and limit their performance when it is removed [103]. The solution to this is to employ different control strategies at different parts of the rehabilitation journey. 'Assist as needed' is suitable for the early stages of recovery, but these should be replaced by more challenge based strategies as performance improves.

2.9 Covid 19 - The Effects of Social Isolation on Rehabilitation

On the 11th of March 2020 the World Health Organisation declared the Covid-19 outbreak to be a global pandemic. As a response to this, alongside an increasing number of infections and deaths, many countries including the UK began a series of national lockdowns. During these lockdowns all non-essential travel was banned and social mixing was highly discouraged or criminalised. Health services were put under considerable strain dealing with the effects of Covid-19, with many services limited or cancelled.

There was a noted decrease in stroke patient admissions, particularly patients presenting with minor symptoms and transient ischaemic attack (TIA) in Italy, France, Germany [104], Canada [9] and Norway [105]. In the UK there was a there was a 12% reduction in the number of admissions for Stroke patients between October 1st 2019 and April 30th 2020 compared with the same period in the 3 previous years [106]. It was noted that quality of care did not decrease for admitted Stroke patients, but there is no data on levels of patient access to rehabilitation or long term outcomes. It is postulated that hospital avoidance is the likely cause of the decrease in Stroke patients rather than a decrease in the number of Strokes.

There is little doubt that social distancing has had an effect on post-Stroke rehabilitation. Indeed, in Canada it was noted that access to rehabilitation care has been significantly reduced [9]. Around half of Stroke survivors in the UK have had therapy appointments cancelled or postponed [8]. These cancellations may have occurred due to an increase in protective measures which require more turnaround time between patients, a reduced capacity in rehabilitation centres due to social distancing requirements, and staff absence due to self-isolating or infection. Further to this, 56% of Stroke patients have not felt safe to attend scheduled appointments. This is likely due to fear of becoming infected with Covid-19, especially considering that Stroke patients are at higher risk [107].

The Covid-19 pandemic has created a need for social distancing and a new paradigm of hospital avoidance due to fear of infection. It was noted that tele-rehabilitation is an effective and well accepted method for providing access to therapy [9], though it was considered that this requires family members and care givers to be given additional training. This is corroborated by [10], with the suggestion that Virtual Reality (VR) technologies and existing computer game systems such as the Nintendo Wii could be used to supplement rehabilitation, though it was noted that most home based exercise require oversight from a physiotherapist. It is clear from the literature that there is a need for a low-cost home based rehabilitation devices, such as MyPAM, especially in the current climate of social distancing.

2.10 Literature Review Summary

There has been much development in the use of robotics to provide rehabilitation for Stroke patients in recent years, which has been prompted by the financial and social implications of an aging population where Stroke is becoming increasingly prevalent. The area of upper limb robotic rehabilitation is dominated by two main classes of devices. The first is exoskeleton-based robotics, and the second is end-effector based devices. These can be further split into Class 1 devices, intended for clinical and lab use, and Class 2 devices, intended to be low cost and installed for home use.

Rehabilitation robotics devices follow an established control strategy. The high-level control strategy is designed to follow established physiotherapy principles which attempt to promote neurofunctional plasticity, though this area is not well understood. A trajectory is generated, and the low-level control strategy describes the specific implementation of position, force, Impedance or Admittance control. Many of the devices utilise some form of interaction control for the low-level control strategy. Interaction control can take the form of Admittance control or Impedance control, though some rehabilitation robotics devices use a hybrid form of the two. The MIT-MANUS, for example, uses Impedance control and the iPAM uses Admittance control.

The literature review has identified research gaps. The requirement was identified to develop low cost solutions for rehabilitation robotics [48], which is significant because the leading commercial solution, the InMOTION Arm, costs in excess of $100,000. It is suggested that future research efforts should focus on a “rigorous comparison of control algorithms” to determine whether there is any clinical benefit to the selection of one over another [49]. There is evidence that the use of rehabilitation robotics can positively affect motor control of targeted muscle groups, but consistent with traditional physiotherapy, there is no consensus on an optimal high-level strategy, low-level implementation or rehabilitation schedule. This is because “mechanisms that support or modulate recovery are not yet fully understood” [25].

The RATULS study found that the use of expensive rehabilitation robotics in a clinical setting provided comparable improvement in upper limb impairment compared with normal care alone, but was not cost effective. This supports the further development of low-cost rehabilitation robotics for home use. Further to this, the Covid-19 global pandemic has created a demand for home based tele-rehabilitation devices due to the need for social distancing.

Chapter 3

Trajectory Generation

In this chapter the trajectory generation strategy for MyPAM is documented. In Section 3.1 the limitations of the trajectory generation strategy of the previous iteration of MyPAM are described, and the chapter objectives are laid out. Section 3.2 documents the derivation of two pairs trajectory functions for MyPAM. The first pair of functions generate a trajectory consisting of equidistant points (Minimum Velocity trajectory) and the second pair of functions generate a Minimum Jerk trajectory. Section 3.3 presents an experiment to determine user preference between a trajectory consisting of equidistant points and a Minimum Jerk trajectory and analyses user motion data to determine the trajectory which promotes a superior user response. Section 3.4 describes a novel method for adjusting a trajectory using local points of attraction or repulsion to allow the creation of more engaging rehabilitation games. Section 3.5 documents an experiment used to validate the use of points of attraction and repulsion to adjust a trajectory.

3.1 Introduction

In the previous iteration of MyPAM the trajectory was generated by the games in the high-level controller (HLC). During operation the game generated equidistant intermediate positions between the start position and the final position of each point-to-point linear reaching movement, and passed these one at a time to the low-level controller (LLC). This approach could be considered flawed in the following ways:

1. A trajectory consisting of equidistant points is consistent with traditional robotic motion, but does not reflect natural human notion. Most rehabilitation robots use a smooth, or normative, trajectory generation strategy. The MIT-MANUS is a notable example of this approach. There is no evidence that a normative trajectory which mimics human motion promotes neurofunctional plasticity [49], but it is considered that a more natural motion may feel more comfortable for the user.

2. The games, whilst intended to run at 30Hz, do not operate at 30 Hz reliably as a result of being dependant on the non-deterministic Microsoft Windows Operating System (OS). There may be instances where the game rate will drop, resulting in incorrect intermediate position data being sent to the controller. Such instances include framerate drops due to graphics issues, unexpected API calls, and unexpected CPU loading due to background services and apps installed by windows updates. Furthermore, at 30 Hz the HLC only generates an intermediate setpoint every 33ms which doesn't adequately utilise the capabilities of the LLC and is not fast enough for effective control.

3. There are occasions where no intermediate points are generated and the final target position is sent to the controller as the next target, for example during some game types and during transitions between different games. This leads to a large difference between the current position and the target position, and large motor demands are generated. The effect of these large differences is the generation of aggressive accelerations and potentially dangerous interaction forces between the patient and the robot.

3.1.1 Chapter Objectives

This Chapter aims to fulfil the following objectives:

Objective 3.1 To derive appropriate functions to generate a smooth trajectory and to generate a trajectory consisting of equidistant intermediate points for any point to point motion for MyPAM.

Objective 3.2 To determine whether there is a user preference between a smooth trajectory or a trajectory consisting of equidistant intermediate points.

Objective 3.3 To determine whether there is a notable difference in user performance when presented with a smooth trajectory or a trajectory consisting of equidistant intermediate points.

Objective 3.4 To design and validate a novel method for adjusting the trajectory with points of attraction or points of repulsion to allow more engaging rehabilitation games.

3.2 Trajectory Generation in MyPAM

In this iteration of the MyPAM the trajectory is generated by the LLC on a National Instruments myRIO, as documented in Chapter 8. This decouples the generation of the trajectory from the processing limitations of a non-deterministic OS and places it on hardware capable of accurate timing. Decoupling the trajectory generation mitigates against the problem documented in point 2 in Section 3.1 above. Instead of transferring each individual setpoint in turn to the LLC, only the final target of a reaching task is transferred allowing the LLC to generate the intermediate setpoints in real time using the desired velocity, the initial position and the target position.

As stated there is no evidence that a normative (or smooth) trajectory provides any rehabilitation benefit. Indeed, there is no evidence that there is a patient preference for a more natural feeling trajectory. For this reason two trajectory strategies were derived for comparison to determine the most appropriate strategy for MyPAM. The first strategy consists of equidistant setpoints between the start point and the target point (Minimum Velocity trajectory). The second consists of a smooth Minimum Jerk trajectory between the start point and the target point.

Minimum Velocity was selected for comparison because it is mathematically the simplest discretised trajectory strategy between two points. Minimum Jerk was selected for comparison because it is the discretised trajectory strategy which most accurately emulates human motion [108], where the ratio of peak velocity to average velocity (R) is most closely matched to that of a human as shown by Table 3.2.

Table 3.2: *Velocity Ratios for Different Trajectories [108].*

Trajectory Type	**Peak Velocity/Average Velocity (R)**
Average Human Arm	circa 1.80
Minimum Velocity	1.00
Minimum Acceleration	1.50
Minimum Jerk	1.88
Minimum Snap	2.19
Minimum Crackle	2.46
Minimum Pop	2.71

Mathematically, generating a smooth trajectory translates to minimising the rate of change of an input, where the input corresponds to the order of the system. For example, a 1st order system denotes a kinematic model where velocities may be arbitrarily specified. This is summarised in the Table 3.3.

Table 3.3: *How the Order of a System Corresponds to the System Input.*

Order of the System	**Input to the System**
1st	Velocity, $\dot{x}$
2nd	Acceleration, $\ddot{x}$
3rd	Jerk, $\dddot{x}$
4th	Snap, $x^{(4)}$
5th	Crackle, $x^{(5)}$
6th	Pop, $x^{(6)}$

3.2.1 Generating a Trajectory consisting of Equidistant Points (Minimum Velocity trajectory)

The function for a Minimum Velocity trajectory was derived by minimising the rate of change of velocity. This was achieved by satisfying the Euler-Lagrange equation shown by Equation 3.1.

$$\frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = 0 \tag{3.1}$$

Where:

$$L = (\dot{x})^2$$

$$\frac{\partial L}{\partial x} = 0 \tag{3.2}$$

$$\frac{\partial L}{\partial \dot{x}} = 2\ddot{x} \rightarrow \frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = -2\ddot{x} = 0 \tag{3.3}$$

Thus the function for acceleration is given by Equation 3.4.

$$\ddot{x}(t) = 0 \tag{3.4}$$

By integrating twice it is possible to identify the function for position, shown by Equation 3.6.

$$\dot{x}(t) = \int \ddot{x}(t)dt = \int 0dt = c_1 \tag{3.5}$$

$$x(t) = \int \dot{x}(t)dt = \int c_1 dt = c_1 t + c_0 \tag{3.6}$$

Substituting the boundary conditions shown by Table 3.4.

Table 3.4: *Trajectory Boundary Conditions - The Start and End Position at the Start and End Time.*

	Position
t=0	a
T=t_f	b

$$x(0) = c_0 = a \tag{3.7}$$

$$x(t_f) = c_1 t_f + c_0 = b \tag{3.8}$$

Solving for coefficients c_1 and c_0.

$$\begin{bmatrix} 0 & 1 \\ t_f & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \tag{3.9}$$

Giving the function shown by Equation 3.11 to evaluate a Minimum Velocity trajectory in the x-direction.

$$x(t) = \left(\frac{b-a}{t_f}\right) t + a \tag{3.10}$$

$$\rightarrow x(t) = x_i + \left(x_f - x_i\right) \frac{t}{t_f} \tag{3.11}$$

Where:

x_i = Initial Position

x_f = Target Position

t_f = Target Time

t = Current Time

This may be extended to two dimensions by independently evaluating in the y-direction, giving the pair of equations shown by Equations 3.11 and 3.12

$$\rightarrow y(t) = y_i + \left(y_f - y_i\right) \frac{t}{t_f} \tag{3.12}$$

3.2.2 Generating a Minimum Jerk Trajectory

The function for a Minimum Jerk Trajectory was derived by minimising the rate of change of Jerk. The method is similar to that shown in Section 3.2.1 for deriving the Minimum Velocity trajectory. The full derivation may be found in Appendix A. The Minimum Jerk trajectory is shown by Equations 3.13 and 3.14.

$$x(t) = x_i + (x_f - x_i)\left(6\left(\frac{t}{t_f}\right)^5 - 15\left(\frac{t}{t_f}\right)^4 + 10\left(\frac{t}{t_f}\right)^3\right) \tag{3.13}$$

$$y(t) = y_i + (y_f - y_i)\left(6\left(\frac{t}{t_f}\right)^5 - 15\left(\frac{t}{t_f}\right)^4 + 10\left(\frac{t}{t_f}\right)^3\right) \tag{3.14}$$

Where:

x_i = Initial Position

x_f = Target Position

t_f = Target Time

t = Current Time

3.3 Experimental Determination of Appropriate Trajectory Strategy - A Preliminary Study

3.3.1 Introduction

There is no published evidence to suggest that a Minimum Jerk trajectory improves rehabilitation outcomes compared with a discretised equidistant point to point trajectory (Minimum Velocity trajectory) as used in traditional robots. Despite this a Minimum Jerk trajectory is often selected as the trajectory generation strategy for rehabilitation robots. An experiment was performed to determine whether:

1. There is a noticeable difference between using a Minimum Jerk trajectory or a Minimum Velocity trajectory for a rehabilitation robot from a user perspective.

2. There is a consensus on user preference between using a Minimum Jerk trajectory or a Minimum Velocity trajectory for a rehabilitation robot.

3. There is a difference in user performance between using a Minimum Jerk trajectory or a Minimum Velocity trajectory for a rehabilitation robot.

3.3.2 Methodology

Twenty healthy participants were presented twice with a series of reaching exercises, which they were assisted to achieve by MyPAM. The reaching exercises followed a star pattern used previously during trials to assess patient progress [109], shown by Figure 3.1. The star pattern was used because it encompasses the majority of the robot workspace and fully explores the dynamics of MyPAM, which in general performs more accurately for reaching tasks that demand motion mainly from one joint, as discussed in Chapter 4.

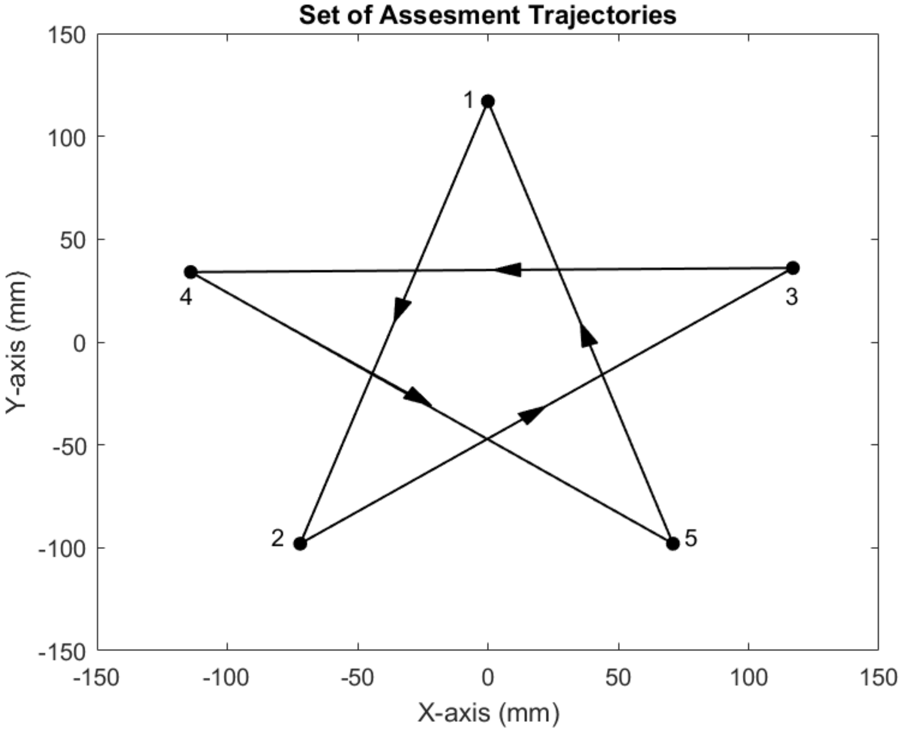

Figure 3.1: *Star Pattern Trajectory Demand Used for Assessment of Participant Performance.*

The two tests were presented in a randomised order and the participants were not informed about which test they were performing until the experiment was over to ensure that there was no bias towards a particular trajectory. One test used the Minimum Jerk Trajectory derived in Section 3.2.2 and one test used the Minimum Velocity trajectory derived in Section 3.2.1, both of which had an average velocity of 50mm/s. Ten participants were presented with the Minimum Jerk trajectory first, and the other ten participants were presented with the Minimum Velocity trajectory first. Each test was performed for one minute each. During each testing phase the next final target was presented to the user as a yellow dot and the next intermediate target was presented as a green dot. Each respective trajectory generated the next intermediate target at a rate of 1000Hz. A new final target was presented to the user when both the end effector of MyPAM and the previous intermediate target were within a 10mm deadzone of the previous final target for one second. When both tests were complete the participants were requested to fill out a questionnaire comprised of the following questions:

1. **Could you feel a difference between the 2 tests?**
2. **Please indicate which trajectory (trajectory 1 or 2) you preferred (if any).**
3. **Please indicate which trajectory (trajectory 1 or 2) felt most natural (if any):**

The motion data for each test were analysed to determine whether which of the two trajectories promoted better user performance. The outcome measures are summarised by Table 3.5.

Table 3.5: *Trajectory Comparison Experiment Outcome Measures.*

Outcome	**Description**
Targets Hit	The number of final targets successfully hit in 1 minute.
Mean Displacement Error	The mean displacement error between the current position of the end effector and the current intermediate target.
Displacement Error Standard Deviation	The standard deviation of displacement error between the current position of the end effector and the current intermediate target.

3.3.3 Results

The paths taken by the twenty participants while tracking the Minimum Jerk trajectory are shown by Figure 3.2. The paths taken by the twenty participants while tracking the Minimum Velocity trajectory are shown by Figure 3.3.

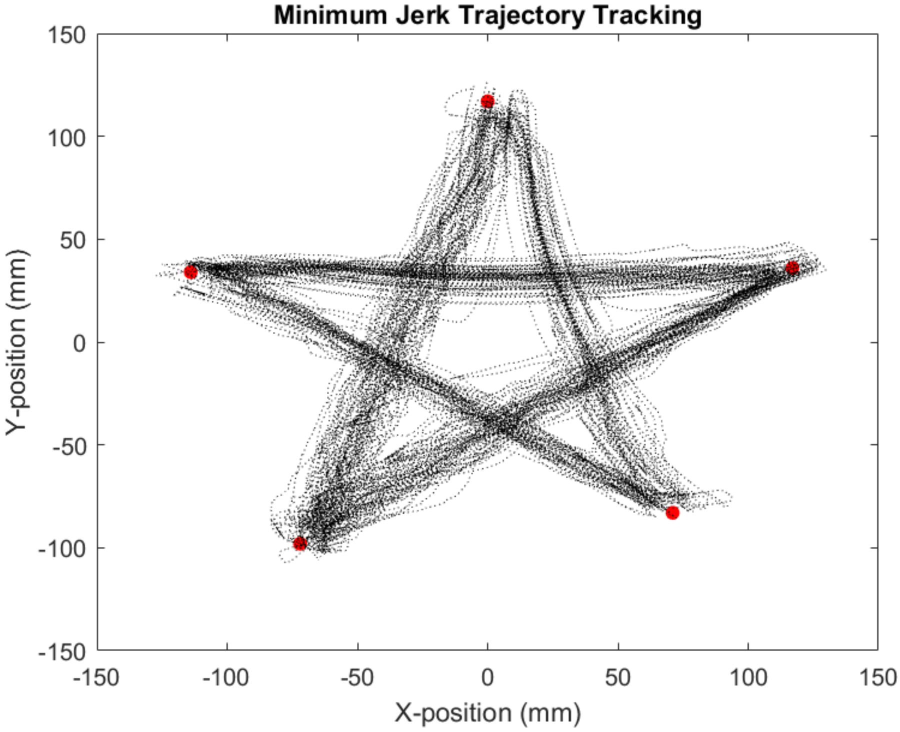

Figure 3.2: *Minimum Jerk Trajectory Tracking Response by Twenty Participants.*

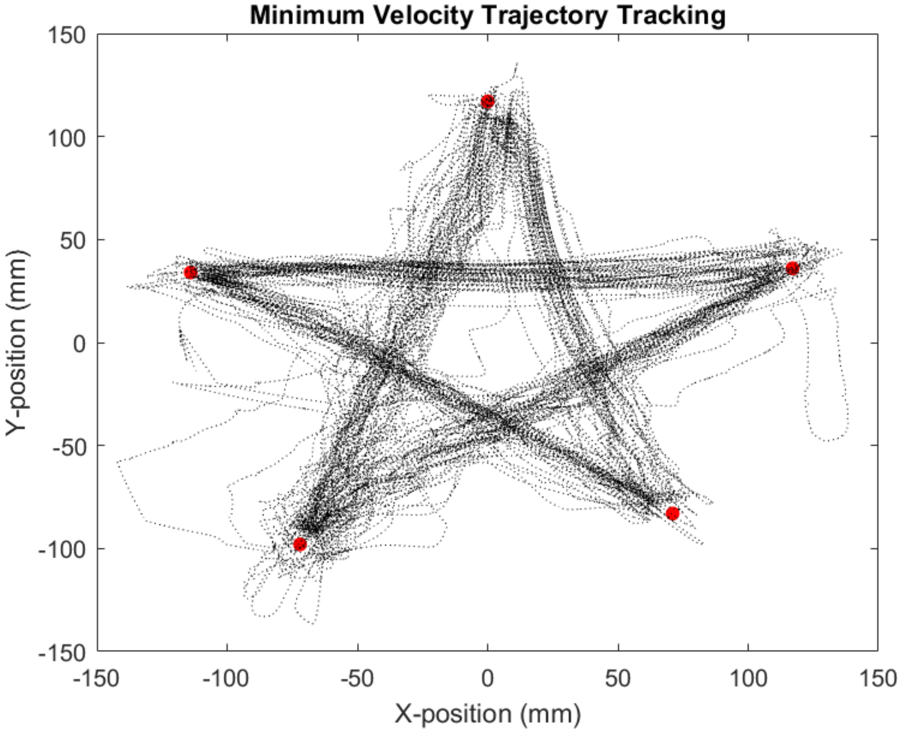

Figure 3.3: *Minimum Velocity Trajectory Tracking Response by Twenty Participants.*

It is observed that in general the participants performed better while tracking the Minimum Jerk trajectory than the equidistant point trajectory, with Figure 3.3 showing divergence from the desired path of a greater magnitude than is observed in Figure 3.2. The results of the questionnaire are shown by Table 3.6.

Table 3.6: *The Results of the Questionnaire Presented to the Twenty Participants*

Question 1 - Can you feel a difference between the two trajectories?		**Question 2 - Did you prefer trajectory 1 or trajectory 2?**		**Question 3 - Did trajectory 1 or trajectory 2 feel most natural?**	
Feel a Difference	90%	Prefer Minimum Jerk	75%	Minimum Jerk Most Natural	65%
Feel No Difference	10%	Prefer Minimum Velocity	15%	Minimum Velocity Most Natural	15%
		No Preference	10%	Neither	20%

The analytical results of the trajectory comparison are shown by Table 3.7.

Table 3.7: *The Analytical Results Comparing the Trajectories of Twenty Participants*

	Minimum Jerk Trajectory	**Minimum Velocity Trajectory**
Mean Displacement Error (mm)	7.7	11.0
Displacement Error Standard Deviation (mm)	5.7	9.7
Mean Targets Hit in 1 Minute	12.6	11.1

3.3.4 Discussion

It is observed on Figure 3.3 that the paths taken by the 20 participants diverge more when tracking the Minimum Velocity trajectory than when tracking the Minimum Jerk trajectory on Figure 3.2. This observation is supported by data shown in Table 3.7, which shows that the mean displacement error was 30% lower and the displacement error Standard Deviation was 41% lower when tracking the Minimum Jerk trajectory compared with the Minimum Velocity trajectory. Further to this it is observed that greater overshoot is present on Figure 3.3 than on Figure 3.2, particularly at targets 2 and 5.

Greater position accuracy and lower overshoot when tracking the Minimum Jerk trajectory may be attributed to the acceleration/deceleration phases at the start and end of each reaching movement. The presence of an acceleration phase allowed the participants to build up to the maximum velocity at the midpoint of each movement, meaning that the ratio of current velocity to distance to next setpoint is less than that when tracking the Minimum Velocity trajectory.

Similarly the presence of a deceleration phase means that participants were able to slow down in the approach to the final target when tracking the Minimum Jerk trajectory, preventing overshoot of the magnitude observed when tracking the Minimum Velocity trajectory. Essentially, the manner in which the Minimum Jerk trajectory discretises the intermediate setpoints of a trajectory acts in a way synonymous with the differential gain of a PID controller by lowering the control demand as the final target is approached.

The mean number of final targets hit during one minute were 12% lower when tracking the Minimum Velocity trajectory than when tracking the Minimum Jerk trajectory. This has occurred because of the conditions required to be met to generate a new final target: a new final target was presented to the user when both the end effector of MyPAM and the previous intermediate target were within a 10mm deadzone of the previous final target for one second. The difference in overshoot observed in Figure 3.3 compared with Figure 3.2 suggests that in general the participants took slightly longer to settle on the final target deadzone whilst tracking the Minimum Velocity trajectory, leading to fewer targets reached in one minute. The difference in mean number of targets hit suggests that the Minimum Jerk trajectory promotes a more accurate user response than the Minimum Velocity trajectory.

The outcome of the questionnaire show that 90% of participants were able to distinguish between the two trajectory strategies. 75% of participants preferred the Minimum Jerk trajectory compared with 15% who preferred the Minimum Velocity trajectory and 10% who had no preference. 65% of participants stated that the Minimum Jerk trajectory felt most natural compared with 15% who stated that the Minimum Velocity trajectory felt most natural and 10% who stated that neither felt most natural. It is interesting to note that not only did the majority of participants prefer the Minimum Jerk trajectory, but the movement data shows that participants performed better when tracking the Minimum Jerk trajectory. This suggests that participants prefer the trajectory which promotes the best performance from them. Indeed, of the five participants who preferred the Minimum Velocity trajectory four of them performed better whilst tracking the Minimum Velocity trajectory and only one of them performed better whilst tracking the Minimum Jerk trajectory.

Unfortunately, a similar conclusion may not be drawn for the participant responses to question three, since there seems to be no distinguishable correlation between participant performance and which trajectory they stated felt most natural. Similarly, there is no distinguishable correlation between trajectory preference and which trajectory felt most natural. Of the five participants who preferred the Minimum Velocity trajectory three of them stated that the Minimum Jerk trajectory felt most natural. This suggests that whilst users tend to prefer the trajectory which promotes the best performance from them, whether or not the trajectory feels natural has little effect on the preference.

There were two participants who were not able to feel a difference between the trajectory strategies, had no preference, and stated that neither trajectory felt most natural. The individual performance data showed that both of these participants hit that same number of targets in the Minimum Jerk trajectory test as the Minimum Velocity trajectory test, with one participant hitting 10 targets each test and the other hitting 11 targets each test. One of these participants performed badly, with a mean displacement error of 12.7mm and 13.3mm for the Minimum Jerk trajectory test and the Minimum Velocity trajectory test respectively. The other participant performed well, with a mean displacement error of 8.18mm and 8.12mm for the Minimum Jerk trajectory test and the Minimum Velocity trajectory test respectively. This suggests that for these two participants a similar performance between the two tests has made it difficult for them to distinguish between the two tests, independent from whether the participant performed well or not.

3.3.5 Conclusion

The Minimum Jerk trajectory promoted a better performance from participants than the Minimum Velocity trajectory. In general, participants preferred the trajectory which promoted the best response from them, and most participants therefore preferred the Minimum Jerk trajectory. In the two cases where a participant was unable to distinguish between the trajectories their performance data was very similar between the two tests. The results of this preliminary study support the use of a Minimum Jerk trajectory as the most appropriate trajectory generation strategy for MyPAM between the two tested trajectories.

It must be noted that this study is unable to answer important questions from a rehabilitation perspective. Since this study tested healthy participants for only a short time, it is difficult to justify an extrapolation of the results to suggest that a Minimum Jerk trajectory is superior to a Minimum Velocity trajectory in terms of the rehabilitation outcome of Stroke patients over an extended period of use, and further studies are necessary to explore this area.

Whilst the majority of the participants stated that they prefer the Minimum Jerk trajectory, there is no obvious correlation between performance and which trajectory strategy felt most natural. Furthermore there is no obvious correlation between preference of trajectory and which trajectory strategy felt most natural. This leads to the conclusion that how a trajectory feels from a user perspective is not important in terms of user performance or preference.

3.4 Adjusting a Discretised Trajectory with Points of Attraction and Repulsion

MyPAM is designed to be used in the home with the patient in isolation, unattended by medical professionals. Such a device is likely to be underutilised without an interesting and engaging suite of computer games which may lead to poor rehabilitation outcomes. It is desirable to be able to adjust a trajectory with points of attraction (Attractors) and points of repulsion (Repulsors) because this allows the creation of more interesting and challenging rehabilitation computer games. More engaging games are likely to increase patient motivation, which has two key benefits:

1. Higher motivation has been linked to better motor learning compared with lower motivation levels for the same amount of exercise [42].

2. Higher motivation ensures greater patient compliance and increased exercise duration and intensity [43].

Work has already been done in the area of adjustment of trajectories, notably Patton created a methodology for using virtual Force Fields to adjust the trajectory demand applied to a user [110]. This adjustment is applied to the end effector depending on its proximity to a force field. It is the opinion of the author, however, that adjustments of this nature should be applied with consideration to the performance of the user. A patient who is performing well may indeed benefit from this increased challenge, but a patient who is not performing well may find the added challenge frustrating or counterproductive. Further to this, in the case of Attraction, there is the potential for a user to become 'stuck' at the Attractive point with the additional risk of promoting instability in the robot control system. For this reason a novel approach has been taken whereby the adjustments are made to the live trajectory as each setpoint is generated in such a way the effects are applied proportionally to the accuracy at which the user is tracking the trajectory. Ie, a user who s tracking the trajectory perfectly will fully notice the effects, but a user who is tracking the trajectory poorly will not notice the effects at all.

The Attractors and Repulsors are applied as an x and y component change to the discretised trajectory, which allows adjustment to a live trajectory from multiple sources such as an external application of Admittance or Impedance. This is shown by Equation 3.15.

$$x(t) = x(t)_u + \delta x_A + \sum \delta x_{ai} + \sum \delta x_{ri} \tag{3.15}$$

Where:

$x(t) = Next\ Intermediate\ Setpoint$

$x(t)_u = Next\ Intermediate\ Setpoint\ From\ Trajectory$

$\delta x_A = Adjustment\ Due\ To\ Admittance$

$\delta x_{ai} = Adjustment\ Due\ To\ Attractor\ i$

$\delta x_{ri} = Adjustment\ Due\ To\ Repulsor\ i$

3.4.1 Forming the Attractive Force

An Attractor is defined here as a point in Cartesian space moves the position of a trajectory setpoint towards it, where the smaller the distance between the Attractor and the next location of the trajectory setpoint the greater the virtual attractive force. Since a relationship between force and distance is being explored, it is useful initially to consider Hooke's Law shown by Equation 3.16.

$$F = k\beta \tag{3.16}$$

Where:

$F = Attractive\ Force$

$k = Virtual\ spring\ constant$

$\beta = Distance\ between\ the\ current\ unadjusted\ setpoint\ and\ the\ Attractor\ position$

Notice that by using Hooke's Law the magnitude of force increases with increasing distance β. This is the opposite of the desired behaviour. An Attractor requires the attractive force to increase with decreasing β, or simply, the closer to the Attractor the greater the attractive force. Adjusting Hooke's Law, as shown by Equation 3.17, produces a more desirable relationship.

$$F = \frac{k}{\beta} \tag{3.17}$$

It can be seen in Figure 3.4 that adjusting Hooke's Law produces the desired relationship where the closer to the Attractor, the greater the Attractive Force.

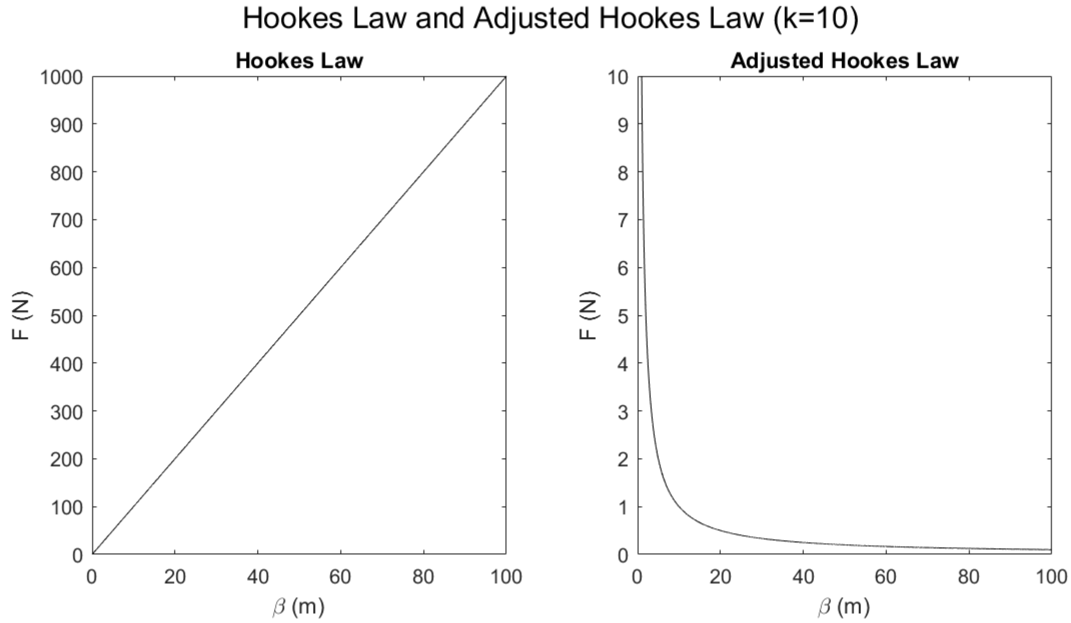

Figure 3.4: ***Left:*** *Hooke's Law.* ***Right:*** *Adjusted Hooke's Law.*

Using adjusted Hooke's law presents issues, since force approaches infinity as β approaches 0, as shown by Equation 3.18. This is a problem because such large virtual forces are likely to introduce instability to the control system in MyPAM.

$$\lim_{\beta \to 0} F = \infty \tag{3.18}$$

The use of an exponential equation produces a desirable relationship between force and distance, but it is necessary to define an exponential relationship where it is possible to define the maximum output force and the effective radius of the Attractor. Thus an exponential equation with a known intercept on the Force-axis is used to create the virtual force, which is shown by shown by Equation 3.19.

$$F = Ae^{\frac{-\beta}{\tau}} \tag{3.19}$$

Where:

$F = Attractive\ Force\ (N)$

$A = Constant\ used\ to\ define\ the\ maximum\ Attractive\ Force\ (N).$

$\beta = Distance\ between\ the\ unadjusted\ setpoint\ and\ the\ Attractor\ position\ (mm).$

$\tau = Constant\ used\ to\ define\ the\ effective\ radius\ of\ the\ Attractor\ (mm).$

Figure 3.5 shows the change in virtual force as β increases using the relationship shown by Equation 3.19. It can be seen that the coefficient A determines the maximum Force, and that as τ increases the force curve flattens.

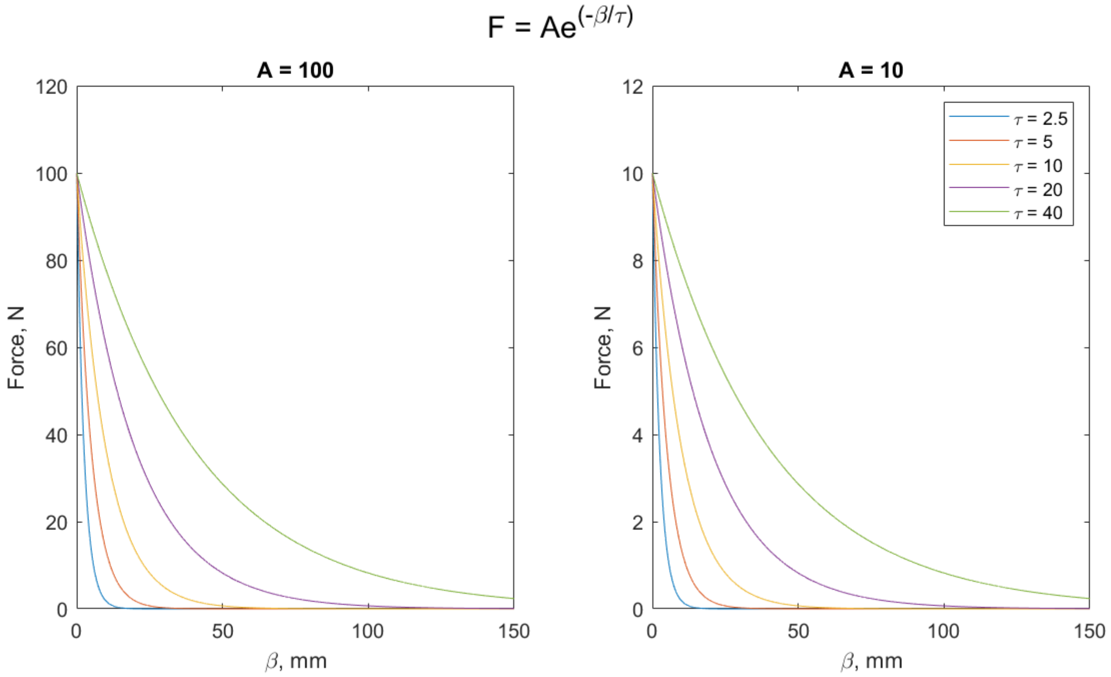

Figure 3.5: *Attractive Force N with Increasing Effective Radius*τ

Coefficient τ is used to define the effective radius of the Attractor. Consider output force to be negligible when it is less than 2% of the maximum force A, which occurs approximately at the distance where $\beta = 4\tau$. The distance-force relationship may now be fully defined where:

1. Coefficient A defines the maximum possible Attractive force, and thus the strength of the Attractor.
2. Coefficient τ defines the effective radius of the Attractor, where $\tau = \frac{r_{desired}}{4}$. Beyond this radius the Force must be set to 0N.

With the Force determined it must be resolved into its x and y components using basic trigonometry as shown by Equations 3.21 and 3.21.

$$F_x = F\cos\theta \tag{3.20}$$

$$F_y = F\sin\theta \tag{3.21}$$

Where:

$F = Attractive\ Force\ (N)$

$F_x = Component\ of\ Force\ in\ the\ X\ Direction$

$F_y = Component\ of\ Force\ in\ the\ Y\ Direction$

$\theta = Angle\ between\ the\ setpoint\ and\ the\ Attractor$

3.4.2 Transforming a Force to a Change in Displacement

The virtual force in the x and y directions must be transformed into a change in displacement in the x and y directions. One method for achieving this is to apply a virtual Admittance filter, which takes the form shown by Equation 3.22.

$$\delta x = \frac{1}{k} F_x \left(1 - e^{-\frac{k}{c}t}\right) \tag{3.22}$$

Where:

$\delta x = Position\ change\ in\ x - direction$

$k = virtual\ spring\ constant.$

$c = virtual\ damping\ constant.$

$t = arbitrary\ time\ constant.$

Examining the exponential component of Equation 3.22 and substituting γ as shown by Equation 3.25.

$$e^{-\frac{k}{c}t} \tag{3.23}$$

$$Let\quad \frac{k}{c}t = \gamma \tag{3.24}$$

$$\rightarrow e^{-\frac{k}{c}t} = e^{-\gamma} \tag{3.25}$$

The exponential component tends to 0 as γ tends to ∞, as shown by Equation 3.26.

$$\lim_{\gamma \to \infty} e^{-\gamma} = 0 \tag{3.26}$$

Substituting Equation 3.26 back into Equation 3.22 it may be observed that the virtual admittance filter simplifies into a simple implementation of Hooke's law as γ tends to ∞, as shown by Equation 3.27.

$$\rightarrow \delta x = \frac{1}{k} F_x (1 - 0) = \delta x = \frac{1}{k} F_x\,(1) = \frac{1}{k} F_x \tag{3.27}$$

The exponential component tends to 1 as γ tends to 0, as shown by Equation 3.28.

$$\lim_{\gamma \to 0} e^{-\gamma} = 1 \tag{3.28}$$

Substituting Equation 3.28 back into Equation 3.22 it may be observed that the virtual admittance filter has no effect, as shown by Equation 3.29.

$$\rightarrow \delta x = \frac{1}{k} F_x (1 - 1) = \delta x = \frac{1}{k} F_x\,(0) = 0 \tag{3.29}$$

It can be seen that using a virtual admittance filter to the change in position due to the attractive force ranges from 0 to a simple implementation of Hooke's Law. There is little benefit to applying a virtual admittance filter, however, since the limited computational resources on the LLC must be preserved for more important control tasks. Furthermore the force/displacement relationship defined by Equation 3.19 already provides a desirable curve with predictable and adjustable parameters. Therefore, the components of force are directly equated to changes in displacement as $F_x = \delta_x$ and $F_y = \delta_y$.

3.4.3 Attractor Point Overshoot

Using Equation 3.19 provides a force curve with predictable and desirable properties. However, when the corresponding changes in displacement in the x and y directions are applied there remains the potential for overshoot if the Attractor is close to the unadjusted trajectory, producing the artifacting shown by Figure 3.6.

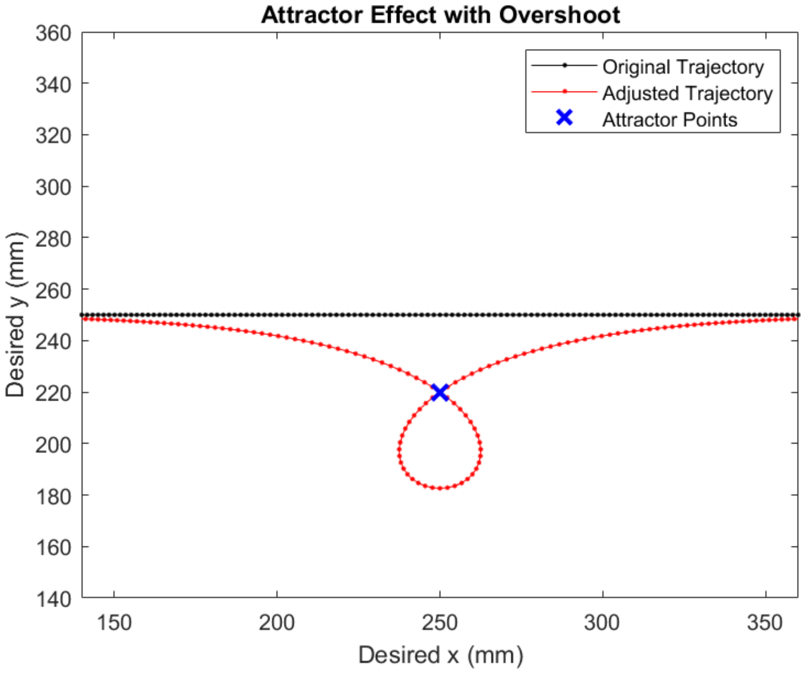

Figure 3.6: *Attractor Point with Artifacting, Which is Caused by Placing an Attractor Too Close to the Unaffected Trajectory.*

This artifacting may introduce instability into the control of the robot due to sudden changes in direction over a short time. When plotting a phasor diagram showing θ_1 (the angle between the current point of the unadjusted trajectory and the Attractor) and θ_2 (the angle between the current point of the adjusted trajectory and the Attractor) it was noticed that $\theta_1 = \theta_2$ except when artifacting was occurring. When artifacting was occurring θ_2 was 180 degrees out of phase from θ_1. This is shown by Figure 3.7.

This allows an adjustment to the algorithm to prevent artifacting. With δ_x and δ_y evaluated, but before they are applied a check is performed to compare θ_1 against θ_2. If $\theta_1 = \theta_2$, the changes in displacement are applied to the trajectory. If $\theta_1 \neq \theta_2$, then δ_x is set equal to the distance between the current location and the Attractor in the x-direction, and δ_y is set equal to the distance between the current location and the Attractor in the y-direction. This ensures that the adjustments to the trajectory will be applied up to the Attractor point but not beyond, preventing overshoot. The result of this is shown by Figure 3.8.

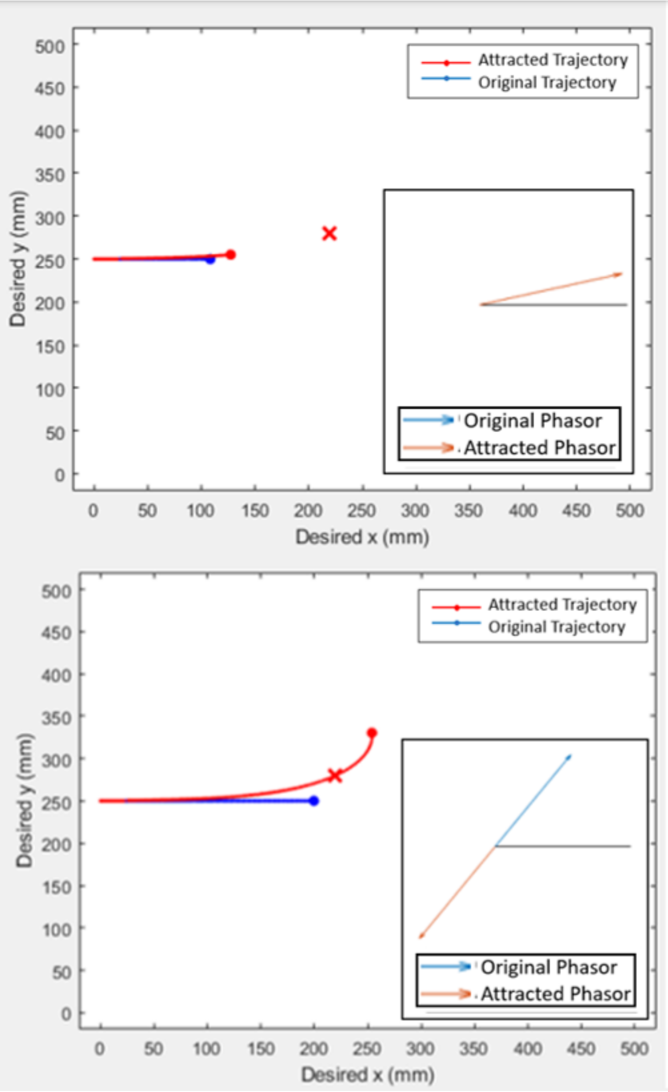

Figure 3.7: *Phasor Diagrams With the Trajectory Adjusted by an Attractor.* ***Above:*** *$\theta_1 = \theta_2$ Occurs When There is No Overshoot.* ***Below:*** *θ_1 and θ_2 180 Degrees Out of Phase, Which Occurs When Overshoot is Present.*

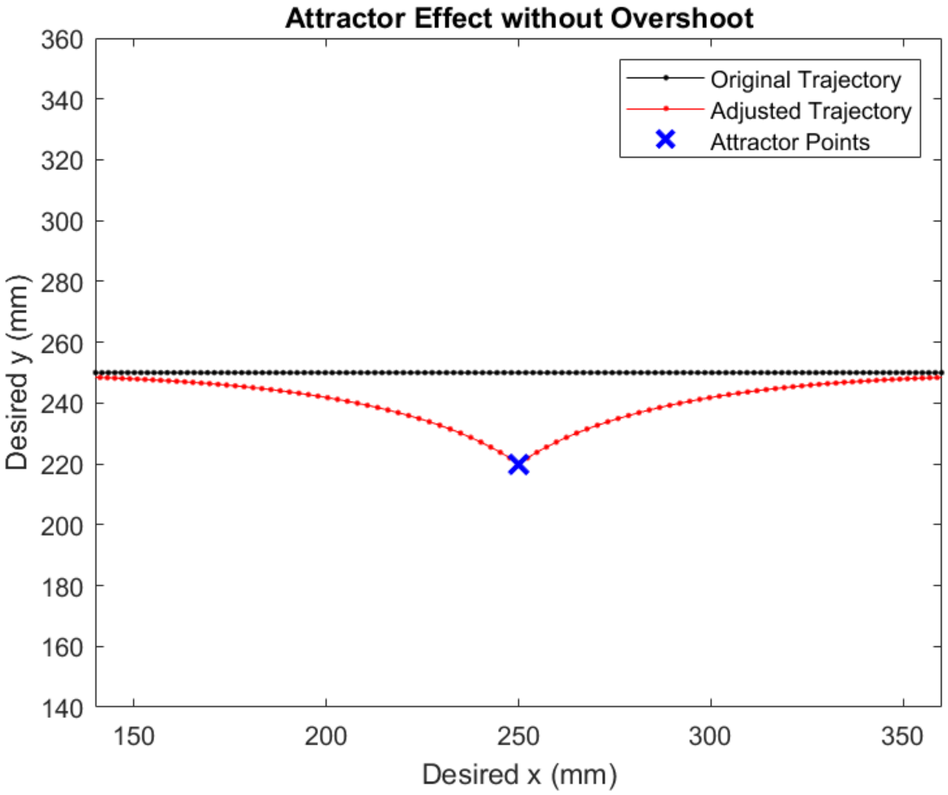

Figure 3.8: *Attractor Point With Artifacting Removed. Note That the Attractor Moves the Path of the Trajectory Towards Itself.*

3.4.4 Repulsors

The implementation of Repulsors requires only a minor adjustment to Equation 3.19, where the output force must be multiplied by -1, shown by Equation 3.30.

$$F = -Ae^{\frac{-\beta}{\tau}} \tag{3.30}$$

Further to this, there is no requirement to prevent artifacting, since in the case of a Repulsor overshoot does not occur. The effects of a Repulsor on the trajectory are shown by Figure 3.9.

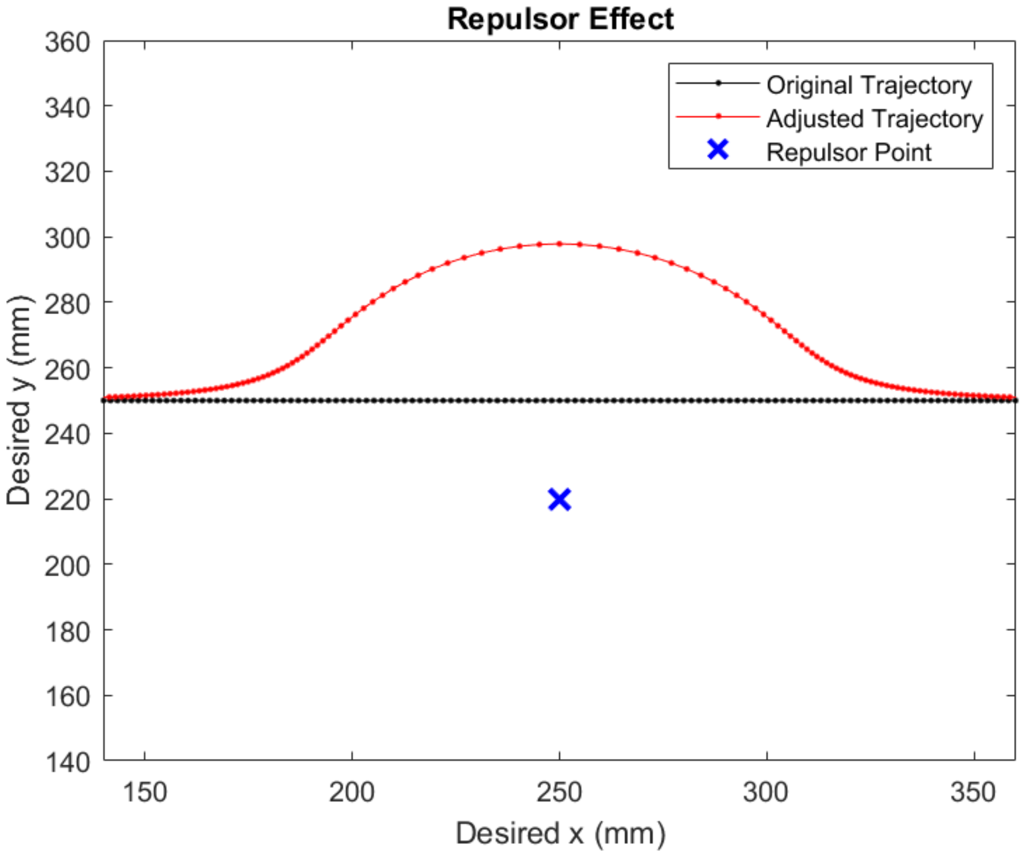

Figure 3.9: *Repulsor Effect. Note That the Repulsor Moves the Path of the Trajectory Away from Itself.*

3.4.5 Multiple Attractors and Repulsors

It is possible to have multiple Attractors and Repulsors affect the trajectory. Each Attractor and Repulsor creates a change in displacement in x and y which are simply summed as shown by Equations 3.31 and 3.32.

$$x_n = x_t + \sum \delta x_i \tag{3.31}$$

$$y_n = y_t + \sum \delta y_i \tag{3.32}$$

Figure 3.10 shows a Minimum Jerk Trajectory being affected by a Repulsor and 2 Attractors. However, it is important to note that when using multiple Attractors and Repulsors, it is possible to create unwanted artifacting seen by Figure 3.11. Whether artifacting occurs when using multiple attractors is a function of a number of variables:

1. Strength of Attractors.
2. Radius of Attractors.
3. Proximity of Attractors to each other.
4. Proximity of Attractors to the unadjusted trajectory.
5. Relative angle between Attractors and the unadjusted trajectory.

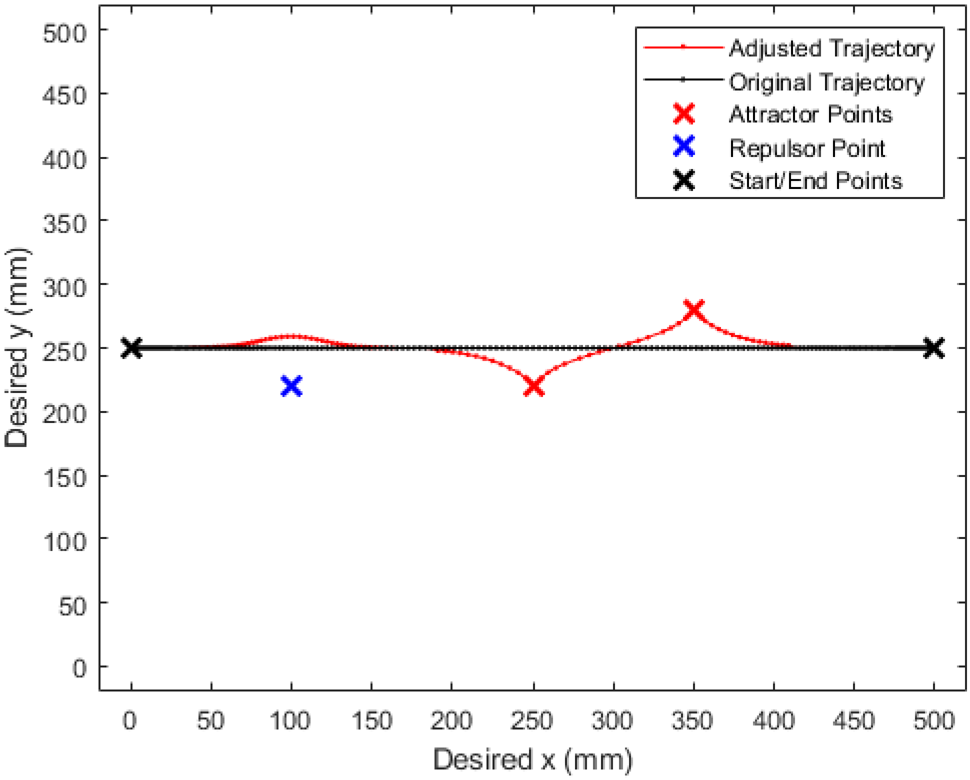

Figure 3.10: *Multiple Attractors Affecting a Trajectory with No Artifacting.*

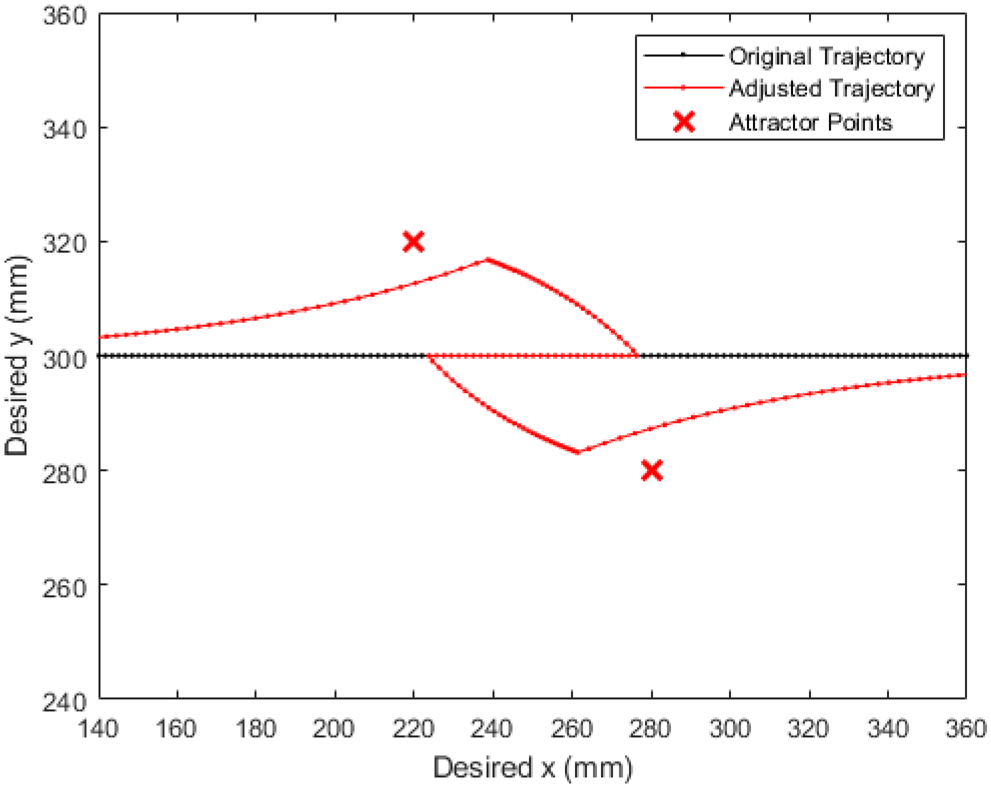

Figure 3.11: *Multiple Attractors Affecting a Trajectory with Artifacting.*

Pragmatically, it is clear that the Attractors should not have either a strength or a radius with the same order of magnitude as the dimensions of the workspace. To prevent artifacting when using multiple Attractors and Repulsors the game designers must meet the following specifications:

1. The maximum strength of an individual Attractor or Repulsor must be the no greater than the magnitude of the radius. **Using Equations 3.19 or 3.30, Coefficient A must be less than or equal to 4τ.**

2. No individual Attractor or Repulsor may be placed perpendicular to the original unaffected trajectory closer in proximity than 30% of its strength. **Distance $\mathbf{d_1}$ (shown by Figure 3.12) must be greater than $0.3\times$ Coefficient A.**

3. Parallel to the original unaffected trajectory, from an angle of $\theta = 0$ degrees to 20 degrees, no individual Attractor or Repulsor may be placed closer in proximity to another Attractor or Repulsor than 30% of the strength of the strongest Attractor or Repulsor. **Distance $\mathbf{d_2}$ (shown by Figure 3.13) must be greater than $0.3\times$ Coefficient A when $\mathbf{0} \leq \theta \leq \mathbf{20}$.**

4. At any angle above 20 degrees, no individual Attractor or Repulsor may be placed closer in proximity to another Attractor or Repulsor than d_2 (shown by Figure 3.13), where d_2 is given by Equation 3.33.

$$\textbf{For } \theta > \mathbf{20}: \quad \mathbf{d_2} \geq \mathbf{A}\left(\mathbf{0.954}\theta^{\mathbf{2}} - \mathbf{0.547}\theta + \mathbf{0.375}\right) \tag{3.33}$$

5. No individual Attractor or Repulsor may be placed within 1 radius proximity to either the start or the end of the unaffected trajectory.

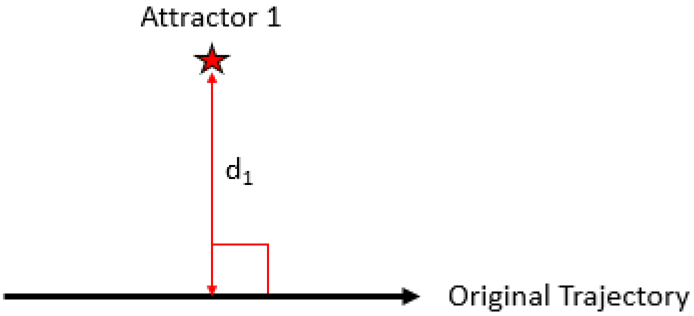

Figure 3.12: *Attractor Specification 2, Showing Distance d_1.*

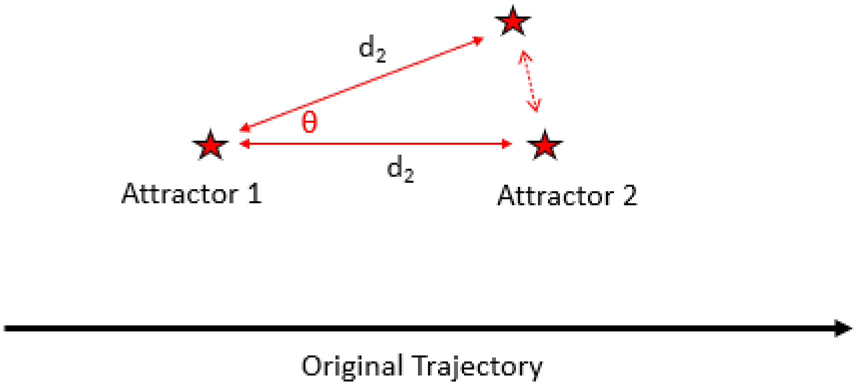

Figure 3.13: *Attractor Specification 3 and 4, Showing Distance* d_2 *and Angle* θ*.*

3.5 Attractor and Repulsor Experimental Validation

3.5.1 Introduction

An experiment was performed to validate the performance of the robot when the trajectory is affected by Attractors and Repulsors. The experiment consisted of three tests:

1. A simple X-direction trajectory with no adjustment to act as the baseline.
2. A simple X-direction trajectory with an Attractor.
3. A simple X-direction trajectory with a Repulsor.

It was necessary to have a baseline for comparison because, as discussed in Section 4.6 in Chapter 4, the effects of inertia and backlash in the joints have a noticeable effect on the trajectory tracking of MyPAM. It is expected that the Attractor and Repulsor will adjust the desired trajectory, but the response of MyPAM to the adjusted trajectory will be imperfect since the response of MyPAM to an unadjusted trajectory is imperfect.

3.5.2 Methodology

A Minimum Jerk Trajectory was generated in the X-direction using the equations defined in Section 3.2.2. The start point of each trajectory was at the default rest position of MyPAM at x=354mm, y=126.2mm. Each trajectory movement consisted of two phases: a movement in the negative x-direction to target 1 at position x=154mm, y=126.2mm followed by a movement in the positive x-direction to target 2 at position x=154mm, y=126.2mm. Three tests were performed with this trajectory. In the first test the trajectory was left unaffected by Attractors or Repulsors to acquire a baseline for comparison.

In the second test the trajectory is tested on the robot whilst affected by an Attractor, with the effect of the Attractor defined by Equation 3.19. The radius of the Attractor is 100mm, the maximum force is 50N and τ = 25mm. In the third test the trajectory is tested on the robot whilst affected by a Repulsor, with the effect of the Repulsor defined by Equation 3.30. The radius of the Repulsor is 100mm, the maximum force is 50N and τ = 25mm. Five repeats were taken for each test and the mean path evaluated. A summary of testing is shown by Table 3.8.

Table 3.8: *A Summary of Tests for Attractors/Repulsors Validation*

	Attractors	Repulsors
Test 1	0	0
Test 2	1	0
Test 3	0	1

3.5.3 Results

The graph given by Figure 3.14 shows the mean response of MyPAM when subjected to a Minimum Jerk trajectory in the X-direction, with the mean X and Y positions against time shown by Figure 3.15.

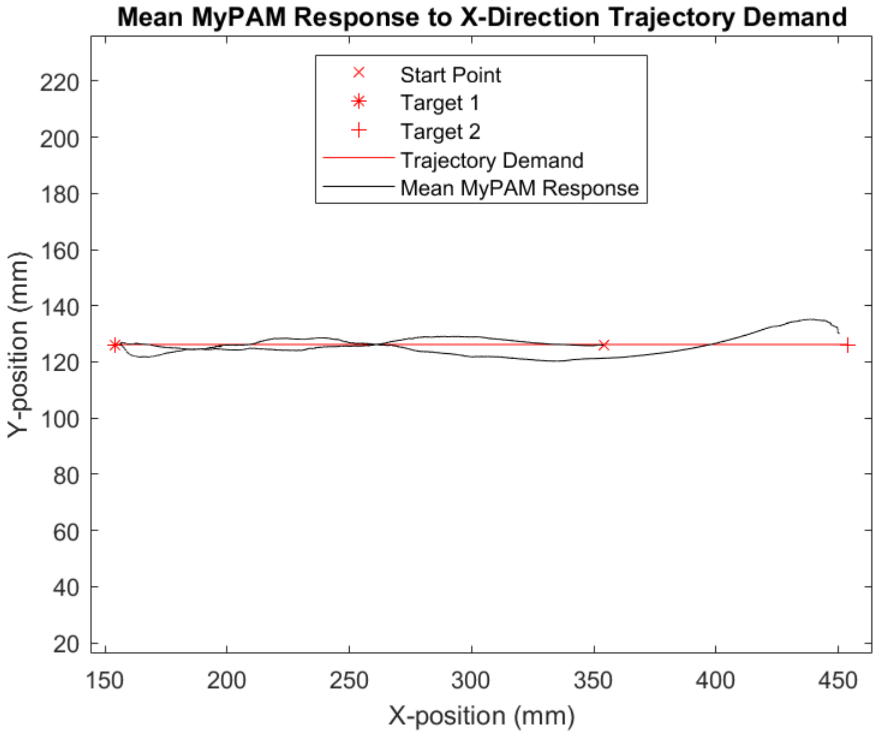

Figure 3.14: *Mean MyPAM X-Direction Minimum Jerk Trajectory Response.*

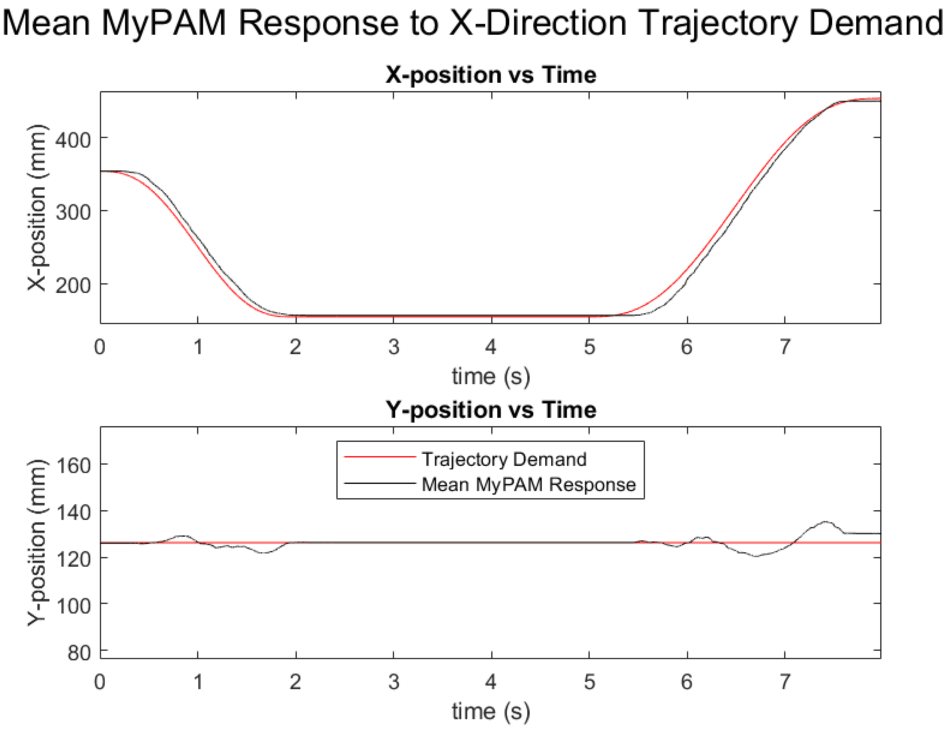

Figure 3.15: *Mean MyPAM X-Direction Minimum Jerk Trajectory Response Against Time.*

The graph given by Figure 3.16 shows the mean response of MyPAM when subjected to a Minimum Jerk trajectory in the X-direction with an Attractor, with the mean X and Y positions against time shown by Figure 3.17.

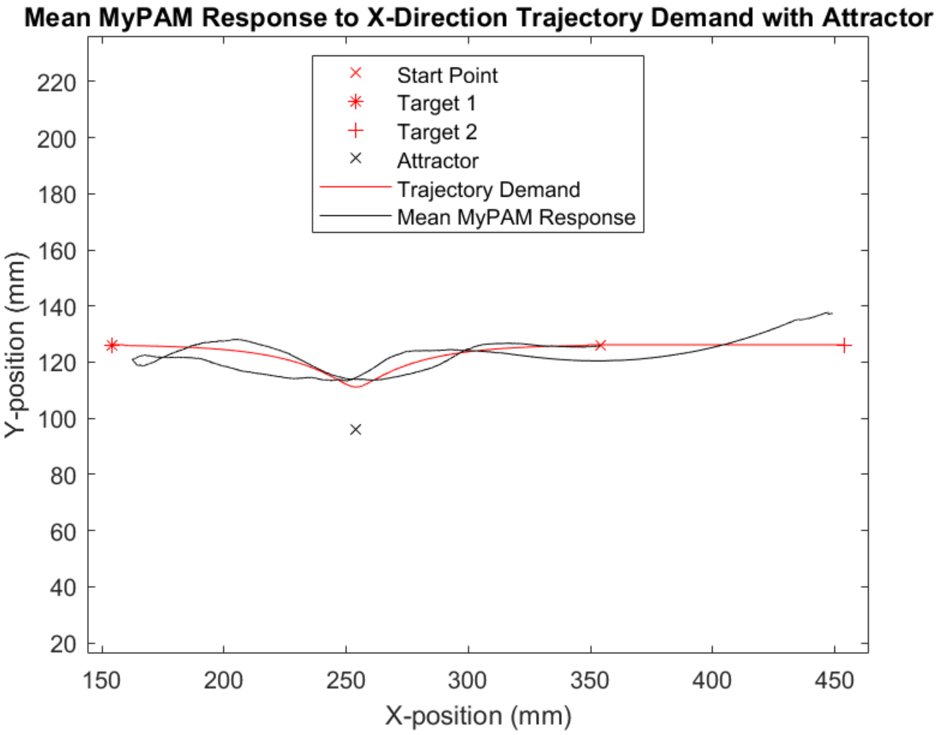

Figure 3.16: *Mean MyPAM X-Direction Minimum Jerk Trajectory with Attractor Response.*

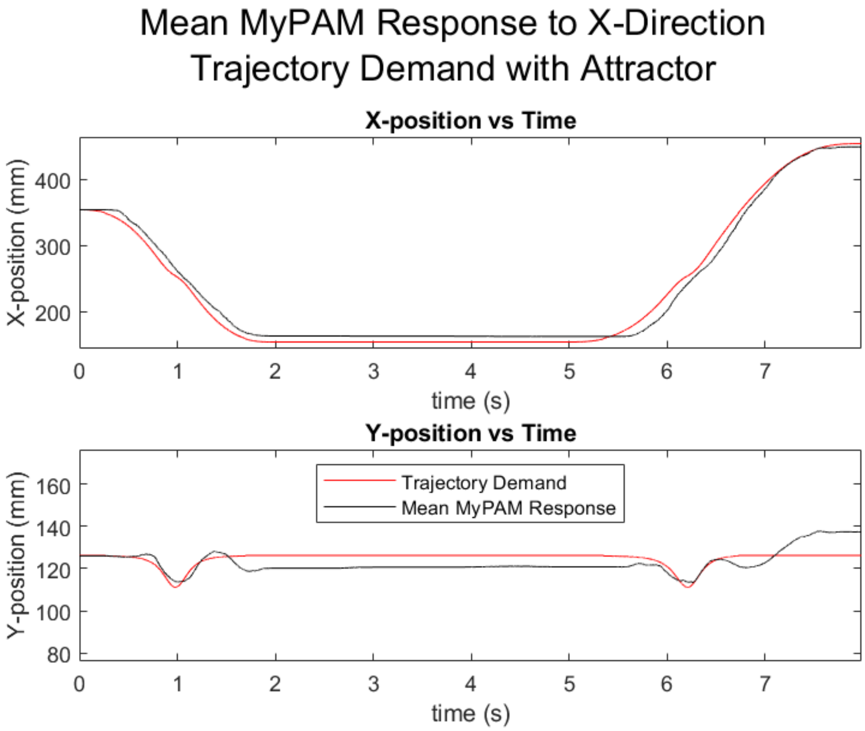

Figure 3.17: *Mean MyPAM X-Direction Minimum Jerk Trajectory with Attractor Response Against Time.*

The graph given by Figure 3.18 shows the mean response of MyPAM when subjected to a Minimum Jerk trajectory in the X-direction with a Repulsor, with the mean X and Y positions against time shown by Figure 3.19.

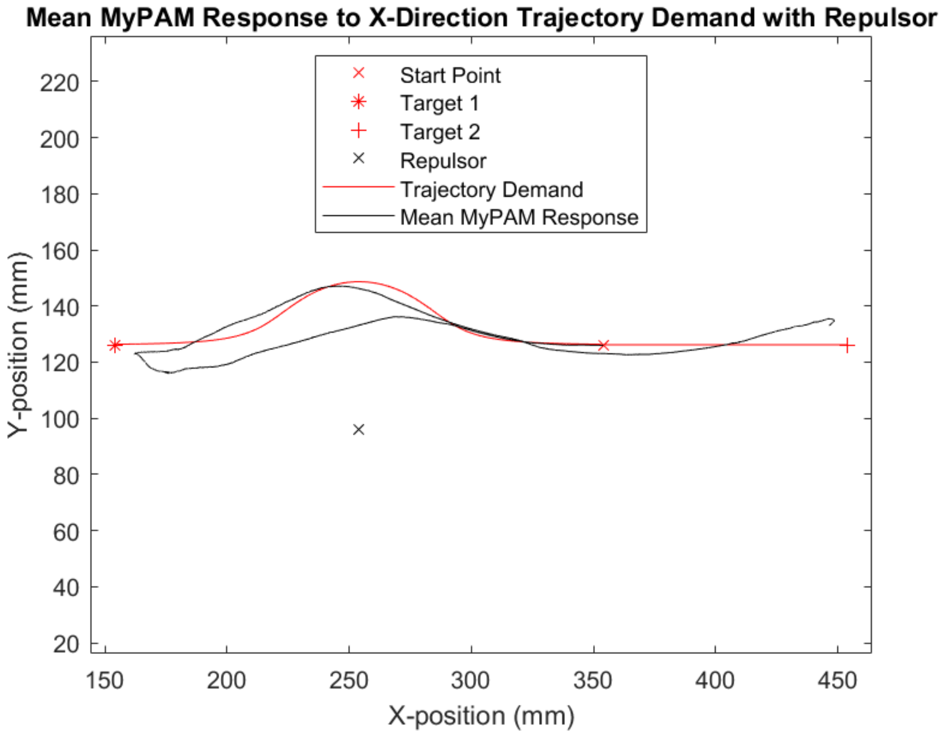

Figure 3.18: *Mean MyPAM X-Direction Minimum Jerk Trajectory with Repulsor Response.*

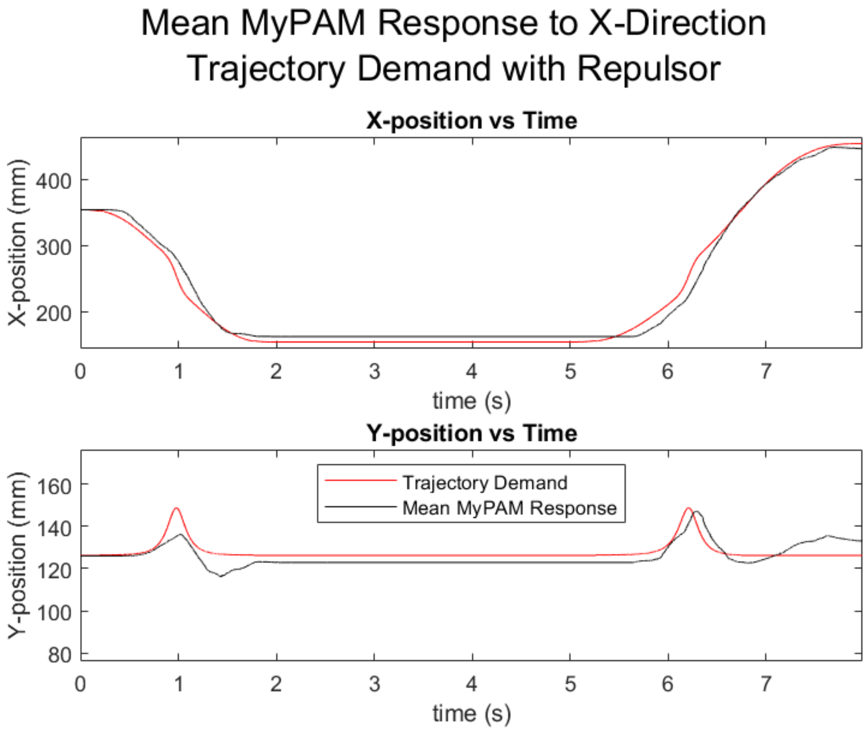

Figure 3.19: *Mean MyPAM X-Direction Minimum Jerk Trajectory with Repulsor Response Against Time.*

3.5.4 Discussion

It is observed on Figures 3.14 and 3.15 for the results of test 1 that MyPAM does not perfectly track the Minimum Jerk trajectory. This occurs because inertial effects are greater for movements of joint 0 than joint 1 and the large amount of backlash in joint 1. This is discussed in detail in Section 4.6 in Chapter 4. It is important to note for later comparison the key events observable through the trajectory tracking due to the effects of the dynamics of MyPAM. In phase 1 of the test 1 the x-position of the end effector of MyPAM closely follows the trajectory demand but with a small amount of lag and shows a small amount of undershoot from target 1 due to the 10mm deadzone around the target.

The y-position of the end effector shows both positive and negative divergence from the desired trajectory of up to 10mm. During phase 2 of test 1 the x-position of the end effector of MyPAM again closely follows the trajectory demand but with a small amount of lag and shows a small amount of undershoot from target 2 due to the 10mm deadzone around the target. The y-position of the end effector shows both positive and negative divergence from the desired trajectory with a smooth sweep of undershoot of the trajectory followed by overshoot of target 2. This smooth sweep is characteristic of the dynamics of the MyPAM and is observable in all x-direction tests in Section 4.6 in Chapter 4.

It is observed on Figures 3.16 and 3.17 that the Attractor has affected the desired trajectory. Further to this, it is observed that MyPAM was able to track the adjusted trajectory. Similar to test 1, in phase 1 of the test 2 the x-position of the end effector of MyPAM closely follows the trajectory demand with a small amount of lag. There is, however, a greater deal of undershoot from target 1 in the x-direction than was observed in test 1. During phase 2 of test 2 the x-position of the end effector of MyPAM again closely follows the trajectory demand but with a small amount of lag, and shows a small amount of undershoot from target 2 due to the 10mm deadzone around the target. The behaviour of MyPAM in the x-position was very similar in phase 2 of test 2 to the behaviour in phase 2 of test 1.

The effect of the Attractor is much more prominent in the y-position, as is expected. During phase 1 of test 2 the y-position of the end-effector follows the desired trajectory closely, but experiences a small amount of overshoot as the desired trajectory returns back to the original path. Following this there is undershoot of target 1. During phase 2 of test 2 the y-position of initially shows poor tracking of the desired trajectory with some lag, before meeting the desired trajectory at the lowest point of the attracted curve and tracking the trajectory closely throughout the rest of the attracted curve. The lag is likely to have arisen due to the backlash in joint 1, which exhibits due to the change in the direction of rotation of the joint. Towards the end of phase 2 of test 2 the y-position of the end effector shows the same characteristic smooth sweep observable in all x-direction tests, with an undershoot of the trajectory followed by an overshoot in the y-position of target 2.

It is observed on Figures 3.18 and 3.19 that the Repulsor has affected the desired trajectory. Further to this, it is observed that MyPAM was able to track the adjusted trajectory. During phase 1 of test 3 the x-position of the end effector of MyPAM tracks the trajectory demand more poorly than was observed in tests 1 and 2. This is likely because the adjusted trajectory moves MyPAM into an area of the workspace from an approach direction where the effects of the dynamics of MyPAM are more pronounced due to the respective demands on joint 0 and joint 1. A similar amount of undershoot from target 1 in the x-direction is present as is observed in test 2. During phase 2 of test 3 the x-position of the end effector of MyPAM again tracks the trajectory demand more poorly than was observed in tests 1 and 2, with more lag than was observed in test 2. There is a similar amount of undershoot from target 2 due to the 10mm deadzone around the target however.

As is the case in test 2, in test 3 the effect of the Repulsor is much more prominent in the y-position. During phase 1 of test 3 the y-position of the end effector of MyPAM tracks the trajectory poorly, demonstrating significant undershoot throughout. Despite this, when comparing test 3 with test 1, the path taken by the end effector is affected by the Repulsor. In phase 2 of test 3 the the y-position of the end effector of MyPAM tracks the trajectory closely and the effects of the Repulsor are pronounced. Towards the end of phase 2 of test 3 the y-position of the end effector shows the same characteristic smooth sweep observable in all x-direction tests, with an undershoot of the trajectory followed by an overshoot in the y-position of target 2.

As predicted the Attractor and the Repulsor have affected the trajectory path in the desired manner, with the Attractor moving the trajectory path towards the Attractor and the Repulsor moving the trajectory path away from the Repulsor. MyPAM tracked the adjusted trajectories satisfactorily, particularly in phase 2 of the tests. The performance of MyPAM was noticeably better when tracking the Attracted trajectory compared with the Repulsed trajectory on phase 1 of the tests because the dynamics of MyPAM are more pronounced in some areas of the workspace depending on the approach direction. Despite this, it is noted that the presence of Attractors or Deflectors have not introduced any instability and when comparing the results of tests 2 and 3 against test 1 the affects of the Attractor and Repulsor are clearly observable.

3.5.5 Conclusion

The use of Attractors and Repulsors affect the trajectory in the desired manner, which allows the game designers to create a suite of rehabilitation games which are more engaging than in previous iterations of MyPAM. This is important because engaging tasks are likely to have positive implications on patient motivation, which in turn leads to improved rehabilitation outcomes. Importantly, MyPAM was capable of tracking the adjusted trajectories, validating the use of Attractors and Repulsors.

It remains the decision of medical professionals how best to use Attractors and Repulsors outside of the perspective of interesting games design, though their use opens up the possibility of implementing alternative high-level Control strategies. It is conceivable that Attractors and Repulsors could be used for Challenge based control methods such as a Resistive strategy used to simulate PNF (as documented in Section 2.4.2), or an Error Amplification Strategy as used in [53].

3.6 Chapter Summary

A pair of functions for generating a trajectory consisting of equidistant intermediate points (Minimum Velocity trajectory) and a pair of functions for generating a Minimum Jerk trajectory were presented in Section 3.2, meeting objective 3.1. Section 3.3 presents an experiment which evaluated the Minimum Velocity and Minimum Jerk trajectories with twenty healthy participants. It was shown that there was a preference for the smoother Minimum Jerk trajectory, meeting objective 3.2, and that the Minimum Jerk trajectory promoted a better performance, meeting objective 3.3. There is no evidence to show that the use of a Minimum Jerk trajectory over a Minimum Velocity trajectory would lead to improved rehabilitation outcomes of Stroke patients, however. But for access to participants and lab space due to Covid 19, a prolonged set of tests using healthy participants would have been performed. The tests would have compared the improvement of kinematic measures of the non-dominant arm whilst using MyPAM, with half the participants guided by a Minimum Jerk trajectory and half guided by a Minimum Velocity trajectory. Section 3.4 presents the derivation of points of Attraction and Deflection, which are used to adjust the path of a discretised trajectory. The use of Attractors and Repulsors is validated in a set of experiments presented in Section 3.5, meeting objective 3.4. The status of the chapter objectives is shown by Table 3.9.

Table 3.9: *Chapter 3 Objectives Status*

Objective	**Description**	**Success?**
3.1	To derive appropriate functions to generate a smooth trajectory and to generate a trajectory consisting of equidistant intermediate points for any point to point motion for MyPAM.	Yes.
3.2	To determine whether there is a user preference between a smooth trajectory or a trajectory consisting of equidistant intermediate points.	Yes.
3.3	To determine whether there is a notable difference in user performance when presented with a smooth trajectory or a trajectory consisting of equidistant intermediate points.	Yes.
3.4	To design and validate a novel method for adjusting the trajectory with points of attraction or points of repulsion to allow more engaging rehabilitation games.	Yes.

Chapter 4

Dynamic Modelling and Analysis

In this chapter detailed kinematic and dynamic models for MyPAM are developed. Section 4.2 details the kinematics of the MyPAM which are used to evaluate the current position of the robot and to obtain the required joint angles to reach a desired position. In Section 4.3 the Jacobian Matrix is derived, which is used to relate the robot joint workspace to the configuration (Cartesian) workspace. The Jacobian Matrix is necessary for impedance control. Section 4.4 documents the development of the dynamic model using the Euler-Lagrange formulation. Section 4.5 details the building of a friction model and deals with integrating the friction model into the complete dynamic model. Section 4.6 documents the creation and validation of a multi-domain dynamic model of MyPAM under different loading conditions using SimScape and SimScape Multibody.

4.1 Chapter Introduction

Modelling is a necessary part of the development of a robot. It is important to understand how a robot will respond to input forces and how the robot will respond to control demands. Several levels of modelling are required to fully understand a robotic system [111], including kinematic and dynamic analyses.

4.1.1 Chapter Objectives

This Chapter aims to fulfil the following objectives:

Objective 4.1 To derive Forward Kinematic and Inverse Kinematic models for MyPAM.

Objective 4.2 To derive the Jacobean Matrix for MyPAM.

Objective 4.3 To obtain a Friction Model for each Joint of MyPAM.

Objective 4.4 To create and validate a dynamic model for MyPAM.

4.2 Kinematics of MyPAM

This section documents the derivation of the Forward and Inverse Kinematics for MyPAM. The Forward Kinematics are used to determine the position of the joints and end-effector given the joint angles. The Inverse Kinematics describe the required joint angles to reach the desired end-effector position.

4.2.1 Forward Kinematics

The Forward Kinematics are derived using Figure 4.1. The joint angles in the MyPAM are calculated from quadrature encoder readings at Joint 0 (the Shoulder) and Joint 1 (the Elbow) of MyPAM.

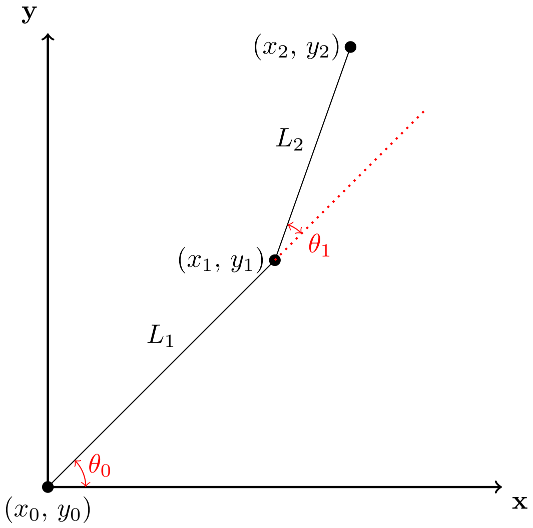

Figure 4.1: *Forward Kinematics for the MyPAM.*

The position of the Shoulder (Joint 0) is located at the origin with coordinates (x_0, y_0), given by Equation 4.1.

$$\boldsymbol{X_0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.1}$$

The position of the Elbow (Joint 1) is located at coordinates (x_1, y_1), given by Equation 4.2.

$$\boldsymbol{X_1} = \begin{bmatrix} L_1 \cos(\theta_0) \\ L_1 \sin(\theta_0) \end{bmatrix} \tag{4.2}$$

The position of the End-Effector (Joint 2) is located at coordinates (x_2, y_2), given by Equation 4.3.

$$\boldsymbol{X_2} = \begin{bmatrix} L_1 \cos(\theta_0) + L_2 \cos(\theta_0 + \theta_1) \\ L_1 \sin(\theta_0) + L_2 \sin(\theta_0 + \theta_1) \end{bmatrix} \tag{4.3}$$

4.2.2 Inverse Kinematics

The Inverse Kinematics are derived using Figure 4.2. The desired position of the End-Effector is the output of the trajectory generation algorithm, which produces a new desired position at a rate of 1kHz.

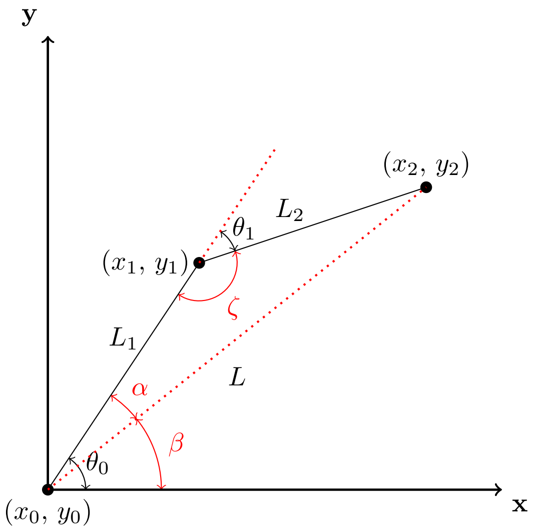

Figure 4.2: *Inverse Kinematics for the MyPAM.*

Deriving the desired angle of Joint 0 (the Shoulder), θ_0:

$$L = \sqrt{{x_2}^2 + {y_2}^2} \tag{4.4}$$

$$\beta = \tan^{\text{-1}}\left(\frac{y_2}{x_2}\right) \tag{4.5}$$

$${L_2}^2 = {L_1}^2 + L^2 - 2L_1 L\cos(\alpha) \tag{4.6}$$

$$\Rightarrow \cos(\alpha) = \frac{{L_1}^2 + L^2 - {L_2}^2}{2L_1 L} \tag{4.7}$$

$$\Rightarrow \alpha = \cos^{\text{-1}}\left(\frac{{L_1}^2 + L^2 - {L_2}^2}{2L_1 L}\right) \tag{4.8}$$

$$\theta_0 = \alpha + \beta \tag{4.9}$$

Thus giving θ_0 as shown by Equation 4.10:

$$\Rightarrow \theta_0 = \cos^{\text{-1}}\left(\frac{{L_1}^2 + L^2 - {L_2}^2}{2L_1 L}\right) + \tan^{\text{-1}}\left(\frac{y_2}{x_2}\right) \tag{4.10}$$

Deriving the desired angle of Joint 1 (the Elbow), θ_1:

$$L^2 = {L_1}^2 + {L_2}^2 - 2L_1 L_2\cos(\zeta) \tag{4.11}$$

$$\Rightarrow \cos(\zeta) = \frac{{L_1}^2 + {L_2}^2 - L^2}{2L_1 L_2} \tag{4.12}$$

$$\Rightarrow \zeta = -\cos^{\text{-1}}\left(\frac{{L_1}^2 + {L_2}^2 - L^2}{2L_1 L_2}\right) \tag{4.13}$$

$$\theta_1 = \pi - \zeta \tag{4.14}$$

Thus giving θ_1 as shown by Equation 4.15:

$$\Rightarrow \theta_1 = \pi + \cos^{\text{-1}}\left(\frac{{L_1}^2 + {L_2}^2 - L^2}{2L_1 L_2}\right) \tag{4.15}$$

4.3 Forming the Jacobian Matrix

The Jacobian Matrix defines the dynamic relationship between two different representations of a system. For MyPAM the position of the robot may be described in Cartesian coordinates or as joint angles. There are two main uses for the Jacobian Matrix for MyPAM. The first use is shown by Equation 4.16, which relates Joint torques to End-Effector Forces. The second use is shown by Equation 4.17, which relates End-Effector velocities to angular velocities of the joints.

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{F} \tag{4.16}$$

$$\dot{\boldsymbol{X}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \tag{4.17}$$

An accepted method for obtaining the Jacobian Matrix of an open chain robotic system is to derive it from Homogeneous Transformation Matrices, which describe the position and orientation of one joint with respect to another.

4.3.1 Assigning Denavit-Hartenberg Frames

Homogeneous Transformation Matrices of an open chain robotic system are obtained by assigning Denavit-Hartenberg frames. Figure 4.3 shows the frames assigned to the MyPAM:

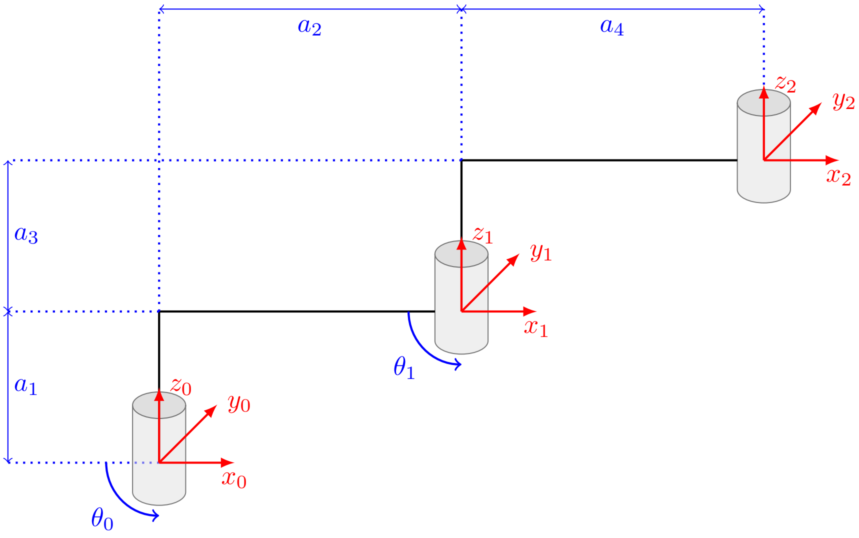

Figure 4.3: *Denavit-Hartenberg Frames Assigned to MyPAM.*

The frames shown in Figure 4.3 allow the population of the Denavit-Hartenberg parameter table, shown in Table 4.2.

Table 4.2: *The Denavit-Hartenberg Parameter Table for the MyPAM.*

Link	$\boldsymbol{\theta}$	$\boldsymbol{\alpha}$	$\boldsymbol{r}$	$\boldsymbol{d}$
1	θ_0	0	a_2	a_1
2	θ_1	0	a_4	a_3

The Homogeneous Transformation Matrices are formed from Equation 4.18 using the Denavit-Hartenberg parameters given by Table 4.2.

$$\boldsymbol{H_n^{n-1}} = \begin{bmatrix} C\theta_n & -S\theta_n C\alpha_n & S\theta_n S\alpha_n & r_n C\theta_n \\ S\theta_n & C\theta_n C\alpha_n & -C\theta_n S\alpha_n & r_n S\theta_n \\ 0 & S\alpha_n & C\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.18}$$

4.3.2 Forming the Homogeneous Transformation Matrices

The Homogeneous Transformation Matrix for transformation from Frame 0 to Frame 1 is shown by Equation 4.19.

$$\boldsymbol{H_1^0} = \begin{bmatrix} C\theta_0 & -S\theta_0 & 0 & a_2C\theta_0 \\ S\theta_0 & C\theta_0 & 0 & a_2S\theta_0 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.19}$$

The Homogeneous Transformation Matrix for transformation from Frame 1 to Frame 2 is shown by Equation 4.20.

$$\boldsymbol{H_2^1} = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & a_4C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & a_4S\theta_1 \\ 0 & 0 & 1 & a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.20}$$

The Homogeneous Transformation Matrix for transformation from the base frame (Frame 0) to the end-effector frame (Frame 2) is obtained by multiplying the previous transformation matrices, producing Equation 4.22.

$$\boldsymbol{H_2^0} = \boldsymbol{H_1^0}\boldsymbol{H_2^1} \tag{4.21}$$

$$\boldsymbol{H_2^0} = \begin{bmatrix} C\theta_0C\theta_1 - S\theta_0S\theta_1 & -C\theta_0S\theta_1 - C\theta_0S\theta_1 & 0 & a_2C\theta_0 + a_4(C\theta_0C\theta_1 - S\theta_0S\theta_1) \\ C\theta_0S\theta_1 - C\theta_0S\theta_1 & C\theta_0C\theta_1 - S\theta_0S\theta_1 & 0 & a_1S\theta_0 + a_4(C\theta_0S\theta_1 - C\theta_1S\theta_0) \\ 0 & 0 & 1 & a_1 + a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.22}$$

The Trigonometric Identities shown by Equation 4.23 and Equation 4.24 may be used to simplify the overall Homogeneous Transformation Matrix, finally producing the complete Homogeneous Transformation Matrix shown by Equation 4.25.

$$\sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta) = \sin(\alpha + \beta) \tag{4.23}$$

$$\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) = \cos(\alpha + \beta) \tag{4.24}$$

$$\boldsymbol{H_2^0} = \begin{bmatrix} C(\theta_0 + \theta_1) & -S(\theta_0 + \theta_1) & 0 & a_2C\theta_0 + a_4C(\theta_0 + \theta_1) \\ S(\theta_0 + \theta_1) & C(\theta_0 + \theta_1) & 0 & a_2S\theta_0 + a_4S(\theta_0 + \theta_1) \\ 0 & 0 & 1 & a_1 + a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.25}$$

4.3.3 Using the Homogeneous Transformation Matrices

The Homogeneous Transformation Matrices contain information about the rotational transformation and the displacement transformation, as shown by Equation 4.26:

$$\boldsymbol{H_n^{n-1}} = \begin{bmatrix} \boldsymbol{R_n^{n-1}} & \boldsymbol{d_n^{n-1}} \\ 0 & 1 \end{bmatrix} \tag{4.26}$$

The rotation transformation matrix for Joint 0 to Joint 1 is obtained by analysing Equation 4.19, shown by Equation 4.27.

$$\boldsymbol{R_1^0} = \begin{bmatrix} C\theta_0 & -S\theta_0 & 0 \\ S\theta_0 & C\theta_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.27}$$

The displacement transformation matrix for Joint 0 to Joint 1 is obtained by analysing Equation 4.19, shown by Equation 4.28.

$$\boldsymbol{d^0_1} = \begin{bmatrix} a_2 C\theta_0 \\ a_2 S\theta_0 \\ a_1 \end{bmatrix} \tag{4.28}$$

The displacement transformation matrix for Joint 0 to Joint 2 is obtained by analysing Equation 4.25, shown by Equation 4.29.

$$\boldsymbol{d^0_2} = \begin{bmatrix} a_2 C\theta_0 + a_4 C(\theta_0+\theta_1) \\ a_2 S\theta_0 + a_4 S(\theta_0+\theta_1) \\ a_1 + a_3 \end{bmatrix} \tag{4.29}$$

The Jacobian Matrix is formed using the information shown in Table 4.3. Since the MyPAM contains only revolute joints only the second column of the table is applicable.

Table 4.3: *Forming the Jacobian Matrix from Homogeneous Transformation Matrices.*

		Prismatic Joints	**Revolute Joints**
Linear Velocity Components	$\dot{x}$ $\dot{y}$ $\dot{z}$	$\boldsymbol{R^0_{i-1}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\boldsymbol{R^0_{i-1}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\boldsymbol{d^0_n} - \boldsymbol{d^0_{i-1}}\right)$
Rotational Velocity Components	ω_x ω_y ω_z	$\boldsymbol{R^0_0} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\boldsymbol{R^0_{i-1}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

The Jacobian Matrix for MyPAM is formulated according to 4.30:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{R^0_0} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\boldsymbol{d^0_2} - \boldsymbol{d^0_0}\right) & \boldsymbol{R^0_1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\boldsymbol{d^0_2} - \boldsymbol{d^0_1}\right) \\ \boldsymbol{R^0_0} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \boldsymbol{R^0_1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \tag{4.30}$$

The value at row 2, column 1 of the Jacobian Matrix is given by Equation 4.31.

$$\boldsymbol{J}_{2,1} = \boldsymbol{R^0_0} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{I} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.31}$$

The value at row 1, column 1 of the Jacobian Matrix is given by Equation 4.33.

$$\left(\boldsymbol{d^0_2} - \boldsymbol{d^0_0}\right) = \begin{bmatrix} a_2 C\theta_0 + a_4 C(\theta_0+\theta_1) \\ a_2 S\theta_0 + a_4 S(\theta_0+\theta_1) \\ a_1 + a_3 \end{bmatrix} - 0 = \begin{bmatrix} a_2 C\theta_0 + a_4 C(\theta_0+\theta_1) \\ a_2 S\theta_0 + a_4 S(\theta_0+\theta_1) \\ a_1 + a_3 \end{bmatrix} \tag{4.32}$$

$$\boldsymbol{J}_{1,1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} a_2 C\theta_0 + a_4 C(\theta_0+\theta_1) \\ a_2 S\theta_0 + a_4 S(\theta_0+\theta_1) \\ a_1 + a_3 \end{bmatrix} = \begin{bmatrix} -a_2 S\theta_0 - a_4 S(\theta_0+\theta_1) \\ a_2 C\theta_0 + a_4 C(\theta_0+\theta_1) \\ 0 \end{bmatrix} \tag{4.33}$$

The value at row 2, column 2 of the Jacobian Matrix is given by Equation 4.34.

$$\boldsymbol{J}_{2,2} = \boldsymbol{R}_{\mathbf{1}}^{\mathbf{0}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} C\theta_0 & -S\theta_0 & 0 \\ S\theta_0 & C\theta_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.34}$$

The value at row 1, column 2 of the Jacobian Matrix is given by Equation 4.36.

$$\left(\boldsymbol{d}_{\mathbf{2}}^{\mathbf{0}} - \boldsymbol{d}_{\mathbf{1}}^{\mathbf{0}}\right) = \begin{bmatrix} a_2C\theta_0 + a_4C(\theta_0+\theta_1) \\ a_2S\theta_0 + a_4S(\theta_0+\theta_1) \\ a_1 + a_3 \end{bmatrix} - \begin{bmatrix} a_2C\theta_0 \\ a_2S\theta_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_4C(\theta_0+\theta_1) \\ a_4S(\theta_0+\theta_1) \\ a_3 \end{bmatrix} \tag{4.35}$$

$$\boldsymbol{J}_{2,1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} a_4C(\theta_0+\theta_1) \\ a_4S(\theta_0+\theta_1) \\ a_3 \end{bmatrix} = \begin{bmatrix} -a_4S(\theta_0+\theta_1) \\ a_4C(\theta_0+\theta_1) \\ 0 \end{bmatrix} \tag{4.36}$$

Finally producing the complete Jacobian Matrix for MyPAM, shown by Equation 4.37.

$$\boldsymbol{J} = \begin{bmatrix} -a_2S\theta_0 - a_4S(\theta_0+\theta_1) & -a_4S(\theta_0+\theta_1) \\ a_2C\theta_0 + a_4C(\theta_0+\theta_1) & a_4C(\theta_0+\theta_1) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{4.37}$$

Equation 4.37 may be tidied up into Equation 4.38.

$$\boldsymbol{J} = \begin{bmatrix} -L_1\sin(\theta_0) - L_2\sin(\theta_0+\theta_1) & -L_2\sin(\theta_0+\theta_1) \\ L_1\cos\theta_0 + L_2\cos(\theta_0+\theta_1) & L_2\cos(\theta_0+\theta_1) \end{bmatrix} \tag{4.38}$$

4.4 Preliminary Dynamic Modelling

A preliminary dynamic model of MyPAM was developed using the Euler-Lagrange formulation, which is shown by Equation 4.39.

$$\mathcal{L}(\theta,\dot{\theta}) = k(\theta,\dot{\theta}) - p(\theta) \tag{4.39}$$

Since the robot is planar the change in potential energy for each joint is zero, thus the Lagrangian may be simplified as shown by Equation 4.40.

$$\mathcal{L}(\theta,\dot{\theta}) = k(\theta,\dot{\theta}) \tag{4.40}$$

The 37 main components of MyPAM were identified, and the mass of each was calculated using a CAD model. The centre of mass of each component was evaluated relative to the relevant rotational axis from the CAD model. The inertia of simplified cylindrical representations of each component was calculated about the relevant rotational axis. The kinetic energy (both translational and rotational) of each component was evaluated and summed using Equation 4.41.

$$k = \sum_{i=1}^{i=n} \left(\frac{1}{2} m_i v_i^2 + \frac{1}{2} I_i \omega_i^2 \right) \tag{4.41}$$

The Lagrangian for MyPAM consists of 55 terms and may be found in Appendix B. The equations of motion were found using Equation 4.42.

$$\tau_i = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} - \frac{\partial \mathcal{L}}{\partial \theta_i}, \;\; i = 1,2 \tag{4.42}$$

The unprocessed equations giving τ_1 and τ_2 may be found in the Appendix B. In the case of serial chain robotics such as MyPAM the equations of motion may be gathered into the form shown by Equation 4.43.

$$\boldsymbol{\tau} = \boldsymbol{M}(\theta)\ddot{\boldsymbol{\theta}} + \boldsymbol{c}(\dot{\theta},\theta) + \boldsymbol{g}(\theta) + \boldsymbol{\tau_f}(\dot{\theta}) \tag{4.43}$$

Where, in Joint space:
$\boldsymbol{\tau}$ = Motor Torque Vector.
$\boldsymbol{M}(\theta)$ = Mass/Inertia Matrix.
$\ddot{\boldsymbol{\theta}}$ = Joint Angle Acceleration
$\boldsymbol{c}(\dot{\theta},\theta)$ = Coriolis and Centrifugal Effects.
$\boldsymbol{g}(\theta)$ = Effects of Gravity (in joint space)
$\boldsymbol{\tau_f}(\dot{\theta})$ = Torque lost to friction effects

MyPAM is planar so there are no gravitational terms. Further to this, the Centrifugal and Coriolis effects may be ignored because the robot moves relatively slowly. However, when moving slowly the effects of friction and stiction are more significant and an accurate friction model is necessary [112], which is dealt with in Section 4.5. For the moment ignoring Torque lost to friction the simplified equation of motion is shown by Equation 4.44.

$$\boldsymbol{\tau} = \boldsymbol{M}(\theta)\ddot{\boldsymbol{\theta}} \tag{4.44}$$

The processed equations giving the simplified torques τ_1 and τ_2 corresponding to Equation 4.44 may be found in Appendix B. The mass/inertia matrix $\boldsymbol{M}(\theta)$ takes the form given by Equation 4.45.

$$\boldsymbol{M}(\theta) = \begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{bmatrix} \tag{4.45}$$

Where:

$$\begin{aligned}
&M_{1,1} = \Big(0.128 + 1.20{L_1}^2 + 0.0693\cos(\theta_0)^2 + 0.0693\cos(\theta_1) + 0.276{L_2}^2\cos(\theta_0)^2 \\
&+0.276{L_2}^2\cos(\theta_1)^2 \text{ -}0.139\cos(\theta_1)^2\cos(\theta_1)^2 + 0.0726L_1\cos(\theta_1) \text{ -}0.0726L_1\sin(\theta_1) \\
&\text{-}0.0726L_1\cos(\theta_1)\cos(2\theta_0) \text{ -}0.0726L_1\sin(2\theta_0)\cos(\theta_1) \text{ -}0.0726L_1\sin(\theta_1)\cos(2\theta_0) \\
&+0.0726L_1\sin(2\theta_0)\sin(\theta_1) + 0.138L_1L_2\cos(\theta_1) \text{ -}0.138L_1L_2\sin(\theta_1) \\
&\text{-}0.552{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2 + 0.139\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1) \text{ -}0.138L_1L_2\cos(\theta_1)\cos(2\theta_0) \\
&\text{-}0.138L_1L_2\sin(2\theta_0)\cos(\theta_1) \text{ -}0.138L_1L_2\sin(\theta_1)\cos(2\theta_0) + 0.138L_1L_2\sin(2\theta_0)\sin(\theta_1) \\
&+0.552{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\Big)
\end{aligned}$$

$$\begin{aligned}
&M_{1,2}=\Big(0.00969 + 0.0693\cos(\theta_0)^2 + 0.0693\cos(\theta_1)^2 + 0.276{L_2}^2\cos(\theta_0)^2 + 0.276{L_2}^2\cos(\theta_1)^2 \\
&\text{-}0.139\cos(\theta_0)^2\cos(\theta_1)^2 + 0.0363L_1\cos(\theta_1) \text{ -}0.0363L_1\sin(\theta_1) \text{ -}0.0363L_1\cos(\theta_1)\cos(2\theta_0) \\
&\text{-}0.0363L_1\sin(2\theta_0)\cos(\theta_1) \text{ -}0.0363L_1\sin(\theta_1)\cos(2\theta_0) + 0.0363L_1\sin(2\theta_0)\sin(\theta_1) \\
&+0.0689L_1L_2\cos(\theta_1) \text{ -}0.0689L_1L_2\sin(\theta_1) \text{ -}0.552{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2 \\
&+0.139\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1) \text{ -}0.0689L_1L_2\cos(\theta_1)\cos(2\theta_0) \\
&\text{-}0.0689L_1L_2\sin(2\theta_0)\cos(\theta_1) \text{ -}0.0689L_1L_2\sin(\theta_1)\cos(2\theta_0) + 0.0689L_1L_2\sin(2\theta_0)\sin(\theta_1) \\
&+0.552{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\Big)
\end{aligned}$$

$$\begin{aligned}
&M_{2,1}=\Big(0.00969 + 0.0693\cos(\theta_0)^2 + 0.0693\cos(\theta_1)^2 + 0.276{L_2}^2\cos(\theta_0)^2 + 0.276{L_2}^2\cos(\theta_1)^2 \\
&\text{-}0.139\cos(\theta_0)^2\cos(\theta_1)^2 + 0.0363L_1\cos(\theta_1) \text{ -}0.0363L_1\sin(\theta_1) \text{ -}0.0363L_1\cos(\theta_1)\cos(2\theta_0) \\
&\text{-}0.0363L_1\sin(2\theta_0)\cos(\theta_1) \text{ -}0.0363L_1\sin(\theta_1)\cos(2\theta_0) + 0.0363L_1\sin(2\theta_0)\sin(\theta_1) \\
&+0.0689L_1L_2\cos(\theta_1) \text{ -}0.0689L_1L_2\sin(\theta_1) \text{ -}0.552{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2 \\
&+0.139\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1) \text{ -}0.0689L_1L_2\cos(\theta_1)\cos(2\theta_0) \\
&\text{-}0.0689L_1L_2\sin(2\theta_0)\cos(\theta_1) \text{ -}0.0689L_1L_2\sin(\theta_1)\cos(2\theta_0) + 0.0689L_1L_2\sin(2\theta_0)\sin(\theta_1) \\
&+0.552{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\Big)
\end{aligned}$$

$$\begin{aligned}
&M_{2,2}=\Big(0.00969 + 0.0693\cos(\theta_0)^2 + 0.0693\cos(\theta_1)^2 + 0.276{L_2}^2\cos(\theta_0)^2 + 0.276{L_2}^2\cos(\theta_1)^2 \\
&\text{-}0.139\cos(\theta_0)^2\cos(\theta_1)^2 \text{ -}0.552{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2 + 0.139\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1) \\
&+0.552{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\Big)
\end{aligned}$$

Note the symmetry across the Mass/Inertia matrix where $M_{1,2} = M_{2,1}$.

This mathematical modelling approach has a number of drawbacks:

1. It was difficult to include sufficient components in the model to simulate MyPAM with acceptable fidelity.
2. Programming the equations of motion manually, using the symbolic maths toolbox in MATLAB for example, is a time consuming and error prone process. This is exacerbated as the number of terms increase as more components are included.
3. The model is only single domain with torques modelled as ideal torques rather than the result of a motor demand, meaning it is difficult to use the model to design or validate the control system.
4. The model created accounts for the MyPAM only in the unloaded condition, with only 2 degrees of freedom. Modelling MyPAM with a human arm connected would require 7 more degrees of freedom. Creating a model with this traditional approach would be time consuming, prone to error, and prohibitively computationally expensive to run.

4.5 Friction Modelling

4.5.1 Introduction

In a simulation of the MyPAM using a very simplified version Equation 4.44, the robot produced perpetual motion for an instantaneous torque input. This demonstrates the importance of including the damping effects of friction into the dynamic model. Friction is difficult to model and is not well understood, but empirically based models exist which detail the effects of dry friction, fluid friction, lubricated friction, skin friction and internal friction. The loads applied to the MyPAM are insufficient to create significant internal deformation, thus internal friction effects are ignored. Fluid friction occurs when layers of viscous fluid move at different velocities and is therefore not relevant to MyPAM. Further to this, the MyPAM does not move fast enough for air resistance to be an issue, so skin friction may be omitted. The MAXON motors are connected to planetary gearheads, which are sealed and lubricated components. There is also unlubricated gearing at each joint of the robot. The model for friction in the MyPAM is therefore expected to consist only of the effects of dry friction and lubricated friction.

Friction effects are commonly categorised as static friction or dynamic friction. Static friction may be presented as a plot of friction levels against steady state velocity and do not provide accurate predictions for non-zero accelerations [113]. Dynamic friction considers friction when the velocities are not steady state. For the purpose of this thesis, only static friction is modelled due to the difficulty in accurately performing experiments with quickly changing variables whilst working from home due to COVID.

4.5.2 Static Friction Identification Experiment

Introduction

The purpose of this experiment was to determine the static friction at each joint. This comprises two separate parts. For this work the term Stiction is used to refer to the friction which must be overcome to initiate movement, whereas the term static friction is used to refer to friction at steady state velocities above 0 $rads^{-1}$.

Methodology

Stiction Identification

For each joint in turn, the motor demand was increased until the joint began to move. The current drawn by the motor was logged at the point just before joint movement was initiated. The output torque of the motor was then evaluated using the motor torque constant, shown by Equation 4.46.

$$\tau = k_t I. \tag{4.46}$$

Where k_t is the torque constant of the motor and I is the current drawn by the motor.

The torque constants for the motors in MyPAM are shown by Table 4.4.

Table 4.4: *Motor and Motor Torque Constant at Each Joint of MyPAM.*

	Joint 0	Joint 1
Motor Type	MAXON RE 40	MAXON DCX32L 24V
Torque Constant k_t	30.2 mNm/A	27.3 mNm/A

The torque at the output shaft at the end of the gear train was then evaluated using Equation 4.47.

$$\tau_{Ci} = k_{ti} I_i \times GR_i \tag{4.47}$$

Where the gear reduction GR is shown by Table 4.5.

Table 4.5: *Gear Reduction at Each Joint of MyPAM*

	Joint 0	Joint 1
Planetary Gear Reduction (GR_1)	15:1	35:1
Further Gear Reduction (GR_2)	8:3	2:1
Overall Gear Reduction (GR)	40:1	70:1

The value for torque given by τ_{Ci} for each motor corresponds to the stiction which must be overcome before joint movement is initiated.

Static Friction Identification

The robot was dismantled so that each joint could be tested independently from the complete system and was free to fully rotate. A Prony Brake was manufactured, which is a simple type of absorption dynamometer historically used to measure the torque output of an engine [114]. Each joint was orientated horizontally in a fixture to prevent the joint from moving but allowing free rotation of the output shaft at the end of the gear train. The brake blocks and Prony Brake were connected to the output shaft and tightened to allow the shaft to rotate at a steady state velocity. By varying the clamping force (and therefore the friction between the shaft and the brake blocks), different output shaft speeds were achieved. The lever arm of the Prony brake was placed upon digital scales which were then zeroed before each test to remove the weight of the lever arm from the results. The output shaft velocity, the motor current and the force applied to the digital scales were logged as each test was performed.

The output torque of each joint was evaluated using equation 4.47. Usable torque was evaluated by multiplying the force measured by the digital scales by the length of the lever arm of the Prony Brake using equation 4.48.

$$\tau_u = F \times l \tag{4.48}$$

Torque lost to static friction at each joint was evaluated by taking the difference between the output torque of the shaft and the usable torque, as shown by Equation 4.49.

$$\tau_{fi}(\dot{\theta}) = \tau_{Ci} - \tau_{ui} \tag{4.49}$$

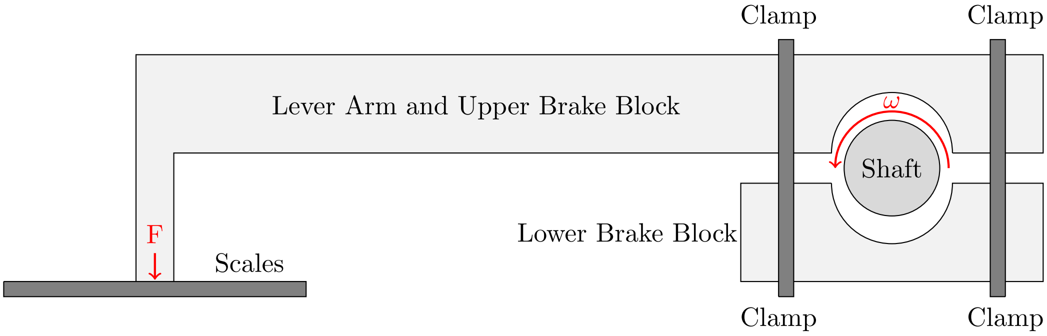

Figure 4.4: *Diagram of the Dynamic Friction Experiment Using Prony Brake Apparatus.*

Results

Stiction Identification

The current draw at motor, torque demand at the motor and torque at the output shaft for each joint before movement is initiated is shown by Table 4.6.

Table 4.6: *MyPAM Joint Stiction Results.*

	Joint 0	Joint 1
Current Draw (A)	0.21	0.15
Torque at Motor (Nm)	0.00634	0.0041
Torque at Output Shaft (Nm)	0.25	0.29

Thus the static friction that must be overcome before movement is initiated is 0.25 Nm at Joint 1 and 0.29 Nm at Joint 2.

Static Friction Identification

The static friction curve for both joints is shown by Figure 4.5, with the stiction results included as a red point at an angular velocity of 0 rad/s.

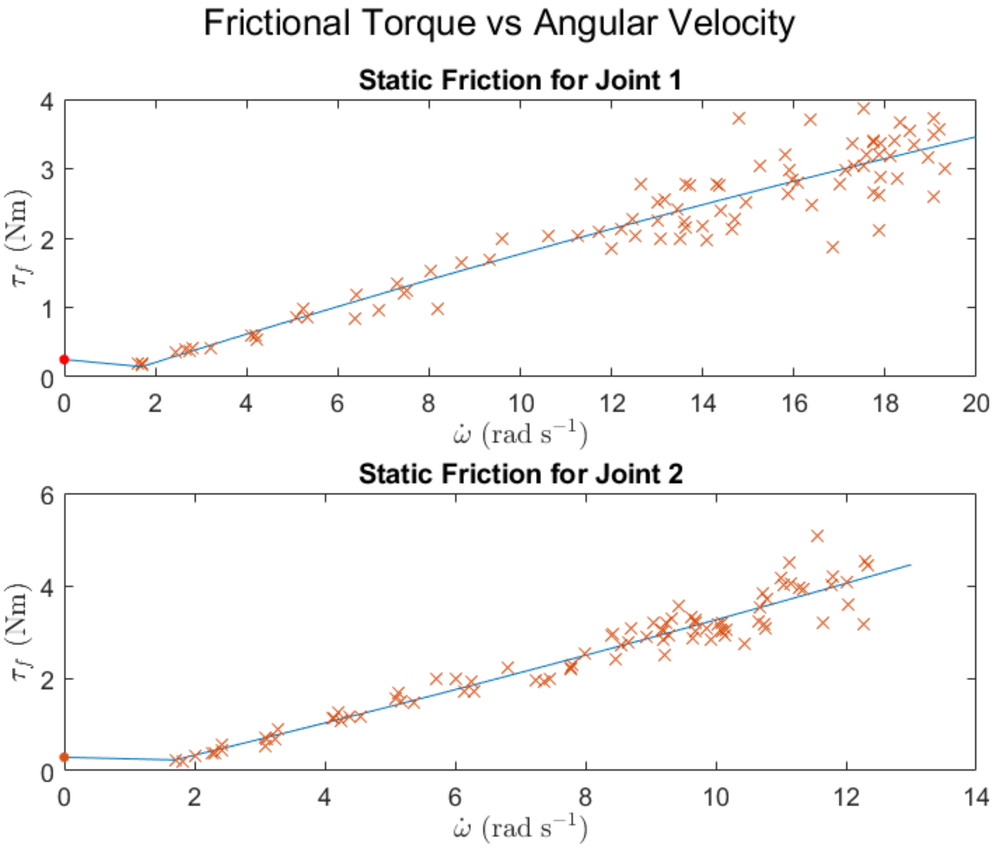

Figure 4.5: *Static Friction for Each Joint of MyPAM.*

The complete combined stiction and static friction model is shown by Equation 4.50.

$$\tau_{fi} = m_i \left| \left(\dot{\theta}_i \right) \right| + c_i \tag{4.50}$$

The coefficients m and c are given by Table 4.7.

Table 4.7: *The Experimentally Determined Static Friction Coefficients.*

$\dot{\theta}_i$	m_i	c_i
$0 < \dot{\theta}_0 < 1.7$	-0.00426	0.25
$\dot{\theta}_0 \geq 1.7$	0.181	-0.0841
$0 < \dot{\theta}_1 < 1.7$	-0.0454	0.29
$\dot{\theta}_1 \geq 1.7$	0.372	-0.456

Discussion

It may be observed that frictional torque is greater in joint 1 than in joint 0. This is expected because joint 1 uses bevel gears to transmit motion through 90 degrees, whereas joint 0 uses only spur gears. Bevel gears are less efficient than spur gears, and therefore have greater frictional losses. Further to this, joint 0 has a 40:1 gear reduction whereas joint 1 has a 70:1 gear reduction. A greater gear reduction has corresponding greater friction.

The shape of the static friction curves shown in Figure 4.5 matches the curves proposed by [113] shown by Figure 4.6, whereby once stiction is overcome at 0 rad s^{-1} there is an initial decrease in friction as angular velocity increases. Once a minimum friction is reached at the Stribeck velocity, friction then increases with increasing angular velocity. It is noted that the static friction curves for joints 0 and 1 of the MyPAM have a less pronounced initial drop in friction, which likely occurs because the dominant friction in the joints is due to the unlubricated contribution of the in-house manufactured bevel and spur gears, rather than the lubricated contribution of the lubricated epicyclic gears which are part of the motor assembly.

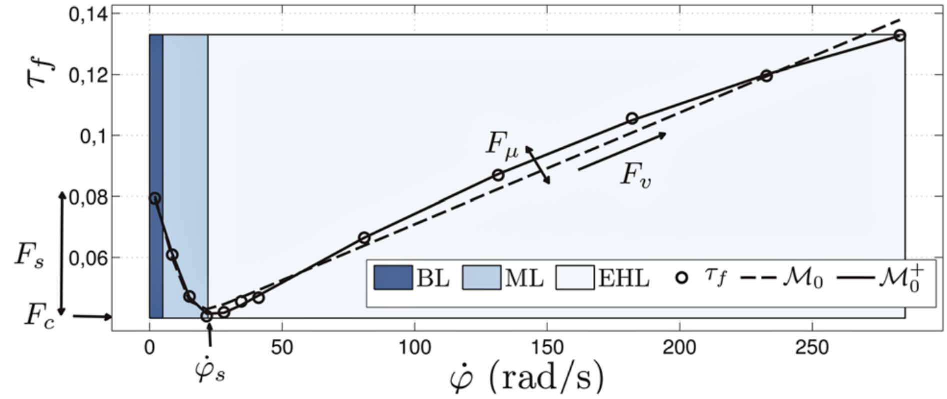

Figure 4.6: *Static Friction Curve [113] Showing the Friction Regimes.*

Where:

F_S = Stansstill Friction Parameter.

F_C = Coulomb Friction.

F_u = Force Related to Non-Newtonian Viscous Friction Behaviour.

F_v = Force Related to Newtonian Viscous Friction Behaviour.

T_f = Frictional Torque.

$\dot{\theta}_s$ = Stribeck Velocity.

$\dot{\theta}$ = Angular Velocity.

BL = Boundary Lubrication Regime.

ML = Mixed Lubrication Regime.

EHL = Elasto-Hydrodynamic Lubrication Regime.

The experimental data points for both joints show good repeatability at low angular velocities, but the greater the angular velocity the greater the spread. This is likely due to the home-made nature of the experiment, which was performed at home during the COVID lockdown. The reliability of the results at higher angular velocities, whilst questionable, does not matter because MyPAM only operates at low angular velocities while in operation. A more important issue is the lack of data at low angular velocities (between 0 and 2 rad s^{-1}) for both joints, which makes it difficult to create a good fit for the static friction curve between the stiction data and the lower friction data around the Stribeck velocity. Indeed, it is not possible to identify the angular velocity at which the static friction is lowest. This has occurred because it was difficult working with home-made equipment. Despite this, at the low angular velocities where it was possible to collect data there is good repeatability suggesting that the friction coefficients are reliable above 1.7 rad s^{-1}.

4.6 Multi-Domain Dynamic Modelling

The model presented in Section 4.4 was formed using the traditional approach to modelling using the Lagrangian formulation. This approach becomes cumbersome as more degrees of freedom are introduced [115], which is the case when modelling the MyPAM with a patient holding the end effector and thereby adding external forces. Already the model was not possible to run due to computational limitations in the unloaded condition, so it was not advisable to continue modelling using the traditional approach.

In recent years simulation based dynamic modelling has grown in popularity due to the increased capability of physics based simulation software packages and the increased computing power available. Adding to this, efficient recursive algorithms have been developed, increasing the performance of simulation software [116]. A multi-domain model of MyPAM was developed using a combination of SimScape and SimScape Multibody by MATLAB. SimScape Multibody allows 3-dimensional dynamic modelling and simulation of mechanical systems, presented within the Simulink environment. The user interface includes a library of configurable blocks representing physical objects which may be placed and connected. SimScape allows single domain modelling of electrical systems

SimScape and Simscape Multibody utilise a physical network approach, whereby block connections communicate information about power [117]. Usefully, an extension library (SimScape Multibody MultiPhysics) enables multi-domain physical effects modelled in Simscape to interact with 3D models created in Simscape Multibody. For MyPAM, this means that it was possible to develop sophisticated motor models in SimScape which were used to drive the rotational joints in the mechanical model in SimScape Multibody. Further to this it was possible to integrate the control scheme, programmed using MATLAB, into the model.

In this section models of MyPAM and a human arm proxy were built using SimScape and SimScape Multibody are presented. The models are validated against real test data generated with MyPAM and the human arm proxy.

4.6.1 Human Arm Proxy

A human arm proxy capable of replicating the seven degrees of freedom of a human arm was used for this work to evaluate the performance of MyPAM under a realistic and repeatable load. The human arm proxy was previously developed for validation of a different upper-limb rehabilitation robot [118]. The human arm proxy emulates Posterior/Anterior Translation and Superior/Inferior Translation at the shoulder using a flexible steel rod secured at a distance away from the shoulder joint. Extension/Flexion of the upper arm is achieved using a rotational joint at the shoulder. Abduction/Adduction are achieved using hinge at the shoulder. Extension/Flexion of the lower arm are achieved by a hinge at the elbow, with the addition of a spring contributing to the stiffness of the elbow joint. External/Internal rotation and Pronation/Supination are achieved with rotational joints, with stiffness of these rotations facilitated by the addition of friction clutches.

A front view of the human arm proxy is shown by Figure 4.7, with four of the seven degrees of freedom illustrated. A side view of the human arm proxy is shown by Figure 4.8, with the remain three degrees of freedom shown.

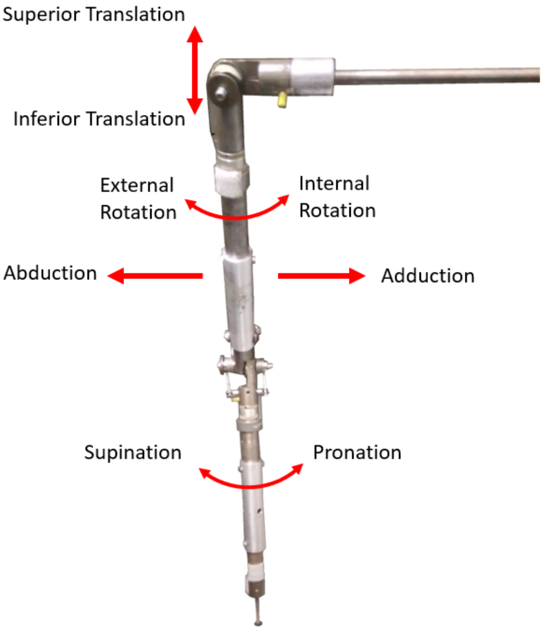

Figure 4.7: *Human Arm Proxy Front View Showing Four of the Seven the Degrees of Freedom.*

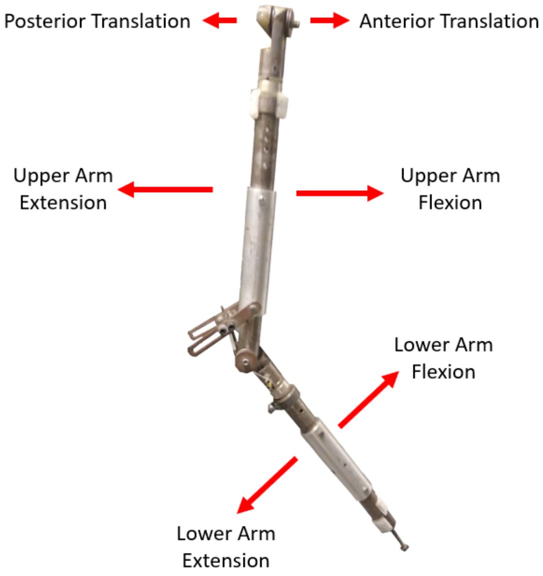

Figure 4.8: *Human Arm Proxy Lateral View Showing the Remaining Three Degrees of Freedom.*

4.6.2 Computational Modelling and Simulation - SimScape Multibody and the Mechanical Domain

The main components of MyPAM identified in Section 4.4 were replicated in SimScape Multibody to create a three-dimensional dynamic model in the mechanical domain. Joint 0 was constrained to rotate about the global origin. Similarly, a 3D dynamic model was created for the human arm proxy. The shoulder of the human arm proxy was constrained to rotate about an appropriate point in the global workspace, and the hand was connected to the MyPAM model with a 6-DoF rotational joint. The modelling procedure for each model followed the workflow:

1. Define the global reference point, coordinate system, and simulation settings by placing the world frame block in parallel with the solver configuration block and the mechanism configuration block;

2. In turn, place and configure a solid body block for a component or place and configure a joint;

3. Connect the block input/output ports, ensuring that rigid body transforms are used where appropriate to translate or rotate frames.

All solid bodies were configured with dimensions and assigned a material density, from which the inertia and mass were automatically calculated. Coordinate frames were assigned at suitable locations for each solid body, which created the input/output connection ports required for components and joints to be connected. Joints 0 and 1 of the MyPAM model were not assigned internal mechanics (friction) because these were instead applied to the motor model for each joint. The two joints were configured to receive a torque input, which was provided by the motor model. All joints in the human arm proxy model were configured with internal mechanics. Finally, Joint 1 and the end-effector of MyPAM were configured to the output position in the global workspace, which was externally logged to MATLAB as the output of each test. The SimScape Multibody model for the unloaded MyPAM is shown by Figure 4.9 and the SimScape Multibody model for MyPAM connected to the human arm proxy is shown by Figure 4.10.

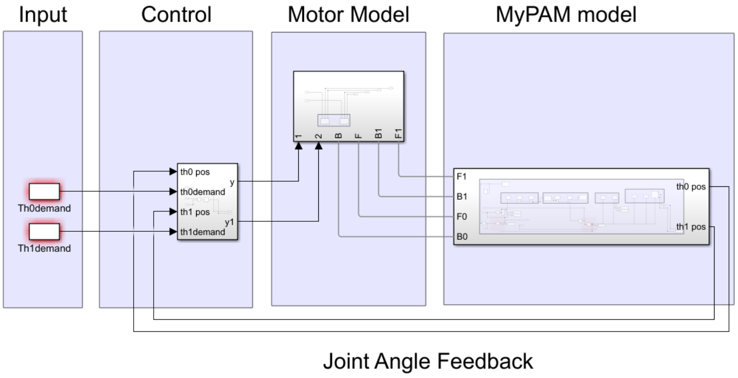

Figure 4.9: *Simscape Multibody Model for MyPAM in the Unloaded Case.*

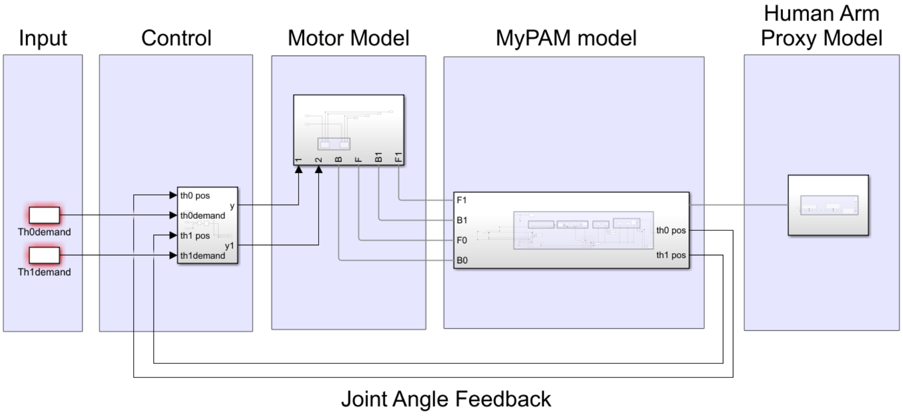

Figure 4.10: *Simscape Multibody Model for MyPAM Connected to the Human Arm Proxy.*

4.6.3 Computational Modelling and Simulation—SimScape and the Electro-Mechanical-Domain

A multi-domain motor model for each powered joint of MyPAM was created in SimScape using a combination of electrical and mechanical blocks. For each powered joint, the motor demand from the control model was provided to a controllable voltage source block. The output of the controllable voltage source block was provided to a DC motor block. The DC motor block in each motor model was configured with the parameters defined in the datasheet for each joint, which are shown by Table 4.8.

Table 4.8: *The DC Motor Block Parameters for Each Joint of MyPAM.*

Parameter	**DC Motor Block (Joint 0)**	**DC Motor Block (Joint 1)**
Armature Resistance (Ohm)	0.299	0.331
Armature Inductance (mH)	0.082	0.103
Torque Constant (mNm/A)	30.2	27.3
No-load Current (A)	0.137	0.164
Nominal Voltage (V)	24	24
Rotor Inertia (gcm^2)	142	72.8

The output of the motor was passed through a gear ratio block configured with the correct gear ratio. A friction block was placed in parallel across the motor and gear blocks to model the friction across the joint. The friction block in each motor model was configured as the friction parameters found in a previous experiment, which are shown by Table 4.9.

Table 4.9: *Friction Parameters For the Motor at Each Joint.*

Parameter	**Joint 0**	**Joint 1**
Breakaway Friction Torque (Nm)	0.25	0.29
Breakaway Friction Velocity (rad^{-1})	0.1	0.1
Coulomb Friction Torque (Nm)	0.18	0.23
Viscous Friction Torque (Nm/rad^{-1})	0.181	0.372

Note that the breakaway friction velocity was set to the default value, which is close to zero. Finally, a conversion block from the SimScape Multibody MultiPhysics library was placed so that the torque output of the motor modelled in SimScape may be provided to the relevant joint of MyPAM modelled in SimScape Multibody. The motor model is shown by Figure 4.11.

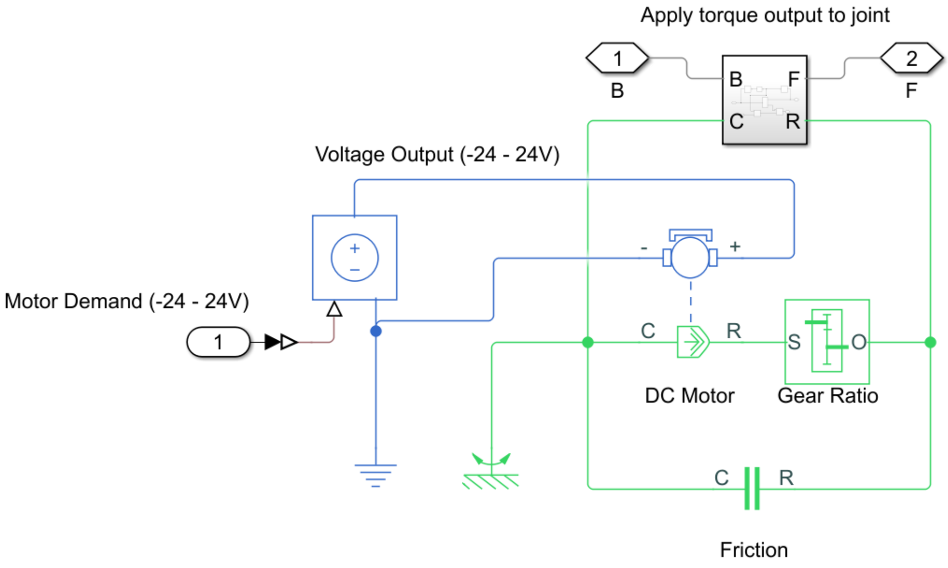

Figure 4.11: *The SimScape Motor Model Showing the Motor Demand Converted into an Applied Torque.*

4.6.4 Computational Modelling and Simulation - MATLAB, Simulink, and the Control Domain

The control scheme was implemented in MATLAB and Simulink and followed the block diagram shown by Figure 4.12.

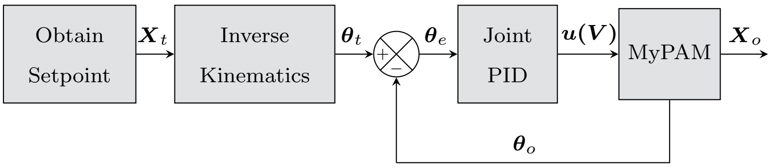

Figure 4.12: *Test Control Strategy Block Diagram, Implemented using MATLAB.*

A discretised Minimum Jerk trajectory between the start and end point of the reaching movement was initially generated using MATLAB to produce a set of Cartesian position vectors for the end-effector, which were used as an input to the whole Simulink model. A MATLAB function block was used in Simulink to perform the inverse kinematics, converting the Cartesian position vectors into joint position demand vectors used for control. Joint position feedback from the MyPAM SimScape Multibody models was used to evaluate the joint position error using standard Simulink mathematics blocks. Simulink PID blocks were used to generate a motor control signal u (in Volts) for each joint using the respective joint position errors. MyPAM uses PI position control only. In the unloaded case for both the robot and the simulation, the gains were $P = 1$, $I = 0.01$. In the loaded case for both the robot and the simulation, the gains were $P = 1.4$, $I = 0.01$.

4.6.5 Simscape Model Testing and Validation

Methodology

A series of tests was performed to validate the performance of the model against the performance of MyPAM. The first pair of tests compared the performance of the MyPAM model against the performance of the MyPAM in the unloaded condition, with no external loading applied. In the first of these tests, a desired trajectory was applied only in the x-direction, and in the second test, a desired trajectory was applied only in the y-direction. The second pair of tests compared the performance of the MyPAM model against the performance of the MyPAM in the loaded condition, with the human arm proxy model connected. The same desired trajectories were applied only in the x-direction and only in the y-direction for each test, respectively. A summary of the testing is provided by Table 4.10.

Table 4.10: *Model Validation Test Summary For MyPAM in the Loaded and Unloaded Conditions Subjected to X and Y Trajectory Demands.*

Test	**MyPAM Loading Condition**	**Trajectory Direction**
1	Unloaded	X-Direction
2	Unloaded	Y-Direction
3	Loaded	X-Direction
4	Loaded	Y-Direction

Figure 4.13 shows MyPAM with the human arm proxy and a simulation of the MyPAM and human arm proxy models.

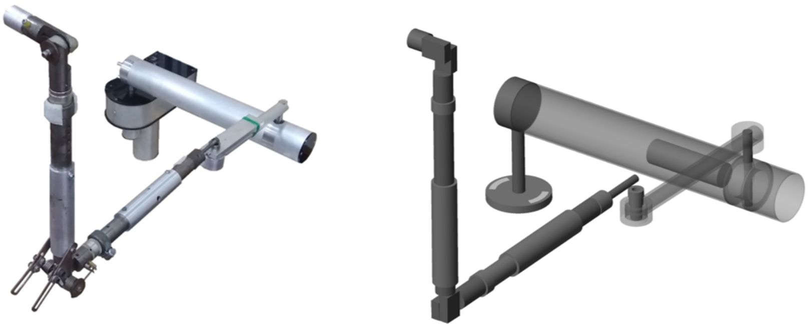

Figure 4.13: ***Left:*** *The Real MyPAM with the Human Arm Proxy.* ***Right:*** *Simulated MyPAM with the Simulated Human Arm Proxy Models.*

4.6.6 Results - Unloaded MyPAM: Tests 1 and 2

The graph shown by Figure 4.14 shows the simulated response and the mean real response of MyPAM when subjected to a minimum jerk trajectory in the x-direction, with the mean x- and y-positions against time shown by Figure 4.15. It may be observed that both responses show a characteristic curve, caused by the small demand of the motor at Joint 0 and the large demand of the motor at Joint 1. The graph shown by Figure 4.16 shows the simulated response and the real responses of MyPAM when subjected to a minimum jerk trajectory in the x-direction, with the x- and y-positions across all repeats against time shown by Figure 4.17.

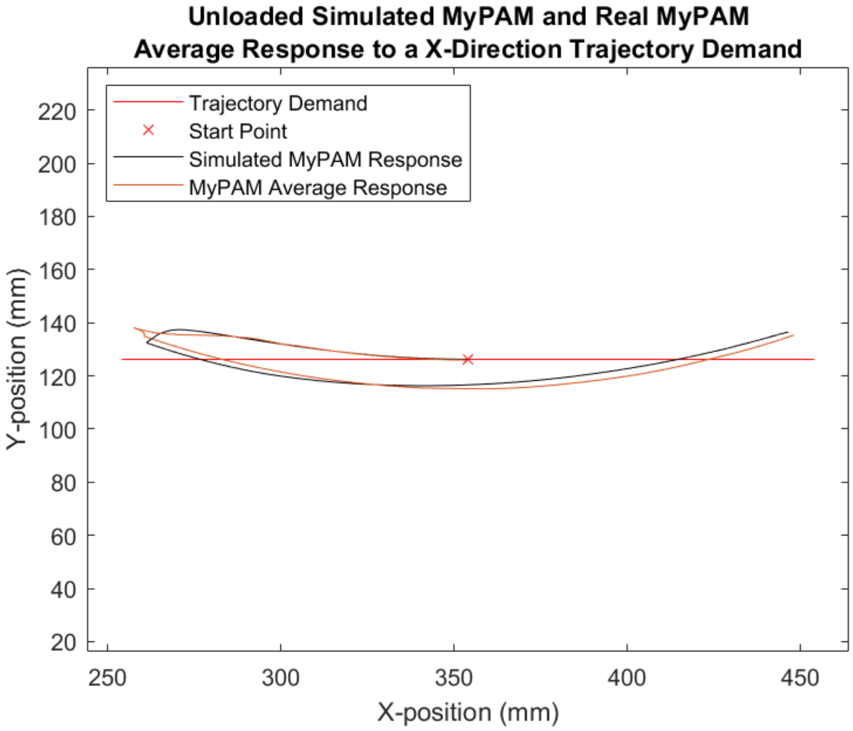

Figure 4.14: *Unloaded Simulated MyPAM and Mean MyPAM X-Direction Trajectory Response.*

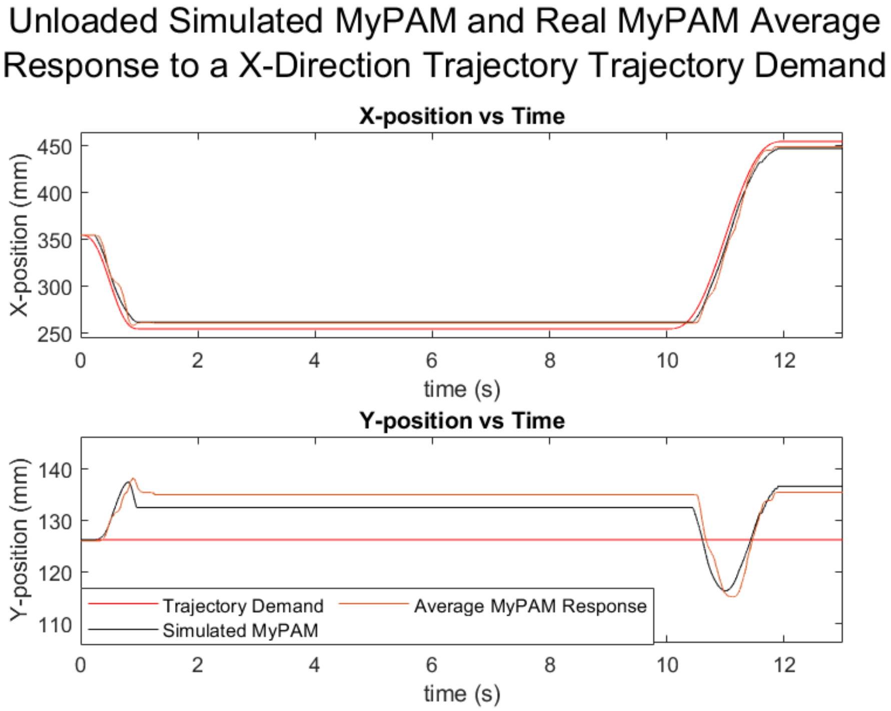

Figure 4.15: *Unloaded Simulated MyPAM and Mean MyPAM X-Direction Trajectory Response Against Time.*

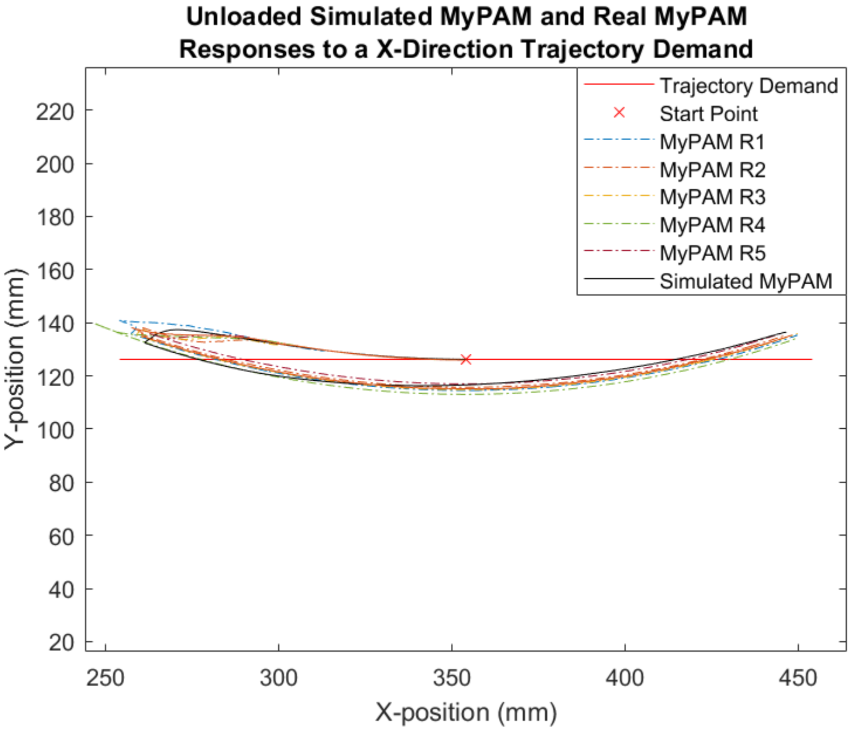

Figure 4.16: *Unloaded Simulated MyPAM and MyPAM X-Direction Trajectory Response.*

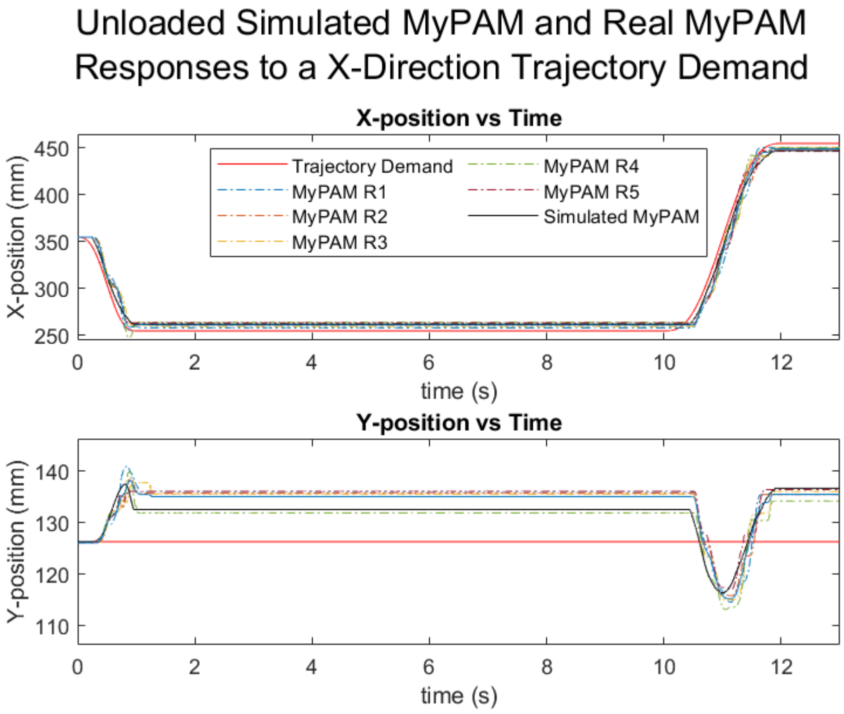

Figure 4.17: *Unloaded Simulated MyPAM and MyPAM X-Direction Trajectory Response Against Time.*

The graph given by Figure 4.18 shows the simulated response and the mean real response of MyPAM when subjected to a minimum jerk trajectory in the y-direction, with the mean x- and y-positions against time shown by Figure 4.19. The graph given by Figure 4.20 shows the simulated response and the real responses of MyPAM when subjected to a minimum jerk trajectory in the y-direction, with the x- and y-positions across all repeats against time shown by Figure 4.21.

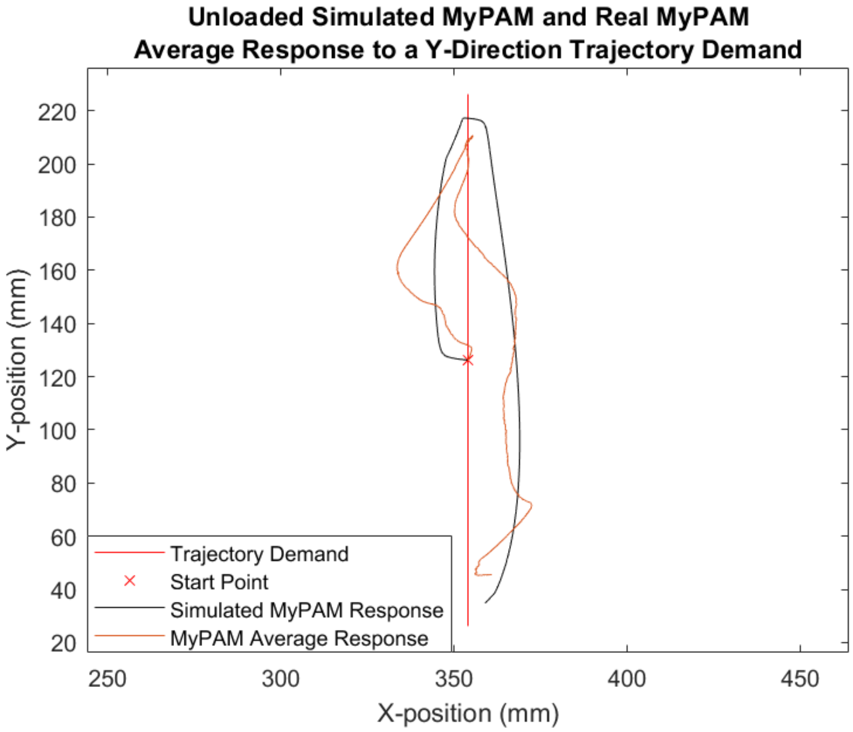

Figure 4.18: *Unloaded Simulated MyPAM and Mean MyPAM Y-Direction Trajectory Response.*

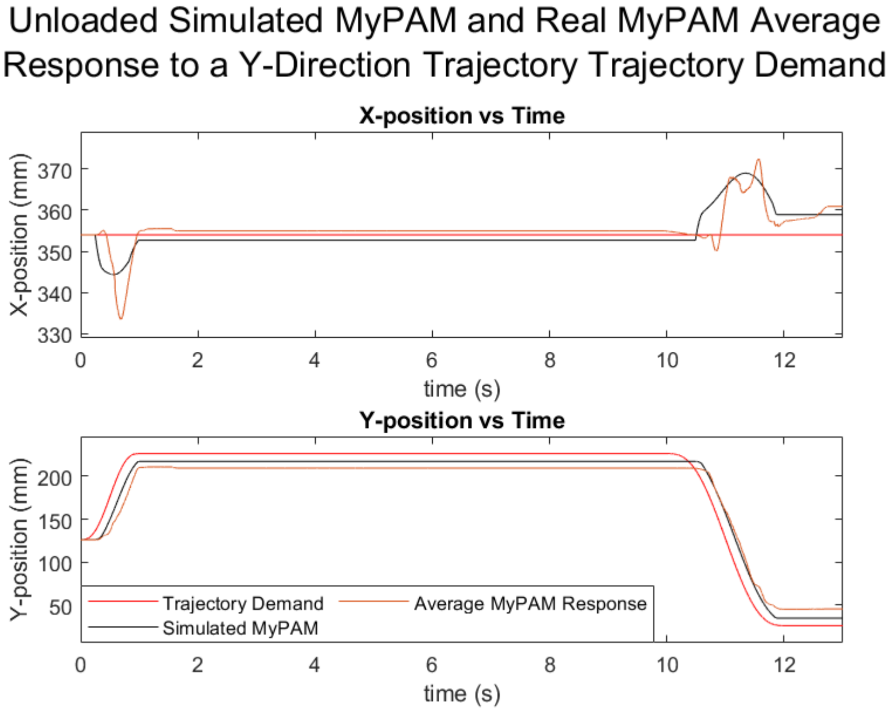

Figure 4.19: *Unloaded Simulated MyPAM and Mean MyPAM Y-Direction Trajectory Response Against Time.*

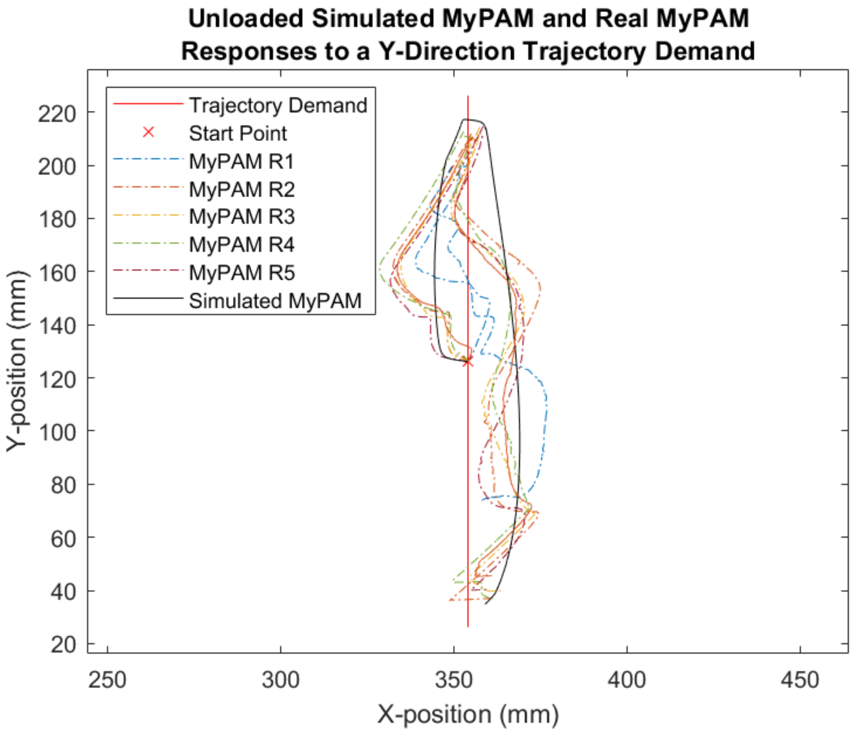

Figure 4.20: *Unloaded Simulated MyPAM and MyPAM Y-Direction Trajectory Response.*

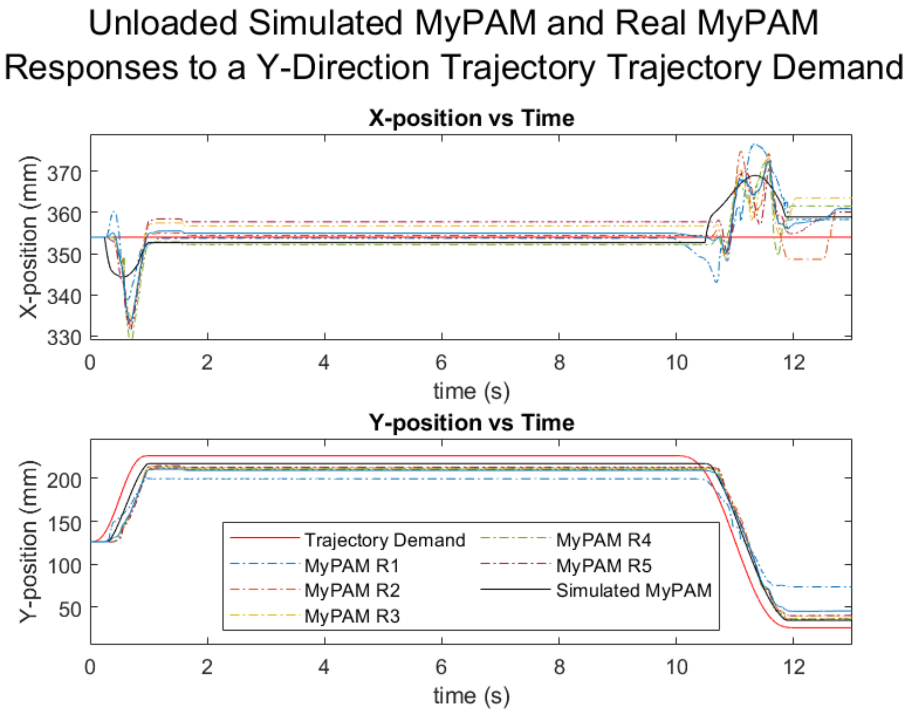

Figure 4.21: *Unloaded Simulated MyPAM and MyPAM Y-Direction Trajectory Response Against Time.*

4.6.7 Results - Loaded MyPAM: Tests 3 and 4

The graph given by Figure 4.22 shows the simulated response and the mean real response of MyPAM when subjected to an x-direction minimum jerk trajectory, with the mean x- and y-positions against time shown by Figure 4.23. It may be observed that the curved response in Figure 4.14 is present. The graph given by Figure 4.24 shows the simulated response and the real responses of MyPAM when subjected to an x-direction minimum jerk trajectory, with the x- and y-positions across all repeats against time shown by Figure 4.25.

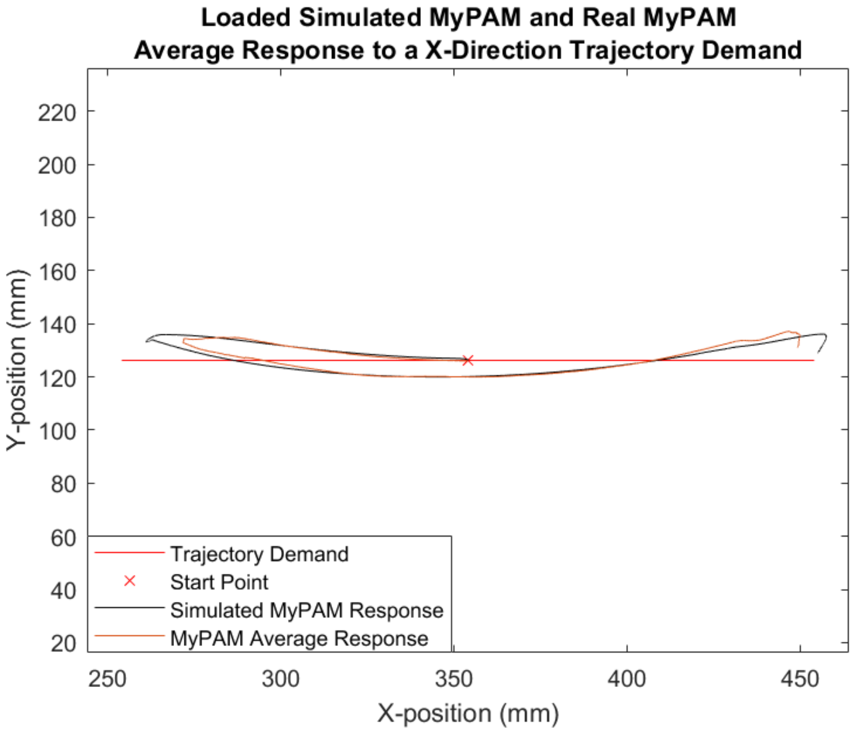

Figure 4.22: *Loaded Simulated MyPAM and Mean MyPAM X-Direction Trajectory Response.*

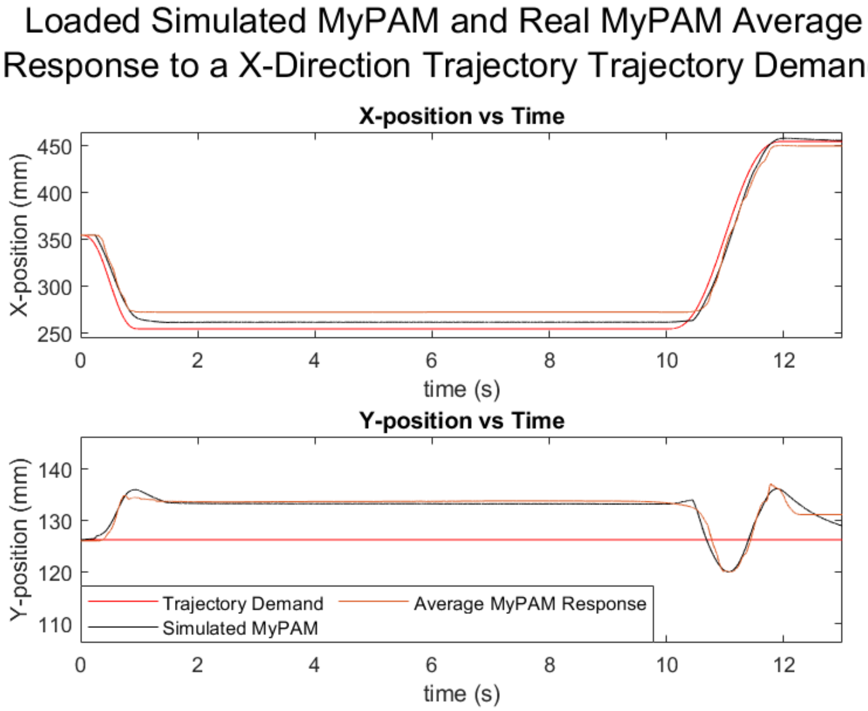

Figure 4.23: *Loaded Simulated MyPAM and Mean MyPAM X-Direction Trajectory Response Against Time.*

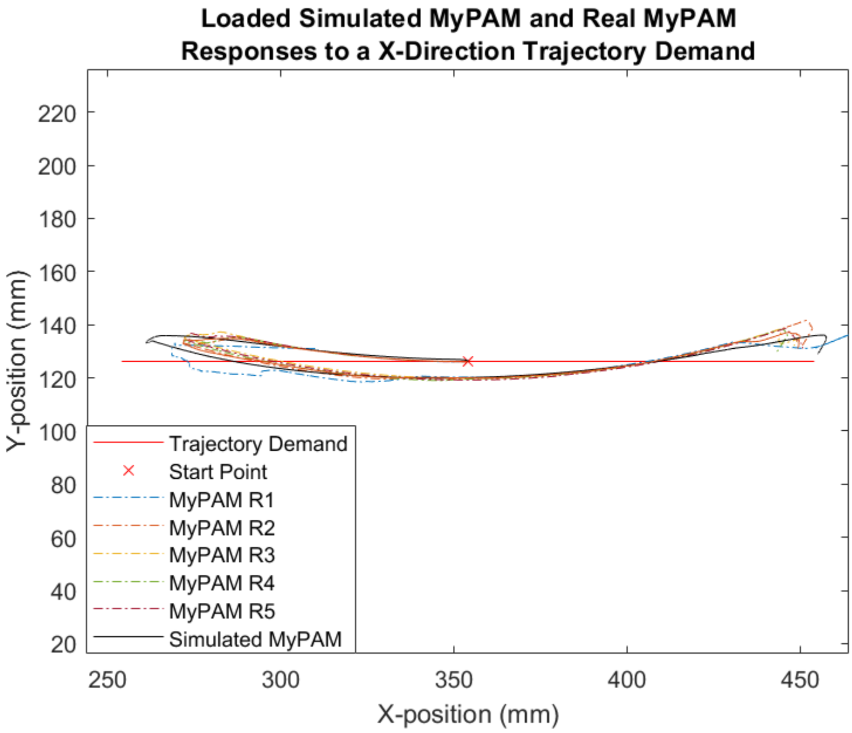

Figure 4.24: *Loaded Simulated MyPAM and MyPAM X-Direction Trajectory Response.*

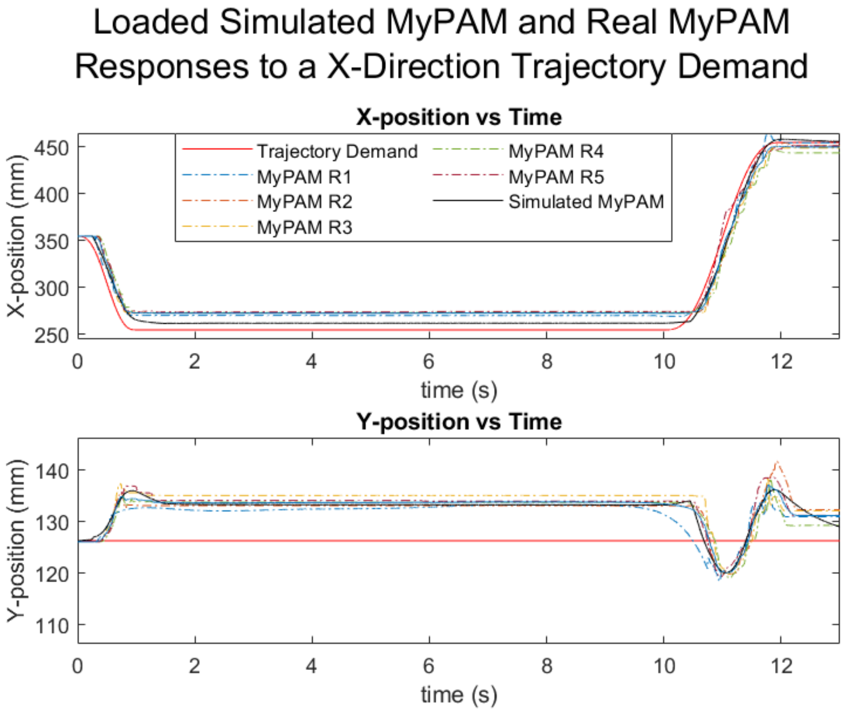

Figure 4.25: *Loaded Simulated MyPAM and MyPAM X-Direction Trajectory Response Against Time.*

The graph given by Figure 4.26 shows the simulated response and the mean real response of MyPAM when subjected to a y-direction minimum jerk trajectory, with the mean x- and y-positions against time shown by Figure 4.27. The graph given by Figure 4.28 shows the simulated response and the real responses of MyPAM when subjected to a y-direction minimum jerk trajectory, with the x- and y-positions across all repeats against time shown by Figure 4.29.

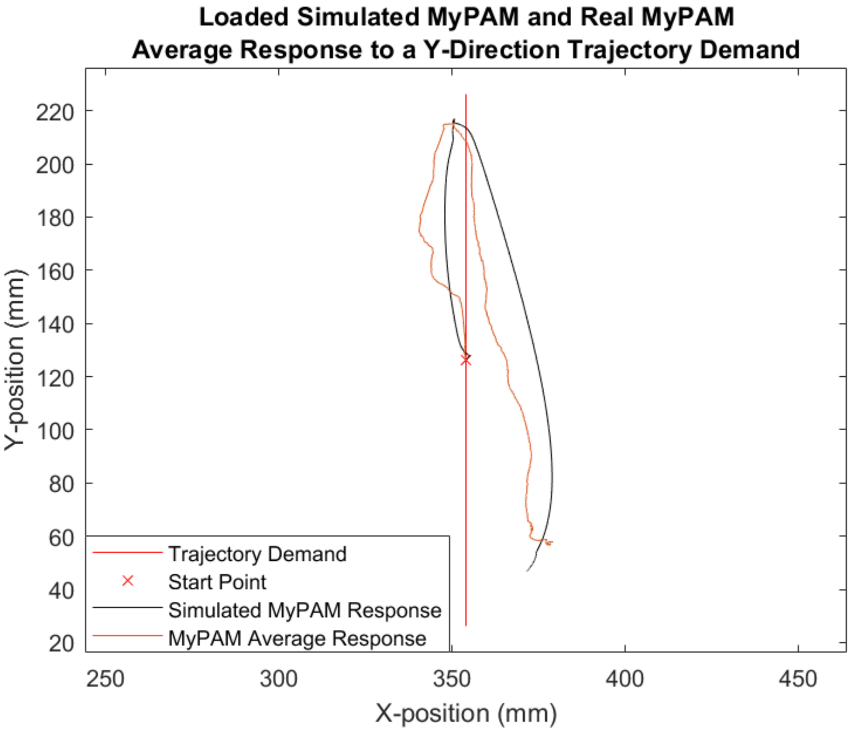

Figure 4.26: *Loaded Simulated MyPAM and Mean MyPAM Y-Direction Trajectory Response.*

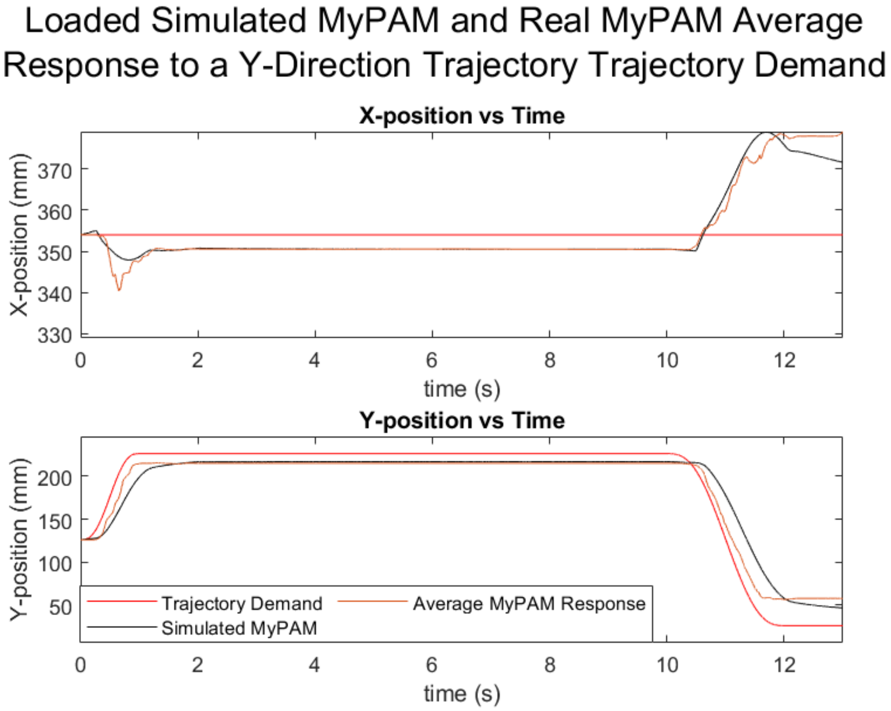

Figure 4.27: *Loaded Simulated MyPAM and Mean MyPAM Y-Direction Trajectory Response Against Time.*

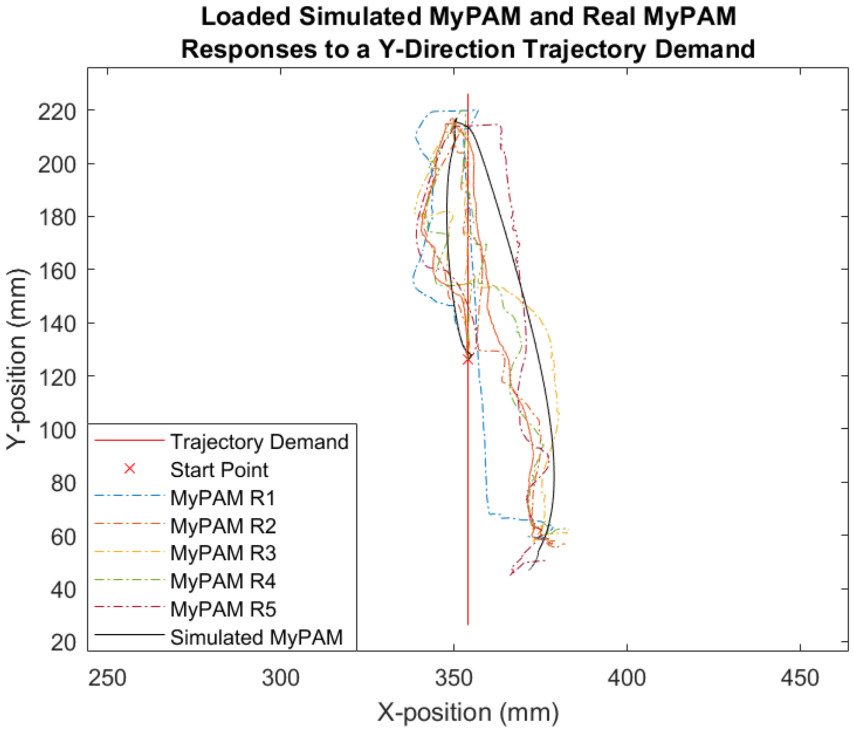

Figure 4.28: *Loaded Simulated MyPAM and MyPAM Y-Direction Trajectory Response.*

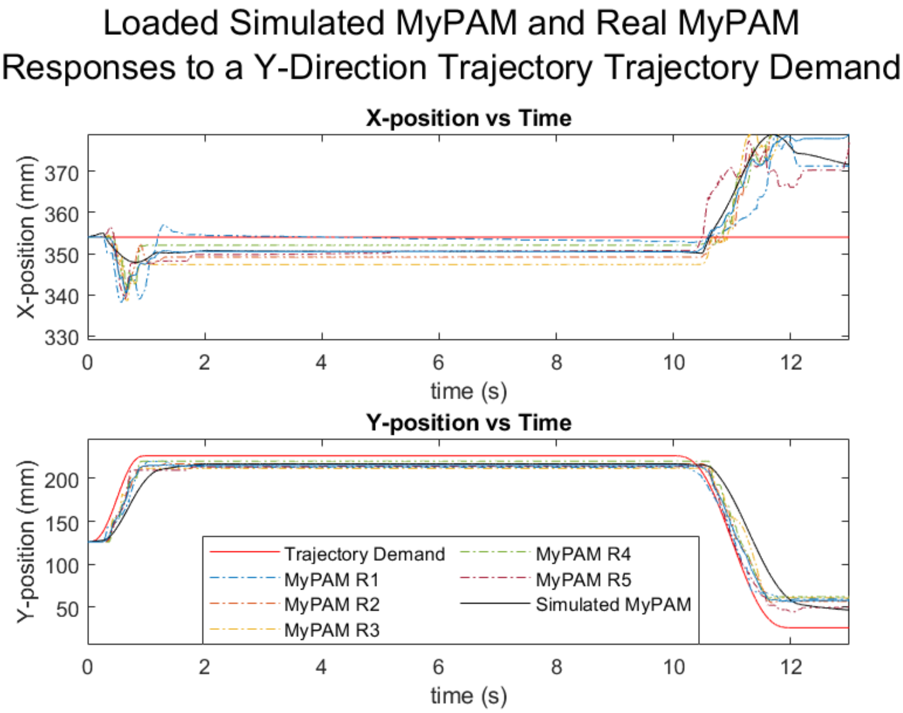

Figure 4.29: *Loaded Simulated MyPAM and MyPAM Y-Direction Trajectory Response Against Time.*

4.6.8 Discussion

In both the unloaded and loaded cases, the x-direction trajectory tracking shown by Figures 4.14 and 4.22 (Tests 1 and 3) is much smoother than the y-direction trajectory tracking shown by Figures 4.18 and 4.26 (Tests 2 and 4) for both the simulated response and the robot response. This is because in this area of the robot workspace, x-direction movement is achieved mainly by rotation of Joint 1, with very little movement required by Joint 0. As expected, the effects of inertia are less pronounced when the majority of the movement occurs at Joint 1 because the majority of the mass of the robot is located between Joint 0 and Joint 1. Similarly, the y-direction trajectory tracking was poorer because most of the movement is achieved moving Joint 0, where the effects of inertia are more pronounced.

In the unloaded case, the x-direction trajectory tracking (Test 1), seen in Figure 4.15, shows good x-position agreement between the simulated response and the robot response, with both closely following the trajectory demand. It should be noted, however, that the robot response shows a small amount of overshoot that the simulated response does not show. There is a similar good x-position agreement between the simulated response and the robot response in the loaded case (Test 3) seen in Figure 4.23, though there is a greater difference between the robot response and the simulated response than is present in the unloaded case. This is likely caused by imperfect modelling of the human arm proxy. The human arm proxy has a greater number of joints than MyPAM, making it considerably more difficult to correctly account for the effects of friction.

The y-position agreement between the robot response and the simulated response for the unloaded x-direction trajectory tracking (Test 1) shown by Figure 4.15 is satisfactory since both lie within the 10 mm dead-zone allowed between the position of the end-effector and the target, though it is noted that the robot response shows greater overshoot and steady-state error than the simulated response. This may be accounted for by the absence of backlash modelling in the MyPAM model. There is a greater y-position agreement between the robot response and the simulated response for the loaded x-direction trajectory tracking (Test 3) shown by Figure 4.23, though the simulated response shows a small overshoot at around 1 s that the robot does not. This is again likely caused by imperfect modelling of friction in the human arm proxy, which added less damping to the simulated response than the human arm proxy added to the robot response.

In the unloaded case, the y-direction trajectory tracking (Test 2), seen in Figure 4.18, shows poorer x-position agreement between the simulated response and the robot response than may be observed in the response to the x-direction trajectory (Test 1). This is likely due to the lack of backlash modelling, which is relatively large at Joint 1. The effect of backlash is pronounced in the robot response because tracking a y-direction trajectory in this area of the robot workspace requires small movements of Joint 1 where the backlash to the x-position demand ratio is high. Despite this, it was observed that the steady-state response in the x-position, seen in Figure 4.19, is closely matched between the robot response and the simulated response. In the loaded case, the x-position agreement between the simulated response and the robot response while tracking the y-direction trajectory (Test 4), seen in Figure 4.27, is poorer than was observed in the unloaded case. This is again likely caused by imperfect modelling of friction in the joints of the human arm proxy model.

In the unloaded case, the y-direction trajectory tracking (Test 2), seen in Figure 4.18, shows reasonable y-position agreement between the simulated response and the robot response, though it was observed that the model tracks the trajectory demand better than the robot. This likely occurred due to the backlash at Joint 1, which caused greater error in the x-position with corresponding inertial effects on the robot. The loaded case (Test 4) shows a better y-position agreement between the simulated response and the robot response, seen in Figure 4.26, due to the damping effects of the friction in the human arm proxy.

It has been noted that SimScape Multibody was poor at modelling instability. This became apparent when attempting to tune the PID gains for the motor controllers using the Ziegler–Nichols method on the unloaded MyPAM model, where it was not possible to introduce instability even with an extremely large derivative gain. Indeed, the model response with large PID gains followed the trajectory perfectly. This means that the PID tuning strategy required a balance of rough tuning with the models to identify sensible gains, followed by finer tuning and validation on the robot. The set of software cannot be relied upon for tuning the control system.

SimScape Multibody does not provide a function for modelling backlash. This was a problem because the backlash in Joint 1 of MyPAM is relatively large. It would be preferred to model the backlash because it has a significant effect on the dynamics, particularly in circumstances where Joint 0 must make frequent changes in direction to track the trajectory. The effects of the backlash on MyPAM can be clearly seen by the jagged motion of the real MyPAM in Figure 4.17. In the same figure, the jagged motion, which would be caused by the effects of the backlash, is absent from the response of the simulated MyPAM, which instead shows a smoother response.

The agreement between the robot response and the model response is acceptable in the primary direction of travel in all cases (i.e., for motion primarily in the x-direction, there is good agreement in the x-position against time). The agreement between the robot response and the model response perpendicular to the primary direction of travel is less good, though it is considered acceptable because this particular region of the workspace was selected for testing as it is where the robot has the most difficulty. It is apparent that the performance of the model more closely matches the performance of the robot when loaded than when unloaded, which is a useful outcome from a rehabilitation perspective since the main purpose of the model is to provide a baseline against which patient data may be compared.

4.6.9 Conclusion

The combination of MATLAB, SimScape, and SimScape Multibody provides an appropriate tool for multi-domain modelling of MyPAM. While the response of the simulation is closer to the response of the robot in the loaded case than in the unloaded case, the model has sufficient accuracy to allow the dynamics of the robot to be accounted for when analysing patient movement data. Importantly, it can be seen that the shape and direction of the response curves produced by the models adequately match the shape and direction of the response curves produced by MyPAM. Modelling the robot using this combination of software allowed the creation of a model with a greater deal of both fidelity and complexity than traditional mathematical modelling alone would allow, though it was noted that the absence of backlash from the model is a limitation. The failure of the models to produce instability in response to a large control input means that this set of software is not appropriate for tuning the control system.

4.7 Chapter Summary

The Forwards Kinematics and Inverse Kinematics were derived for MyPAM in Section 4.2, meeting Chapter objective 4.1. The Forwards Kinematics are used to obtain the position of the end effector from the joint angles. The Inverse Kinematics are used to obtain desired joint angles from a desired end effector position. The Jacobean Matrix was derived Section 4.3, meeting Chapter objective 4.2. The Jacobean Matrix is used for conversion between joint space and Cartesian coordinate space and is particularly useful for Impedance control, where it is necessary to convert a virtual force into a torque demand. In Section 4.5 a model for friction of the joints of MyPAM was developed from experimental data, meeting Chapter objective 4.3. Section 4.4 showed the development of a preliminary dynamic model for MyPAM, though was not possible to validate this model due to the model complexity. Finally, Section 4.6 shows the development and validation of a multidomain dynamic model for MyPAM, meeting Chapter objective 4.4. The multidomain dynamic incorporates a control model, friction effects, the electrical domain of the motors and the mechanical domain of the mechanical structure of the robot. The status of the chapter objectives is shown by Table 4.11.

Table 4.11: *Chapter 4 Objectives Status*

Objective	**Description**	**Success?**
4.1	To derive Forward Kinematic and Inverse Kinematic models for MyPAM.	Yes.
4.2	To derive the Jacobean Matrix for MyPAM.	Yes.
4.3	To obtain a Friction Model for each Joint of MyPAM.	Yes.
4.4	To create and validate a dynamic model for MyPAM.	Yes.

Chapter 5

Grip Sensor

In this chapter the development of an integrated handle/grip sensor for the end effector of the MyPAM is documented. The necessity for a grip sensor is presented in Section 5.1. The grip sensor operation is based on displacing magnets relative to an array of Hall Effects sensors. Section 5.2 details the specifications for the grip sensor. Section 5.3 presents the construction of the grip sensor including component selection and the system architecture. Section 5.4 documents the data processing procedure. Finally, Section 5.5 presents the validation of the grip sensor.

5.1 Chapter Introduction

For patient safety when using the robot it is necessary that there be some mechanism to ensure that the patient is grasping the handle before trying to run the games. As shown in Chapter 4 the dynamics of MyPAM are different depending on the presence of the interaction of a patient, and therefore two controllers are necessary if using position control only as discussed in Chapter 7. A grip sensor is required to determine which controller to use. On previous iterations of the MyPAM a capacitive sensor was used to detect the user. A significant issue was the tendency of the capacitive sensor to drift to zero, which meant that after around 1 minute the system could no longer detect the user. To re-initiate movement the patient was required to release the handle and then re-grasp it, which can be difficult to do for a patient who has disability in the hand and is disruptive to the rehabilitation workflow.

The experimental validation of the grip sensor presented in this Chapter were completed at home during the COVID-19 lockdown. Please note that there was no access to research facilities during this time.

5.1.1 Chapter Objectives

This Chapter aims to fulfil the following objectives:

Objective 5.1 To develop a low cost grip sensor to act as a handle for MyPAM.

Objective 5.2 To validate the performance of the grip sensor when detecting grasp.

Objective 5.3 To validate the performance of the grip sensor when detecting squeeze.

5.2 Grip Sensor Specifications

Whilst capacitive sensing was used in past iterations of MyPAM, there were issues with consistency. It is desirable that a grip sensor is designed which may be further extended in the future for use as a low cost integrated force sensor/handle, since measuring the interaction force between the patient and the end effector is a necessary component for Admittance Control and 6-axis force sensors are prohibitively expensive. To this end a novel approach was taken, using an array of magnets offset from Hall Effects Sensors by a deformable elastomer, inspired by the success of MagOne [119]. MagOne is consists of a tri-axis Hall Effects sensor surrounded by a deformable elastomer containing a neodymium magnet, such that the magnet was offset from the Hall Effects sensor.

1. The grip sensor must be sensitive to a human grasp, even that of someone with poor grip.
2. The grip sensor must detect 3 levels of contact: no contact, grasp and squeeze.
3. The grip sensor must be low cost so as not to dramatically increase the cost of MyPAM.
4. The grip sensor must be further expandable as a force sensor, for use as an input to the Admittance filter described in Chapter 7.

It is useful to note that grip strength varies due to a number of factors including age, gender and level of disability. The purpose of the grip sensor for MyPAM is not to measure a patient's grip strength but rather to detect the presence of a user at all. For this reason it is designed to be sensitive to touch without attempting to measure the strength of grasp. Extending the functionality of the grip sensor to detect a squeeze allows the sensor to be used as an input to the MyPAM, which may be considered similar to a mouse click and would be useful as an input to the games.

5.2.1 Justifying the Approach

There are many alternative ways in which the work presented in this chapter, with examples including the use of limit switches or even pushbutton sensors. To that end this section aims to justify the design choice made to create a grip sensor of a construction inspired by the success of the MagOne tactile sensor. A key demand of the grip sensor was the possibility to extend its use to that of a low force sensor (which is further documented in Chapter 6). This immediately rules out the use of pushbuttons or limit switches as the sensing mechanism for grip. A further justification is that the use of limit switches or small pushbuttons would detect only 2 distinct levels of grip (grip or no contact), as opposed to the desired three levels (no contact, grasp and squeeze). Another reasonable approach would have been to create the force sensor by another methodology and simply use software limits to detect the three desired levels of contact. There are limitations on this however, namely the desired footprint of the handle, the space available inside the handle volume, and cost.

Since the force sensor was intended to also be the handle for the end effector (an integrated handle/sensor), the footprint of the device had to be small enough to be held comfortably, possibly for long periods of time and possibly by a patient with disability of the hand. Further to this are the safety constraints for a medical device like MyPAM, which limits the use of industrial 6-axis force sensors. The reason for this is that an industrial force sensor would be required to be placed at the base of the handle. The handle and elbow link had been designed in such a way as to mitigate the risk of finger trapping, but the addition of an industrial force sensor (which has a larger diameter than desirable) increases the risk of finger trapping. The cost of an industrial sensor was also too high, and of a magnitude similar to the rest of the entire robot.

The space available inside the handle volume was limited, which made the use of the MagOne methodology suitable. Other methods were considered however, but were ultimately rejected for increasing the size of the handle too much. This included the use of Flexiforce sensors, but the diameter of the sensor pad and the width of the electronic tracks were considered too large (especially considering at least 4 sensors would be required to measure force with sufficient accuracy in 2 degrees of freedom). It was also considered to manufacture a module to hold strain gauges, which is the traditional force-sensing paradigm. However, the cost of CNC machining for the module was again considered high, and the calibration that would be required for each sensor was considered likely to be equal in difficulty compared with a force sensor made using the MagOne methodology.

As a final note, it was considered that the opportunities offered by the MagOne methodology were as yet unexplored and presented an exciting opportunity for research. It was noted, however, that there was no guarantee for success and thus it was decided to develop a grip sensor initially to validate this new approach before beginning the much more difficult task of developing the force sensor.

5.3 Grip Sensor Construction

5.3.1 Physical Construction

The grip sensor consists of a rigid aluminium core embedded with an array of four single-axis analogue Hall Effect sensors (Honeywell SS490 Series), which connects to the end effector housing of the robot. Strong neodymium magnets were connected to four equally spaced carrier plates which surrounded the inner core. A hyperelastic silicone separation layer was cast between the carrier plates and the inner core such that the application of force allows the carrier plates and magnets to be independently moved relative to the Hall Effects sensor array. Figure 5.1 shows the grip sensor and a cross-section of the grip sensor. Figure 5.2 shows an exploded view of the grip sensor.

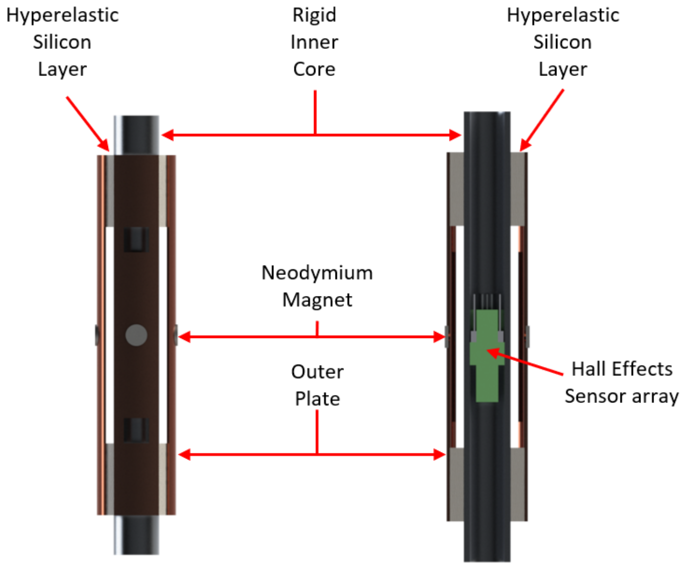
Hyperelastic Silicon Layer
Rigid Inner Core
Hyperelastic Silicon Layer
Neodymium Magnet
Outer Plate
Hall Effects Sensor array

Figure 5.1: ***Left:*** *A Solidworks Render of the Grip Sensor.* ***Right:*** *A Solidworks Render of the Grip Sensor Cross Section.*

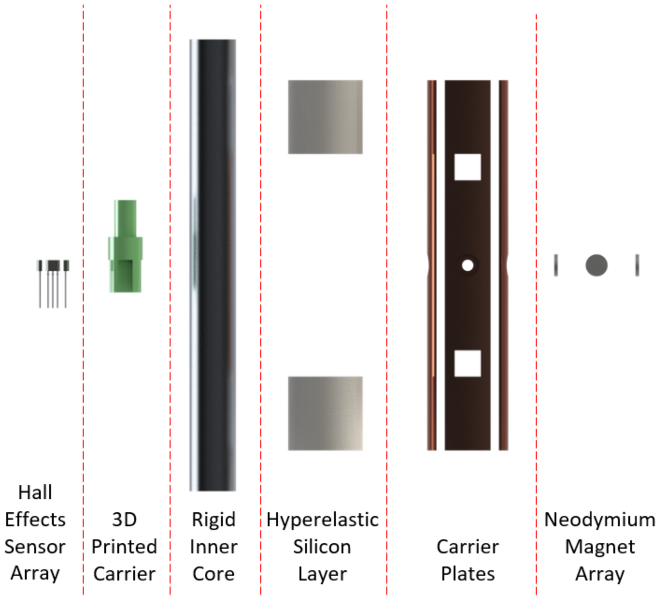

Figure 5.2: *A Solidworks Render of the Grip Sensor Components.*

5.3.2 Silicone Selection

EcoFlex was selected for the hyperelastic silicone since EcoFlex has been successfully used in the past for a tactile sensor application [119]. A range of EcoFlex exist, which cure with a range of stiffnesses. The selection of the most appropriate silicone product was not immediately apparent, however, because EcoFlex products are not traditional engineering materials and as such do not have any material properties published. Further to this, whilst EcoFlex products have been utilised for various applications in some published scientific work [120][121][122][123], either the necessary material properties have either been left unpublished or there is significant disagreement in the published data. This disagreement is highly likely to have have arisen due to differences in mixing ratios and test environments. To this end four physical prototypes were built to find the most appropriate composition, two using EcoFlex 0010 and two using EcoFlex 0050. The prototypes consisted of inner and outer hollow tube sections with different compositions and volumes of EcoFlex cast in between. It should be noted that EcoFlex 0010 is significantly softer, more elastic, and more compliant than EcoFlex 0050, which is the stiffest in the range. The composition of the test pieces is shown by Figure 5.3.

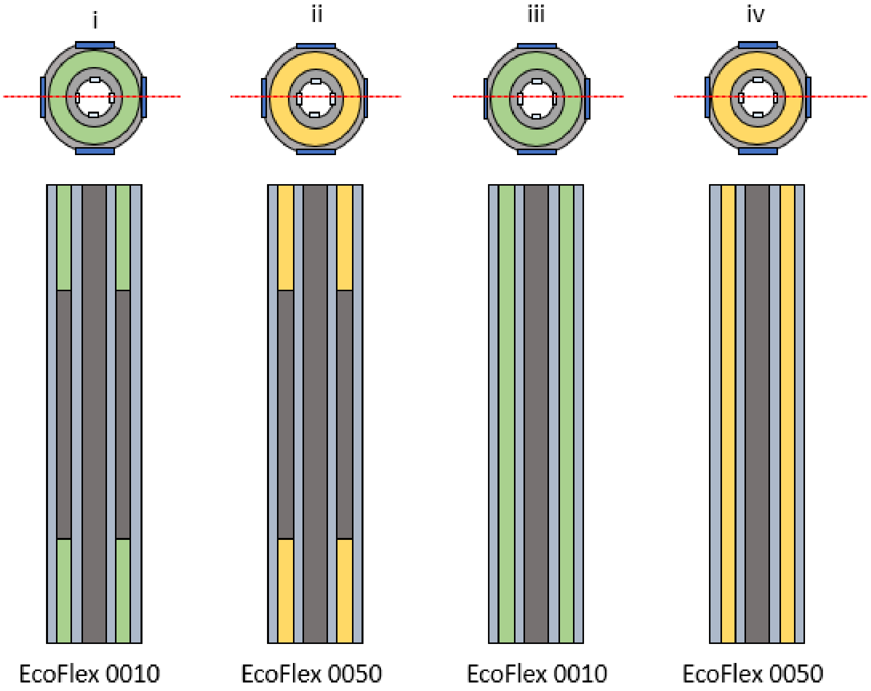

Figure 5.3: *The Composition of the Four Physical Test Pieces Using Different Composition and Layouts of the Hyper-Elastic Silicone.*

Test piece *i* used EcoFlex 0010 which was cast 20mm from each end of the test piece, leaving a void throughout the centre of the test piece. Test piece *ii* had a similar composition to test piece i, but used EcoFlex 0050. Test piece *iii* used EcoFlex 0010 cast throughout the entire length of the test piece. Test piece *iv* used EcoFlex 0050 cast throughout the entire length of the test piece.

It was found that for test pieces *ii*,*iii* and *iv* the outer core did not displace relative to the inner core sufficiently to create a usable difference in the Hall Effect Sensor measurement with the application of small forces. Test piece *i* however provided the desired displacement with the application of small forces, so this configuration was used to to build the grip sensor.

5.3.3 System Architecture, Data Acquisition, Signal Conditioning and Data Transfer

Much of the electronics and communication design choices were informed by the system architecture of MyPAM, which meant that it was necessary to acquire and filter data at source before sending a data packet to the myRIO for utilisation. Data was acquired from the Hall Effect Sensor array using a Teensy 3.2 with an acquisition rate of 1kHz, since this could be directly powered by the myRIO. A hardware based low-pass filter was constructed with a cutoff frequency of 1.5khz for each Hall Effects Sensor to reduce electrical noise. The raw sensor data was built into a datapacket and transmitted from the Teensy to the myRIO using the I^{2}C protocol at a rate of 1kHz. This achieved three aims:

1. Communication could be achieved using only two digital I/O pins on the myRIO, rather than requiring four Analogue pins (of which there are fewer available).

2. Standardising communication to the myRIO from peripheral devices such as the grip sensor simplifies the programming architecture of the low level controller.

3. Data transferred by I^{2}C is more robust to noise than data transferred as raw analogue voltage measurements, since the data packet could be injected with parity bits to check the correct transmission of the message upon receipt.

The system architecture is shown by Figure 5.4.

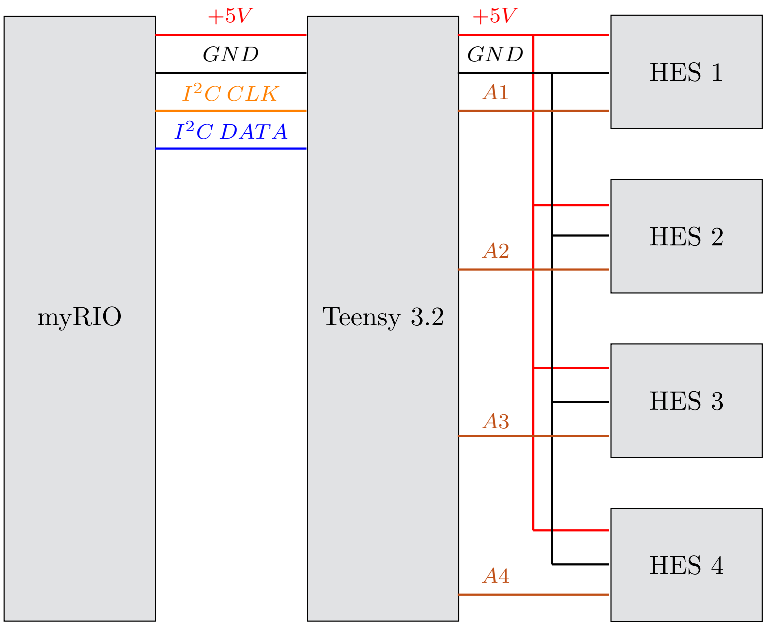

Figure 5.4: *The System Architecture of the Grip Sensor Connection to MyPAM.*

5.4 Data Processing

Each of the Hall effects sensors in the array outputs a voltage in the presence of a magnetic field, which is measured in raw bits ranging from 0-1023. In the presence of a negative magnetic field, the sensors output between 0-511. In the presence of a positive magnetic field the sensors output between 513-1023 bits. In the presence of no magnetic field the sensors output 512. With the grip sensor manufactured initial testing was used to determine the baseline reading of each sensor when the grip sensor was ungrasped. Due to the signal conditioning the baseline readings were stable and each sensor did not deviate from 540 bits, representing the detection of a positive magnetic field as expected.

As presented in the specifications in Section 5.2 it was necessary that the grip sensor was able to detect 3 levels of contact: no contact, grasp and squeeze. The threshold for grasp detection was set to a deviation of any Hall Effects sensor reading of more than 2 bits, which represents a voltage change of only 0.0098V. This was selected as the threshold because testing showed this as the level at which the inertia of moving the grip sensor did not trigger a false reading. The threshold for squeeze detection was set to a deviation of any Hall Effects sensor reading of more than 10 bits. This threshold was found by testing to provide measurements which are distinct from grasp alone, but which are achievable with minimal gripping effort. Since the four magnet carrier plates move independently of each other, crossing either of these two thresholds in any of the four Hall Effects sensor readings were registered as grasp detection or squeeze detection respectively.

5.5 Grip Sensor Validation

5.5.1 Introduction

Two experiments were performed to validate the performance of the grip sensor. The first experiment tests the performance of the grip sensor in grip mode only. The second experiment tests the performance of the grip sensor in the bi-functional grip/squeeze mode to determine that a distinction could be made between a grasp and a squeeze. The grip sensor is very sensitive to grasping and registers grip at very low (sub 1N) forces, so testing apparatus was manufactured to assist with testing.

The testing apparatus consisted of a glove with a flexible wire connected to the palm. The wire was positioned in such a way that grasping the grip sensor pushed the wire against a small metal plate, completing a circuit in a similar way to a limit switch. A small spring returns the wire to the open position when no grasping is occurring. The normally open circuit was tied low using a 10kΩ resistor and was connected to a digital input pin of the myRIO. When grasping, the glove outputs a high signal, and when nothing is grasped the glove outputs a low signal. A similar circuit was constructed such that connecting the forefinger of the glove to the thumb creates a connection, which was used to register a squeeze event.

5.5.2 Methodology

Grip Mode

The grip sensor was grasped and released for a few seconds at a time over a period of 20 seconds whilst wearing the testing glove. The direction from which the grip sensor was grasped was changed with each grasp to ensure that the sensor registered a grasp for the full 360 degree sweep. Three repeats were performed, with a mixture of short grasps and longer grasps.

Bi-functional Mode

The grip sensor was grasped, squeezed and released for a few seconds at a time over a period of 20 seconds whilst wearing the testing glove. During the squeeze phase the forefinger and thumb of the adapted glove were pressed together to register the squeeze event. The direction from which the grip sensor was grasped and squeezed was changed with each grasp phase to ensure that the sensor registered a squeeze for the full 360 degree sweep. Three repeats were performed, with a mixture of short grasps and longer grasps.

5.5.3 Results

Figure 5.5 shows the results of the three repeats with the grip sensor tested for grasp only. The blue lines show the response of the testing glove and the red lines show the response of the grip sensor.

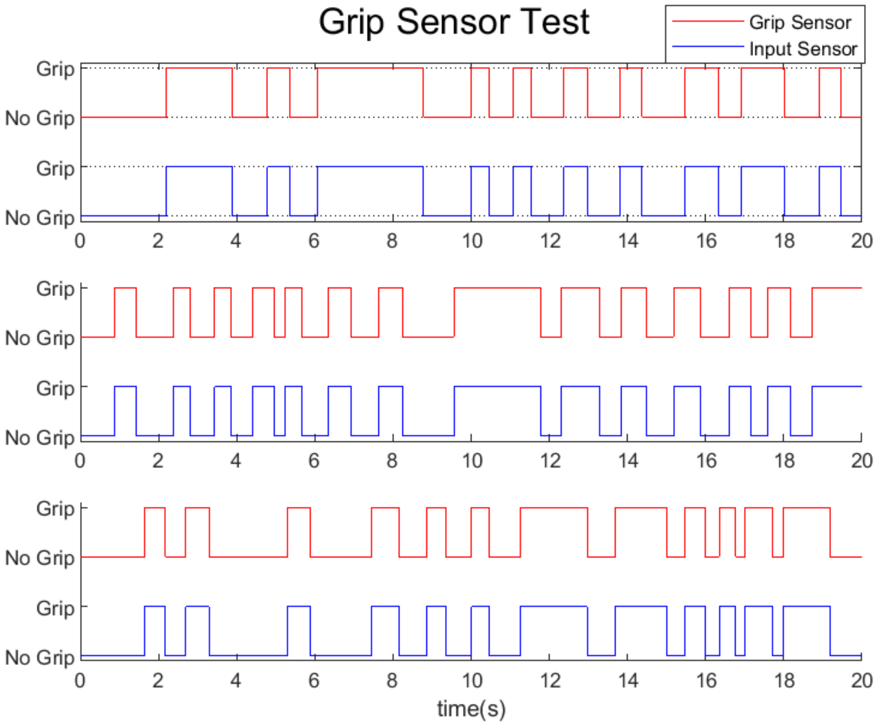

Figure 5.5: *Grip sensor Experimental Results - Grip or No Grip.*

Figure 5.6 shows the results of the three repeats with the grip sensor tested for grasp/squeeze in bi-functional mode. The blue lines show the response of the testing glove and the red lines show the response of the grip sensor.

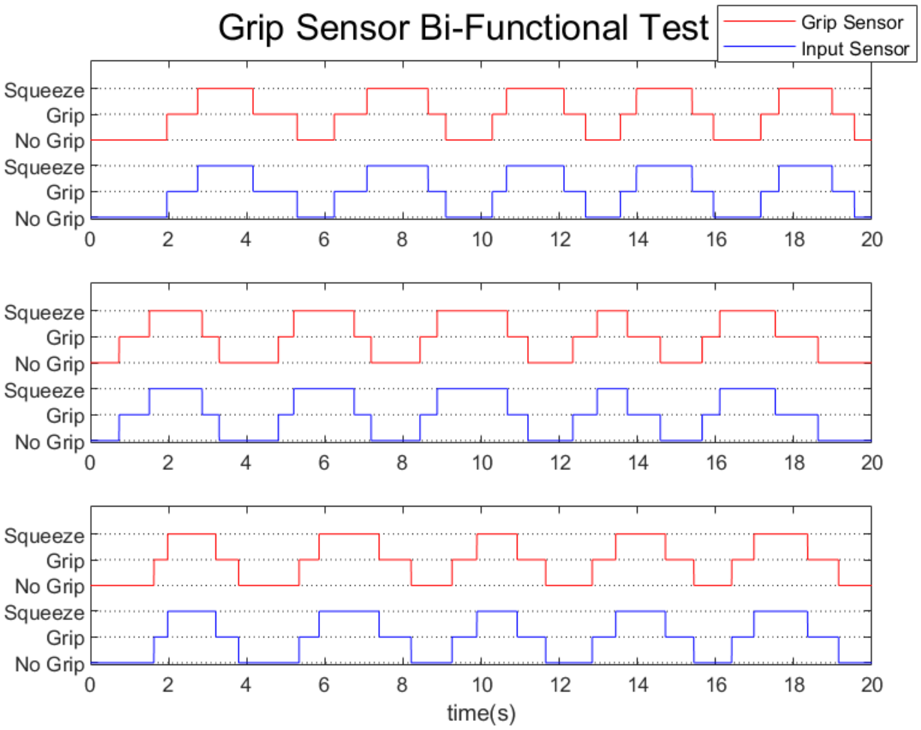

Figure 5.6: *Grip Sensor Bi-functional Experimental Results - Grip, No Grip, or Squeeze.*

5.5.4 Discussion and Conclusion

It can be observed in Figure 5.5 that the grip sensor robustly detects a grasp. All three repeats show agreement between the adapted glove and the grip sensor. No false readings may be observed, and it can be seen that the grip sensor registers a grasp at the same time as the adapted glove and a release at the same time as the adapted glove for every grasp. The success of the grip sensor allows it to be used to select between the two position control schemes for MyPAM. One position control scheme is used when there is no patient grasping the end effector, and the second is used when there is a patient grasping the end effector. This is important because these distinct use cases have different dynamics, and safe operation of a single control scheme in one use case may introduce instability in the other use case.

It can be observed in Figure 5.6 that the grip sensor detects a grasp at the same time as the adapted glove and a release at the same time as the adapted glove for every grasp. Similarly it may be observed that all three repeats show agreement between the adapted glove and the grip sensor during the squeeze phase. This is a useful outcome because it allows the grip sensor to be used as an input device with MyPAM, which allows the creation of more engaging rehabilitation games.

The success of the grip sensor warrants an adaptation of the grip sensor into a low cost 2-axis force sensor which works using a similar methodology. This is a useful outcome because measuring the interaction force between the patient and the MyPAM is necessary for implementing Admittance control.

5.6 Chapter Summary

In this chapter a grip sensor was developed and validated. Section 5.3 documents the construction of the sensor, meeting objective 5.1. The grip sensor takes the form of four carrier plates embedded with neodymium magnets offset using a hyperelastic silicone separation layer offset from a rigid aluminium core embedded with an array of Hall effects sensors. Application of force moves the carrier plates relative to the array of Hall Effects sensors. The performance of the grip sensor to detect grasp and squeeze is validated in Section 5.5, meeting objectives 5.2 and 5.3 and the sensor performed well for both tasks. The grip sensor robustly detects the presence of a user and successfully distinguishes between grip and squeeze. The design and construction of the grip sensor facilitates further exploration for use as a low-cost 2-axis force sensor, which is documented in Chapter 6. The grip sensor is constructed from low-cost components with a price of circa £30. The status of the chapter objectives is shown by Table 5.2.

Table 5.2: *Chapter 5 Objectives Status*

Objective	Description	Success?
5.1	To develop a low cost grip sensor to act as a handle for MyPAM.	Yes.
5.2	To validate the performance of the grip sensor when detecting grasp.	Yes.
5.3	To validate the performance of the grip sensor when detecting squeeze.	Yes.

Chapter 6

Force Sensor

In this chapter the grip sensor developed in Chapter 5 is adapted into an integrated 2DoF force sensor/handle for the end effector of MyPAM. Section 6.1 introduces the necessity of a force sensor to measure the interaction between the patient and the MyPAM and lays out the chapter objectives and force sensor specifications. Section 6.2 describes the construction of a force sensor based on the use of an array of single-axis Hall Effects sensors. Section 6.3 describes the training and selection of a Neural Network to convert the Hall effects sensor array data into a 2-dimension force reading. Section 6.4 documents the experiments used to validate the performance of the single-axis Hall Effects based force sensor.

6.1 Chapter Introduction

To implement Admittance control or Impedance control it is necessary to measure the interaction force between the patient and the end effector. Since the MyPAM is a planar device it is only required to measure the x and y components of force. The majority of rehabilitation robots which require a force sensor use industrial off-the-shelf force sensors. For example, the iPAM used the 6-axis ATI Mini40 [45] with a cost of around $5500 for each measured joint. Off-the-shelf force sensors can be obtained which measure force in 1 axis or in 6 axes, but no product exists which measures force in 2-axes. Common guidance for robotics suggests that in order to measure force in more than one axis a 6-axis force sensor should be used. 6-axis sensors are too expensive for a low-cost robot however.

To measure force in 2 axes at a low cost it is clear that a bespoke device must be developed. There has been success in recent years in the development of low-cost tactile sensors, such as MagOne [119]. Indeed, the grip sensor developed in Chapter 5 is based on the sensing methodology used by MagOne. MagOne consists of a tri-axis Hall Effects sensor surrounded by a deformable elastomer containing a magnet, such that the magnet was offset from the Hall Effects sensor. Whilst the performance of this tactile sensor is acceptable, the complex relationship between the magnetic field and force required a considerable effort to characterise. This was achieved by using a combination of simulation, optimisation techniques and genetic programming [124] . It is useful to note that the performance parameters for a tactile sensor are very different to those of the 2-axis force sensor required by MyPAM.

Despite the difficulty in characterising the relationship between magnetic field and force, the use of Hall Effects Sensors for force sensing is promising as a solution for developing the end effector force sensor for MyPAM. MagOne costs only £10, provides reliable and accurate performance and has a very small footprint. Further to this design guidelines are provided for creating Hall Effect based tactile sensors [119], but the relevance of these design guidelines for such a different application is uncertain. Based on the success of MagOne, the success of the grip sensor developed in Chapter 5, and the provided design guidelines this chapter documents the adaptation of the grip sensor into a 2-axis force sensor.

6.1.1 Force Sensor Specifications

The MyPAM is designed to be low cost, aesthetically appealing, and have an appropriately sized footprint in order to fit into the home of a patient. This means that the Force sensor selected for the MyPAM must meet the following specifications:

1. The Force sensor must measure force in the global X and Y directions.

2. The Force sensor must be low cost so as to not substantially increase the cost of the device.

3. The Force sensor must not substantially increase the footprint of the device.

4. The Force sensor must not substantially alter the aesthetic appeal of the device.

6.1.2 Chapter Objectives

This Chapter aims to fulfil the following objectives:

Objective 6.1 Develop a low-cost integrate force sensor/handle to measure force in 2 axes..

Objective 6.2 Measure the performance of the force sensor against known static loads to determine viability of the product.

Objective 6.3 Measure the performance of the force sensor against known dynamic loads to determine viability of the product.

6.2 Single-axis Hall Effects Sensor Array Based Force Sensor

6.2.1 Introduction

The success of the integrated grip-sensor/handle developed in Chapter 5, which had excellent sensitivity to grip, suggested that a force sensor exploiting an array of Single-axis Hall Effects sensors may be successful. To this end a single-axis Hall Effects Based force sensor was developed. The force sensor was subjected to both static and dynamic tests in order to validate performance.

6.2.2 Physical Design

Similar to the grip sensor, the single-axis Hall Effects Based force sensor consists of a rigid aluminium core embedded with an array of four single-axis analogue Hall Effect sensors (Honeywell SS490 Series), which connects to the end effector housing of the robot. Instead of attaching neodymium magnets to individual carrier plates as was done when constructing the grip sensor, a rigid outer core embedded with neodymium magnets was placed around the inner core and a hyperelastic silicone separation layer was cast between the 2 cores, such that an applied force allows the outer core to displace relative to the inner core. This arrangement is shown by Figure 6.1.

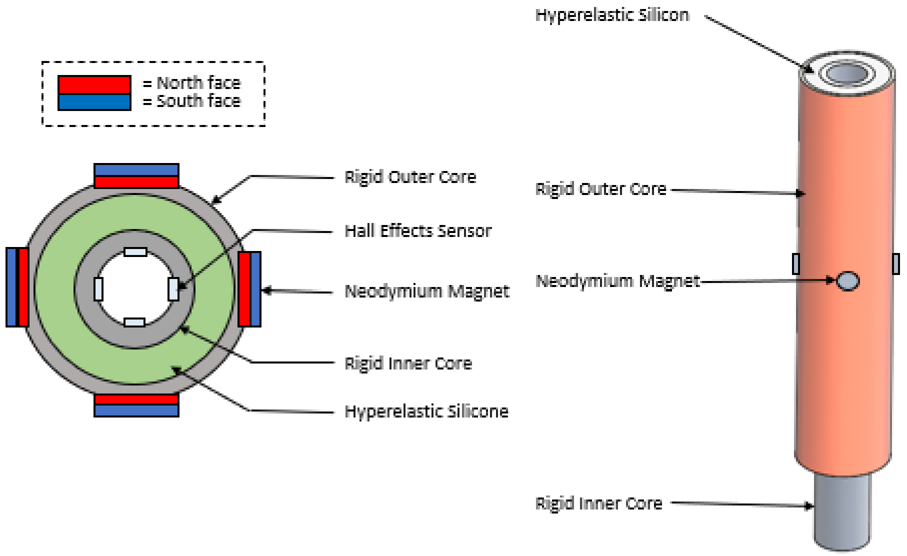

Figure 6.1: ***Left:*** *A Cross sectional View of Force Sensor.* ***Right:*** *An Isometric View of Force Sensor*

EcoFlex 0010 was used as the silicone separation layer, since it performed well for the grip sensor. The system architecture, data acquisition procedure, signal conditioning, and data transfer protocol remained the same as the grip sensor.

6.2.3 Converting Hall Effects Data to a Force Measurement

A Neural Network was used to characterise the relationship between Hall Effects data and input force because of the difficulty presented in building a mathematical model. The hyperelastic silicone displaces non-linearly under load and, as stated in Section 5.3.2 in Chapter 5, there is little agreement in published data for material properties. The mixing and casting process used to create the silicone is difficult to reproduce exactly. This means that any material properties identified from test pieces would not be reproducible. An added difficulty in modelling the force sensor is presented by the complex magnetic field caused by 4 strong magnets in a 3D arrangement. Further to this, machine learning was used to characterise the displacement/force relationship for the MagOne tactile sensor [124] .

6.3 Training the Neural Network for the Single-axis Hall Effects Based Force Sensor

6.3.1 Introduction

A Neural Network must be trained before it may be used. Training data must be obtained which consists of a linked set of known inputs and outputs, which is used to train the network using a process known as back-propagation. Once a network has been trained it can be used to predict an output when provided with a previously unseen input. Two sets of training data were used for this work and multiple Neural Networks with different topologies were trained. The first set of training data consisted of static force measurements, where the force sensor data was acquired when loaded with a series of known static forces and allowed to settle between each measurement. The second set of training consisted of dynamic force measurements, where the force sensor data was acquired when loaded with a series of known dynamic forces which constantly changed.

A Neural Network consists of multiple layers of linked neurons. The input layer accepts the input data and passes the data to a hidden layer. Each neuron in the the hidden layer applies an activation function and passes the results of this processing to the next layer, which may be another hidden layer or the output layer. The output layer provides the solution to the problem that the Neural Network has been trained to solve. Between each layer a set of weights and biases are applied. Back-propagation trains the network by adjusting the weights and biases. The toplogy of a generic Neural Network is shown by Figure 6.2.

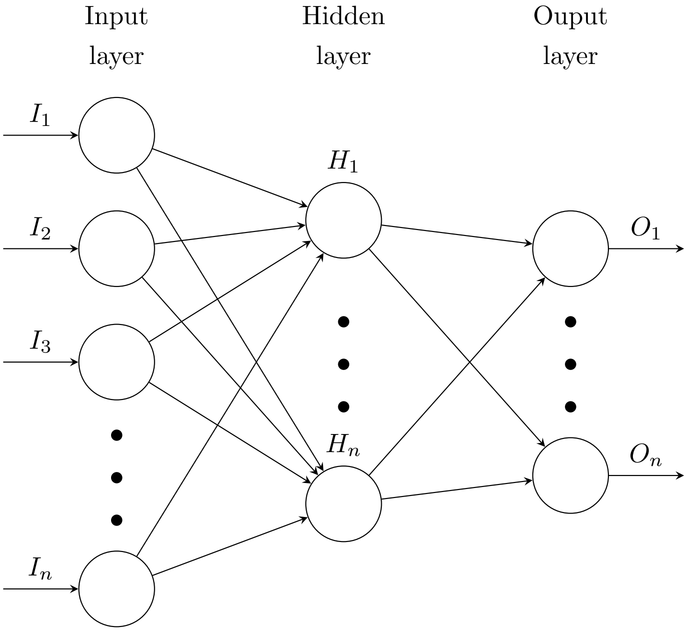

Figure 6.2: *The Topology of a Generic Neural Network.*

Topology selection for a Neural Network is an ill-defined problem, relying on a heuristic iterative approach. The size of the input layer of the Neural Network is informed by the application. For this particular problem the input layer consists of four neurons because the force sensor relies on a measurement from four Hall Effects sensors. Since a Neural Network is used as a classification tool, whereby each output neuron outputs a probability (ranging from 0-1), the size of the output layer depends on the number of possible classifications. The output layer in this particular problem consists of 202 output neurons, which represent a range of -50N to +50N in 1N increments in the x-direction and -50N to +50N in 1N increments in the y-direction. The x and y components of force may be obtained from the Neural Network by observing the output layer and selecting the x neuron and y neuron with the highest probability rating. The optimum size and number of hidden layers cannot be determined, so a number of Neural Networks were trained and compared.

6.3.2 Acquiring Neural Network Training Data - Static Force

A test rig was constructed with an inner frame capable of being moved relative to an outer frame. The force sensor could be mounted in the test rig in such a way that one frame was connected to the inner rigid core and one frame connected to the outer rigid core. This meant that the application of a known force applied to one frame of the test rig was able to displace the inner core of the force sensor relative to the outer core. Further to this, the force sensor could be rotated in the test rig to change the loading direction in the x-y measurement plane. The test rig setup is shown by Figure 6.3.

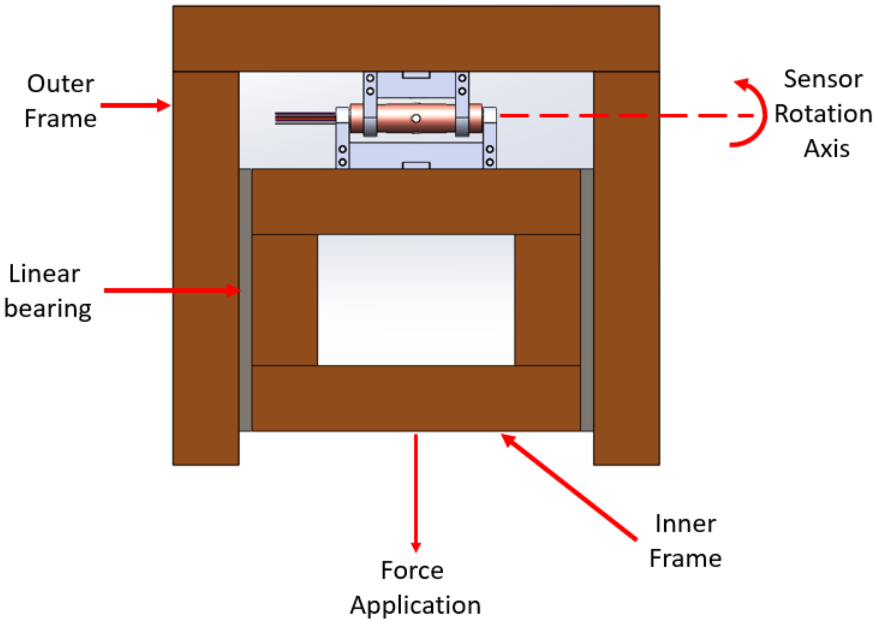

Figure 6.3: *The Static Load Test Rig. The Inner Frame Moved Relative to the Outer Frame, Applying a Known Load to the Force Sensor.*

The force sensor was mounted in the test rig and a quadrature encoder was connected to measure the loading angle. The force sensor was loaded from 0N to 50N in 1N increments and the Hall Effects Sensor data were logged alongside the x and y components of loading force, which were resolved from the loading force and the loading angle. The Force Sensor was then unloaded and rotated by 10 degrees. The process was repeated until measurements had been obtained for a full 360 degree rotation. Ten repeats were performed of the full procedure (each consisting of circa 100 000 samples), with the force sensor dismounted from the test rig and remounted between each repeat to minimise the effect of the starting conditions on the full dataset. All sets of data were combined to be used as the training data for the static Neural Network.

6.3.3 Acquiring Neural Network Training Data - Dynamic Force

A test rig was constructed which rigidly held a JR3 Multi-Axis Force-Torque Sensor and allowed the single-axis Hall Effects Based force sensor to be mounted in such a way that any force applied to the single-axis Hall Effects Based force sensor was also measured by the JR3 Multi-Axis Force-Torque Sensor. This arrangement is shown by Figures 6.4 and 6.5.

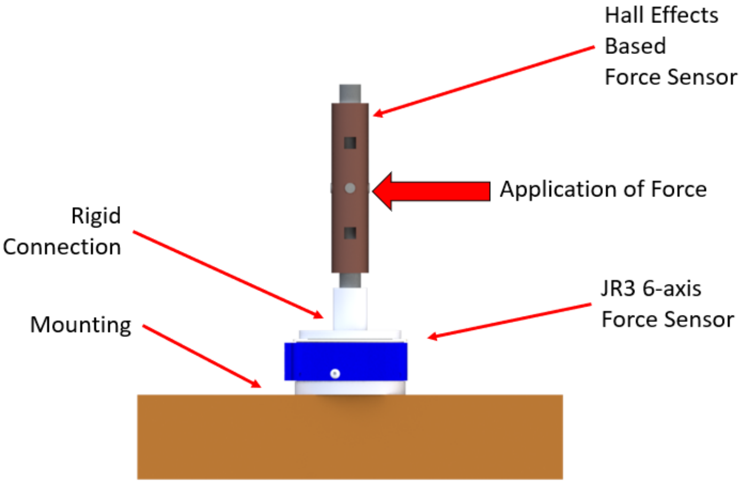

Figure 6.4: *The Dynamic Load test Rig - Front View*

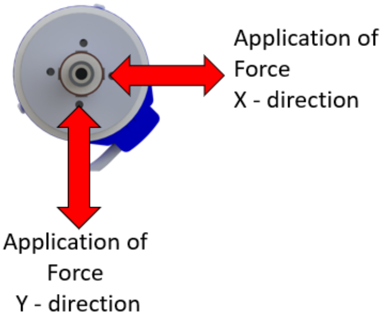

Figure 6.5: *The Dynamic Load test Rig - Top View*

The JR3 Multi-Axis Force-Torque Sensor was mounted in the test rig and calibrated. The single-axis Hall Effects Based force sensor was mounted to the JR3 Multi-Axis Force-Torque Sensor. The Hall Effects Sensor array data from the single-axis Hall Effects Based force sensor and the x and y loading data from the JR3 Multi-Axis Force-Torque Sensor were logged whilst a series of dynamic loads were applied from all directions for 100 seconds, producing 100 000 samples. Ten repeats were performed of the full procedure, with the force sensor dismounted from the test rig and remounted between each repeat to minimise the effect of the starting conditions on the full dataset. All sets of data were combined to be used as the training data for the dynamic Neural Network.

6.3.4 Processing the Training Data

It was necessary to pre-process both sets of training data into a form that a Neural Network could accept. For the input data: The Hall Effects Sensor data were formed into a matrix with the elements scaled to between 0 and 1. The input force data were used to form the expected output vector. The expected output vector was 202 elements in length and contained all zeros, apart from the two elements corresponding to the x and y components of force which were set to one. For example, for the case that the components of input force were 20N in the x-direction and -30N in the y-direction all elements in the expected output vector were set to 0, apart from the elements corresponding to 20N in the x-direction and -30N in the y-direction which were set to 1. Post processing the output of the Neural Network required selecting the x-neuron and the y-neuron with the highest probability to identify the x and y components of input force.

6.3.5 Neural Network Selection

A number of Neural Networks were trained with the static force training data in an effort to optimise the network topology whilst being respectful of the need to conserve computational resources, since the final Neural Network must run on the myRIO alongside the rest of the robot control program. The differences in topology between Neural Networks included varying the number of hidden layers and varying the number of Neurons in each layer. The Levenberg-Marquardt was used as the training algorithm since it is faster at training than many other algorithms provided there is sufficient memory available. The optimised toplogy found for the Static Force Neural Network was again used whilst training the dynamic force Neural Network, since it was a hugely time consuming process to train multiple networks.

The topology, training time and Mean Squared Error of the Neural Networks trained with the static load training data are shown by Table 6.2.

Table 6.2: *The Single Axis Neural Network Topology, Training Time and Mean Squared Error for the Nine Neural Networks.*

Neural Network	Hidden Layer 1 Neurons	Hidden Layer 2 Neurons	Hidden Layer 3 Neurons	Hidden Layer 4 Neurons	Hidden Layer 5 Neurons	Hidden Layer 6 Neurons	Hidden Layer 7 Neurons	Hidden Layer 8 Neurons	Training Time (hrs)	Mean Squared Error
1	1000								81	6.01×10^{-7}
2	500	500							120	5.20×10^{-7}
3	202	202	202						91.5	5.31×10^{-7}
4	202	202							73.75	5.50×10^{-7}
5	50	50	50	50					66	8.25×10^{-7}
6	10	50	100	202					91	1.33×10^{-6}
7	202	404	202						105.25	1.67×10^{-6}
8	10	20	40	60	80	100	150	202	135	9.13×10^{-6}
9	202	150	100	80	60	40	20	10	138.5	1.90×10^{-6}

In every case the Neural Network training was ended because the training gradient had become too small, meaning that increasing amounts of time were required to obtain increasingly small Neural Network performance gains. Neural Network 4 was selected as the optimised Network topology for testing, since it provided a balance between performance and computer resource preservation. A Neural Network was trained using the Static Force training data and second Neural Network was trained using the Dynamic Force training data.

6.4 Single-axis Hall Effects Based Force Sensor Validation Experiment

6.4.1 Introduction

To evaluate the proficiency of the force sensor a pair of experiments were performed, the first evaluating static loading and the second evaluating dynamic loading. For each test the respective trained Neural Network was built into the force sensor architecture and the response of the force sensor was tested against known loads.

6.4.2 Methodology

Static Force Validation

The Single-axis Hall Effects Based Force Sensor was loaded into the Static force test rig described in Section 6.3.2 and the loading procedure used to obtain the Static Force training data set was repeated. The predicted components of force in the X-direction and Y-direction output by the trained Neural Network were logged alongside the loaded components of force in the X-direction and Y-direction for comparison.

Dynamic Force Validation

A similar procedure was followed as above. The Single-axis Hall Effects Based Force Sensor was loaded into the Dynamic force test rig described in Section 6.3.3 and the loading procedure used to obtain the Dynamic Force training data set was repeated. The predicted components of force in the X-direction and Y-direction output by the trained Neural Network were logged alongside the loaded components of force in the X-direction and Y-direction for comparison.

6.4.3 Results

The response of the force sensor to static loading is shown by Figure 6.6.

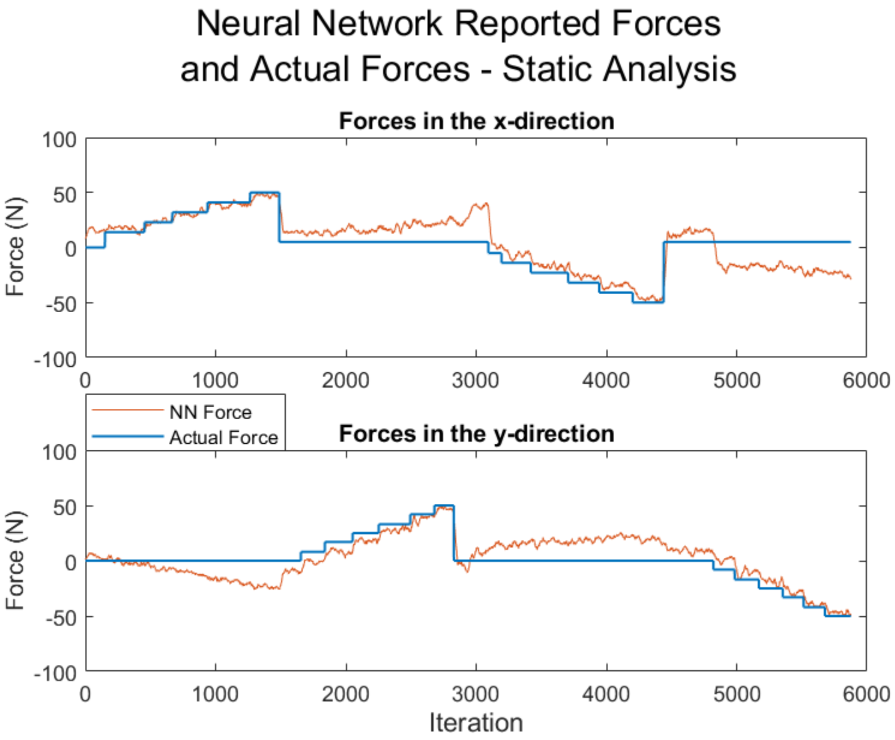

Figure 6.6: *The Response of the Force Sensor with a Trained Neural Network to Static Loading*

The response of the force sensor to dynamic loading is shown by Figure 6.7.

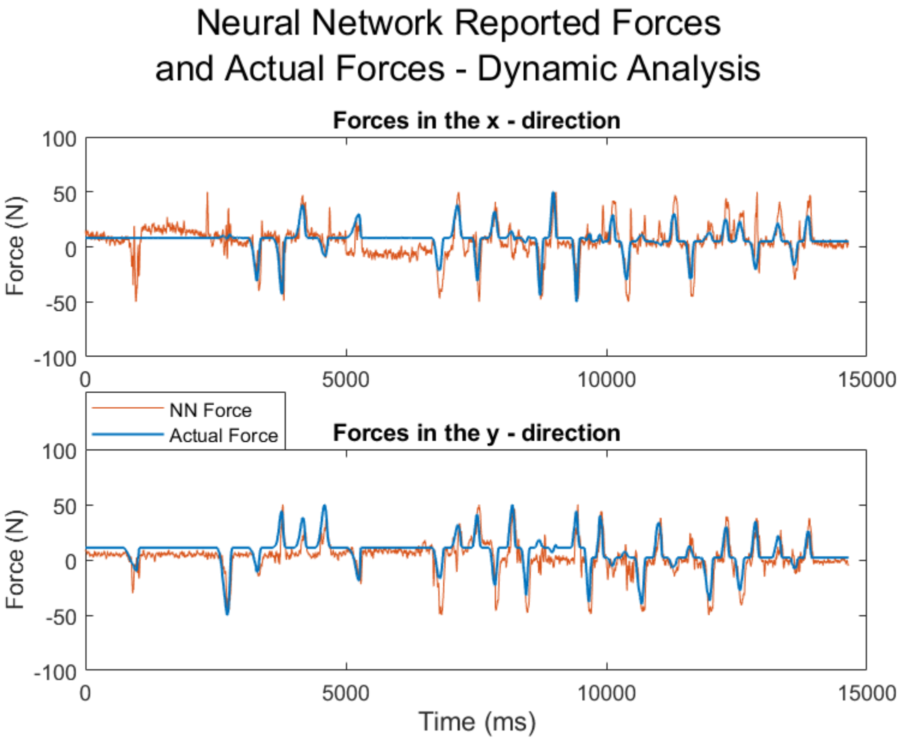

Figure 6.7: *The Response of the Force Sensor with a Trained Neural Network to Dynamic Loading*

6.4.4 Discussion

The ability of the Neural Network to accurately characterise static force data was poor, though it is observed that the force reported by the Neural Network follows the trend of the input force in the principle loading direction. Observing figure 6.6 it can be seen that there are 4 main phases in the loading data. Phase 1 occurs between iteration 1 and circa iteration 1500, where the load in the x-direction increases in force and the load in y-direction is 0N. Phase 2 occurs between circa iteration 1500 and circa iteration 3000, where the load in the x-direction is 0N and the load in the y-direction increases. Phase 3 occurs between circa iteration 300 and circa iteration 4500, where the where the load in the x-direction decreases in force and the load in y-direction is 0N. Finally phase 4 occurs between circa iteration 4500 and circa iteration 6000, where the load in the x-direction is 0N and the load in the y-direction decreases.

In phase 1 the response of the Neural Network shows reasonable agreement with the loading data in the x-direction, but it is observed that in the y-direction the Neural Network response decreases to around -20N as the load in the x-direction increases. This cross-loading response of the Neural Network may be observed in each stage of static load testing. It is apparent that a load in one direction affects the response of the Neural Network in both directions. It is theorised that this may be occurring because the magnetic field is too complex to be properly measured by the array of single axis Hall Effects Sensors. Another concern is that in both the x-direction or the y-direction the Neural Network does not accurately report low forces. It is observed in the x-direction in phase 1, for example, that the Neural Network reports a force of around 10N instead of 0N between iteration 0 and iteration 150 with an input force of 0N. Similar behaviour is observed in the beginning of phase 2 and phase 4. The final point of concern is the significant error seen in phase 4, with a large drop of around 30N reported in the x-direction by the Neural Network at around iteration 4900.

The results of the dynamic test displayed by Figure 6.7 also show that the ability of the Neural Network to accurately characterise static force data was poor, it is noted that the force reported by the Neural Network follows the trend of the input force in the principle loading direction. Similar to the results of the static testing it may be observed that there exists some cross-loading, where load in one direction affects the response of the Neural Network in both directions. This may be seen most clearly at around iteration 1000 where a loading force in the y-direction only produces a good response from the Neural Network in the y-direction and also a large response also in the x-direction.

The dynamic testing results highlight an issue not observed in the results of the static loading results. It may be seen that the forces reported by the Neural Network frequently saturate. This is particularly observed, for example, in the force reported in the x-direction from iteration 10000 to the end of testing. In this testing region the actual loading in the x-direction peak at around 30N and -30N but the Neural Network reports forces which peak at 50N and -50N, which is the maximum magnitude of force that the neural network is trained to characterise.

The inability of the single-axis Hall Effects based force sensor to accurately characterise the loading force presents a safety issue for rehabilitation robotics, since measured forces are used in Admittance control schemes to adjust the motor demands. Unexpected changes to motor demands are likely to introduce instability which in turn may impart dangerous forces onto the user. It is suspected that the magnetic field produced by strong magnets in a 3D arrangement is too complex to adequately characterise using an array of Hall Effects sensors which measure the magnetic field in only direction. This is supported by the presence of cross-loading in the forces reported by the Neural Networks, which suggests that the Hall Effects data is not detailed enough to decouple the x-direction and y-direction forces.

In its current form the single-axis Hall Effects based force sensor is not suitable for use as the integrated end-effector/force sensor for MyPAM.

6.5 Tri-Axis Hall Effects Based Force Sensor

6.5.1 Introduction

The single-axis Hall Effects Based force sensor failed to measure the applied force with sufficient accuracy, both in the static and the dynamic tests. As discussed in Sections 6.4.4 it was theorised that this may be because measuring the magnetic field at only 1 plane normal to each Hall Effects Sensor was not sufficient to characterise the complex 3 Dimensional magnetic field. To this end a similar force sensor was manufactured which used Hall Effects Sensors capable of measuring the magnetic field in 3 planes. It was theorised that measuring the magnetic field in local x, y and z directions for each sensor (a total of 12 measurements) would provide sufficient measurement density to characterise the force-displacement relationship.

6.5.2 Physical Design

The physical design of the Tri-axis Hall Effects based force sensor is derived from the single-axis Hall Effects Based force sensor. The operation method remains unchanged, whereby an outer rigid core embedded with strong magnets moves relative to an inner core embedded with Hall effects Sensors with the application of force. Similarly, the overall geometry and dimensions remain unchanged and the same EcoFlex 00-10 material is used as the deformable separation layer. The internal electronics are more complex, however, since the tri-axis Hall Effects Sensor has a microchip based form-factor which requires interaction through either the I^{2}C or SPI protocols rather than simply outputting an analogue voltage like the Single Axis Hall Effects Sensor. A carrier PCB was created to mount the tri-axis Hall Effects Sensor and facilitate communication through I^{2}C. The internal rigid core was redesigned to provide space for wiring and to securely mount the PCBs, whilst retaining sufficient stiffness to resist bending upon the application of force.

The force sensor assembly uses an aluminium rod as a central core to provide stiffness. 3D printed inserts were used to mount the Tri-axis Hall Effects Sensor PCBs, whilst leaving sufficient space for internal wire routing. Mounted to the 3D printed inserts were an upper and lower section of Aluminium tube. This completed the rigid inner core of the sensor. A rigid outer casing was placed around the inner core assembly and the silicone separation layers were cast using EcoFlex 00-10. Finally, strong Neodymium magnets were attached to the outer core. Figure 6.8 shows an exploded view of the complete assembly, and Figure 6.9 shows a cross section of the complete assembly.

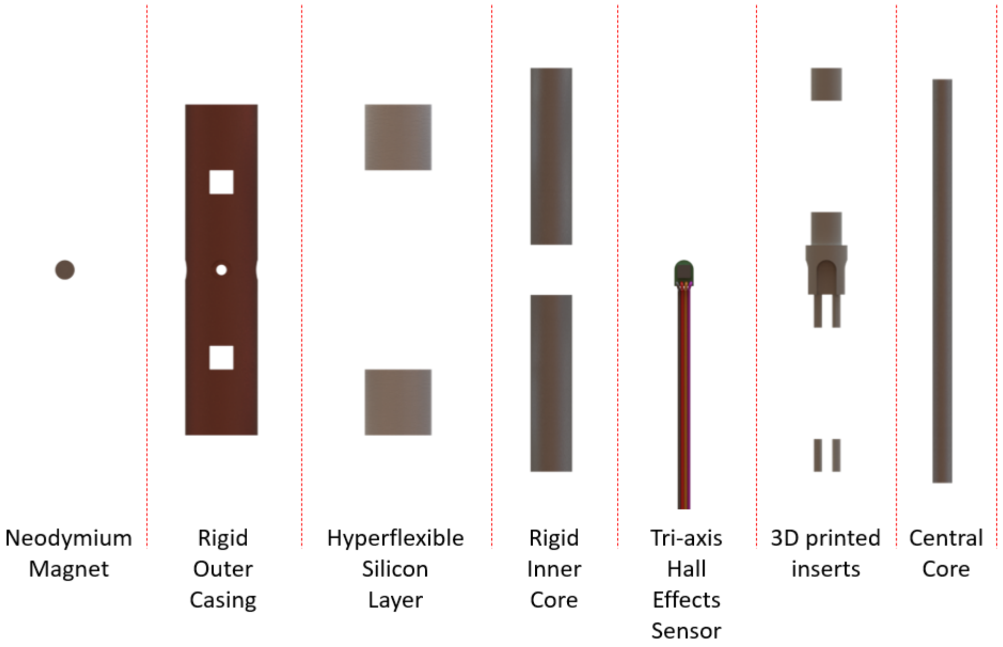

Figure 6.8: *A Solidworks Render of the Force Sensor Components*

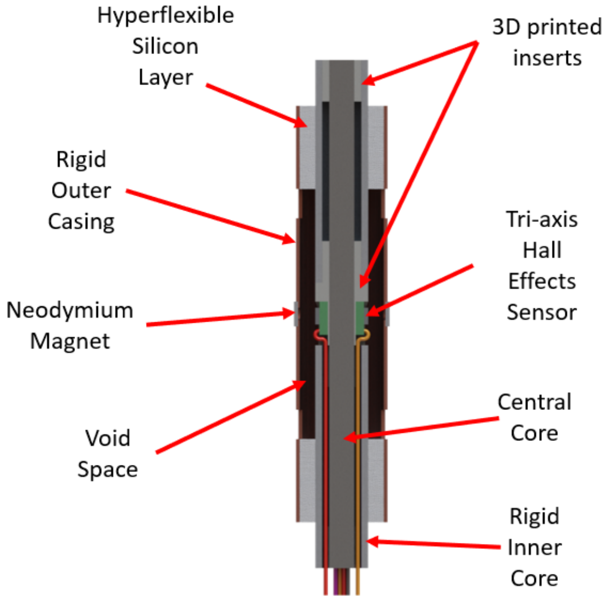

Figure 6.9: *A Solidworks Render of the Force Sensor Cross Section*

6.5.3 System Architecture, Data Acquisition, Signal Conditioning, and Data Transfer

The system architecture was similar to that of the single-axis Hall Effects Based force sensor. Data was acquired from the Hall Effect Sensor array using a Teensy 3.2 with an acquisition rate of 1kHz using the I^{2}C communications protocol. The chip based form factor of the Tri-axis Hall Effects Sensors meant that it was not possible to build a hardware filter, since the output of the chips were digital as opposed to analog. To this end, a digital filter (Hanning moving average filter) was programmed on the Teensy as part of the data pre-processing requirements. The pre-processed data was then transmitted to the myRIO using a separate I^{2}C communication bus. The system architecture is shown by Figure 6.10.

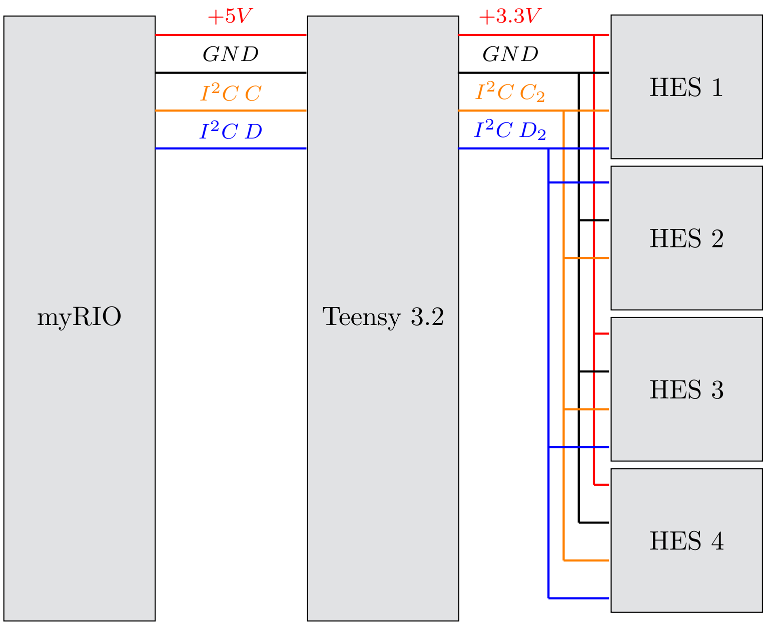

Figure 6.10: *The System Architecture of the Tri-axis Hall Effects Sensor Array.*

6.6 Training the Neural Network for the Tri-axis Hall Effects Based Force Sensor

The Neural Network training process was similar to the process used for the Single-axis Hall Effects Based force sensor, with the main difference in the size of the input layer. A static force dataset consisting of circa 1000000 samples was acquired using the same procedure and equipment discussed in Section 6.3.2 for the purpose of NN topology selection. The input layer for the Tri-axis Hall Effects Based Force Sensor consisted of 24 neurons corresponding to the MSB and LSB of each axis of measurement for each sensor. The size of the output layer remains unchanged at 202 output neurons representing a range of -50N to +50N in 1N increments in the x-direction and -50N to +50N in 1N increments in the y-direction.

The topology, training time and Mean Squared Error of the Neural Networks are shown by Table 6.3. On inspection it may be observed that a shallower Neural Network with a many neurons performs better than a deeper Neural Network with fewer neurons. It may also be seen that a 2 layer network offers the best performance, with reasonable insensitivity to the number of neurons. It may also be observed that the MSE (Mean Squared Error) for every neural network is much poorer than was the case when training the single-axis Hall Effects Based force sensor by three or four orders of magnitude.

Table 6.3: *The Tri-Axis Neural Network Topology, Training Time and Mean Squared Error for the Nine Neural Networks.*

Neural Network	Hidden Layer 1 Neurons	Hidden Layer 2 Neurons	Hidden Layer 3 Neurons	Hidden Layer 4 Neurons	Hidden Layer 5 Neurons	Hidden Layer 6 Neurons	Hidden Layer 7 Neurons	Hidden Layer 8 Neurons	Training Time (hrs)	Mean Squared Error
1	1000								22	0.00157
2	500	500							32	0.00137
3	202	202	202						36	0.00143
4	202	202							13	0.00141
5	50	50	50	50					13.5	0.00172
6	20	20	20	20	20	20	20	20	16.5	0.00438
7	202	404	202						11.5	0.00266
8	24	26	28	30	40	70	100	202	19.5	0.00215
9	24	22	20	16	14	8	4	2	14	0.00371

In every case the Neural Network training was abandoned before reaching the desired MSE target due to the test data divergence, which is seen the performance plot for Neural Network 7 shown by Figure 6.11. This divergence of MSE between the training data and the test data is known as overfitting and has a number of potential causes including noisy data and too complex a Neural Network. Techniques to prevent overfitting, include simplifying the NN and applying regularisation to the weight training function, were applied but no improvements could be made. For the purposes of testing Neural Network 2 was selected. A Neural Network was trained using the Static Force training data and second Neural Network was trained using the Dynamic Force training data, consisting of 1000 000 samples and acquired according to the methodically documented is section 6.3.2.

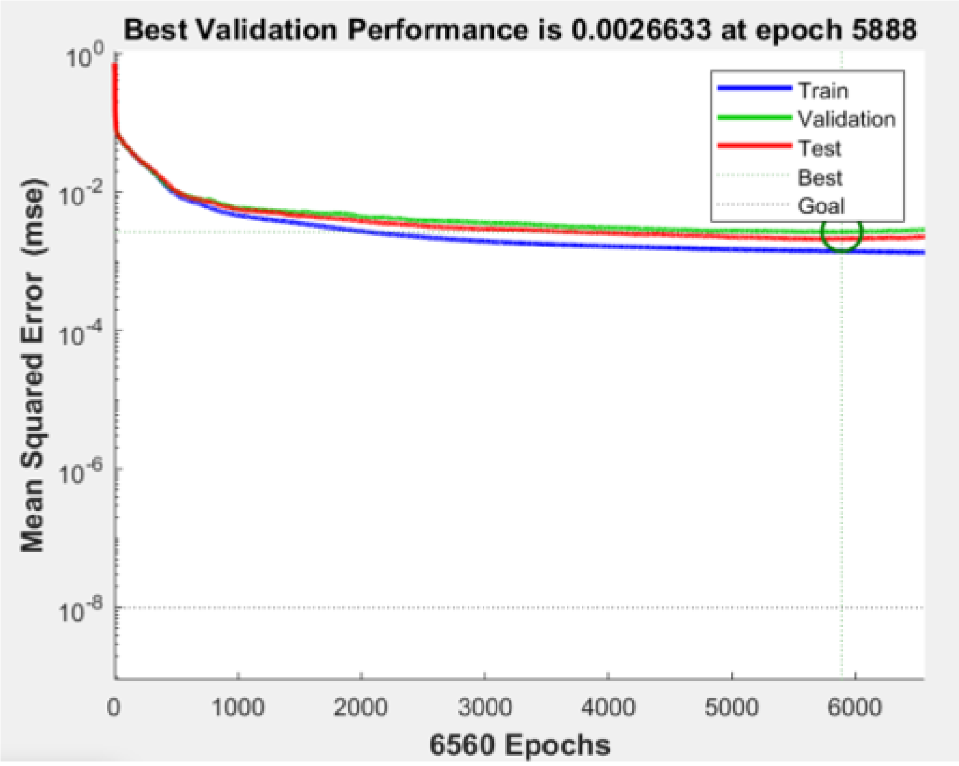

Figure 6.11: *Test Data Divergence During Neural Network Training. Note That the Mean Squared Error for the Test Data Increases After Epoch 5888 While the Mean Squared Error for the Training Data Continues to Decrease.*

6.7 Tri-axis Hall Effects Based Force Sensor Validation Experiment

6.7.1 Introduction

To evaluate the proficiency of the force sensor a pair of experiments were performed, the first evaluating static loading and the second evaluating dynamic loading. For each test the respective trained Neural Network was built into the force sensor architecture and the response of the force sensor was tested against known loads. Given that the MSE were four orders of magnitude worse than were observed when training the Neural Networks for the single-axis Hall Effects Based force sensor it was anticipated that the performance of the tri-axis Hall Effects Based force sensor would be significantly worse.

6.7.2 Methodology

Static Force Validation

The tri-axis Hall Effects Based Force Sensor was loaded into the Static force test rig described in Section 6.3.2 and the loading procedure used to obtain the Static Force training data set was repeated. The predicted components of force in the X-direction and Y-direction output by the trained Neural Network were logged alongside the loaded components of force in the X-direction and Y-direction for comparison.

Dynamic Force Validation

A similar procedure was followed as above. The tri-axis Hall Effects Based Force Sensor was loaded into the Dynamic force test rig described in Section 6.3.3 and the loading procedure used to obtain the Dynamic Force training data set was repeated. The predicted components of force in the X-direction and Y-direction output by the trained Neural Network were logged alongside the loaded components of force in the X-direction and Y-direction for comparison.

6.7.3 Results

The response of the force sensor to static loads is shown by Figure 6.12.

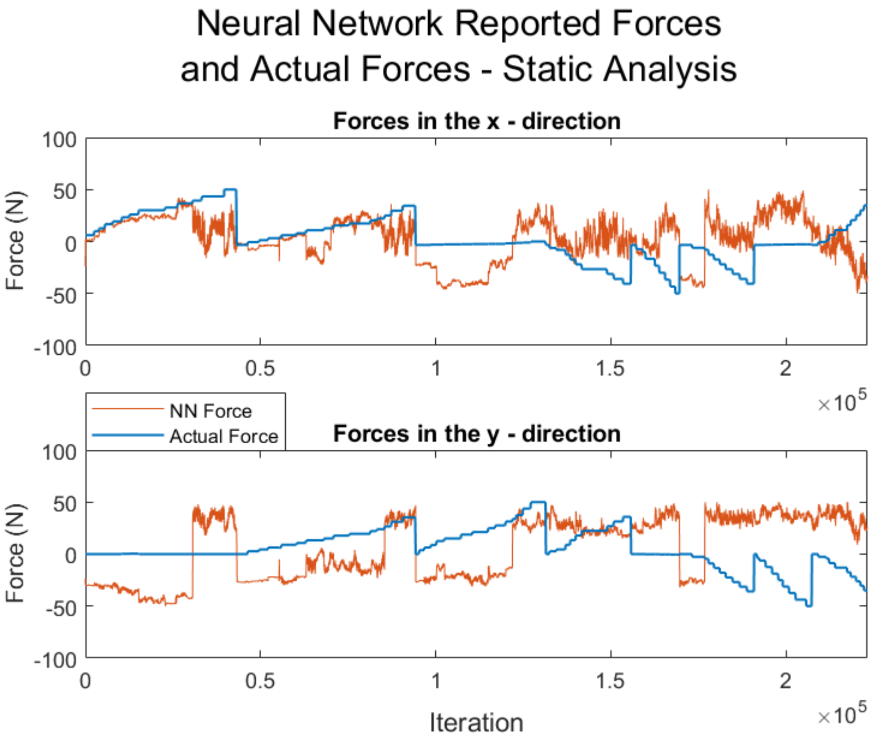

Figure 6.12: *The Response of the Force Sensor with a Trained Neural Network to Static Loading*

The response of the force sensor to dynamic loads is shown by Figure 6.13.

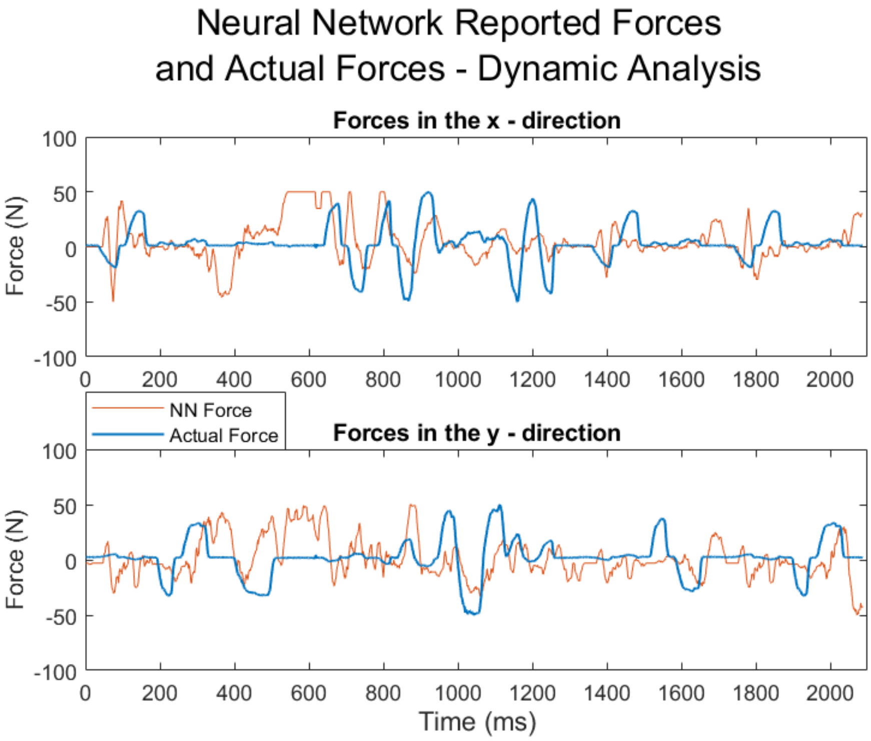

Figure 6.13: *The Response of the Force Sensor with a Trained Neural Network to Dynamic Loading*

6.7.4 Discussion

The static test results displayed by Figure 6.7 show small periods where the tri-axis Hall Effects-based force sensor response poorly follows the trend of the input force, the best example may be observed in the x-direction between iteration 0 and 30000. However, the tri-axis Hall Effects-based force sensor response mostly shows seemingly random noise with a little apparent relation to the input force. In the y-direction the tri-axis Hall Effects-based force sensor response mainly fluctuates between large negative predictions of force and large positive predictions of force and shows no relation to the input force. The dynamic test results displayed by Figure 6.13 shows few periods where the response of the tri-axis Hall Effects-based force sensor follows the trend of the input force, and mostly shows a large and seemingly random error.

In all cases the ability of the Neural Networks to characterise the force data was poor. The target Mean Squared Error during training was 1×10^{-8} and none of the Neural Networks could be trained beyond a Mean Squared Error of magnitude 1×10^{-3}. This is significantly worse than the Mean Squared Errors of magnitude 1×10^{-7} observed in the Neural Networks trained with the 1 DoF Sensors. In each case the training was ended due to the divergence of the validation data, shown by the characteristic positive gradient of the validation data curve after epoch 5888 in Figure 6.11. This is typical of overfitting of a NN, which may have a number of causes:

1. **The NN attempts to predict a trend in excessively noisy data** [125, 126].

   Sensor measurement noise was noted during the data acquisition phases for the static and dynamic testing. The tri-axis Hall Effects sensors, as previously mentioned in Section 6.5.3 are interacted with via I^2C, requiring that a digital filtering method was rather than simply building a hardware filter. The sensor array outputs twelve measurements (twenty four if it is taken into account that each measurement is read as two separate bytes from different registers on the chip) and it is likely the case that the noise has had some effect on the accuracy of the Hall Effects data, since there are twelve different values which may contain variation for the same force input, making it difficult for the NN to characterise the input force.

2. **The NN is too complex** [127].

   The complexity of the NN is unlikely to be the cause of overfitting in this case, since overfitting was observed in every NN trained during the optimisation phase described in section 6.6, even NN1 which contained just a single hidden layer. It was considered that the NNs had potentially reached local minima rather than their respective global minima. This was considered unlikely, since different multiple NNs had reached similar conclusions in terms of the order of magnitude of the MSE. A further test was however carried out whereby NN2 was retrained with the static force test data for an additional five times (over a period of around three weeks). No further improvement of the MSE was gained and the overfitting was observed each time.

3. **Undersampled training data** [128].

   Both the static and dynamic NNs were provided with a training dataset containing 1 000 000 samples. It is difficult to ascertain whether the datasets are sufficient for training the tri-axis Hall Effects-based force sensor, but in practical terms increasing the size of the dataset would increase training time. Indeed, training the best performing NN for force sensor took 120 hours with an input layer of only 4 neurons since there was no early stopping. In the case that increasing the size of the training dataset of the tri-axis Hall Effects sensor based should prevent overfitting and thus prevent early stopping there is no way to predict how long the training would take. Further to this there is no way to evaluate an appropriate size for the training dataset.

A highly likely cause of the failure of the force sensor is the movement of the outer rigid core relative to the inner rigid core. Since the force sensor is attempting to measure force in 2 axes, it was only necessary that the outer core more relative to the inner core in 2 axes. The motion is poorly constrained, however, leaving the outer core able to move relative to the inner core in 6 axes. This is likely the cause of overfitting also, which is fundamentally caused by a mismatch in the complexity of the NN compared with the complexity of the model its attempting to characterise. In its current form the tri-axis Hall Effects based force sensor is not suitable for use as the integrated end-effector/force sensor for MyPAM.

6.8 Progress Made and Future Work

The bulk of the work performed in this Chapter and Chapter 5 occurred at home away from the laboratory environment due to the Covid 19 lockdown, which limited the potential progress on the force sensor and directed the calibration approach towards a 'back box' Neural Network. The grip sensor presented in Chapter 5 was successful at characterising three distinct states. It is conceivable that this success could be extended to the 202 separate states represented by the output layer of the Neural networks which were tested, and it is unfortunate that the relative movement of the cores was not sufficiently constrained to only the two axes of measurement. The first obvious next step is to revisit the mechanical design of the integrated handle/sensor to ensure that all relative movement of the inner and outer cores is constrained to only the axes in which force is desired to be measured. From here there are clear routes to progress this work in the future.

The first route is to sufficiently characterise the mechanical properties of the hyperelastic silicone and the magnetic properties of the magnets in that complex arrangement. Following this there should be a period of modelling and analysis to fully characterise the complete assembly, allowing the measurements from the Hall Effects Sensors to be directly converted to force measurements in the form shown by Equation 6.1:

$$F_x, F_y = f(Sensor_1, Sensor_2, Sensor_3, Sensor_4) \tag{6.1}$$

Characterising the assembly is likely to be a difficult process because of the variable properties of the silicone, but should a sufficiently robust procedure be designed for mixing and casting the silicone it is certainly possible. It must be noted that the characterisation of MagOne required considerable effort. However, there would likely remain the need for a calibration procedure of each complete assembly before use due to small differences and tolerances in the manufacture of each unit.

The second route is that the experimental procedure detailed in this chapter be repeated, with the hope of a better outcome once the cores of the assembly are properly constrained. The third route is to explore a different machine learning methodology, with Genetic Algorithms as a good potential candidate. In practice there is no reason that all three routes shouldn't be pursued simultaneously.

6.9 Chapter Summary

In this Chapter the grip sensor developed in Chapter 5 was adapted into a 2-axis force sensor intended for use as the integrated end-effector/force sensor for MyPAM. Section 6.2 presents the first iteration, based on an array of single axis Hall Effects sensors. Section 6.3 presents the training of the Neural Network used to convert the Hall Effects data into a 2-axis force output. Section 6.4 presents a validation experiment where it was determined that the single-axis Hall Effects based force sensor was not capable of characterising force well enough to be suitable for MyPAM. Section 6.5 presents the second iteration, based on an array of tri-axis Hall Effects sensors. Section 6.6 presents the training of the Neural Network used to convert the Hall Effects data into a 2-axis force output. Section 6.7 presents a validation experiment where it was determined that the tri-axis Hall Effects based force sensor was not capable of characterising force and is not suitable for use with MyPAM. All three of the Chapter objectives were met, and it was determined that the sensor was not yet capable of measuring force accurately and was inappropriate for use with MyPAM at this current time. The status of the chapter objectives is shown by Table 6.4.

Table 6.4: *Chapter 6 Objectives Status*

Objective	Description	Success?
6.1	Develop a low-cost integrate force sensor/handle to measure force in 2 axes.	Yes.
6.2	Measure the performance of the force sensor against known dynamic loads to determine viability of the product.	Yes.
6.3	Measure the performance of the force sensor against known dynamic loads to determine viability of the product.	Yes.

Chapter 7

Control

In this Chapter the control strategy of MyPAM is documented. Section 7.1 introduces the control problem for MyPAM, discusses the difficulty in control design and lays out the chapter objectives. Section 7.2 presents the Position Control strategy. Section 7.3 presents the Admittance Control Scheme, including the derivation of the Admittance filter and Admittance Control implementation. Section 7.4 presents the validation of the Admittance Control scheme and shows that it is able to regulate the position demand in response to force inputs. Section 7.5 presents a mechanism used to protect against instability in MyPAM, with experiments validating the operation of the instability protection shown in Section 7.6.

7.1 Introduction

Stable control of a rehabilitation robot such as MyPAM is of critical importance, since if MyPAM enters an unstable state it may impart dangerous or painful forces onto the patient. The main difficulty in designing a control system of a rehabilitation robot is the unpredictable nature of external inputs, namely the interaction between the robot and the patient.

For a linear system there are well defined procedures for designing a control system. Using Classical control theory, the first stage of controller design requires obtaining the equations of motion of the system [129]. A transfer function representing the system model in the frequency domain may be obtained from the equations of motion using Laplace transforms. The transfer function is used to analyse the response of a system and is necessary for the design of a controller. The Classical control theory methodology only works for Single Input Single Output (SISO) linear systems however. Modern Control theory may be used for Multiple Input Multiple Output (MIMO) linear systems, allowing a State Space description to be formed [130]. A State Space description presents the the dynamics of the system as a set of coupled 1st order differential equations, known as state variables, with a set of algebraic equations which combine the state variables into physical output variables. State Space theory operates entirely in the time domain.

Analysing MyPAM produces a model which is non-linear, as shown in Chapter 4, which means that the modelling techniques described above are not applicable for designing the control system of MyPAM. Techniques exist for the control of non-linear dynamic systems. One accepted method for non-linear control is to linearise the behaviour of the plant and to apply traditional linear control methods, but this is only effective around a chosen operating point, with large deviations often leading to instability. Linearising around a fixed point requires the point to remain unchanged. For example, consider the case of an inverted pendulum. There exist two fixed points in the workspace where the pendulum will remain stable with no input forces required. The first is vertically downwards and the second is vertically upwards. In this simple example the workspace is small and the presence of gravity allows intuitive deduction of the location of appropriate fixed points for linearisation. In the case of MyPAM there are two main difficulties in linearising around a fixed point. Firstly, in the workspace there is no fixed point appropriate for all possible trajectories. Secondly, it is not possible to predict the magnitude or the direction of disturbances introduced by a patient holding the end effector and applying force. It is for this reason that rehabilitation robots generally select Impedance or Admittance control as the control paradigm as discussed in Chapter 2. Admittance and Impedance control rely on the use of force sensors, which add significant cost to the robot.

7.1.1 Chapter Objectives

This Chapter aims to fulfil the following objectives:

Objective 7.1	Describe the Position Control scheme for MyPAM.
Objective 7.2	Design and validate an Admittance control scheme for MyPAM.
Objective 7.3	Design and validate instability protection for MyPAM.

7.2 Position Control

Previous iterations of MyPAM have successfully implemented position control to assist the patient to perform reaching exercises. PI control was used to minimise the error between the position of the end effector and each successive setpoint of the target trajectory. A derivative gain was not used because it introduces instability, which is likely due to the discretised Minimum Jerk Trajectory which generates a new setpoint at a rate of 1kHz.

As modelled and described in Section 4.6 in Chapter 4, there are two distinct operating conditions for MyPAM which have different dynamics. The first operating condition is when there is no user holding the end effector of MyPAM (the unloaded condition). The second operating condition is when there is a user holding the end effector of MyPAM (the loaded condition). The development of the grip sensor in Chapter 5 allows MyPAM to distinguish the presence of a user and apply different gains to account for the different dynamic condition. The PI gains were tuned heuristically and are $P = 1, I = 0.1$ in the unloaded condition and $P = 1.4, I = 0.1$ in the loaded condition. The position control scheme follows the process illustrated by Figure 7.1.

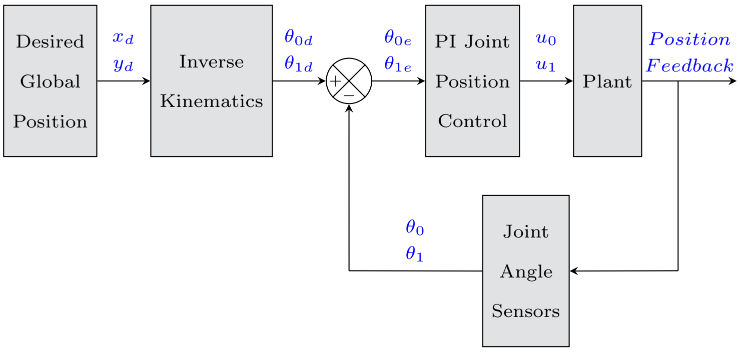

Figure 7.1: *The Position Control Scheme Block Diagram for MyPAM.*

The Position Control scheme follows the process:

1. The next setpoint x_d, y_d is generated by the trajectory generation algorithm.
2. Inverse kinematics are applied to the position demand x_d, y_d, creating the joint position demands θ_{0d}, θ_{1d}.
3. The joint position demands θ_{0d}, θ_{1d} are compared with the actual joint positions θ_0, θ_1 to calculate the joint position errors θ_{0e}, θ_{1e}.
4. PI control is applied independently to each of the joint position errors θ_{0e}, θ_{1e}, producing the motor demands u_0, u_1.
5. The motor demands u_0, u_1 are applied to MyPAM.
6. The position of the joints of MyPAM are measured by the encoders, producing θ_0, θ_1.

The Position Control scheme when loaded has been extensively tested in Chapter 3, where twenty participants tested two different trajectory generation strategies, and Chapter 4 during validation of the model. The Position Control scheme when unloaded has been tested in Chapter 3 when testing the effect of Attractors and Repulsors on the trajectory and in Chapter 4 during validation of the model.

7.3 Admittance Control

Admittance control is frequently used as the low-level control strategy for a rehabilitation robot because it solves the traditional control problem of disturbance rejection. In position control alone the force applied by the user to end-effector is treated as an unwanted disturbance, and a position control system which is insufficiently robust may enter an unstable state as a response. The benefit of using Admittance control is that the patient interaction force is used as a control input, rather than being considered as a disturbance which must be rejected. This allows the use of only one control regime for MyPAM, based on the position control loop of the unloaded condition described in Section 7.2 as opposed to the two control regimes selected by the measurement of a grip sensor.

7.3.1 Deriving the Admittance Filter

The Admittance filter takes the form shown by Equation 7.1.

$$x_i = \frac{F_x}{ms^2 + cs + k} \tag{7.1}$$

Where:

$x_i = Change\ in\ x\ position$

$F_x = x\ component\ of\ force$

$m = Virtual\ mass$

$c = Virtual\ damping\ coefficient$

$k = Virtual\ spring\ constant$

Mass may be omitted since inertial effects are undesirable, thus a simplified spring-damper arrangement is considered. Consider the input force $F_x(t)$ as a step input with magnitude γ, shown by Equation 7.2.

$$F(S) = \mathcal{L}\{f(t)\} = \frac{\gamma}{s} \tag{7.2}$$

The transfer function G(S) for a spring-damper system takes the form shown by Equation 7.3.

$$G(S) = \frac{A}{1 + \tau s} \tag{7.3}$$

Where:

$A = 1/k$

$\tau = c/k$

The output X(S) is therefore given by Equation 7.4.

$$X(S) = F(S)G(S) = \frac{A\gamma}{s(1 + \tau s)} \tag{7.4}$$

The outcome of the partial fraction decomposition is given by Equation 7.7.

$$\frac{A\gamma}{s(1 + \tau s)} = \frac{D}{s} + \frac{E}{1 + \tau s} \tag{7.5}$$

$$\frac{A\gamma s(1 + \tau s)}{s(1 + \tau s)} = \frac{Ds(1 + \tau s)}{s} + \frac{Es(1 + \tau s)}{1 + \tau s} \tag{7.6}$$

$$A\gamma = D(1 + \tau s) + Es \tag{7.7}$$

The outcome of collecting terms is given by Equation 7.8.

$$A\gamma = D + s(D\tau s + E) \tag{7.8}$$

Equating the coefficients s^0 is shown by Equation 7.9.

$$D = A\gamma \tag{7.9}$$

Equating the coefficients s^1 is shown by Equation 7.11.

$$0 = D\tau + E = A\gamma\tau + E \tag{7.10}$$

$$\rightarrow E = -A\gamma\tau \tag{7.11}$$

Producing the Admittance relationship shown by Equation 7.14.

$$X(S) = \frac{A\gamma}{s(1+\tau s} = \frac{A\gamma}{s} - \frac{A\gamma\tau}{1+\tau s} \tag{7.12}$$

$$\frac{A\gamma}{s} - \frac{A\gamma\tau}{1+\tau s} = \frac{A\gamma}{s} - \frac{A\gamma\tau}{1+\tau s}\frac{1/\tau}{1/\tau} = \frac{A\gamma}{s} - \frac{A\gamma}{s+1/\tau} \tag{7.13}$$

$$x(t) = \mathcal{L}^{-1}\{X(S)\} = A\gamma\left(1 - e^{-\frac{1}{\tau}t}\right) \tag{7.14}$$

Substituting $A = 1/k$ and $\tau = c/k$ produces final Admittance filter function shown by Equation 7.15.

$$x_i(t) = \frac{1}{k}F_x\left(1 - e^{-\frac{k}{c}t}\right) \tag{7.15}$$

$x_i = Change\ in\ x\ position$

$F_x = x\ component\ of\ force$

$c = Virtual\ damping\ coefficient$

$k = Virtual\ spring\ constant$

The Admittance filter shown by Equation 7.15 describes the reaction to a force input of a first order system over time. For this to be useful it is necessary for the force input to remain constant for at least the length of the settle time, which is usually evaluated as equal to 4 times the time constant τ. Indeed, consider Figure 7.2 which shows the application of Equation 7.15, where it may be observed that the input force must be held constant for two seconds to be considered settled.

In the context of a rehabilitation robot such as MyPAM this doesn't make a lot of sense since the force input will change unpredictably and is highly unlikely to be constant. Further to this it is not apparent what value of time t to use as an input to the Admittance filter. Two options were considered for the selection of the value for time t:

1. Monitor the force input and if it is the same for a number of iterations (within limits) increase the value for time t from $t = 0$. If the force changes, reset the value to $t = 0$.
2. Apply a static constant value for time t.

Option one above was rejected for two reasons. First, the implementation would add unnecessary programming complexity and it is important to preserve the limited computational resources of the myRIO where possible. More importantly, however, option one was considered likely to render the Admittance filter useless since the input force is likely to vary considerably during use of MyPAM. A quickly varying force input would result in the Admittance filter always acting at time $t = 0$, thus not modulating the the target position at all since $x_i(t) = 0$ when $t = 0$.

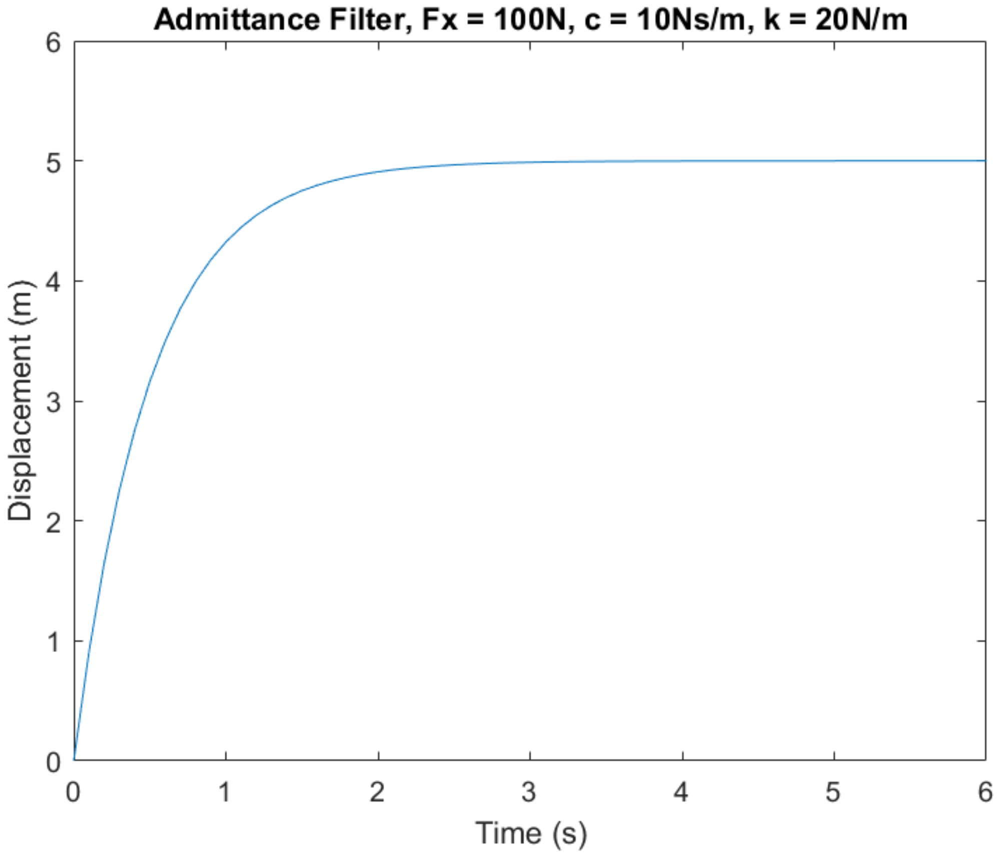

Figure 7.2: *Admittance Filter Response - Displacement Against Time.*

Considering option two above requires exploration of an acceptable value for time t. Applying time $t = 0$ has the undesirable result that $x_i = 0$, as is the likely outcome when considering option one. Applying a very large value for time t results in the Admittance filter acting simply as a virtual spring (an implementation of Hooke's law), since as time t tends to infinity the exponential component of Equation 7.15 tends to 0. This method was used in the MEMOS system, a rehabilitation robot designed to be low cost in an approach similar to MyPAM [72]. Applying time $t = 1$ means that the output of the Admittance filter depends of the values of coefficients c and k. This method was used in iPAM [79], where coefficient c was used as an input to adjust the level of assistance provided to the patient. In MyPAM the level of assistance is already adjusted by a gain (ranging from 0% - 100%) applied to the motor demands u_0, u_1, so a second method of assistance adjustment is unnecessary. Since there already exists a method to modulate assistance level in MyPAM, the Admittance filter was set as a simple virtual spring arrangement to preserve computational resources. A separate filter is used for the global x and global y directions, as shown by Equations 7.16 and 7.17 respectively.

$$x_i = \frac{1}{k} F_x \tag{7.16}$$

$$y_i = \frac{1}{k} F_y \tag{7.17}$$

7.3.2 Interaction Force Measurement

Admittance Control relies on measurement of the interaction force between the patient and the end effector. The force sensor developed in Chapter 6 was not satisfactory, so for the purpose of testing the Admittance control scheme for MyPAM a JR3 6-axis Force/Torque sensor was used. The JR3 was rigidly mounted to the end-effector so that the measurement of force in the y-direction was parallel to Link 2 of MyPAM, as shown by Figure 7.3.

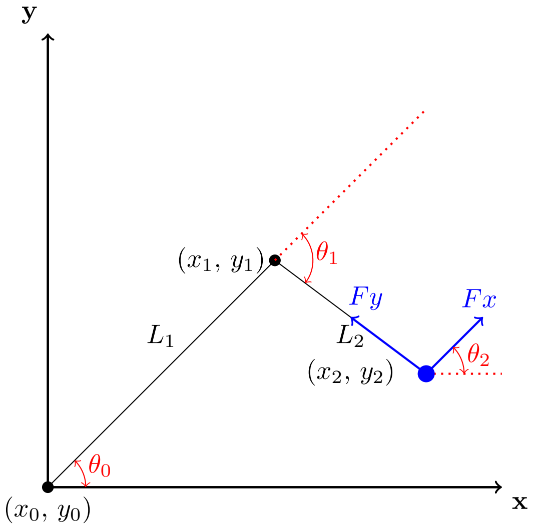

Figure 7.3: *The Location and Orientation of the Force Sensor at the End Effector of MyPAM.*

It was necessary to resolve the force sensor measurements relative to the global coordinate system. The angle θ_2 is given by Equation 7.18.

$$\theta_2 = \theta_0 + \theta_1 + \frac{\pi}{2} \tag{7.18}$$

The resolved forces in the global coordinate system are given by Equations 7.20 and 7.20.

$$RF_x = F_x \cos\theta_2 - F_y \sin\theta 2 \tag{7.19}$$

$$RF_y = F_y \cos\theta_2 + F_x \sin\theta 2 \tag{7.20}$$

7.3.3 Implementation of Admittance Control

Admittance Control relies on an inner Position Control loop which is modulated by an out force compensation loop. The block diagram for the Admittance Control scheme in MyPAM is shown by Figure 7.4, where the inner Position Control loop is that which was previously described in Section 7.2.

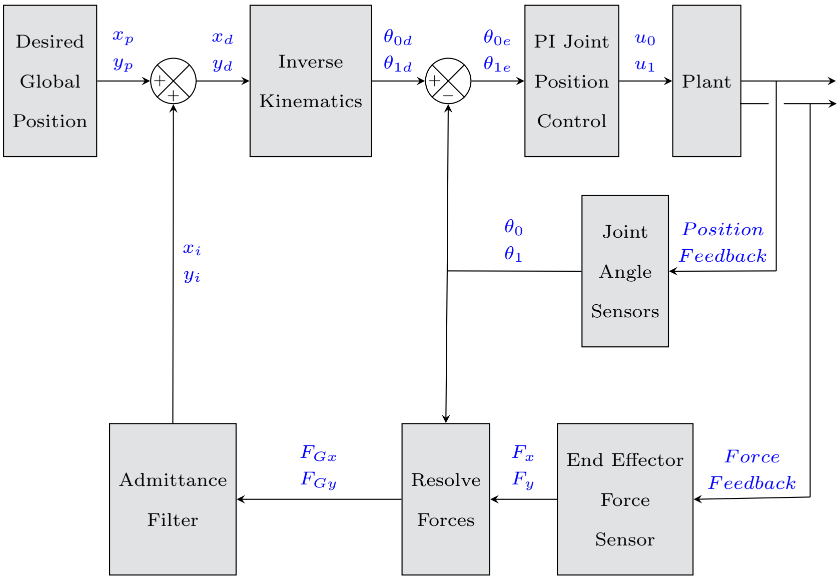

Figure 7.4: *The Admittance Control Scheme Block Diagram for MyPAM.*

The Admittance Control scheme follows the process:

1. The next setpoint x_p, y_p is generated by the trajectory generation algorithm.
2. The output of the Admittance filter x_i, y_i is applied, adjusting the setpoint into the new position demand x_d, y_d.
3. Inverse kinematics are applied to the position demand x_d, y_d, creating the joint position demands θ_{0d}, θ_{1d}.
4. The joint position demands θ_{0d}, θ_{1d} are compared with the actual joint positions θ_0, θ_1 to calculate the joint position errors θ_{0e}, θ_{1e}.
5. PI control are applied independently to each of the joint position errors θ_{0e}, θ_{1e}, producing the motor demands u_0, u_1.
6. The motor demands u_0, u_1 are applied to MyPAM.
7. The position of the joints of MyPAM are measured by encoders, producing θ_0, θ_1.
8. The force measured at the end effector of MyPAM are measured, producing local components of force F_x, F_y.
9. The global components of force F_{Gx}, F_{Gy} are resolved using the local components of force F_x, F_y and joint positions θ_0, θ_1.
10. The Admittance filter is applied to the global components of force F_{Gx}, F_{Gy}, producing the change in position demand x_i, y_i.

7.4 Experimental Validation of Admittance Control

7.4.1 Introduction

MyPAM was subjected to testing to validate the effect of the Admittance filter in the Admittance Control scheme defined in Section 7.3.3. Since the Admittance filter consists of only a virtual spring, adjusting the value of coefficient k should alter the position demand in a predictable way. Two tests were performed. The first test was used to identify an appropriate value for the virtual spring constant k and the second test was used to compare the response of MyPAM in the loaded condition to three different control strategies when connected to the human arm proxy. In the second test it was expected that the Admittance Control scheme would modulate the intermediate target positions of the Minimum Jerk trajectory, resulting in an undershoot of the final target of a magnitude similar to the constraining resulting from connection to the human arm proxy.

7.4.2 Methodology

Two tests were performed to evaluate the performance of the Admittance Control. The first test was a static test used to identify an appropriate value for the virtual spring constant k. MyPAM was positioned at its usual start position ($x = 354mm, y = 126.2mm$) and its movement was constrained. Force was applied to the end effector and the resolved components of input force and resultant change in displacement as the output of the admittance filter were logged. Three values of coefficient k were tested: $k = 1$, $k = 10$ and $k = 100$. The error between change in displacement output from the Admittance filter and the expected change in displacement was evaluated.

The second test was a dynamic test. The Human Arm Proxy previously used in the model validation in Section 4.6 of Chapter 4 was connected to the end effector of MyPAM. MyPAM was placed in its default start position ($x = 354mm, y = 126.2mm$). The target point was set to $x = 354mm, y = 126.2mm$ and a discretised Minimum Jerk trajectory was used as the trajectory generation strategy. A deadzone with radius 10mm was placed at the target point. Three control conditions were tested, all in the loaded condition with the human arm proxy connected to MyPAM:

1. The Position Control scheme with gains $P = 1$ $I = 0.1$, which was designed for the unloaded condition. This was necessary since it provides a baseline to compare the Admittance Control scheme which uses the Position Control scheme with gains $P = 1$ $I = 0.1$ as the inner loop.

2. The Position Control scheme with gains $P = 1.4$ $I = 0.1$, which was designed for the loaded condition. This was necessary since it provides a similar use case to compare the Admittance Control scheme against.

3. The Admittance Control scheme with inner position loop gains $P = 1$ $I = 0.1$ and virtual spring constant $k = 1$.

In the second test five repeats were taken for each control system response. Position data and the end effector force data were logged throughout.

7.4.3 Results

The results for Admittance test 1 are shown by Figures 7.5,7.6,7.7,7.8,7.9 and 7.10.

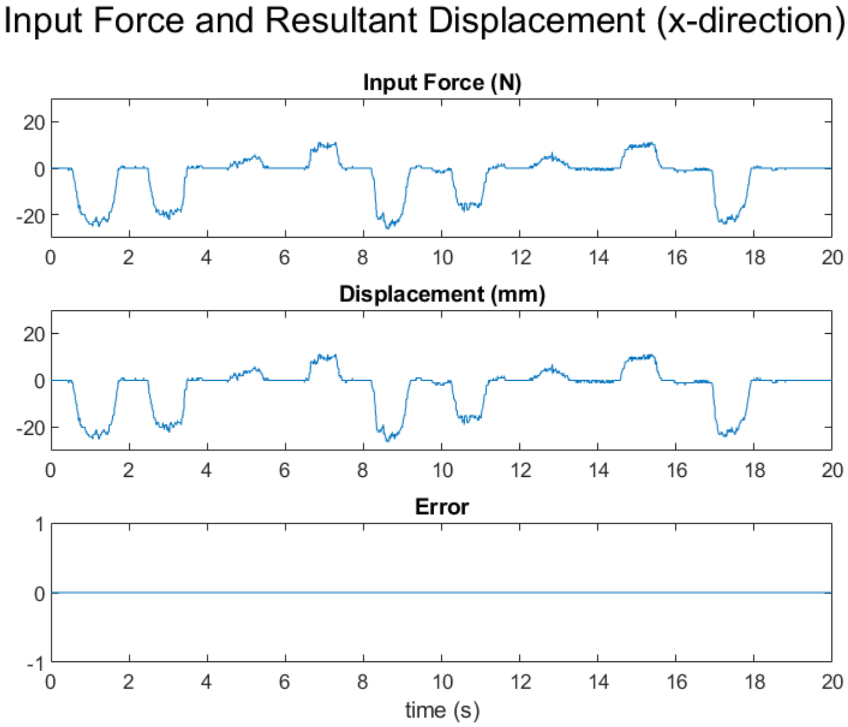

Figure 7.5: *The X component of Input Force and Resultant Change in X-Displacement Demand for Admittance (k=1)*

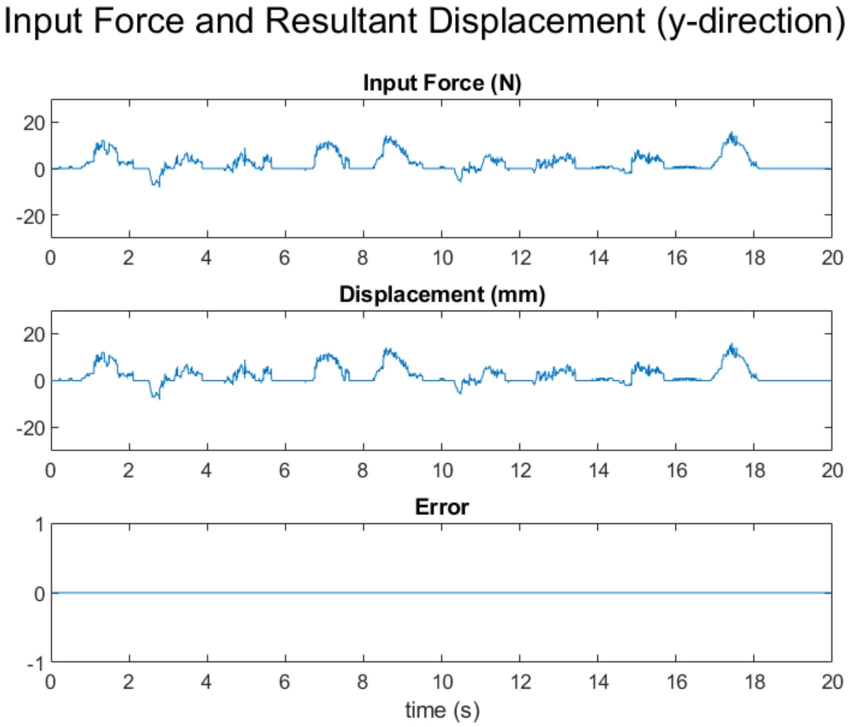

Figure 7.6: *The Y component of Input Force and Resultant Change in Y-Displacement Demand for Admittance (k=1)*

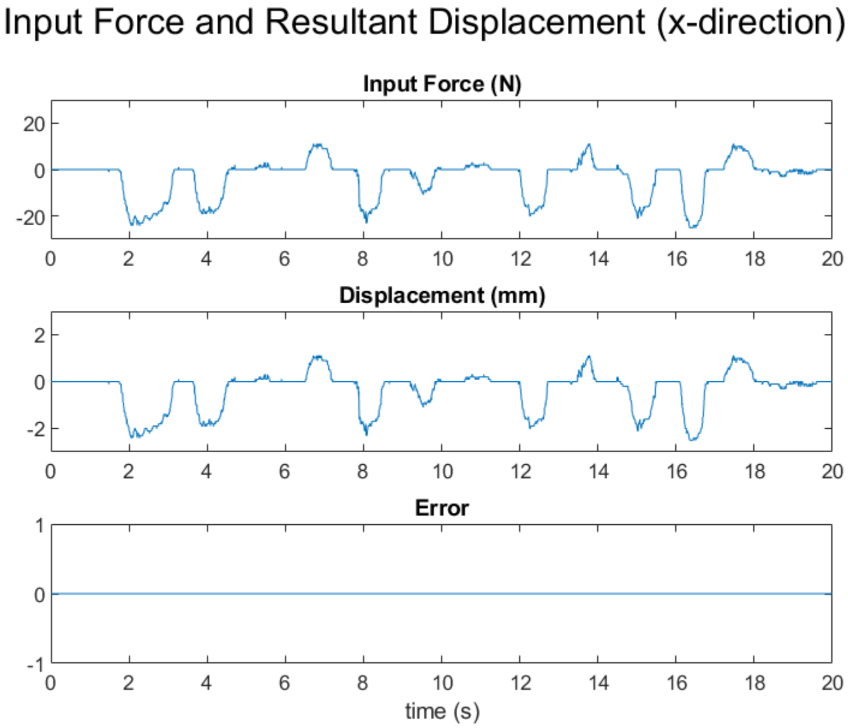

Figure 7.7: *The X component of Input Force and Resultant Change in X-Displacement Demand for Admittance (k=10)*

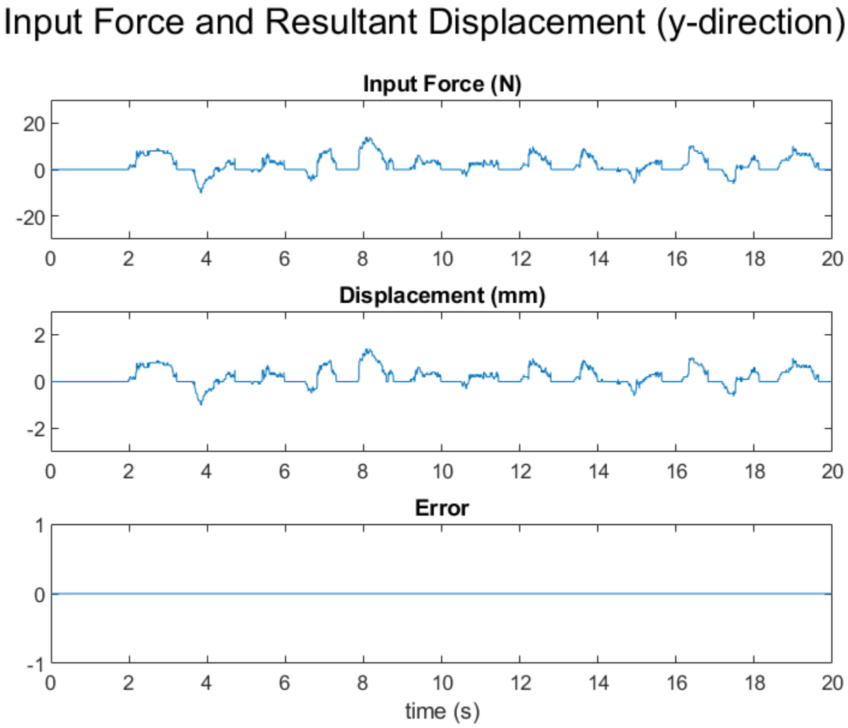

Figure 7.8: *The Y component of Input Force and Resultant Change in Y-Displacement Demand for Admittance (k=10)*

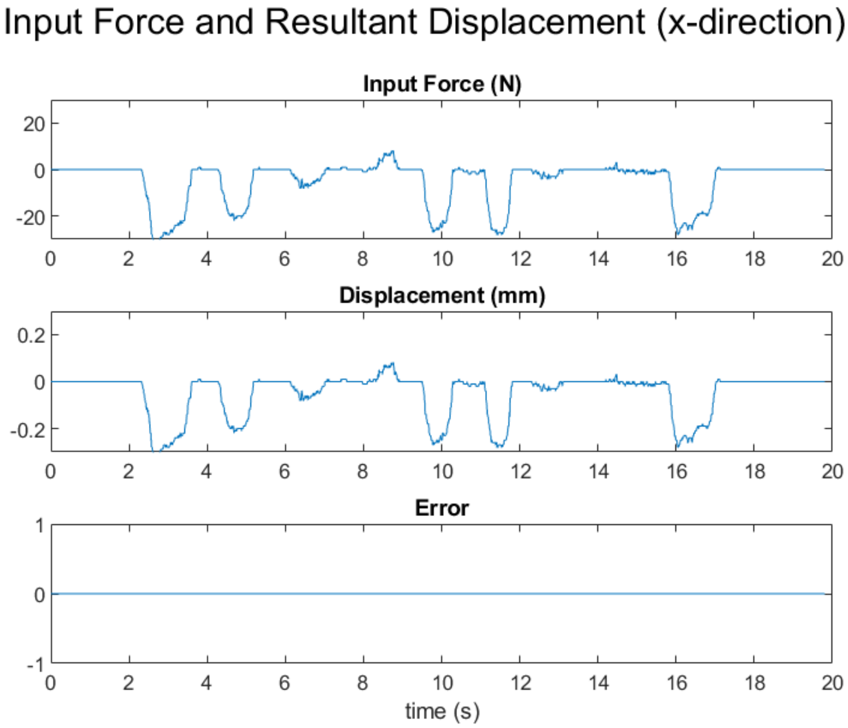

Figure 7.9: *The X component of Input Force and Resultant Change in X-Displacement Demand for Admittance (k=100)*

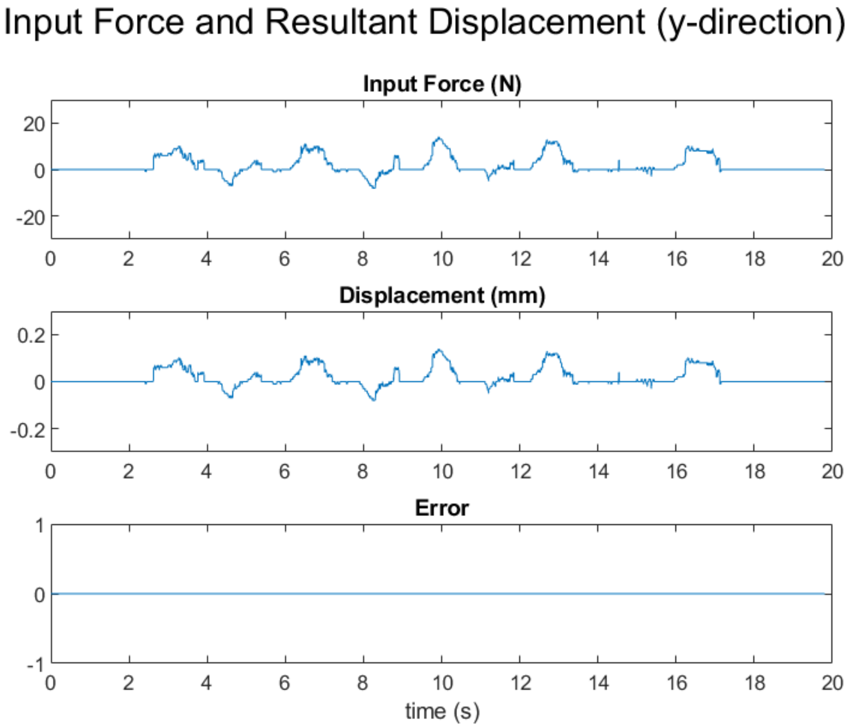

Figure 7.10: *The Y component of Input Force and Resultant Change in Y-Displacement Demand for Admittance (k=100)*

The mean responses of MyPAM with three control schemes to a trajectory demand is shown by Figure 7.11

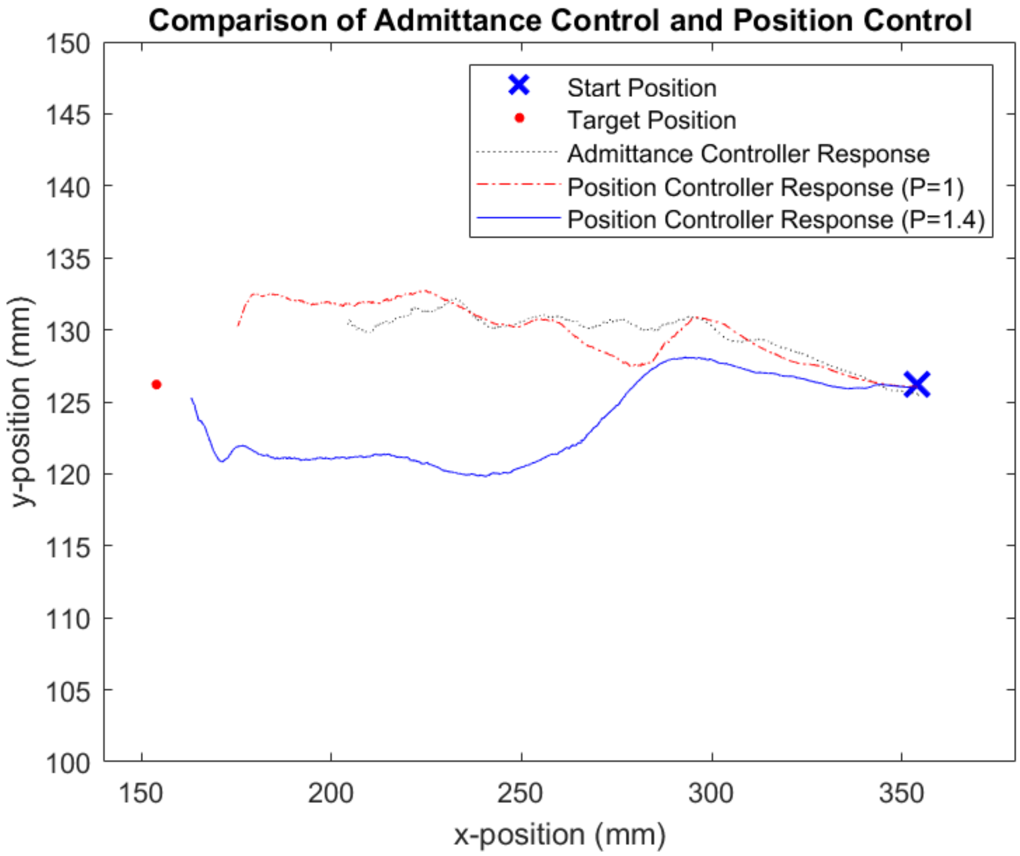

Figure 7.11: *The Mean Responses of MyPAM with Three Control Schemes to a Trajectory Demand.*

The mean end effector force components and mean final displacement errors of the 5 repeats at the settle point of each control scheme test (the location at which no further progress towards the target location) are shown by Table 7.2

Table 7.2: *The Settling State of MyPAM - Showing the Force Measured at the End Effector Due to Motor Demand and the Position Error.*

Control Scheme	X -Component of Force (N)	Y-Component of Force (N)	X-position Error (mm)	Y-Position Error (mm)
Position (P = 1)	53	21	21.3	4.2
Position (P = 1.4)	58	27	9.2	0.7
Admittance	37	14	50.2	4.2

7.4.4 Discussion and Conclusion

The results shown by Figures 7.5,7.6,7.7,7.8,7.9 and 7.10 show that the Admittance filter works correctly, with the error the between change in displacement output from the Admittance filter and the expected change in displacement as zero in all cases. As expected when examining the Admittance filter implementation shown by Equations 7.16 and 7.17 the higher the value of coefficient k, the lower the change in displacement demand in response to a force input. A high value for coefficient k which produces very little change in the displacement demand it is not useful from a control perspective since it does very little to modulate the position demand and the Admittance filter is essentially rendered useless. For this reason a value of $k = 1$ was selected as appropriate for the Admittance control scheme in MyPAM.

Whilst the results of test 1 were useful for determining an appropriate value for the virtual spring constant in the Admittance filter, the results of test 2 are more useful in the validation of Admittance Control for MyPAM. Observing the mean paths tracked by MyPAM in Figure 7.11 it may be seen that only the Position Control Scheme ($P = 1.4$) reached the target deadzone. This is because the motor demands were large enough to overcome the resistive forces in the system (the stiffness of the human arm proxy and friction in the motors and the human arm proxy), but this came at the expensive of the largest interaction forces. The Position Control Scheme ($P = 1.$) was not able to reach the target deadzone and experienced the second largest interaction forces. The motor demands were not sufficient to overcome the resistive forces in the system, yet the forces were still 43% higher in the x-direction and 50% higher in the y-direction than experienced with the Admittance Control scheme.

Observing Table 7.2 it is clear that the interaction force is lower for the Admittance Control scheme than for either of the Position Control Schemes. This is because the Admittance Control Scheme has modulated the position demand as a result of the interaction force, which is the desired outcome. The caveat to this is that the Admittance Control scheme has produced the greatest position error. In terms of Control this could be considered a poor outcome, but from a patient perspective there is little benefit to achieving the desired target at the expense of painful forces applied to the arm. This could have two repercussions:

1. Excessive interaction force may cause additional injuries.
2. Excessive interaction forces may be a demotivating factor to the patient, disincentivising them from further using MyPAM and having negative effects on rehabilitation outcomes.

The Admittance Control Scheme is therefore suitable only for patients with upper limb disability rather than full disability. This highlights limitations present in the methodology used for test 2. The human arm proxy is a passive device, which emulates the worst case scenario of a completely disabled patient. While it is useful to analyse the response in this case, in practice a completely disabled patient is unlikely to be using MyPAM. Indeed, in trials of previous versions of MyPAM a patient was excluded from the trials because of severe paresis [11]. Full validation of the Admittance Control Scheme requires a further study with active participants, which would compare the Admittance Control scheme against the Position Control Scheme in a similar manner to the trajectory comparison study presented in Chapter 3.

7.5 Instability Protection

A critical control property for MyPAM is stability of the control system, since if the robot enters an unstable state it may impart dangerous or painful forces onto the patient. When using Position Control only there two distinct dynamic situations with different PI gains. Whilst control is stable when the MyPAM is moving in the unloaded mode with the correct gains, there exists two possibilities for entering an unstable state. The first is a failure of the grip sensor to distinguish the presence of a user correctly. This may lead to the use of the loaded MyPAM PI gains, which are too high for the unloaded state. The second is an excessive force input in the Loaded case. To this end instability protection was added to the low level control system of the robot, which removes power to the motors should the robot enter an unstable state and is configurable to trigger a full system shutdown.

Whilst it is clear by personal observation when the robot has entered an unstable state, it is necessary to define instability for MyPAM. Further to this, it was important that the instability definition be computationally efficient to preserve the limited resources on the control hardware and therefore not throttle the processing rate of the control loop. The output of the Position Control system defined in Section 7.2 are two motor demands u_0 and u_1 which may be either positive or negative depending on the direction of the motor demand. In practice the motor control board for each motor requires a PWM (Pulse Width Modulation) signal, ranging from 0% to 100%, and a demand direction defined as HIGH or LOW. Instability for MyPAM is defined as three or more changes in the direction demand of either motor in 250ms or less, which was determined to be sufficient to prevent unstable oscillation around a setpoint without affecting normal use of MyPAM.

Consideration must be taken to ensure that the instability protection is not too sensitive, however. There are some directions of travel in the workspace in MyPAM which promote a non-optimal path due to backlash in the joints of the robot. The instability protection must not interfere with the normal operation of MyPAM but must be robust enough to stop the device should any instability occur.

7.6 Experimental Validation of Instability Protection

7.6.1 Introduction

The implementation of instability protection was necessary to protect users of MyPAM from the unacceptable accelerations and interaction forces caused by instability. To this end, two tests were performed to validate the instability protection scheme defined in Section 7.5. The first test involved producing an unstable state in MyPAM and monitoring the response of the system. The second test was carried out to ensure that the instability protection did not interfere with the normal use of MyPAM, since the backlash in the joints causes non-optimal paths in some directions of travel.

7.6.2 Methodology

Test 1 MyPAM was positioned at its usual start position ($x = 354mm, y = 126.2mm$). The target for MyPAM was set at the same position ($x = 354mm, y = 126.2mm$) such that any deviation from the start point would result in the motors applying restoring forces. The low level controller was set with a high proportional gain of $P = 2$ so that any deviation from the setpoint would result in an unstable oscillatory response. Force was applied to the end effector and quickly released, displacing the end-effector and requiring the MyPAM to apply motor demands. MyPAM was allowed to enter an unstable state and monitored to ascertain correct operation of the instability protection. Three repeats were performed, with the system reset to the initial position each time before starting. During testing the position of the end effector, the direction demand of the motors, and the status of the safety shutoff marker were logged.

The position data of the end effector were analysed and split into path components which corresponded to either an obvious change in direction or a change in direction of either of the motor demands. The direction of motor demand and safety shutoff status were examined at the end point of each path component to ascertain whether the Instability Protection had worked correctly. This test was repeated three times.

Test 2 With MyPAM in the unloaded condition and using the Position Control Scheme $P = 1$ $I = 0.1$, the star pattern trajectory defined in Section 3.3 in Chapter 3 was applied for 1 minute. The position data and direction change count were logged throughout and success was defined as a complete minute of testing with no shutdown due to the instability protection. This test was repeated three times.

7.6.3 Results

Figures 7.12, 7.13, 7.14, 7.15, 7.16, 7.17, 7.18 and 7.19 show the trajectory of the end-effector of MyPAM and the individual components of the movement during instability test 1 repeat 1. Figures 7.20, 7.21, 7.22, 7.23, 7.24, 7.25, 7.26 show the trajectory of the end-effector of MyPAM and the individual components of the movement during instability test 1 repeat 2. Figures 7.27, 7.28, 7.29, 7.30, 7.31, 7.32, 7.33 and 7.34 show shows the trajectory of the end-effector of MyPAM and the individual components of the movement during instability test 1 repeat 3. Test 1 repeat 1 lasted for 400ms, with the beginning of phase 7 (the point where the system was deemed unstable) reached in 235ms. Test 1 repeat 2 lasted for 380ms, with the beginning of phase 6 reached in 232ms. Test 1 repeat 3 lasted for 355ms, with the beginning of phase 7 reached in 202ms.

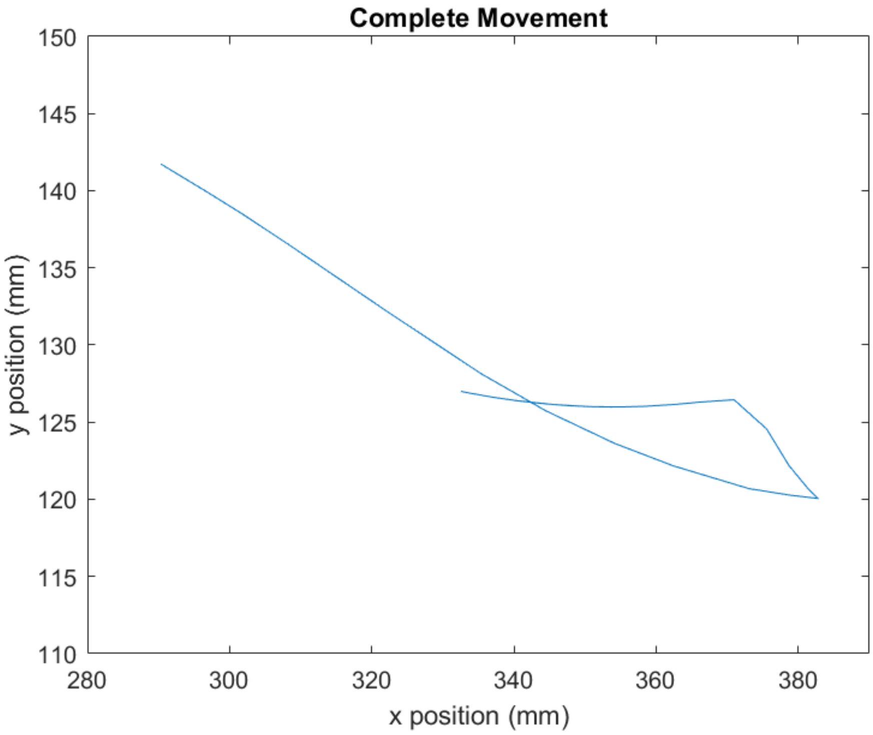

Figure 7.12: *The Complete Path Travelled of the End Effector During Instability Test 1.*

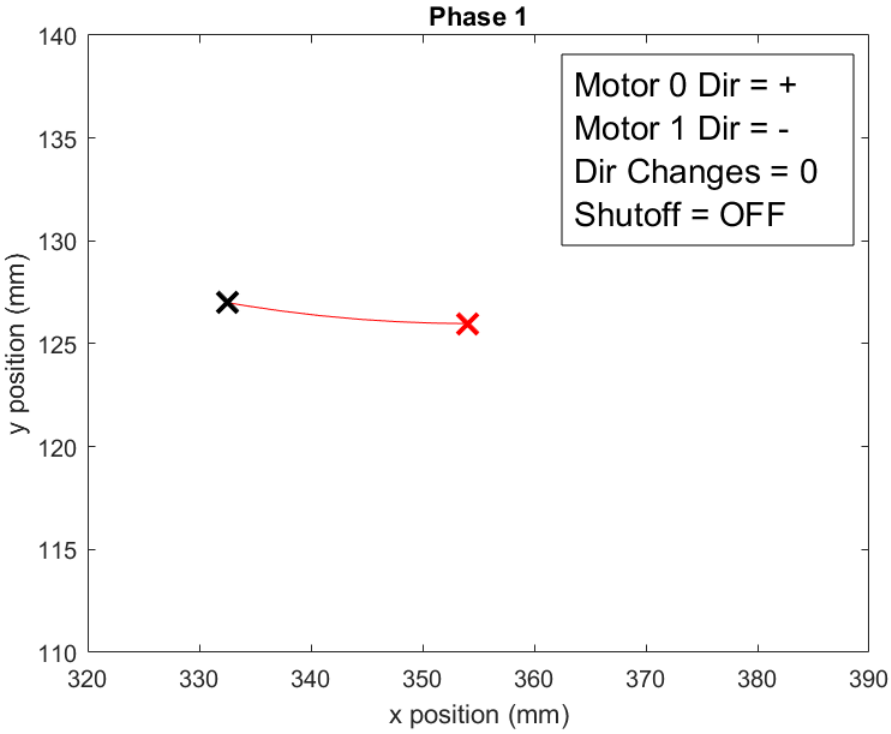

Figure 7.13: *Trajectory Component 1 of the End Effector During Instability Test 1.*

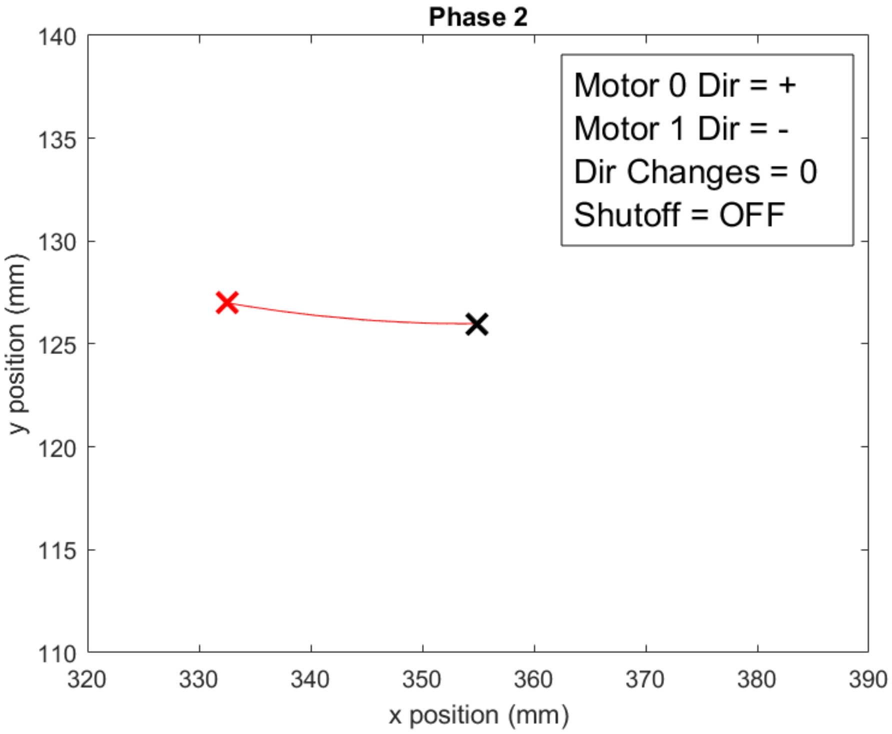

Figure 7.14: *Trajectory Component 2 of the End Effector During Instability Test 1.*

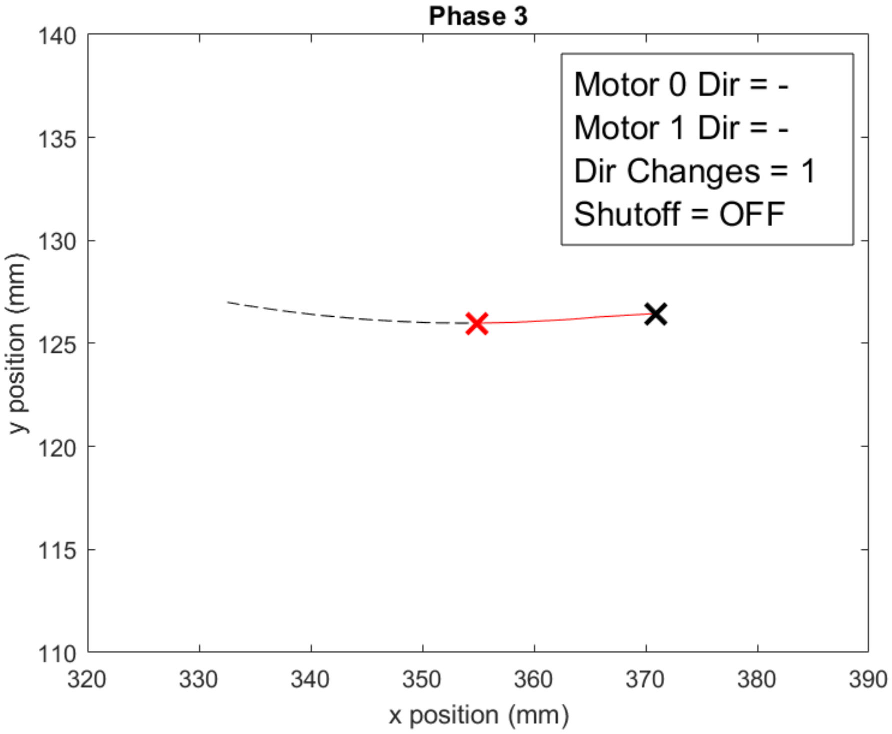

Figure 7.15: *Trajectory Component 3 of the End Effector During Instability Test 1.*

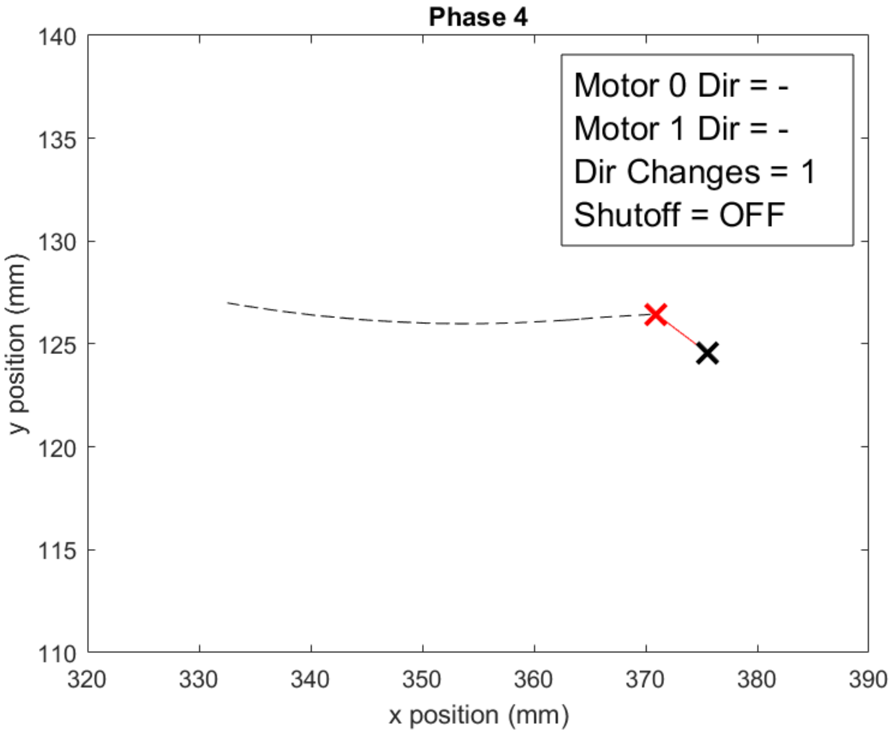

Figure 7.16: *Trajectory Component 4 of the End Effector During Instability Test 1.*

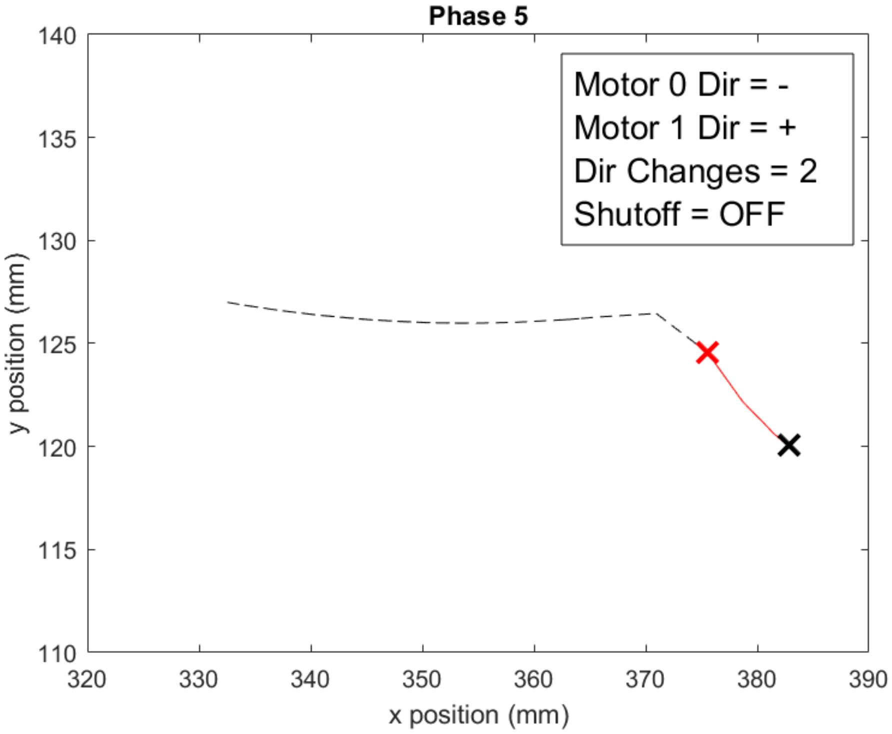

Figure 7.17: *Trajectory Component 5 of the End Effector During Instability Test 1.*

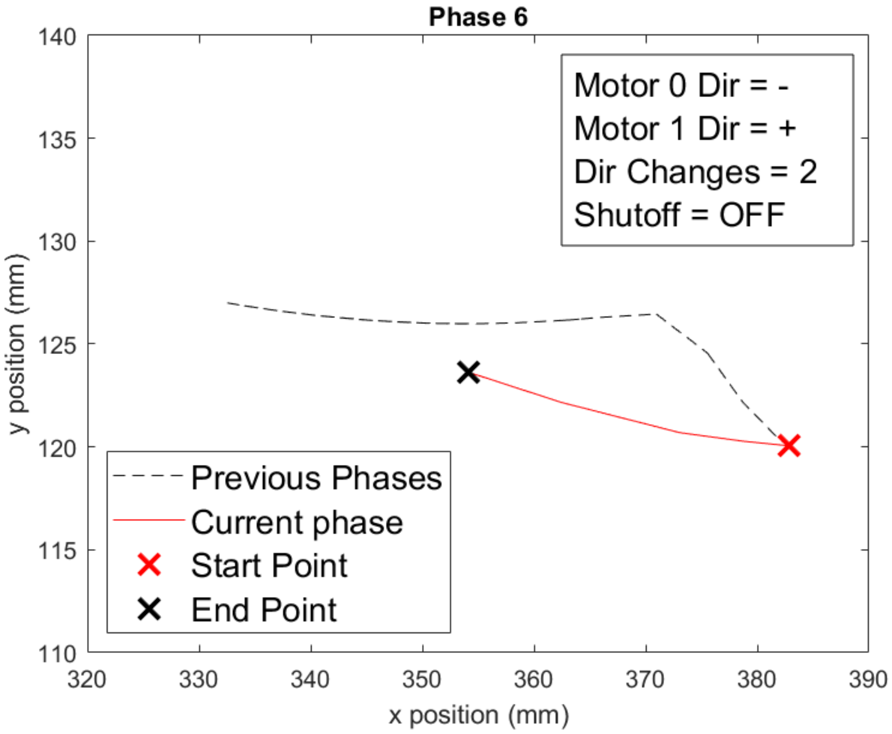

Figure 7.18: *Trajectory Component 6 of the End Effector During Instability Test 1.*

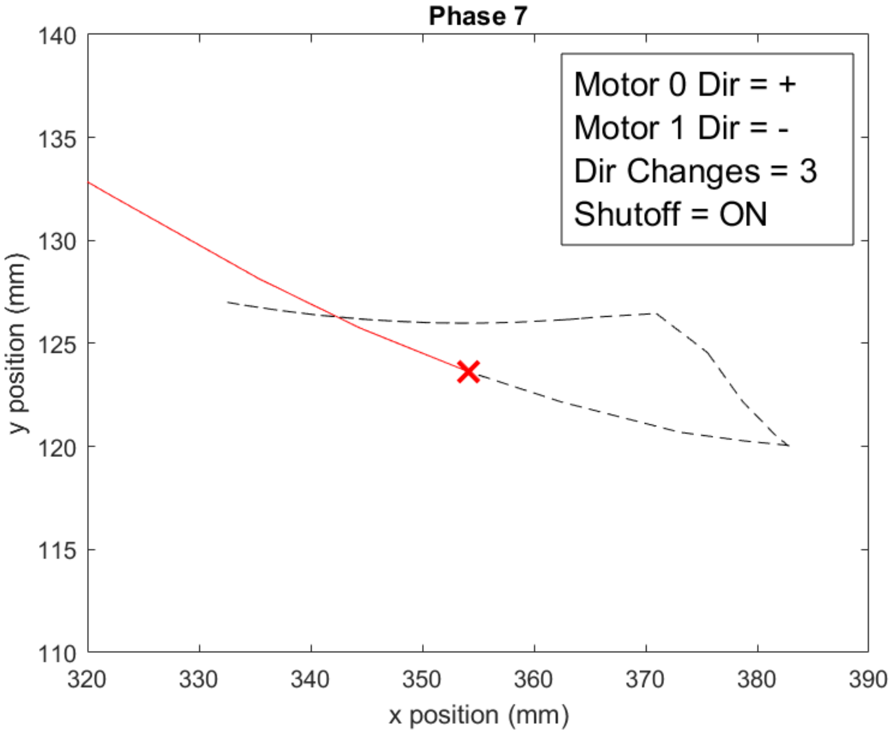

Figure 7.19: *Trajectory Component 7 of the End Effector During Instability Test 1.*

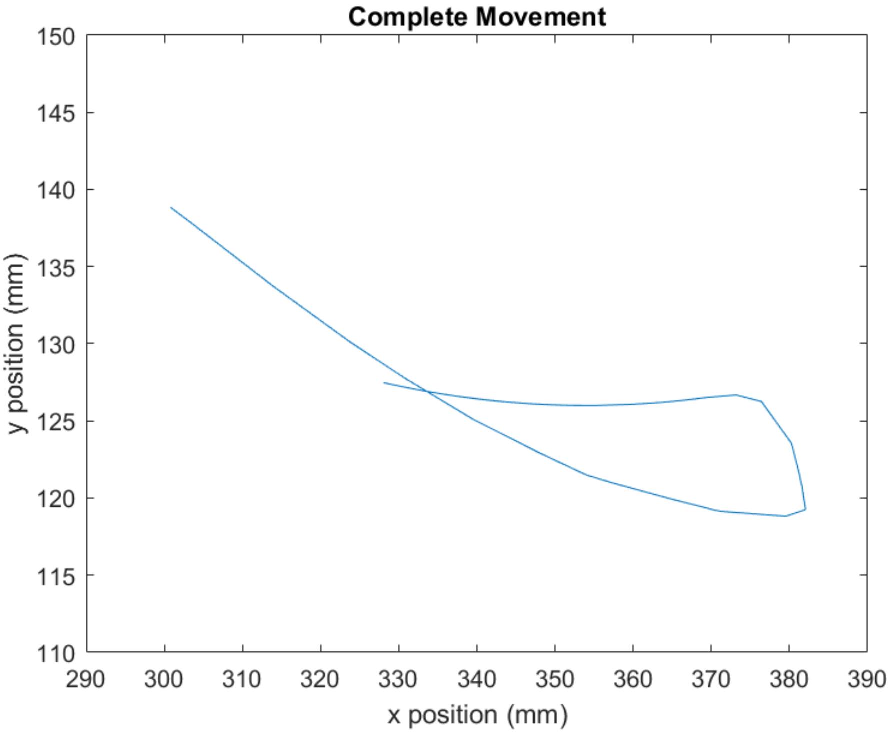

Figure 7.20: *The Complete Path Travelled of the End Effector During Instability Test 2.*

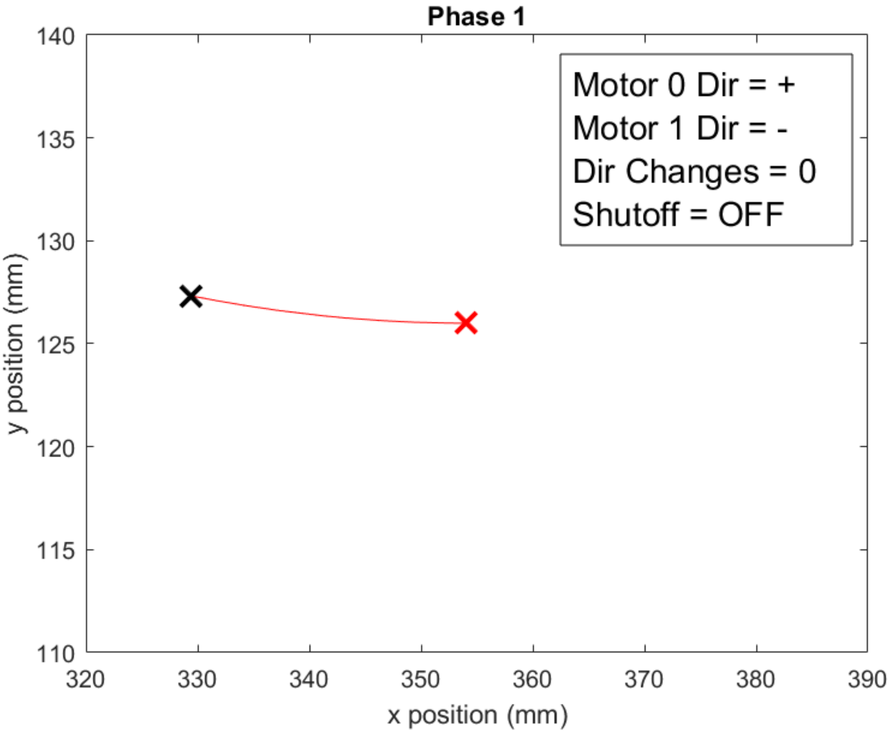

Figure 7.21: *Trajectory Component 1 of the End Effector During Instability Test 2.*

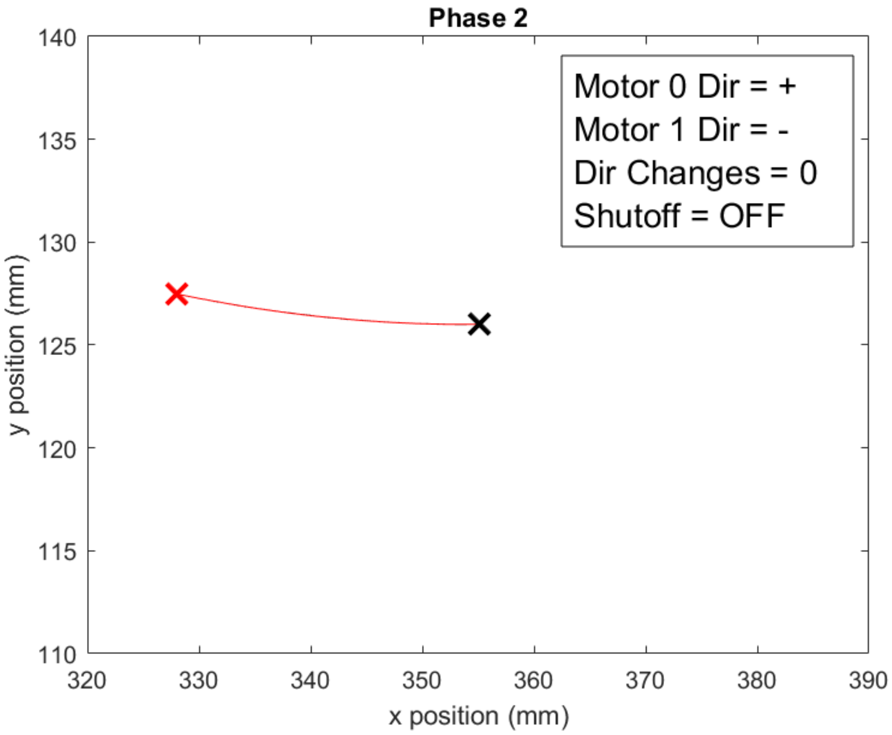

Figure 7.22: *Trajectory Component 2 of the End Effector During Instability Test 2.*

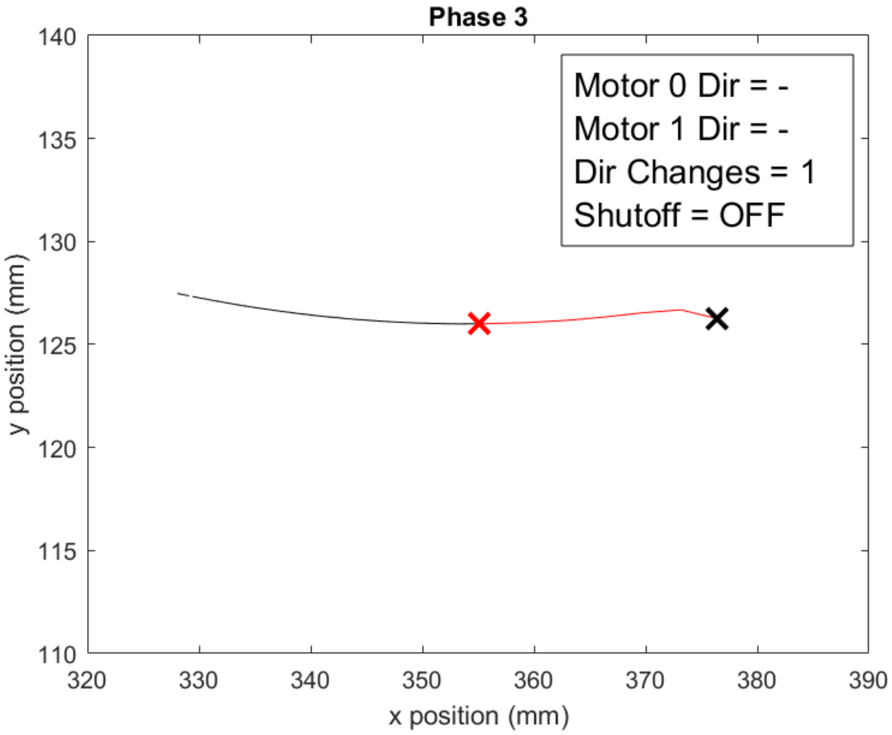

Figure 7.23: *Trajectory Component 3 of the End Effector During Instability Test 2.*

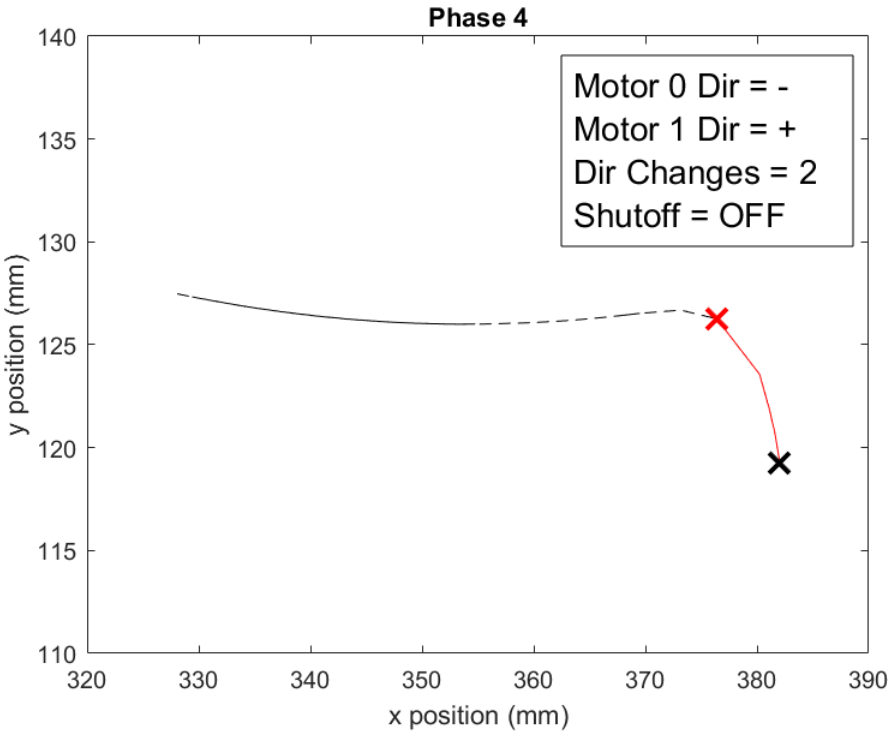

Figure 7.24: *Trajectory Component 4 of the End Effector During Instability Test 2.*

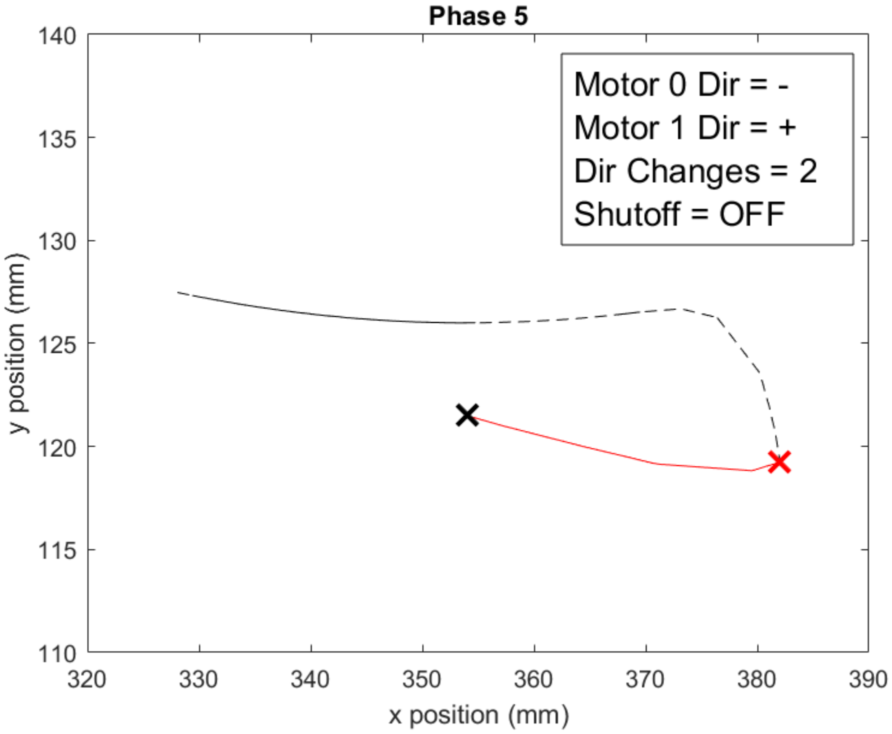

Figure 7.25: *Trajectory Component 5 of the End Effector During Instability Test 2.*

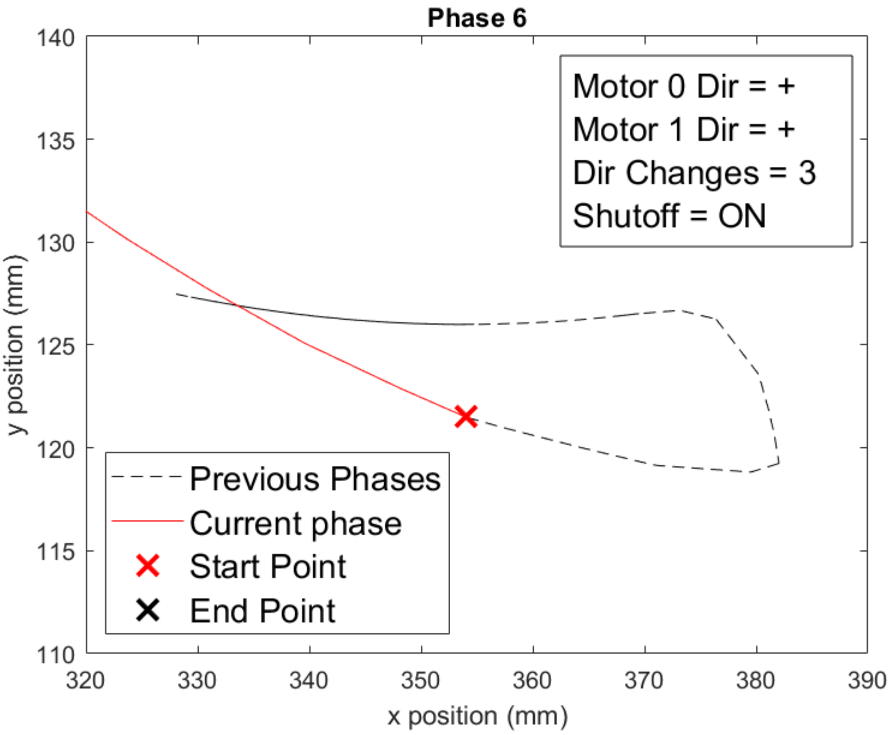

Figure 7.26: *Trajectory Component 6 of the End Effector During Instability Test 2.*

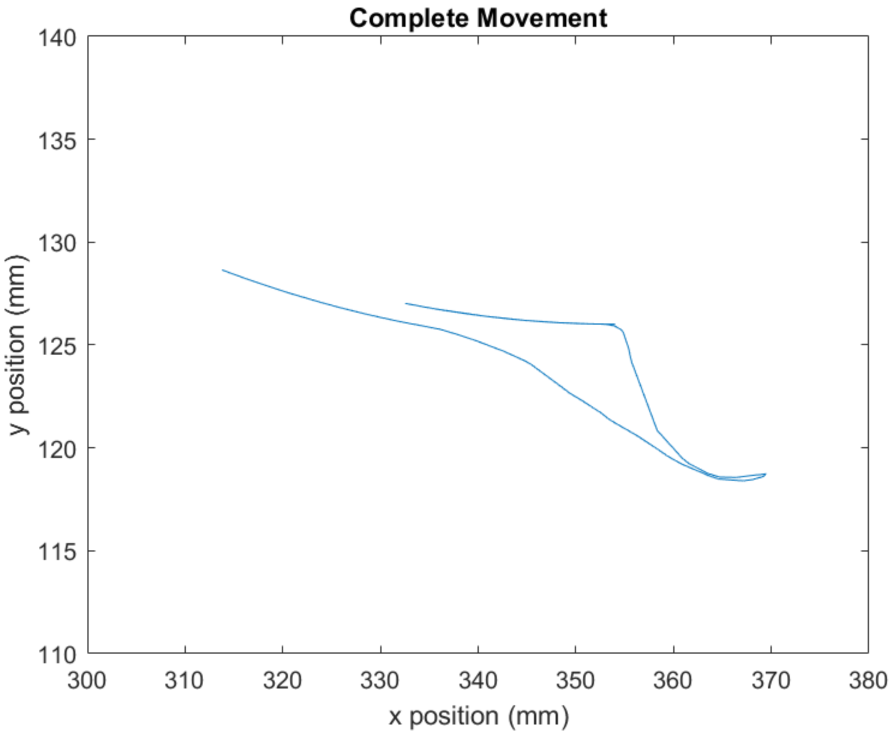

Figure 7.27: *The Complete Path Travelled of the End Effector During Instability Test 3.*

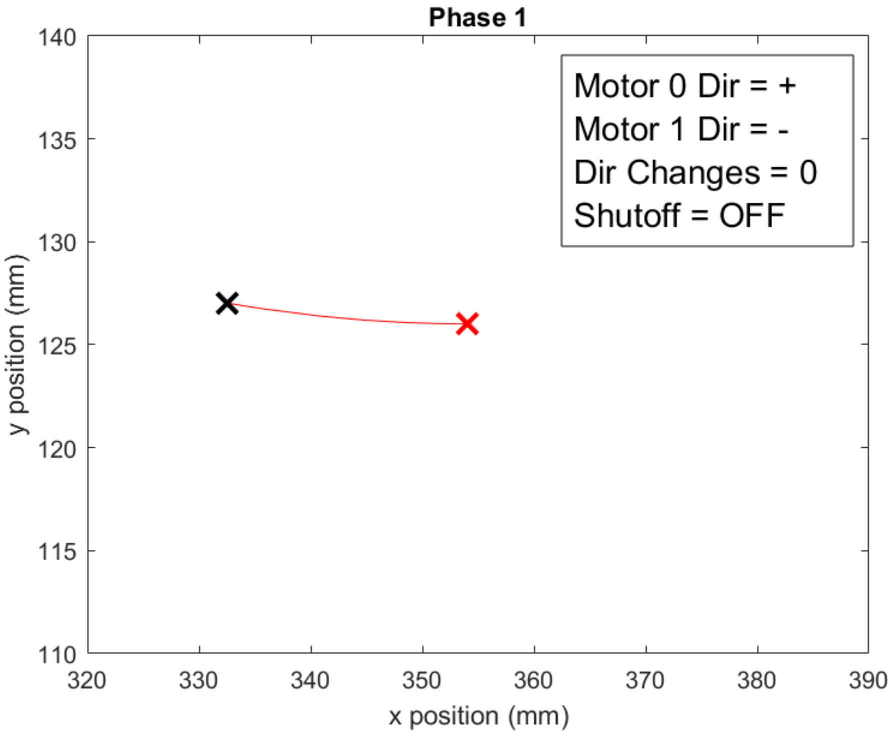

Figure 7.28: *Trajectory Component 1 of the End Effector During Instability Test 3.*

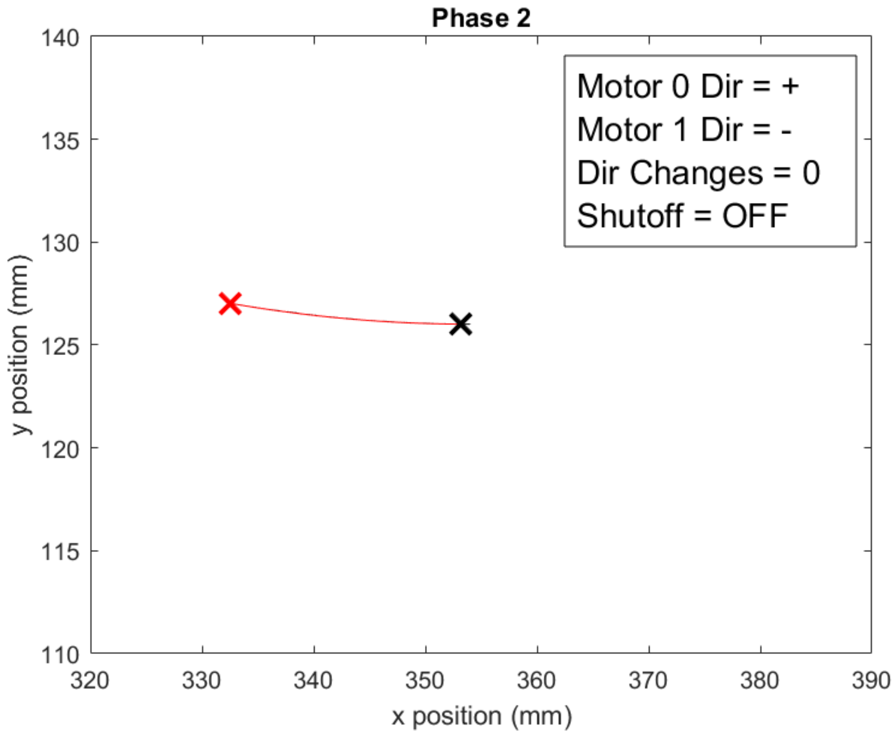

Figure 7.29: *Trajectory Component 2 of the End Effector During Instability Test 3.*

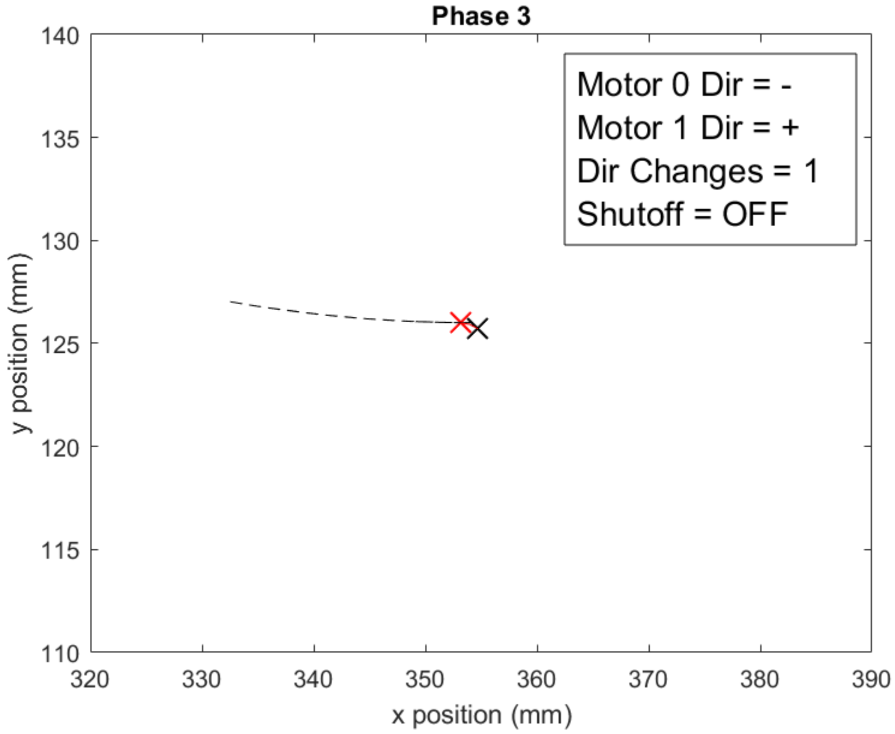

Figure 7.30: *Trajectory Component 3 of the End Effector During Instability Test 3.*

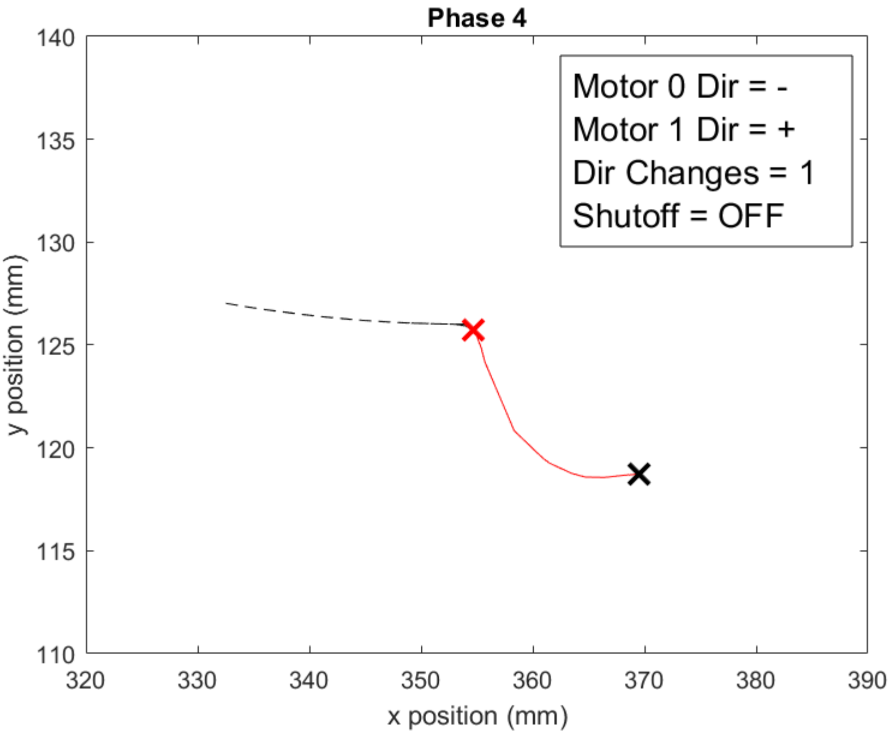

Figure 7.31: *Trajectory Component 4 of the End Effector During Instability Test 3.*

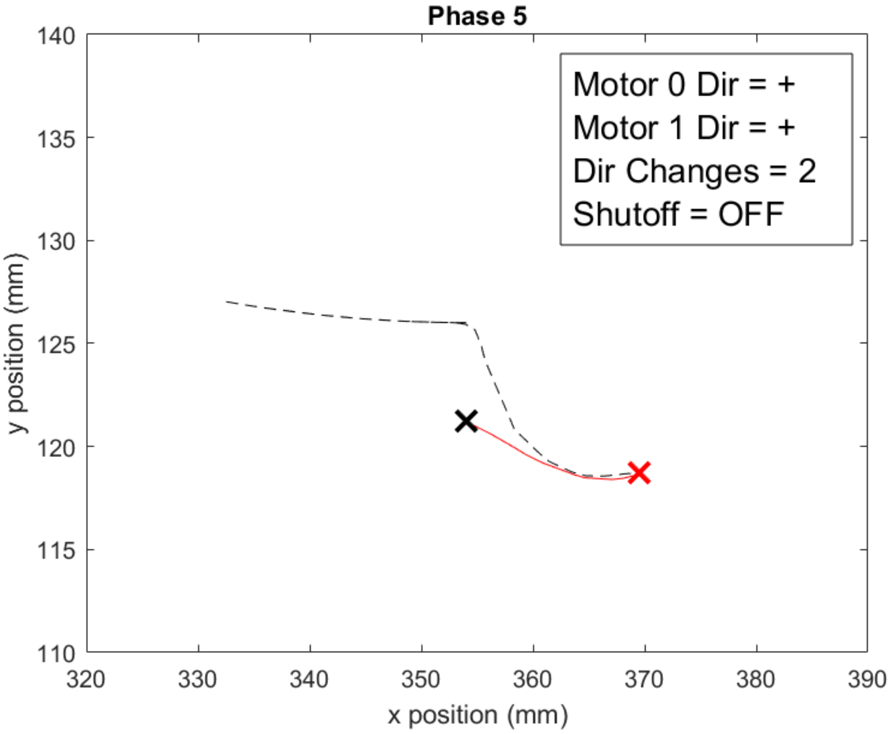

Figure 7.32: *Trajectory Component 5 of the End Effector During Instability Test 3.*

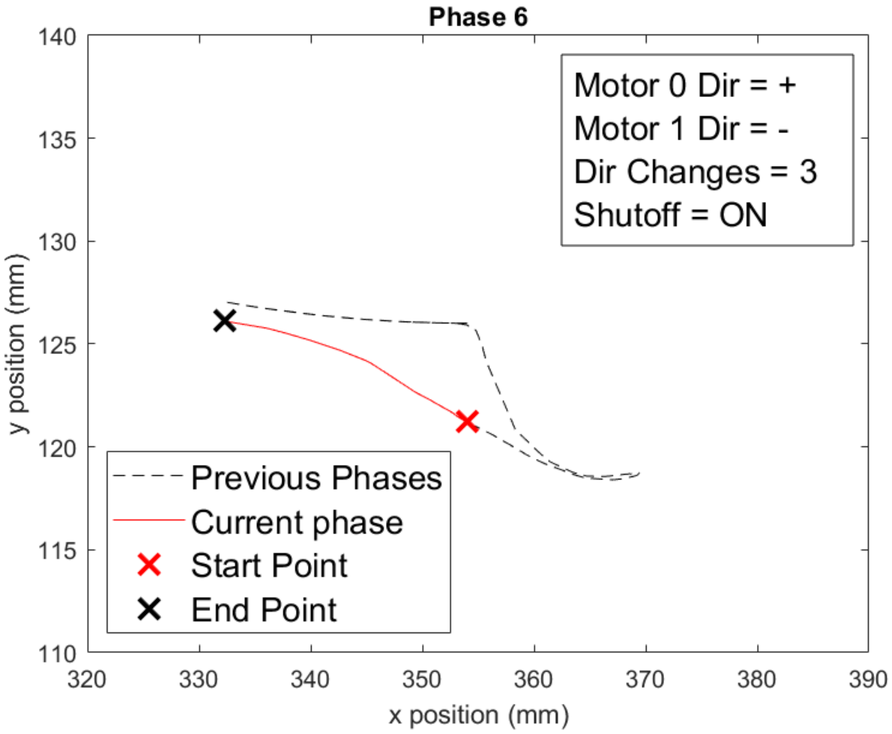

Figure 7.33: *Trajectory Component 6 of the End Effector During Instability Test 3.*

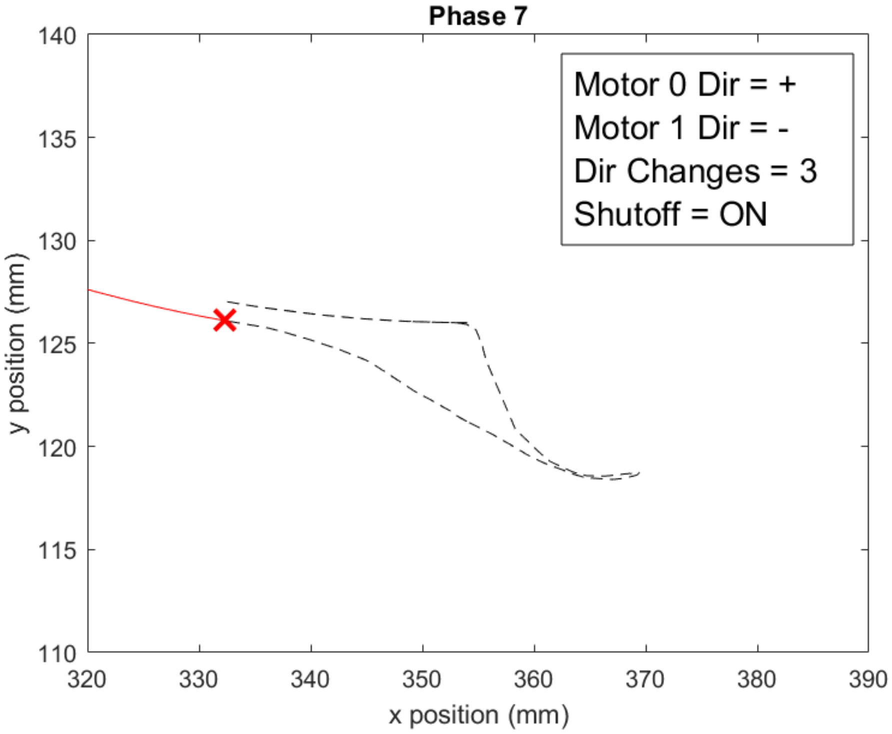

Figure 7.34: *Trajectory Component 7 of the End Effector During Instability Test 3.*

The trajectory tracking by MyPAM using the Position Control Scheme for 1 minute is shown by Figures 7.35, 7.36 and 7.37.

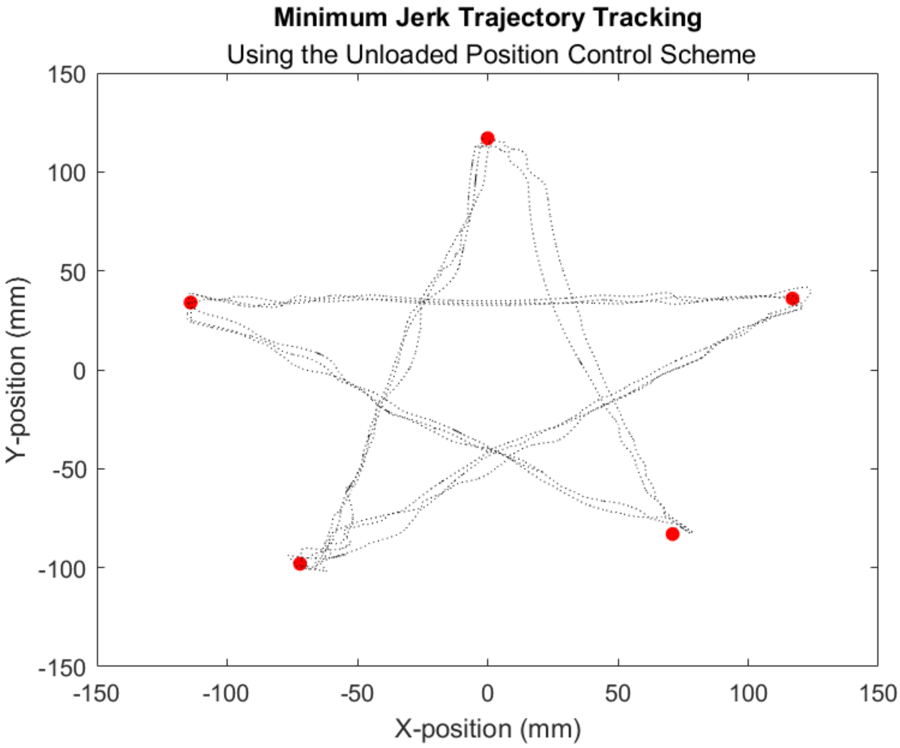

Figure 7.35: *MyPAM Tracking the Star Trajectory For 1 Minute - Repeat 1.*

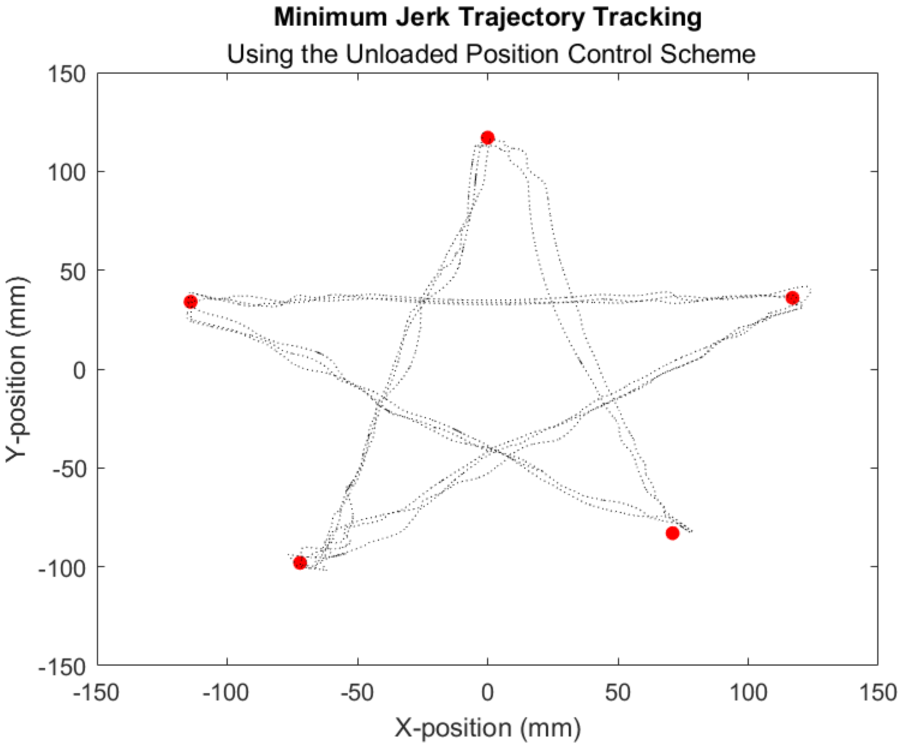

Figure 7.36: *MyPAM Tracking the Star Trajectory For 1 Minute - Repeat 2.*

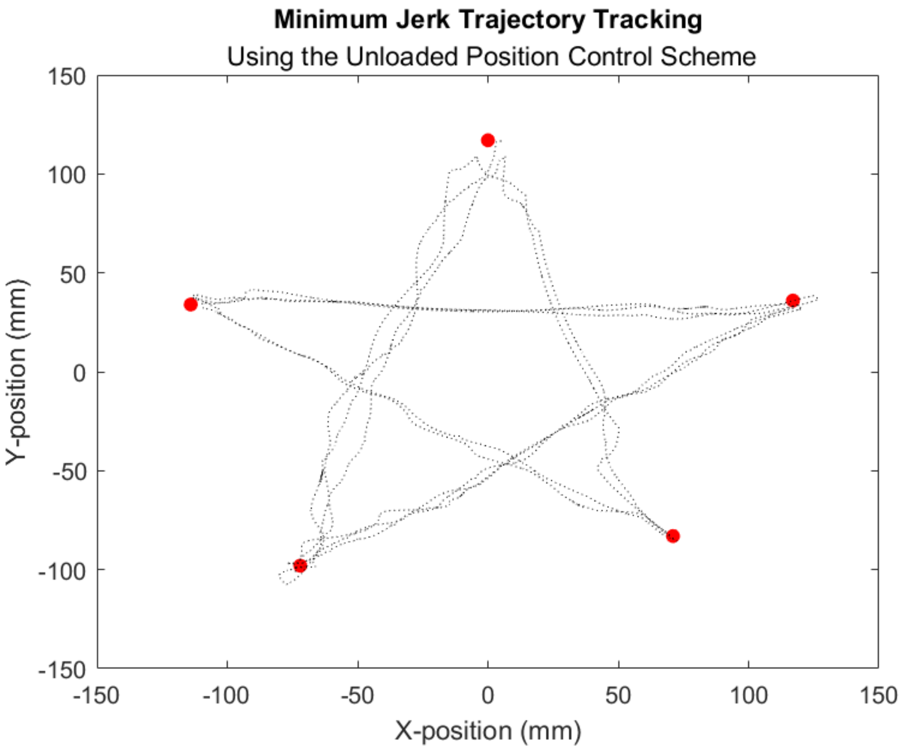

Figure 7.37: *MyPAM Tracking the Star Trajectory For 1 Minute - Repeat 3.*

The maximum number of direction changes in 250ms over 1 minute of Trajectory Tracking is shown by Table 7.3.

Table 7.3: *Peak Direction Changes in 250ms Over 1 Minute of Trajectory Tracking.*

Repeat	**Peak Direction Changes in 250ms**
Repeat 1	2
Repeat 2	2
Repeat 3	2

7.6.4 Discussion and Conclusion

Observing Figures 7.12 and 7.20 it may be seen that the trajectory the end-effector of MyPAM is similar in repeats 1 and 2 of test 1. Figure 7.27 shows that that the trajectory the end effector of MyPAM is not as similar in repeat 3 of test 1, which is likely to have been caused by a difference in the loading direction of the initial force required to produce the instability. It is noted, however, that all three repeats produced three changes in the direction of the motor demands in under 250ms, activating the instability protection and setting the motor demands to 0%.

It is interesting to note that the overshoot shown by Figures 7.12, 7.20 and 7.27, which occurs after the instability protection is activated and the motor gains are set to 0%, was larger for repeats 1 and 2 than for repeat 3 of test 1. It is apparent that the initial loading direction has had a noticeable effect on trajectory of the end-effector in a manner synonymous with a double pendulum. Most importantly the instability protection has correctly identified the instability in the system and the motor shutoff has been activated in all three tests.

Observing the graphs shown by Figures 7.35, 7.36 and 7.37 it may be seen that the response of MyPAM shows some oscillation across the straight line trajectory demand. this is particularly noticeable in Figure 7.35 between targets 1 and 2. It is noted, however that all three tests completed the full minute, and the instability protection was not activated, with each test producing a maximum of 2 direction changes in 250ms as shown by Table 7.3.

The Instability protection operates as intended, with instability correctly identified in all three repeats of the first test and not erroneously identified in any of the three repeats of the second test.

7.7 System Level Control

The developed controller sits well within the Low-level control structure when referring to Figure 2.4. It is important to note that by decoupling the developed controller from the HLC (in contrast to the previous iteration of MyPAM, where the trajectory generation was performed by the HLC), many more possibilities are presented for gamification of reaching tasks since the game developers need not consider the specific implementation. Further to this the developed controller accepts the onus of patient safety both by monitoring for instability events and by preventing large distances between setpoints, which was again a limitation of the previous iteration of MyPAM.

This particular controller enables the use of several specific rehabilitation techniques depending on parameters set by the Mid-level controller. The Bobath technique, whereby patients are assisted through the initialisation and completion of tasks, could be applied with MyPAM by adjusting the assistance gains close to the start and end of each reaching movement. Muscle strengthening exercises may indeed also be implemented using the MyPAM by applying a negative assistance gain, thus applying a resistance against which a user must act. Again, this would involve the Middleware setting the gain. In should be noted that this particular technique is appropriate only for patients who have remaining some use of the affected limb, and is likely to be useful further down the rehabilitation timeline for home based activity.

The approach promoted by Carr and Shepphard is also a candidate for the controller described in this section, particularly when coupled with the HLC which is designed to provide stimulating visual feedback and provide cognitive challenges. Fundamentally, MyPAM is good at generating repetitive reaching tasks, and assisting (or indeed resisting) patients to achieve them. The controller described here is only a part of the overall system, and is flexible enough in its design to replicate a number of rehabilitation techniques. It is the decision of the physiotherapist when to apply each particuclar technique, which is achievable by setting gains in the Middleware.

7.8 Chapter Summary

The Position Control strategy is presented in Section 7.2, meeting chapter objective 7.1. The Admittance Control filter is derived and the Admittance Controller is designed in Section 7.3, with validation of the Admittance Control strategy shown in Section 7.4, meeting chapter objective 7.2. Section 7.5 presents a mechanism used to protect against instability in MyPAM, with experiments validating the operation of the instability protection shown in Section 7.6, meeting Objective 7.3. The status of the chapter objectives is shown by Table 7.4.

Table 7.4: *Chapter 7 Objectives Status*

Objective	**Description**	**Success?**
7.1	Describe the Position Control scheme for MyPAM.	Yes.
7.2	Design and validate an Admittance control scheme for MyPAM.	Yes.
7.3	Design and validate instability protection for MyPAM.	Yes.

Chapter 8

System Integration

This chapter presents the design and architecture of the software infrastructure surrounding the Low Level Controller (LLC). The purpose of the LLC in the MyPAM system architecture, data handling strategies and chapter objectives are laid out are explained in Section 8.1. The LLC consists of a number of modules with distinct tasks, which are documented in this chapter using system diagrams presented using the Unified Modelling Language (UML) standard [131]. Section 8.2 describes the Software Control Module, which is responsible for controlling the startup and shutdown procedure of the LLC. Section 8.3 documents the Communications Module, which is responsible for handling external communications between the LLC and the Mid-Level Controller (MLC). Section 8.4 describes the Error Module, which is responsible for handling software errors in the LLC, and initiating safe shutdown in the event that an unexpected error occurs. Section 8.5 documents the Logging Module, which is responsible for logging to an external file the important events of the operation of the LLC. Section 8.6 describes the operation of the Data Acquisition Module, which is responsible for processing MyPAM sensor data. Section 8.7 details sensor and actuator Input and Output (I/O) on the Field Programmable Gate Array (FPGA), which is responsible for low level hardware I/O. Section 8.8 documents the Control Module, which handles trajectory generation and robot control.

The main aim of this Chapter is to discuss the system architecture of MyPAM, with a large focus on the Low Level Controller.

8.1 Introduction

A rehabilitation robot requires a high level controller to handle task generation, a trajectory generator to handle task encoding, and a low level controller to handle task implementation as discussed in Chapter 2.4.1. The implementation of this control hierarchy demands reliable communication between these dependent sets of software to transfer the necessary data. High level Control in MyPAM is achieved by gamification, whereby the task generation is performed by a set of games written in Unity. The task data generated by the games is passed to a Mid Level Controller (MLC), which is responsible for formatting and logging all data and handling communication between the games and the Low Level Controller (LLC). The LLC is comprised of a set of software, written using LabVIEW and hosted on a National Instruments MyRIO, which handles trajectory generation and low level control. Figure 8.1 shows a high level a system diagram detailing the control hierarchy for MyPAM.

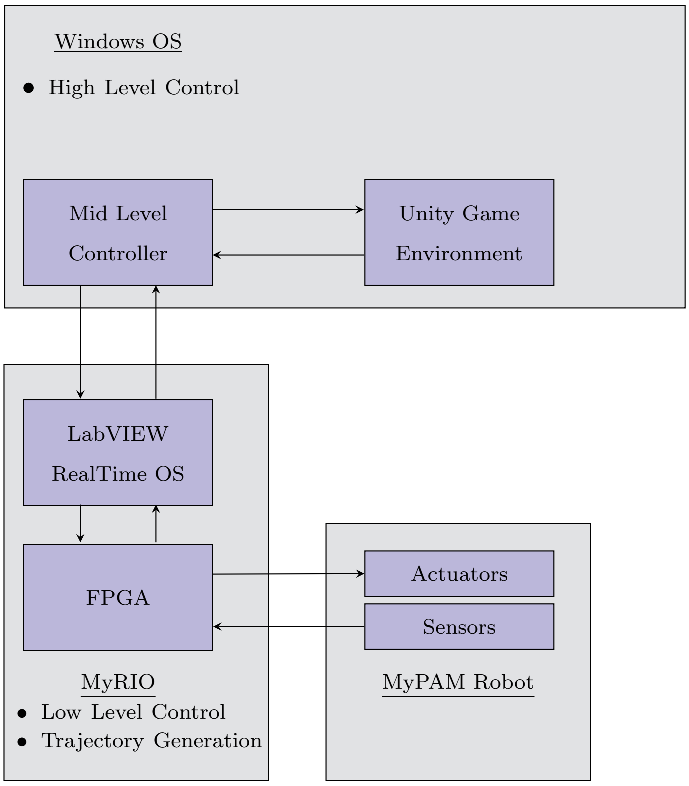

Figure 8.1: *The System Diagram for MyPAM Control Hierarchy.*

Much of the work in this thesis focuses on Low-level control, which for this iteration of MyPAM is programmed on the myRIO. The software architecture surrounding the vital components of low level control is designed and presented in this Chapter.

8.1.1 High Level Controller

The MyPAM has a number of High Level Controllers, a number which may indeed expand in the future. Referring to Figure 2.4, the HLC is simply a medium to convert rehabilitation goals into tasks. For MyPAM, this entails defining the target position of a reaching task, and displaying the target position and current position of the end effector as an interesting game. With a simple set of specifications any developer can create a game, which is added to the game library upon review and acceptance. This modular approach has been purposely designed with the aim of creating a diverse library of stimulating games which may be cycled though to ensure maximum participant motivation and engagement, since it is well understood that this promotes better rehabilitation outcomes. The HLC does not make any control or trajectory decisions, which is an improvement upon previous iterations of MyPAM. The particular game (or HLC) or sequence of games launched during a session is determined by the Middleware (or MLC).

8.1.2 Mid level Controller

The Mid Level Controller is the hub of the software chain for MyPAM, about which all other sets of software (and even some hardware) are designed to be easily interchangeable or replaceable. The MLC doesn't fit neatly into the rehabilitation robotics hierarchy model presented by Figure 2.4, but can be considered to be in control of both the HLC and the LLC. The Middleware is responsible for providing an interface for the physiotherapist to set important parameters including:

1. Assistance Level (0-100%).
2. Strength of Attractors and Repulsors (Weak/Medium/Strong).
3. Allowable number of Attractors and Repulsors per reaching task.
4. Allowable game types.

Importantly, the Middlware provides an interface for the physiotherapist to monitor session data and analytics.

Another key role of the Middleware is to launch each HLC instance and to activate the LLC, providing an interface for each to communicate the necessary data. Modularity has been the key design consideration for the MLC, with infrastructure in place for a library of HLCs and easily replaceable control hardware (ie, the myRIO can be easily swapped out for a sufficiently capable microcontroller or indeed even a programmable logic controller, provided a suitable control algorithm is programmed).

The final key design feature of the MLC is communication with a web server, so that a fleet of MyPAMs can be accessed by one centrally located physiotherapist, though this has not yet been implemented.

8.1.3 Low Level Controller Software Architecture Overview

The MyRIO is well suited for hosting the LLC because it contains a Real Time Operating System (RTOS) and a Field Programmable Gate Array (FPGA), alongside reliable methods to communicate between the two. The RTOS differs from a standard operating system (such as Microsoft Windows) in that it can guarantee precise timing for high priority tasks (such as control tasks) with no risk of a task scheduler introducing high levels of timing jitter. The FPGA is an embedded platform which implements a hardware circuit. An FPGA provides high speed data processing with low latency, and as such is ideal for interfacing with actuators and sensors for robot control.

The software architecture of the LLC is designed with patient safety as the first priority. To achieve this the code parallelism inherent when programming with LabVIEW is fully exploited, with software modules responsible for different parts of the controller implementation running asynchronously in parallel with low coupling. Each module consists of software units responsible for only one job each. Coupling is a term used to describe the level of interdependency between software modules. Highly coupled software contains modules which are highly interdependent, a situation which may lead to deadlocking. Deadlocking occurs when two modules prevent each other from accessing a required resource, essentially halting the program in a potentially unsafe state. Low coupling was achieved in the LLC by designing the modules to run asynchronously, in parallel, and with a defined startup and shutdown procedure and by careful data handling. The main components of the LLC are:

1. Program Control Module: Controls the start and stop procedure of the LLC. This is important infrastructure for the LLC software, but does not specifically define any robot control components.
2. Communications Module: Handles communication between the MLC and the LLC.
3. Error Module: Performs safety checks and error handling for the LLC. This is important for patient safety.
4. Logging Module: Logs the main events to an external file each time the LLC is run.
5. Data Acquisition Module: Handles communication between the RTOS and the FPGA, and is expandable to handle communication between the RTOS and external hardware devices.
6. Control Module: Handles low level control including trajectory generation and determining motor demands.
7. FPGA: Handles sensor input, input signal conditioning, and actuator output between the LLC and MyPAM.

Each of the software modules running on the RTOS have the same State Machine design pattern. Only three states are present: an initialisation phase, a running phase and an end phase. In the initialisation phase of each module variables are initialised and communication methods established. The initialisation phase lasts for only one loop iteration, before moving on to the running phase. Each module remains in its running phase until a shutdown event is received, after which it enters the end phase for one iteration. In the end phase of each module all data communication are safely shutdown, variables are released, and default variables are written to global variables where appropriate.

8.1.4 Software Testing

Well designed software is subjected to a series of tests during development. Unit testing refers to the use of a test framework whereby each unit of code (usually responsible for only one process) is subjected to a pass or fail unit test. A unit test involves writing code which is external to the project that passes a set of input arguments a code unit, and compares the actual output the expected output. Periodically during development the full suite of unit tests will be run to ensure that each unit continues to meet specifications. Integration testing involves running large sections of the developed software to ensure the expected interaction between modules behaves as anticipated. System testing involves running a complete project to ensure that all requirements are met.

During development of the LLC each software module was written modularly, with individual units of code designed to perform only one job. Each code unit was developed using a Test-Driven Development (TDD) methodology, whereby a unit test was developed with pass/fail parameters for each unit of code before the unit was written. Each unit of code was then developed until it would pass its unit test. A summary of the unit tests may be found in Appendix C. When all the units and modules had been developed integration testing was performed to validate the data communication and module interfaces. Integration testing was a two-stage process. In the first stage each software module was independently tested to ensure that all intramodule dataflow worked correctly and that all software units correctly worked when combined into a module. In the second stage the integration between software modules was tested to ensure that intermodule communication worked correctly. The final stage of testing was System testing, where the completed LLC was run to ensure that data communication between the MLC and the LLC worked correctly.

8.1.5 Data Handling in the Low Level Controller

In the LLC data is classified as either inter-module communication data or control data. Control data is the data used for control of the robot and includes the position targets received from the MLC, sensor data from the robot joints, and motor demands to be written to the robot's motors. Inter-module communication data comprises message or event data used for communication between modules. All data are contained in appropriate data structures determined by the data use cases. Four data handling strategies are used to ensure the desired low coupling between modules: Queues, Events, LabVIEW Clusters (equivalent to Structs in C and C++) in a Shared Variable Engine, and Global variables for communication between the FPGA and modules running on the RTOS.

Message Queues

Queues are used to communicate messages in a many to one relationship, whereby messages may be enqueued from any of the modules and are dequeued and handled by only one module. When a message is available in a queue an event is fired which alerts the receiving module to dequeue and handle the message. This behaviour is native to LabVIEW and ensures that the receiving module is not constantly polling the queue, thus preventing unnecessary computational overhead. Preserving conserving computational resources is important to allow all modules to operate within the desired timing specifications necessary for robot control. Three message queues are used in the LLC. The first is used to send messages to the Program Control Module. The second is used to send messages to the Logging Module. The third is used to send messages to the Error Module. A message may be low priority or high priority. Low priority messages are dequeued in the order in which they are queued. A high priority message is enqueued at the front of the queue so that it is dequeued and handled before other messages in the queue.

Event Handling

Event messaging is used for communication in a one-to-many relationship, whereby an event is generated by one module and is received and acted on by all other modules. Two main events exist in the LLC, which are generated by the Program Control Module. The first event is used at startup, where the Program Control Module sends a message which alerts the other modules to begin operating. The second event is used at shutdown, where the Program Control Module sends a message which alerts the other modules to cease operating. Further to this, LabVIEW natively uses events to alert the receiving node of a queue when a message is present.

Shared Variable Engine

The Shared Variable Engine (SVE) is a globally scoped data structure containing the control data as two sets of global variables: a LabVIEW Cluster for input data and a LabVIEW Cluster for output data. Input data is used for storing the most recent data obtained from the MLC and output data is used for storing the most recent data to be transmitted to the MLC. Elements of the input and output data are also used by the Low Level Control Module for robotic control. No data is stored long term in the LLC, which means that as soon as new data is available the old data is overwritten. This ensures that the control data used to control the robot is always the most recent data available. Using Globally Scoped data in this way poses the potential risk of race conditions. Race conditions occur when multiple sets of parallel code attempt to overwrite a variable at the same time, leaving the variable with an unknown value. In the LLC each variable may only be overwritten from one code location but read from many, which eliminates the risk of race conditions.

FPGA/RTOS Communication

Data is passed between the FPGA and the RTOS using shared global variables. Each variable is written to in only one location, eliminating the risk of race conditions. Data is not stored, so the value held at each global variable is the most recent set of data. This is very similar to the Shared Variable Engine, with the only difference being that data are not stored in data structures but instead one global variable exists as a raw data type for each data item on the FPGA. This is because it is computationally expensive and comparatively slow to store and transmit from a large data structure on the FPGA. FPGA resources are extremely limited and must be preserved for I/O tasks.

8.1.6 Chapter Objectives

This Chapter aims to fulfil the following objectives:

Objective 8.1 Document each code module developed for the Low Level Controller and discuss its relevance in the Control hierarchy.

Objective 8.2 Fully describe and justify the data communication methods used for intermodule communication, intramodule communication, and communication between the Low Level Controller and the Mid Level Controller.

Objective 8.3 Produce Unified Modelling Language Sequence Diagrams and Activity Diagrams to define processes in each software module.

8.2 Software Control Module

8.2.1 Overview and Responsibilities

The Software Control Module is responsible for controlling the startup procedure and shutdown procedure of the LLC. All of the modules start at the same time in parallel when the LLC begins, but they do not enter their respective initialisation phases until the Software Control Module has broadcasted the 'Start' Event. The order in which startup occurs is important to prevent other modules from accessing resources which haven't yet been initialised. For example, the SVE must be initialised by the Program Control Module before another module attempts to access it or a critical error will be thrown.

8.2.2 Intermodule Data Communication

The Program Control Module is responsible for determining the lifespan of the event system, the SVE, and an intermodule message queue. The event system is used to broadcast the 'Start' event and the 'Shutdown' event to all other modules. The SVE is used as a globally scoped data structure containing control data. The message queue is used by other modules to send a message to the Program Control Module.

8.2.3 Operating procedure

Figure 8.2 shows the UML Activity diagram and Figure 8.3 shows the UML Sequence diagram for the Program Control Module. It can be seen that most of the work done by the Program Control Module occurs during the Initialisation and the End Phases, with the Running Phase dominated solely by awaiting a shutdown request message. It is important to note that a shutdown request may be obtained by the Program Control Module either from a message enqueued into the Program Control module Queue or from the Shared Variable Engine, either of which will end the Running Phase of the Program Control Module and initiate the shutdown procedure. To prevent polling and unnecessary computational overhead the operation of the program Control module during the Running Phase is event driven. This means that an event is fired when a shutdown request is present at either source which prompts the Program Control Module to obtain and handle the shutdown request.

8.2.4 Relevance in the Control Hierarchy

This module plays a supporting role in the overall control hierarchy of MyPAM. It doesn't perform any control function in terms of the rehabilitation robotics control hierarchy presented by Figure 2.4, but it is necessary in the infrastructure of the LLC.

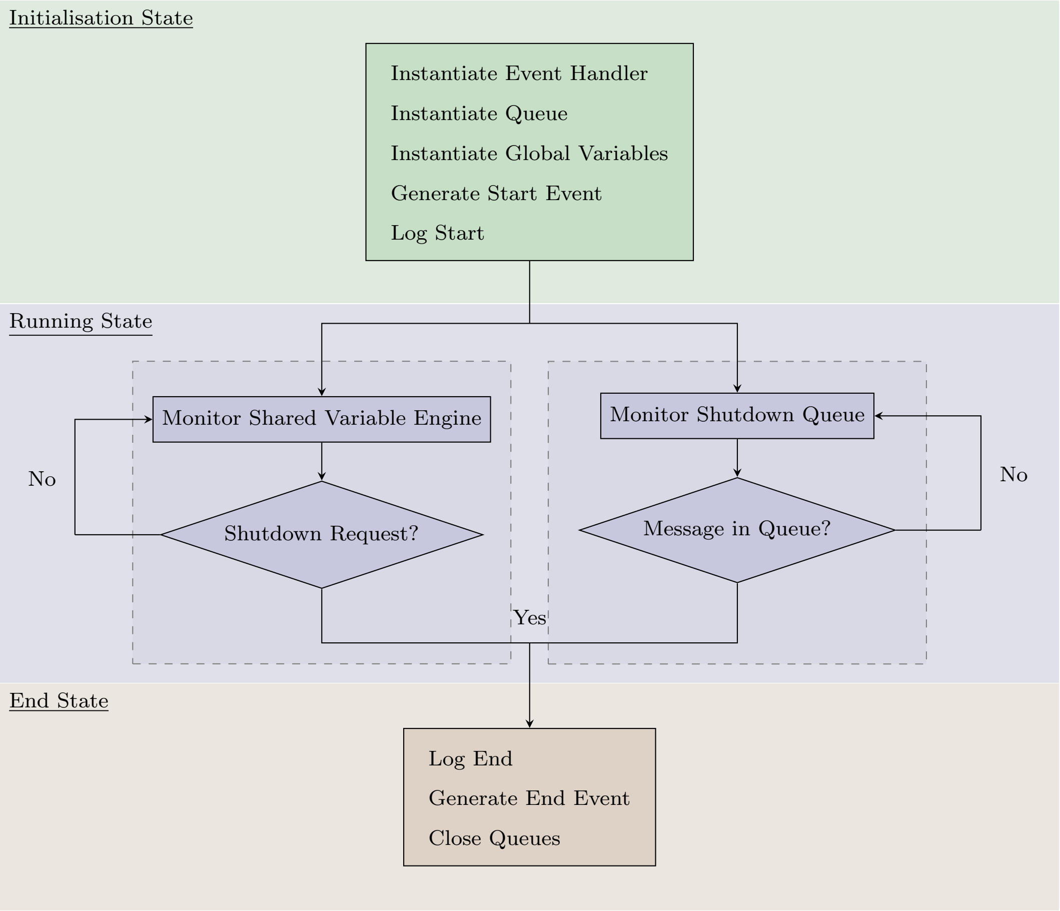

Figure 8.2: *UML Activity Diagram for the Program Control Module.*

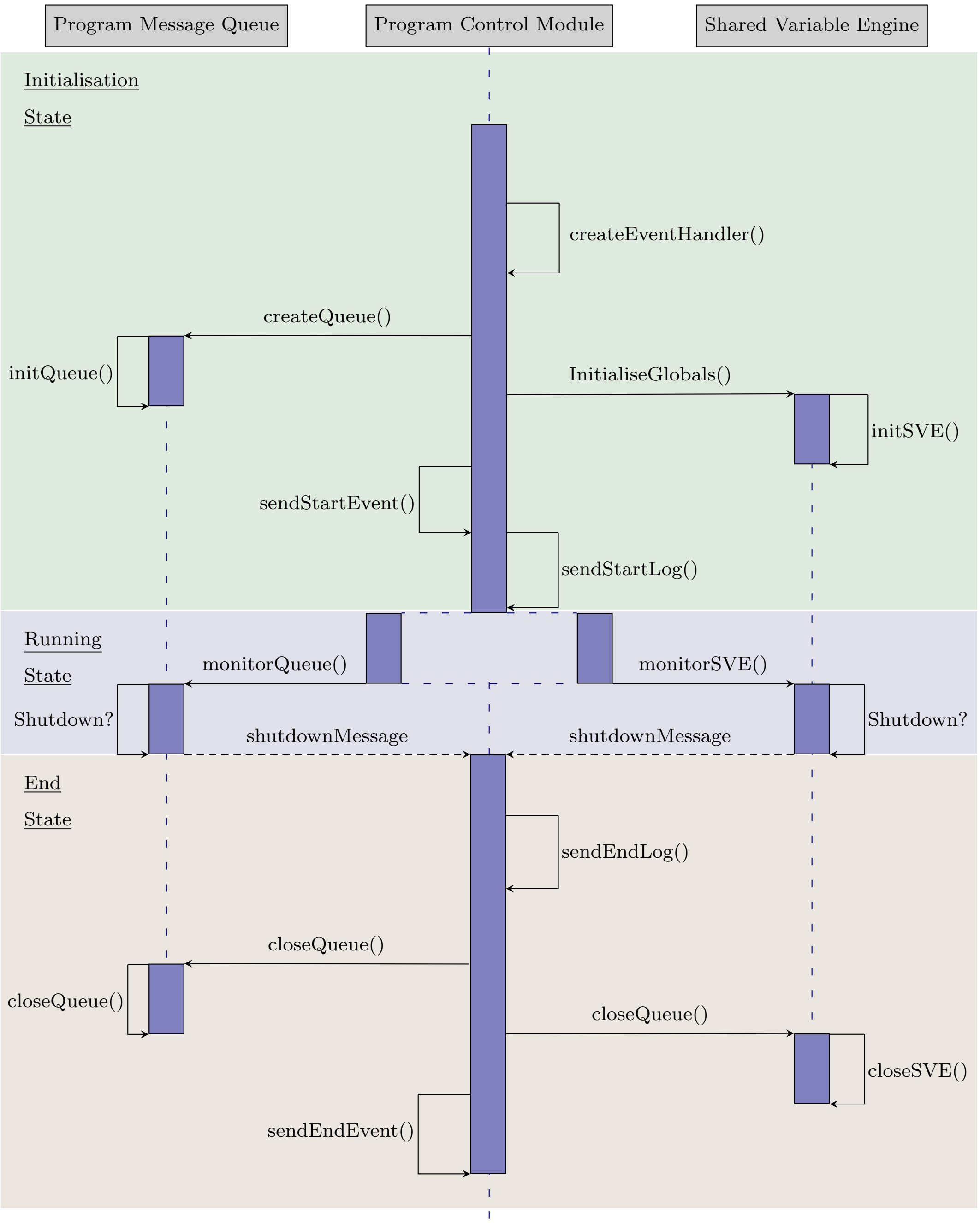

Figure 8.3: *UML Sequence Diagram for the Program Control Module.*

8.3 Communications Module

8.3.1 Overview and Responsibilities

The Communications Module is responsible for handling external communications between the LLC and the MLC. External communications are achieved by User Datagram Protocol (UDP). In the previous iteration of MyPAM communication was achieved using Transmission Control Protocol (TCP), resulting in a highly coupled software system. Unlike TCP, UDP does not perform handshaking to ensure receipt of data. Similar to intermodule communication in the LLC, loose coupling between the LLC and the MLC is preferred because it prevents code deadlocking. Two separate UDP channels are used: the first is used to transmit position and status data from the LLC to the MLC and the second is used to receive desired target and shutdown request data from the MLC.

8.3.2 Intermodule Data Communication

The Communications Module is not responsible for determining the lifespan of any internal communications structures.

8.3.3 Operating procedure

Figure 8.4 shows the UML Activity diagram for the Communications Module. Figure 8.5 shows the UML Sequence diagram for the output parallel operation, where data is sent via UDP from the LLC to the MLC. Figure 8.6 shows the UML Sequence diagram for the input parallel operation, where data is received via UDP from the MLC. The Initialisation Phase for the Communications Module does not begin until the start event is generated by the Program Control Module. During the Initialisation Phase the UDP transmission and receiving ports are opened. During the Running Phase the Communications Module has two parallel operations: one transmitting JSON data to the MLC and one receiving JSON data from the MLC. The Shutdown Phase is initiated when an event is generated by the Program Control module. During the Shutdown Phase both ports are closed.

8.3.4 Relevance in the Control Hierarchy

This module enables communication between the High level Controller and the Low Level Controller in the rehabilitation robotics control hierarchy presented by Figure 2.4. It allows position setpoints to be passed to the LLC for robotic movement and position data to be passed back to the HLC for display to the user as a game.

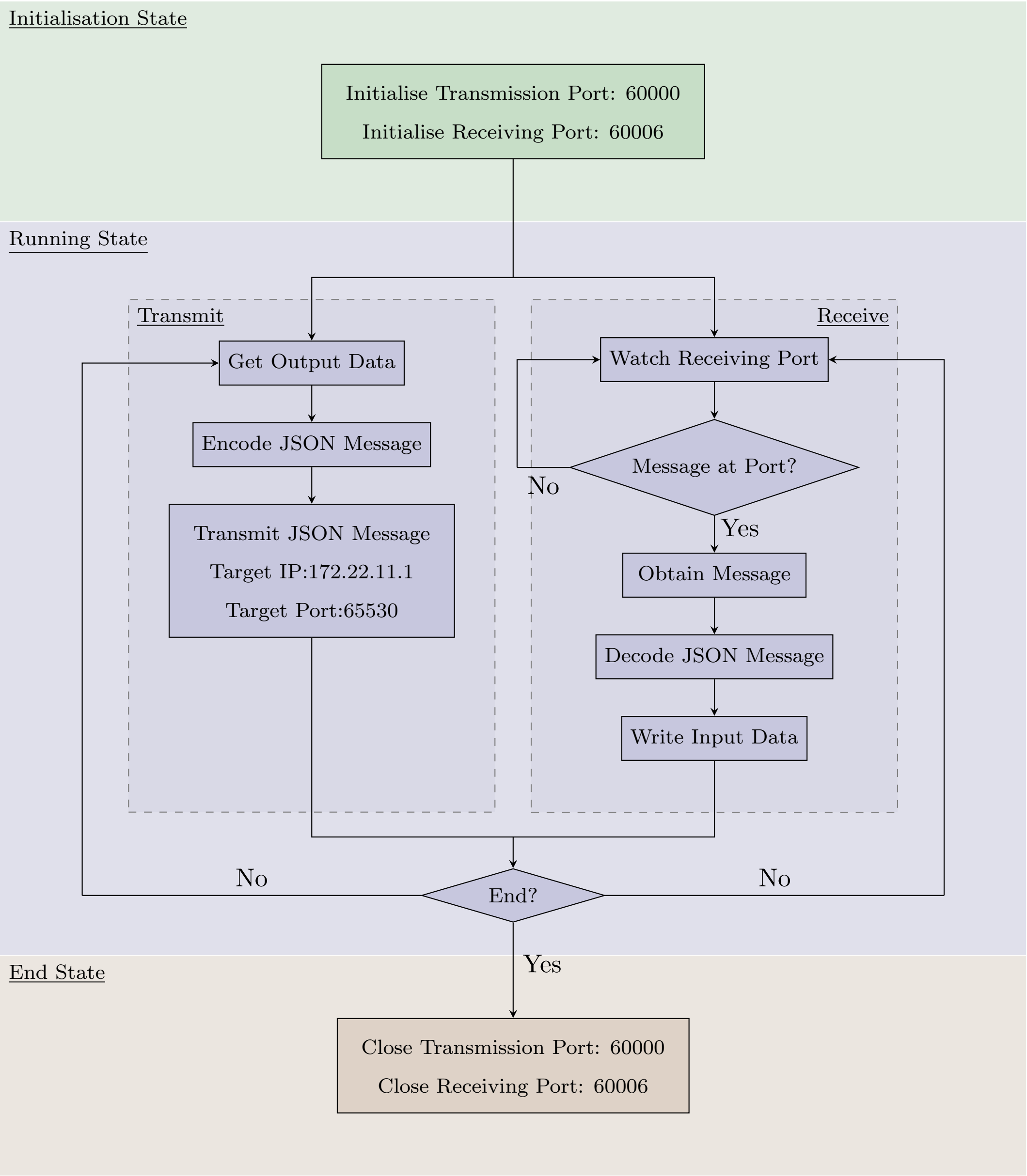

Figure 8.4: *UML Activity Diagram for the Communications Module.*

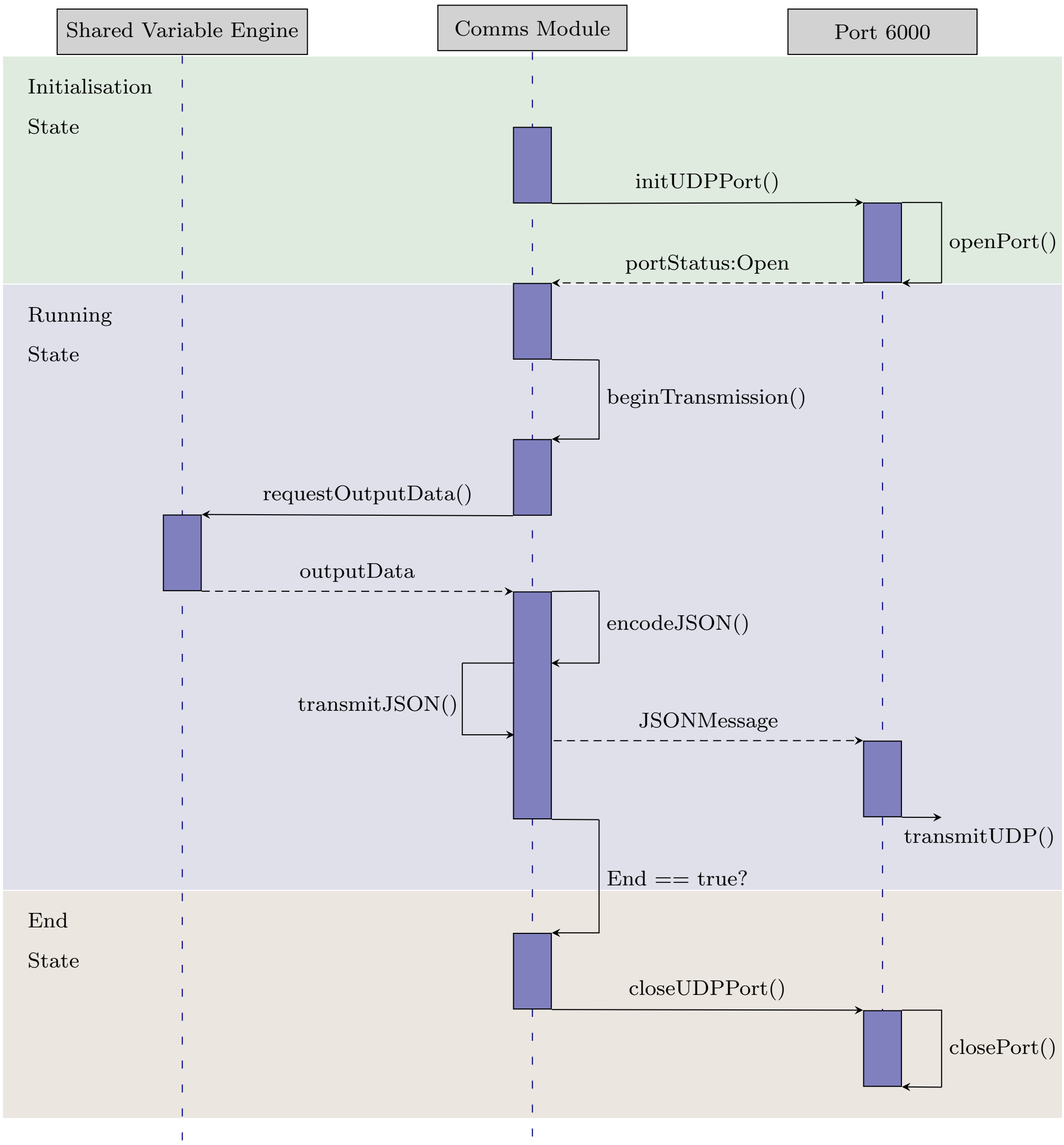

Figure 8.5: *UML Sequence Diagram for the Output Parallel Operation of the Communications Module.*

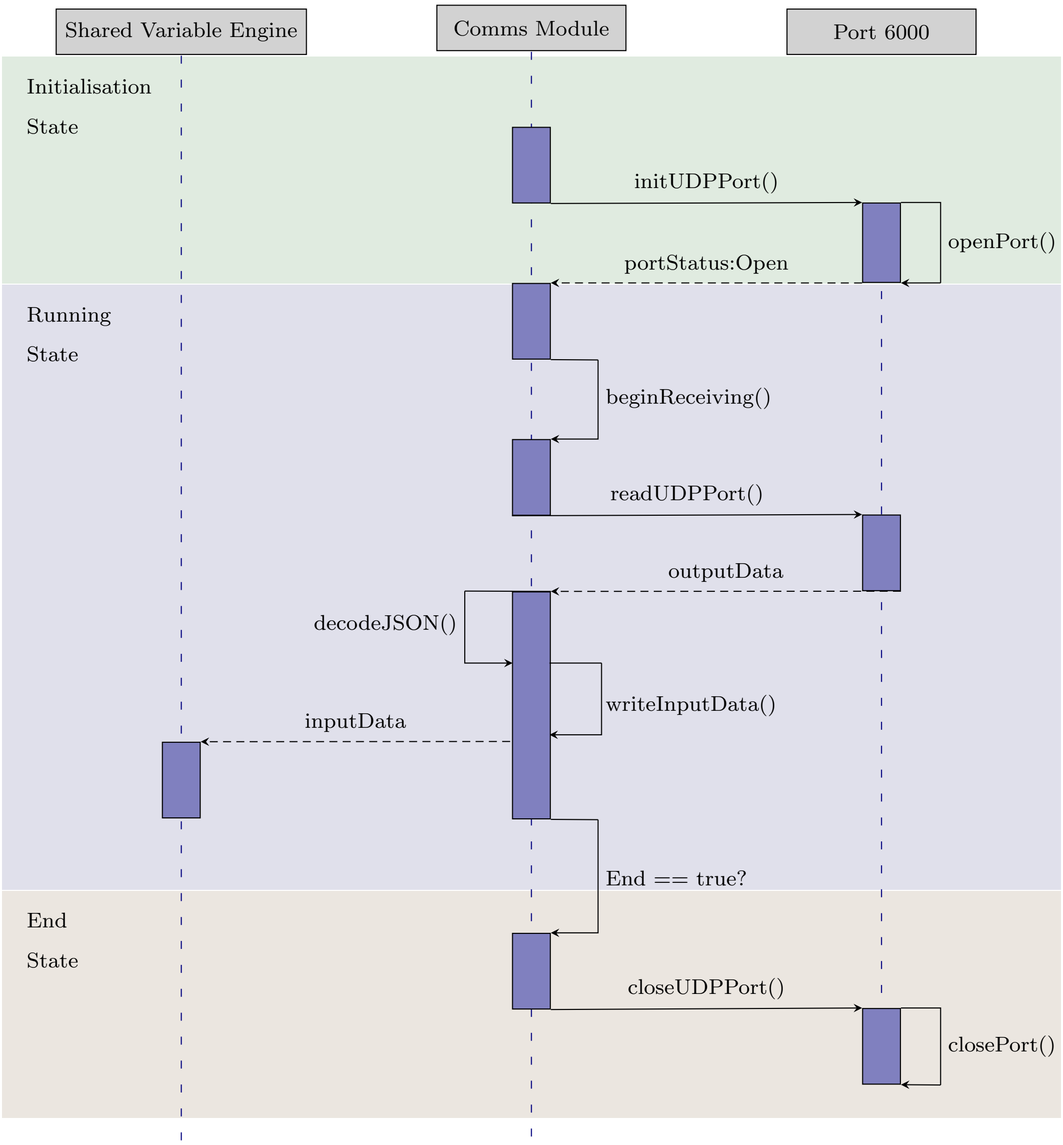

Figure 8.6: *UML Sequence Diagram for the Input Parallel Operation of the Communications Module.*

8.4 Error Module

8.4.1 Overview and Responsibilities

The Error Module is responsible for safety in the LLC. Considerable effort has been made to identify and mitigate potential code errors in LLC, with common errors already handled at source. A good example of this may be found in the Communications Module, where LabVIEW Error 56 is generated when reading the receiving UDP port if there is no message available, which will halt the operation of the LLC in a potentially dangerous state. In this event it is programmed for the LLC to ignore this error and continue operation.

8.4.2 Intermodule Data Communication

The Error Module is responsible for determining the lifespan of the Error Module Queue, which allows any other module to alert the Error Module of an unexpected error.

8.4.3 Operating procedure

Figure 8.8 shows the UML Activity diagram for the Error Module. Figure 8.9 shows the UML Sequence diagram for the Error Module. Figure 8.7 shows the UML Activity diagram for an external module enqueuing an error message to the Error Module Message Queue.

During the Initialisation Phase the Error Queue is created. During the Running Phase the Error Module awaits a message in the Error Queue. If there occurs an unexpected (and therefore unhandled) error in any module the details of this error are enqueued as a formatted error message into the Error Module Queue and handled by the Error Module. When a message is available in the Error Queue an event fires which prompts the Error Module to dequeue the message. The Error Module then enqueues a High priority Log message into the Logging Module Queue and enqueues a shutdown request into the Program Control Queue. This ensures that the details of the error are logged for further analysis and that a safe shutdown procedure occurs. When an error message is handled by the Error Module the Error Module Queue is flushed to remove all other messages. This ensures the fastest possible safe shutdown of the LLC. During the End Phase the Error Module Queue is destroyed.

8.4.4 Relevance in the Control Hierarchy

This module plays a supporting role in the overall control hierarchy of MyPAM. It doesn't perform any control function in terms of the rehabilitation robotics control hierarchy presented by Figure 2.4, but it is necessary in the infrastructure of the LLC.

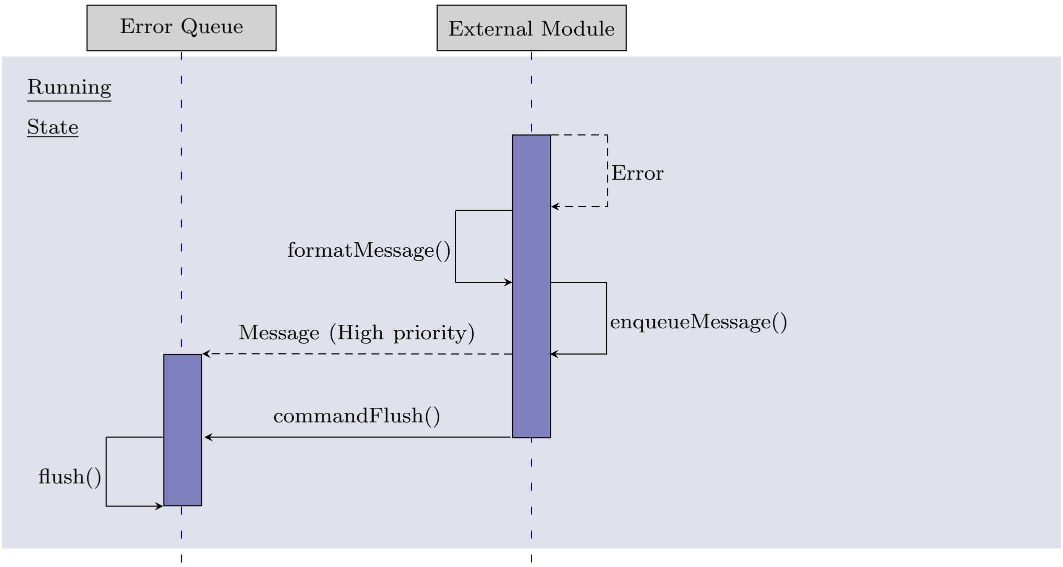

Figure 8.7: *UML Sequence Diagram for an External Module Enqueuing an Error Message to the Error Module Message Queue.*

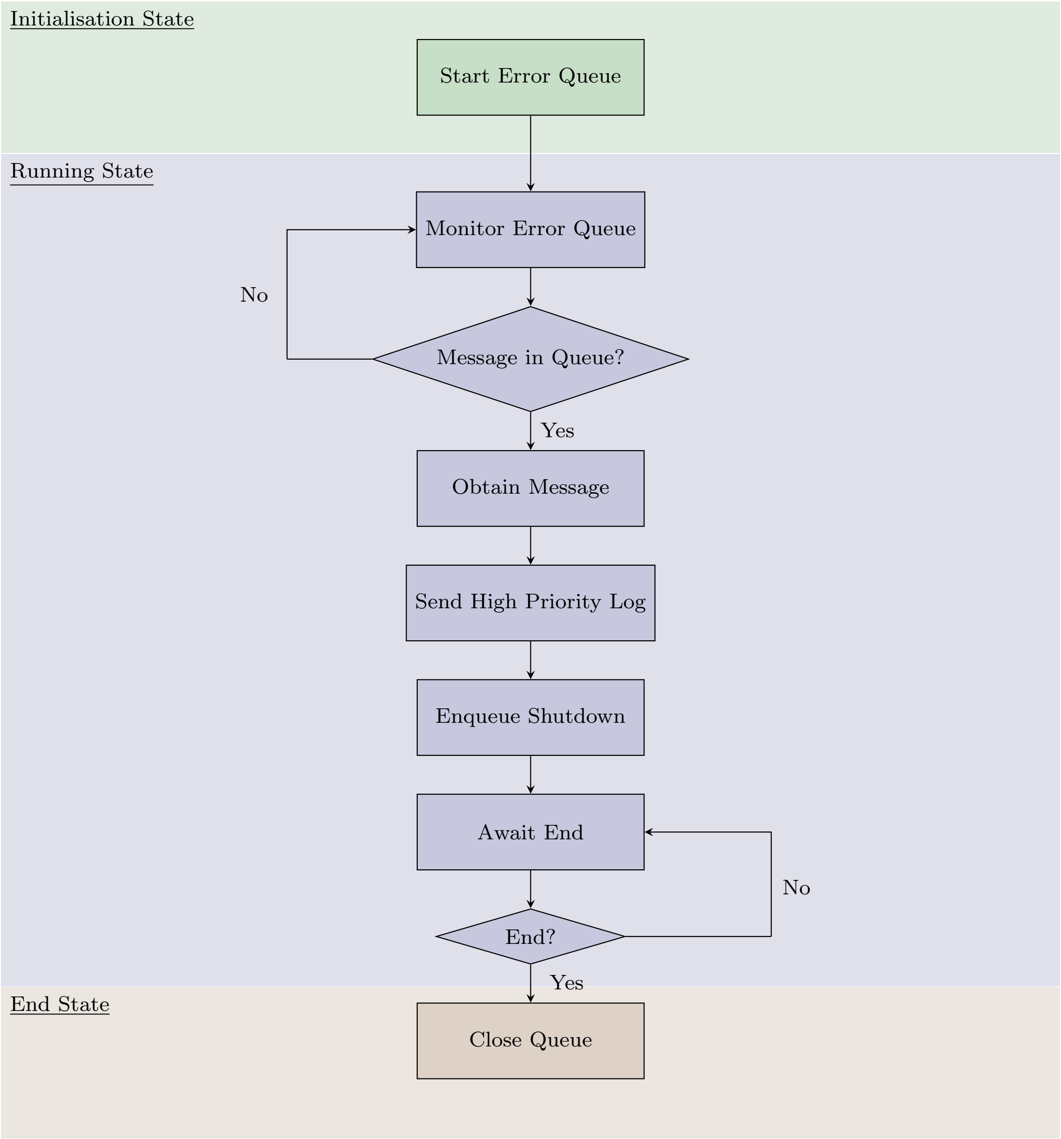

Figure 8.8: *UML Activity Diagram for the Error Module.*

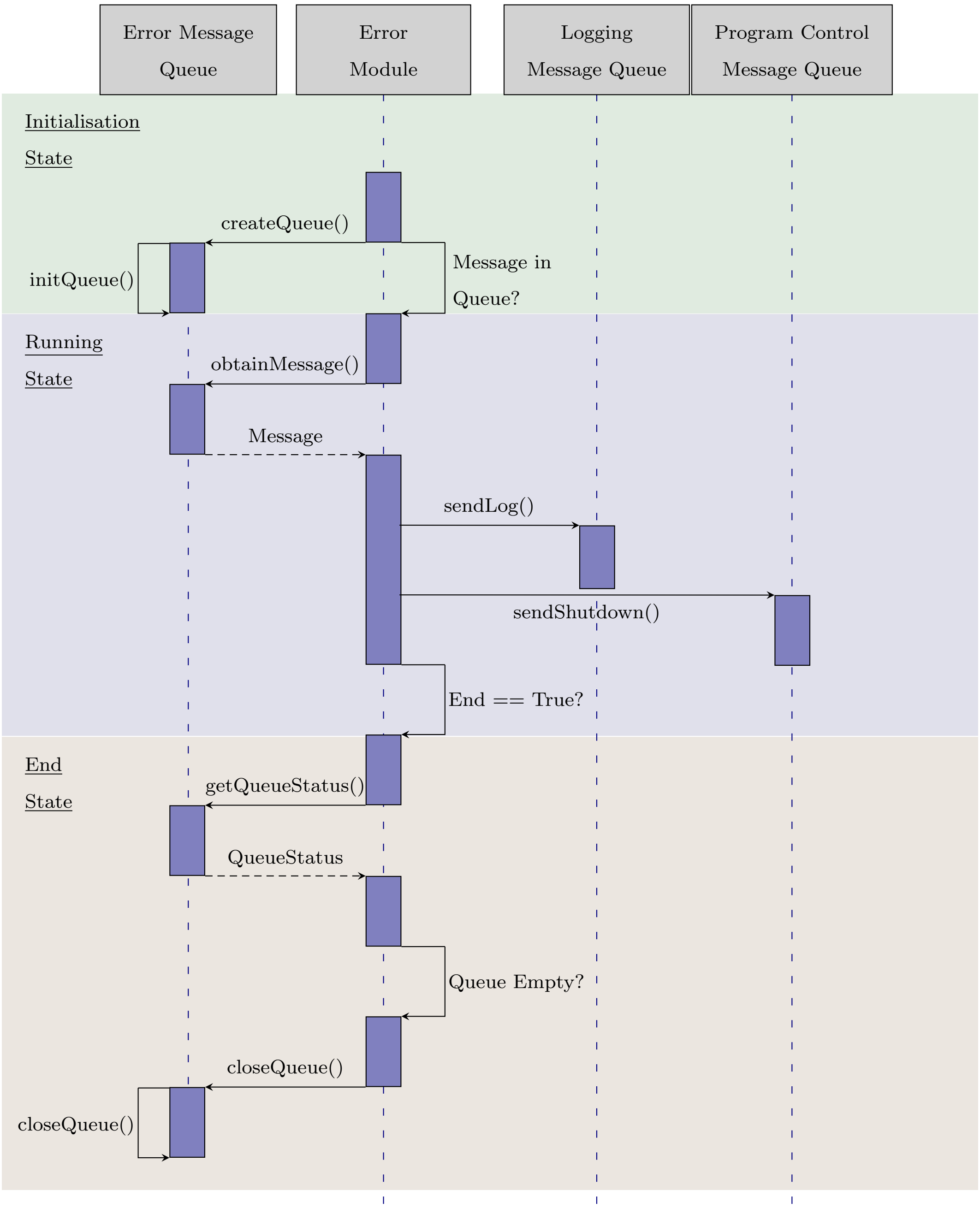

Figure 8.9: *UML Sequence Diagram for the Error Module.*

8.5 Logging Module

8.5.1 Overview and Responsibilities

The Logging Module is responsible for writing to an external file the main events of the operation of the LLC. Each module enqueues a low priority message to the Logging Module Queue when the startup event is received from the Program Control Module, when the Initialisation Phase is complete and the Running Phase begins, and when the End Phase is begun. The Error Module enqueues a high priority message to the Logging Module Queue if an error has occurred. When a high priority message is enqueued the Logging Module Queue is cleared of all low priority messages to ensure that the safe shutdown procedure is not held up. It is important to maintain a log for each running session of MyPAM because it allows the module location and location in the sequence of operations of unexpected errors to be determined. This in turn allows the root cause error to be determined and ultimately prevented from occurring again.

8.5.2 Data Communication

The Logging Module is responsible for determining the lifespan of the Logging Module Queue, which allows all other modules to send a message for logging.

8.5.3 Operating procedure

Figure 8.10 shows the UML Sequence diagram for an external module enqueuing a low priority message to the Logging Module Message Queue. Figure 8.11 shows the UML Sequence diagram for an external module enqueuing a high priority error message to the Logging Module Message Queue. Only the Error Module will send a high priority message in the event of an error.

Figure 8.12 shows the UML Activity diagram and Figure 8.13 shows the UML Sequence diagram for the Logging Module.

During the Initialisation Phase the Logging Module Queue is created. During the Running Phase the logging Module awaits a message in the Logging Module Queue. When a message is present it is dequeued, formatted and written to the log. During the End Phase the Logging Module Queue is destroyed.

8.5.4 Relevance in the Control Hierarchy

This module plays a supporting role in the overall control hierarchy of MyPAM. It doesn't perform any control function in terms of the rehabilitation robotics control hierarchy presented by Figure 2.4, but it is necessary in the infrastructure of the LLC.

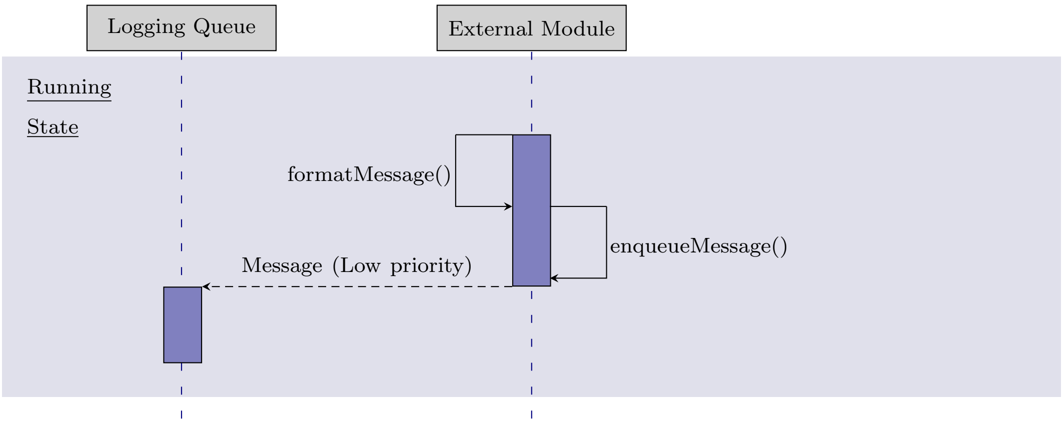

Figure 8.10: *UML Sequence Diagram for an External Module Enqueuing a Low Priority Message to the Logging Module Message Queue.*

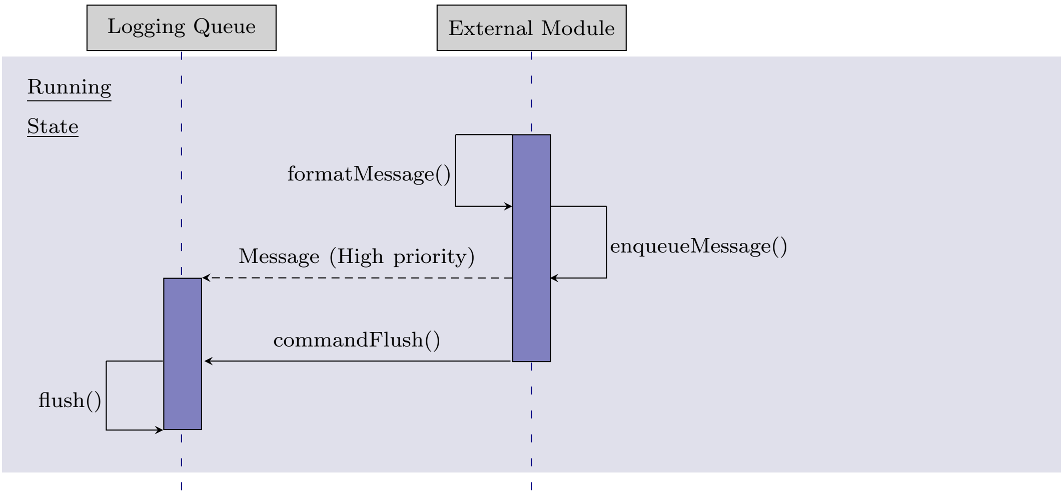

Figure 8.11: *UML Sequence Diagram for an External Module Enqueueing a High Priority Error Message to the Logging Module Message Queue.*

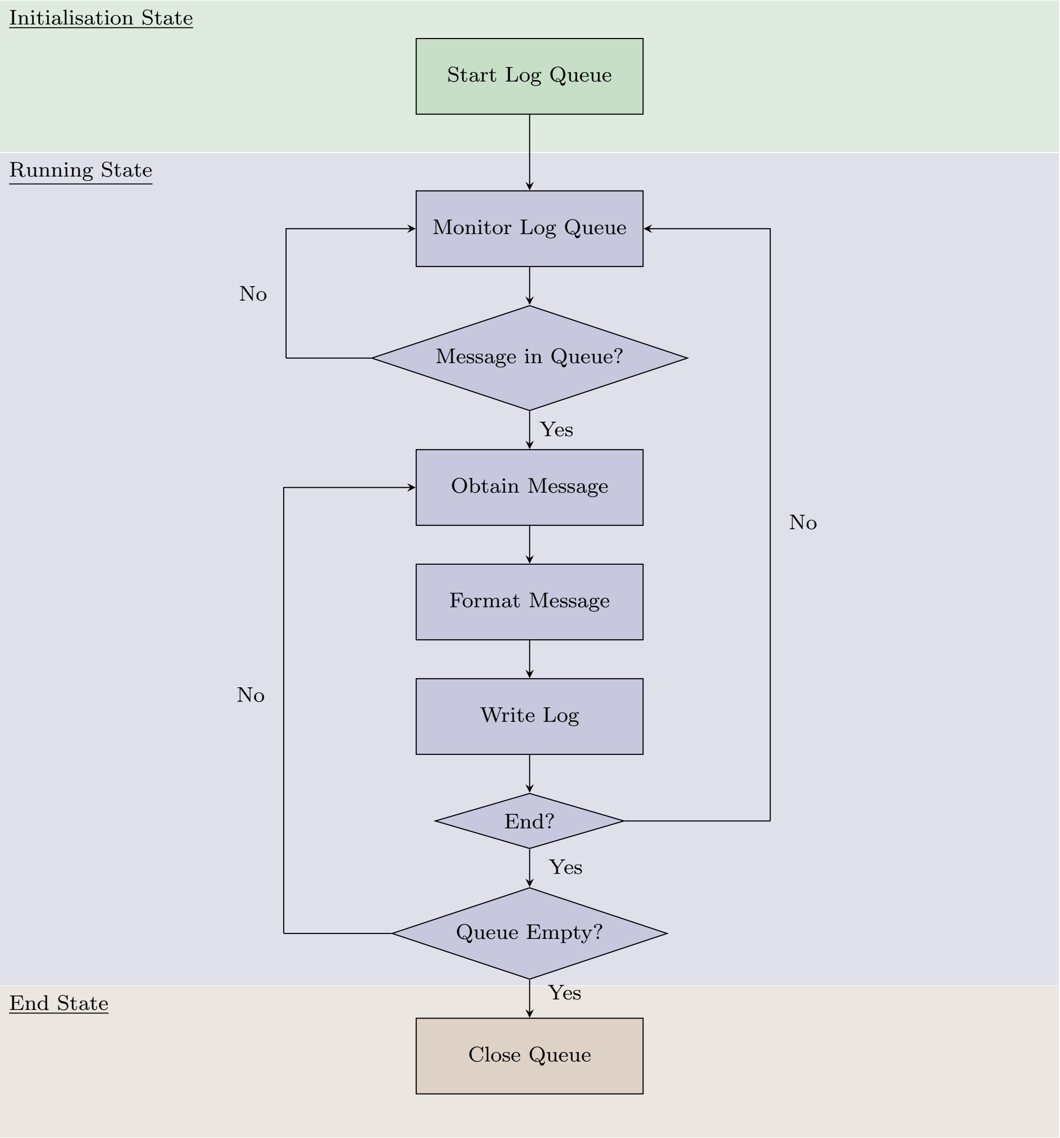

Figure 8.12: *UML Activity Diagram for the Logging Module.*

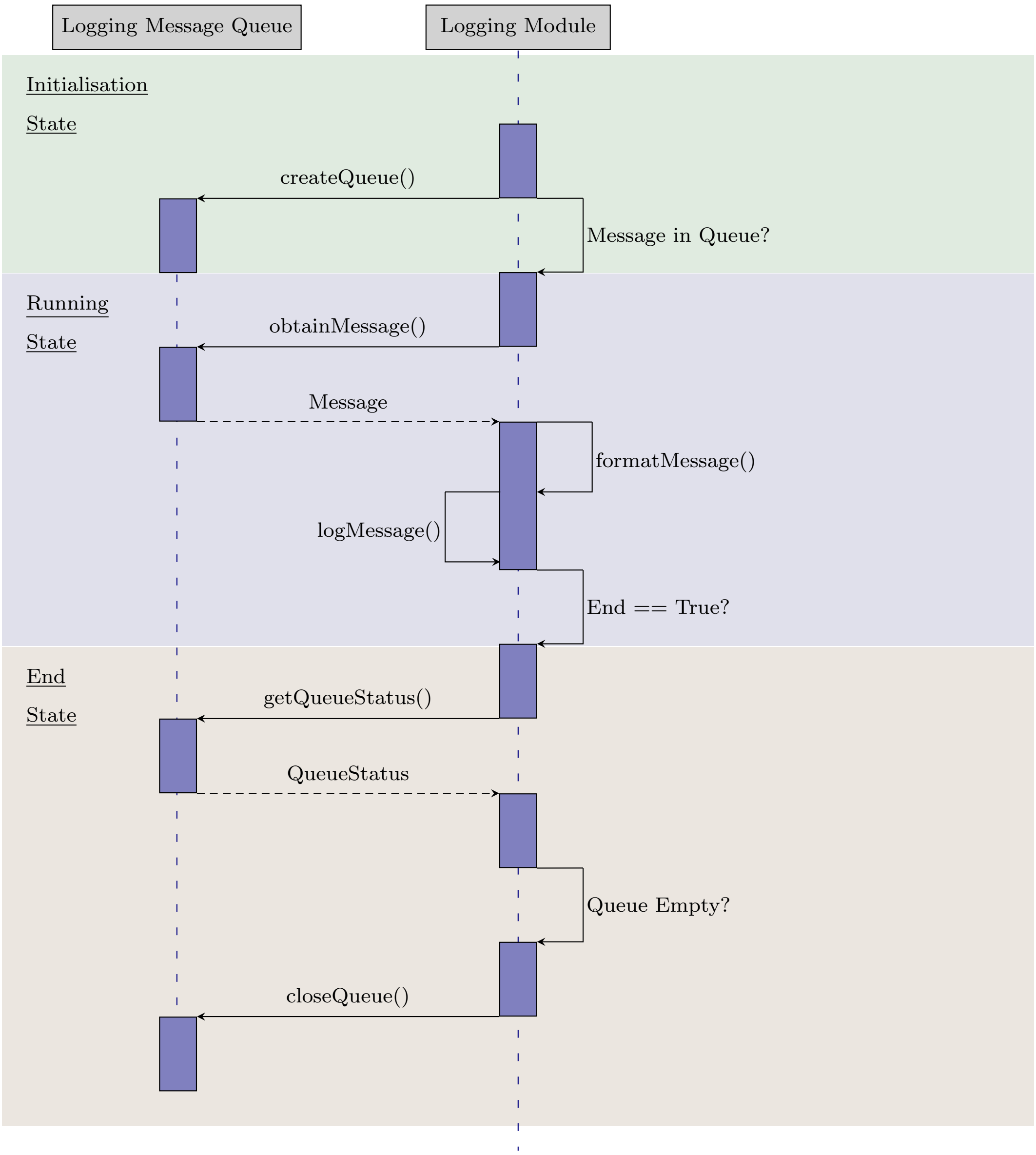

Figure 8.13: *UML Sequence Diagram for the Logging Module.*

8.6 Data Acquisition Module

8.6.1 Overview and Responsibilities

The Data Acquisition Module is responsible for processing all MyPAM sensor data. Most sensor data is acquired by the FPGA, but resource limitations in the FPGA leave it unable to fully process the data. The Data Acquisition Module converts the encoder counts into joint angles and performs forward kinematics to obtain the X and Y position of all joints. Note that the Data Acquisition Module has been designed with application future scalability in mind, such that external data acquisition equipment may be easily interfaced (for example, a force sensor connected to a separate microcontroller).

8.6.2 Intermodule Data Communication

The Data Acquisition Module is responsible for determining the lifespan of shared global variables used for transferring data from the FPGA to the LLC.

8.6.3 Operating procedure

Figure 8.14 shows the UML Activity diagram and Figure 8.15 shows the UML Sequence diagram for the Data Acquisition Module. During the Initialisation Phase the shared global variables are created. During the Running Phase the shared global variables are sampled at a rate of 1kHz to obtain the encoder count for each joint. The encoder counts are processed into joint angles and global X and Y positions of the joints, which are written to the SVE. During the End Phase the shared global variables are destroyed.

8.6.4 Relevance in the Control Hierarchy

This module monitors the sensing and actuation of the robot control signals, and thus serves as part of the 'sensing/actuation' block in the rehabilitation robotics control hierarchy presented by Figure 2.4. Note this this is distinct from the FPGA in that it has been build modularly for future expansion. The FPGA deals with raw analog or digital signals, but this module may deal with data communicated though higher level protocols such as UART, SPI or I2C. A good example of this implemented is found in Figure 5.4 in Chapter 5 and Figure 6.10 in Chapter 6.

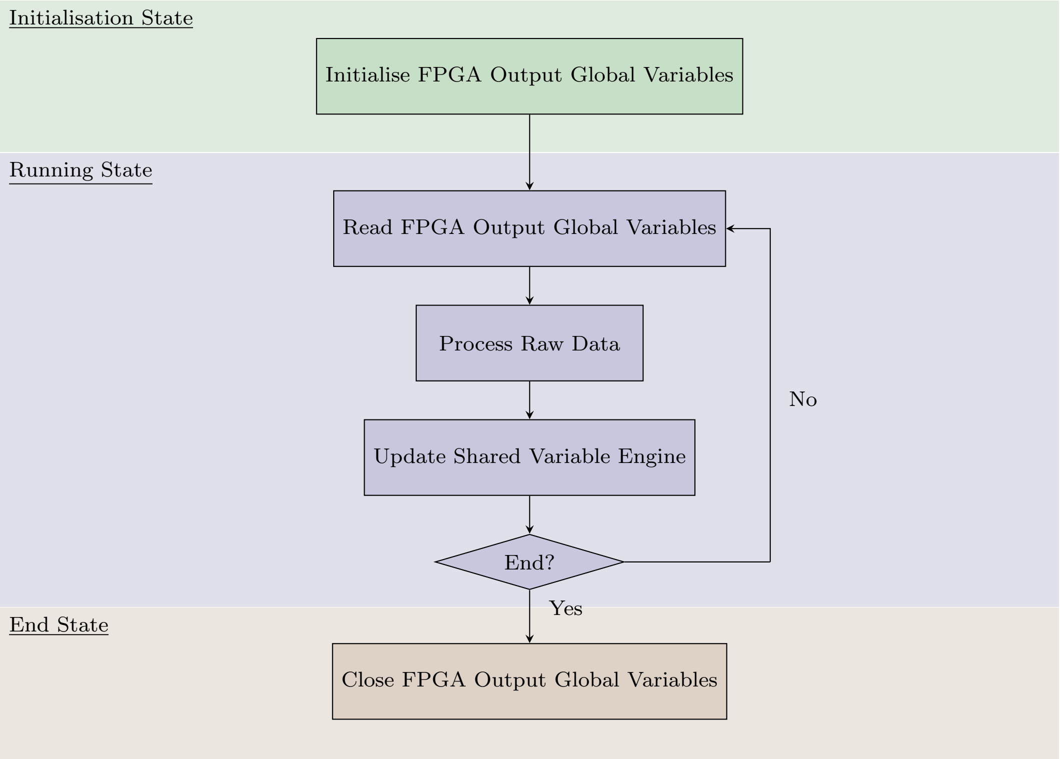
Initialisation State
Initialise FPGA Output Global Variables
Running State
Read FPGA Output Global Variables
Process Raw Data
Update Shared Variable Engine
End?
No
Yes
End State
Close FPGA Output Global Variables

Figure 8.14: *UML Activity Diagram for the DAQ Module.*

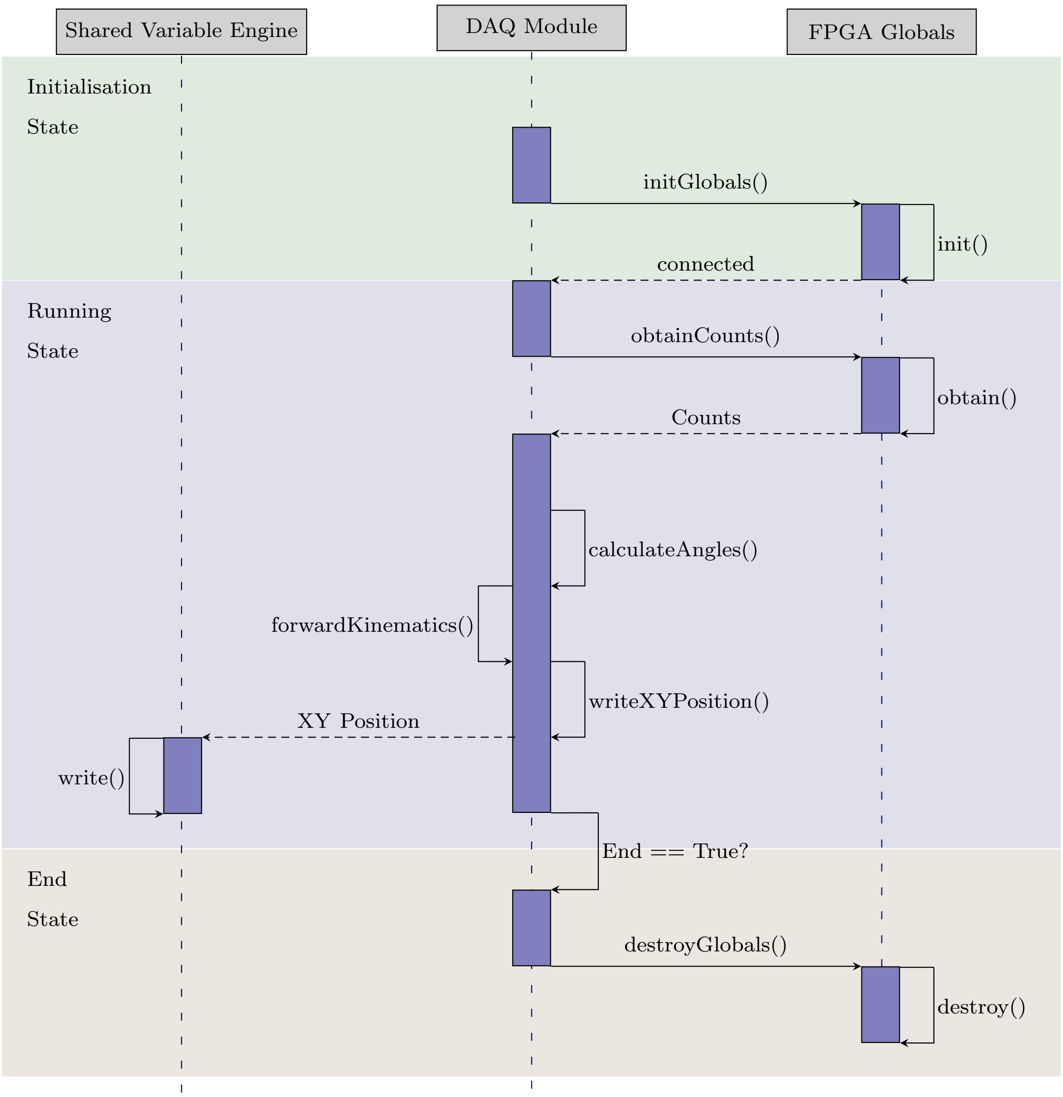

Figure 8.15: *UML Sequence Diagram for the DAQ Module.*

8.7 Sensor and Actuator Input/Output on the FPGA

8.7.1 Overview and Responsibilities

The FPGA on the myRIO is responsible for handling sensor input and actuator output on the robot. Note that the FPGA is always running when the myRIO is powered, there is no startup or shutdown procedure beyond writing default zero values to the motors. The inherent parallelism of the FPGA is exploited to achieve eight parallel operations occurring simultaneously at a rate of 25MHz:

1. Output the desired direction of rotation signal to motor 0 controller.
2. Output the desired direction of rotation signal to motor 1 controller.
3. Output the enable/disable signal to motor 0 controller.
4. Output the enable/disable signal to motor 1 controller.
5. Generate and output the PWM signal to motor 0 controller.
6. Generate and output the PWM signal to motor 1 controller.
7. Read Phases A and B from encoder 0 and update the encoder count.
8. Read Phases A and B from encoder 1 and update the encoder count.

8.7.2 Reading the Encoders

Rotary encoders produce noisy output oscillations due to switch bounce. Switch bounce refers to a very fast mechanical bounce of the components of the embedded switches in the encoder as they settle into their new position, causing the underlying circuit to be opened and closed several times. This produces excess counts, affecting the accuracy of the joint angle measurement. Switch bounce in the encoders was countered using two strategies. The first strategy was the implementation of a simple digital filter, whereby a state change in an encoder phase was registered only if it had been stable for 10 clock cycles (400 ns). The second strategy involved a decoding method, whereby the encoder phase state changes are used to increment/decrement a count, which was robust to unexpected changes.

Decoding strategy

The outputs an encoder are in quadrature, with the edge transitions of Phase A offset from the edge transitions of Phase B, as shown by Figure 8.16. Only one edge may transition at any period in time, which usefully means that only one edge may bounce at any time.

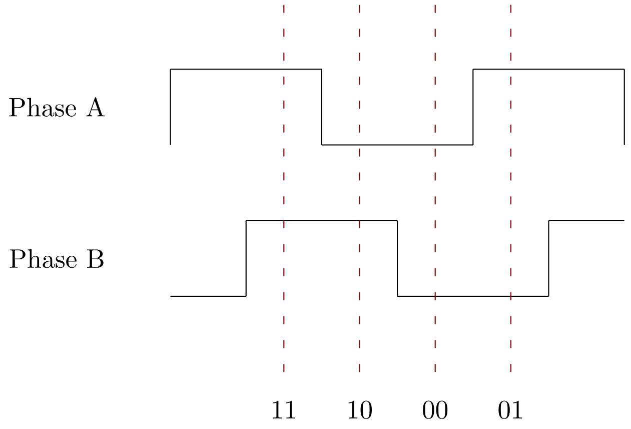

Figure 8.16: *Encoder Phases in Quadrature.*

A 2-bit binary number was constructed using Phase A for the least significant bit (LSB) and Phase B for the most significant bit (MSB). There exist only eight valid state transitions:

Clockwise	11 → 10 → 00 → 01 → 11
Anticlockwise	01 → 00 → 10 → 11 → 01

A 4-bit binary number was formed with the two MSBs formed of the 2-bit number from the previous states and the two LSBs are formed of the 2-bit number from the current state. This means that eight numbers exist which represent a valid state change, shown by Table 8.3.

Table 8.3: *Valid State Changes For Encoder Measurements.*

Clockwise	Anticlockwise
1110	0100
1000	0010
0001	1011
0111	1101

A look up table (LUT) was formed which is used to return 0,1 or -1 to adjust the encoder count depending on the transition. The input to the LUT is the 4-bit number and the output is the count adjustment. The LUT is shown by Table 8.4.

Table 8.4: *The Quadrature Encoder Decoding Look Up Table (LUT).*

4-bit Number	Valid State Transition?	Direction	Output
0000	No	X	0
0001	Yes	CW	1
0010	Yes	ACW	-1
0011	No	X	0
0100	Yes	ACW	-1
0101	No	X	0
0110	No	X	0
0111	Yes	CW	1
1000	Yes	CW	1
1001	No	X	0
1010	No	X	0
1011	Yes	ACW	-1
1100	No	X	0
1101	Yes	ACW	-1
1110	Yes	CW	1
1111	No	X	0

Thus, at iteration n for example, a 4-bit number of 0001 would indicate that a clockwise rotation has occurred and that the encoder count must be incremented. This decoding method is robust to bounce, as shown by the case study below.

Decoding Case Study

In this section a case study examines the output of the LUT for a clockwise step: 11 → 10. For a clockwise step the encoder count should increment by 1.

A clockwise step with no bounce is shown by Table 8.5. Note that the net adjustment to the encoder count is +1.

Table 8.5: *Quadrature Encoder Decoding Case Study - A Clockwise Step with No Bounce.*

Iteration	i	i+1
Number Generated	xx11	1110
LUT Output		+1

A clockwise step with one bounce in Phase A is shown by Table 8.6. Note that the net adjustment to the encoder count is +1.

Table 8.6: *Quadrature Encoder Decoding Case Study - A Clockwise Step with One Bounce.*

Iteration	i	i+1	i+2	i+3
Number Generated	xx11	1110	1011	1110
LUT Output		+1	-1	+1

A clockwise step with two bounces in Phase A is shown by Table 8.7. Note that the net adjustment to the encoder count is +1.

Table 8.7: *Quadrature Encoder Decoding Case Study - A Clockwise Step with Two Bounces.*

Iteration	i	i+1	i+2	i+3	i+4	i+5
Number Generated	xx11	1110	1011	1110	1011	1110
LUT Output		+1	-1	+1	-1	+1

A clockwise step with two bounces in Phase A is shown by Table 8.8. Due to the sampling rate a bounce return is missed (ie, iteration *i+3* from Table 8.7 is omitted). Note that the net adjustment to the encoder count is +1.

Table 8.8: *Quadrature Encoder Decoding Case Study - A Clockwise Step with Two Bounces and One Bounce Return Missed.*

Iteration	i	i+1	i+2	i+3	i+4
Number Generated	xx11	1110	1011	1111	1110
LUT Output		+1	-1	0	1

8.7.3 Data Communication

Data is exchanged between the FPGA and the LLC using two sets of global variables. The first set is created by the Data Acquisition Module and is used for transferring sensor data from the FPGA to the LLC. The second set is created by the Control Module and is used for transferring actuator data from the LLC to the FPGA.

8.7.4 Operating procedure

Figure 8.17 shows the UML Sequence Diagram for the FPGA parallel operation for enabling or disabling of a motor. Note that this operation occurs in parallel with all other FPGA operations and is duplicated identically for both motor 0 and motor 1. Figure 8.18 shows the UML Sequence Diagram for the FPGA parallel operation for writing the direction of rotation to the motor. Figure 8.19 shows the UML Sequence Diagram for the FPGA parallel operation for writing PWM to a Motor. Figure 8.20 shows the UML Sequence Diagram for the FPGA parallel operation for reading the encoder and updating the encoder count.

8.7.5 Relevance in the Control Hierarchy

This module monitors the sensing and actuation of the robot control signals, and thus serves as part of the 'sensing/actuation' block in the rehabilitation robotics control hierarchy presented by Figure 2.4.

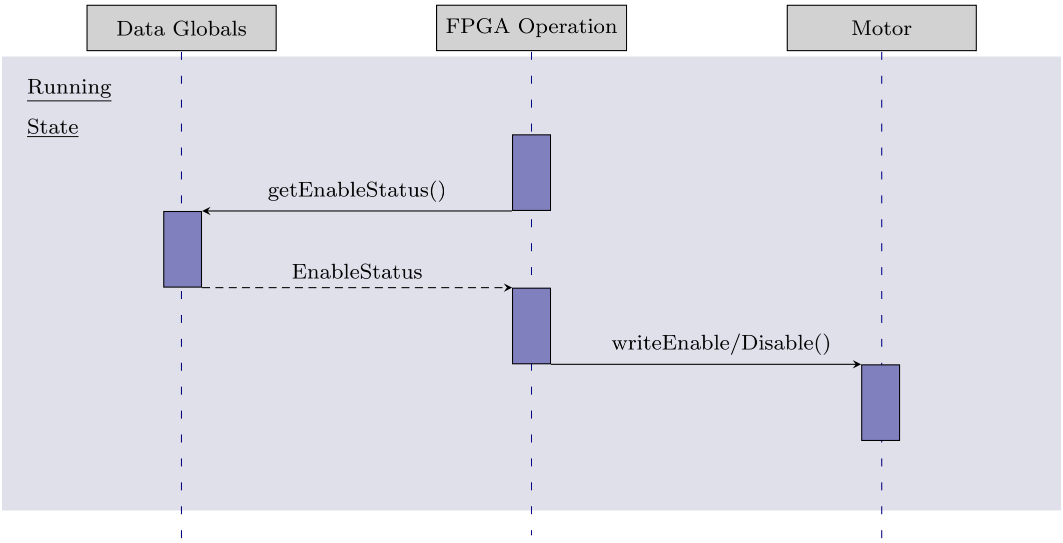

Figure 8.17: *UML Sequence Diagram for the FPGA Parallel Operation for Enabling or Disabling a Motor.*

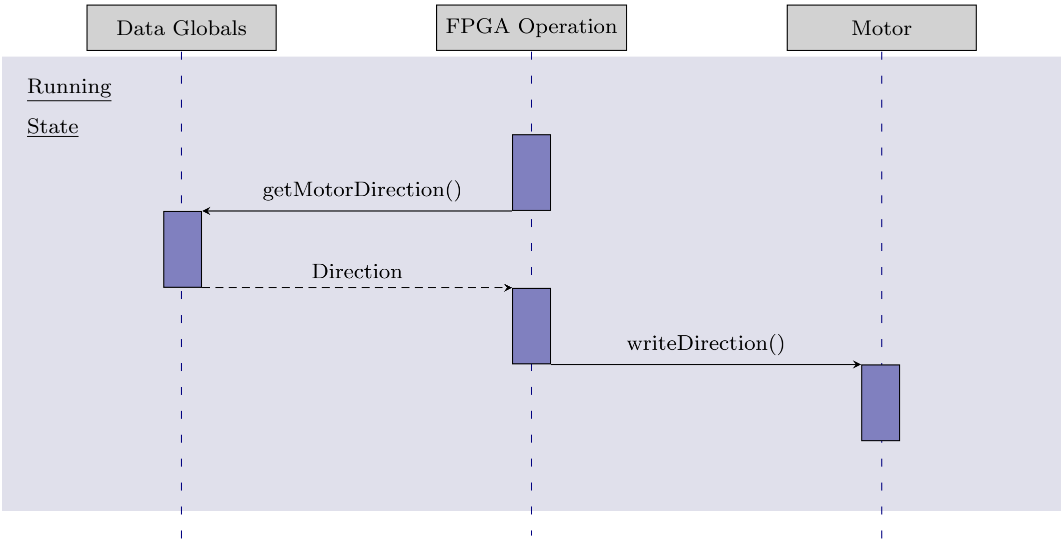

Figure 8.18: *UML Sequence Diagram for the FPGA Parallel Operation for Writing the Direction of Rotation to a motor.*

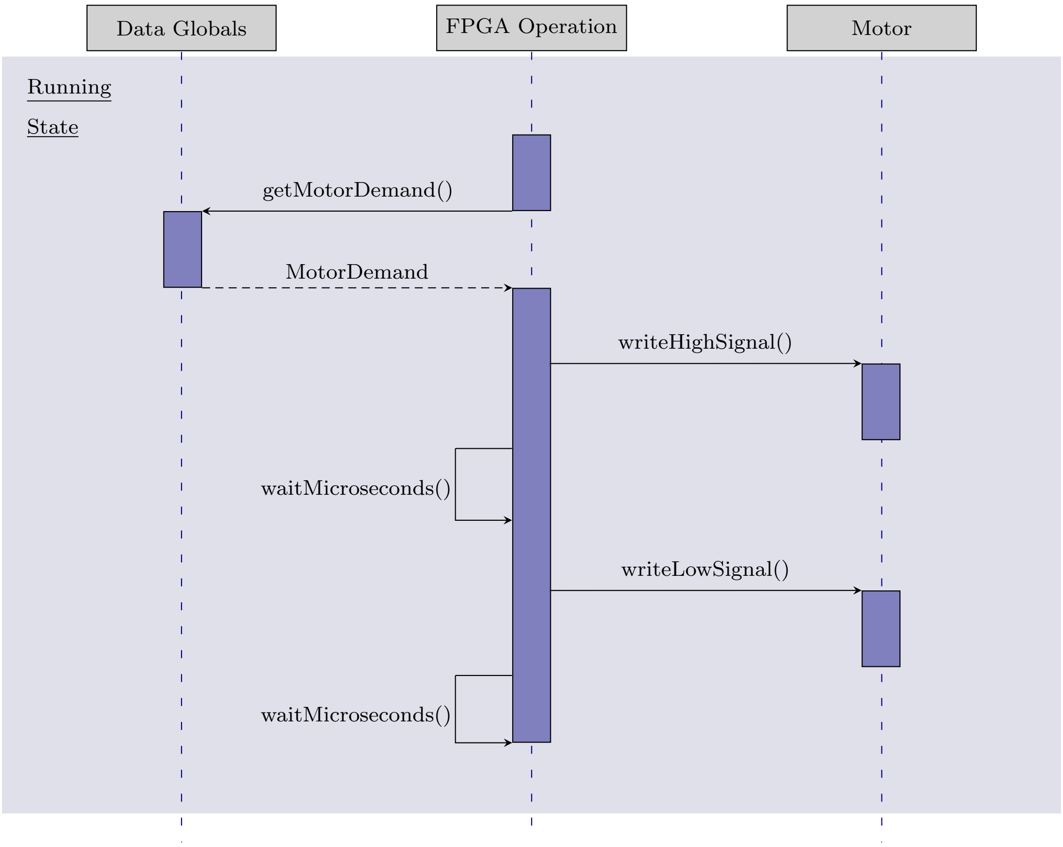

Figure 8.19: *UML Sequence Diagram for the FPGA Parallel Operation for writing PWM to a Motor.*

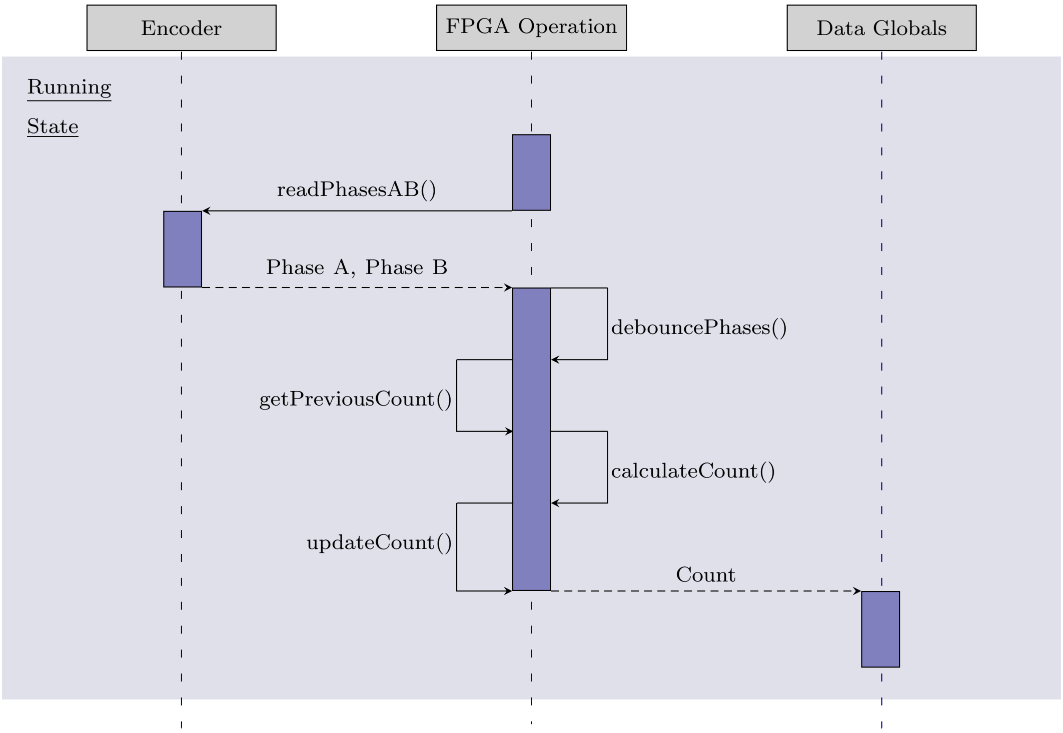

Figure 8.20: *UML Sequence Diagram for the FPGA Parallel Operation for Reading the Encoder and Updating the Encoder Count.*

8.8 Control Module

8.8.1 Overview and Responsibilities

The Control Module is responsible for generating the next trajectory setpoint, implementing robot control, and transmitting the desired motor values to the FPGA.

8.8.2 Intermodule Data Communication

The Control Module is responsible for determining the lifespan of shared global variables used for transferring data from the LLC to the FPGA.

8.8.3 Operating procedure

Figure 8.21 shows the UML Activity diagram and Figure 8.22 shows the UML Sequence diagram for the Control Module. During the Initialisation Phase the shared global variables are created and written with default values. During the Running Phase the control data is read from the SVE. The current position of the end effector is compared with the target position to determine whether any control needs to be performed. In the event that no control is necessary, default motor values are written to the FPGA. If the target position is not equal to the current position the next setpoint is evaluated in parallel with any adjustment due to Admittance and Attractors/Repulsors. Following this, inverse kinematics (derived in Chapter 4 are performed to obtain desired joint angles and PID control (detailed in Chapter 7 is performed to determine the necessary motor demands. The motor demands are written to the FPGA. During the End Phase default motor values are written to the FPGA before the global variables are destroyed.

8.8.4 Relevance in the Control Hierarchy

This module performs the majority of the work described in the other parts of the thesis, specifically:

1. The Minimum Jerk trajectory generation derived in Chapter 3.
2. The Attractors and Repulsors derived in Chapter 3.
3. The Inverse Kinematics derived in Chapter 4.
4. The Admittance Control Scheme derived in Chapter 7 and shown by the block diagram in Figure 7.4.

The generation of the next setpoint of the trajectory occurs at a rate of 1kHz and is described by the pair of Equations 8.2 and 8.2:

$$x(t) = x_i + (x_f - x_i)\left(6\left(\frac{t}{t_f}\right)^5 - 15\left(\frac{t}{t_f}\right)^4 + 10\left(\frac{t}{t_f}\right)^3\right) + \delta x_A + \sum \delta x_{ai} + \sum \delta x_{ri} \tag{8.1}$$

$$y(t) = y_i + (y_f - y_i)\left(6\left(\frac{t}{t_f}\right)^5 - 15\left(\frac{t}{t_f}\right)^4 + 10\left(\frac{t}{t_f}\right)^3\right) + \delta y_A + \sum \delta y_{ai} + \sum \delta y_{ri} \tag{8.2}$$

Where:

$x(t) = Next\ Intermediate\ Setpoint$

$x_i = Initial\ Position$

$x_f = Target\ Position$

$t_f = Target\ Time$

$t = Current\ Time$

$\delta x_A = Adjustment\ Due\ To\ Admittance$

$\delta x_{ai} = Adjustment\ Due\ To\ Attractor\ i$

$\delta x_{ri} = Adjustment\ Due\ To\ Repulsor\ i$

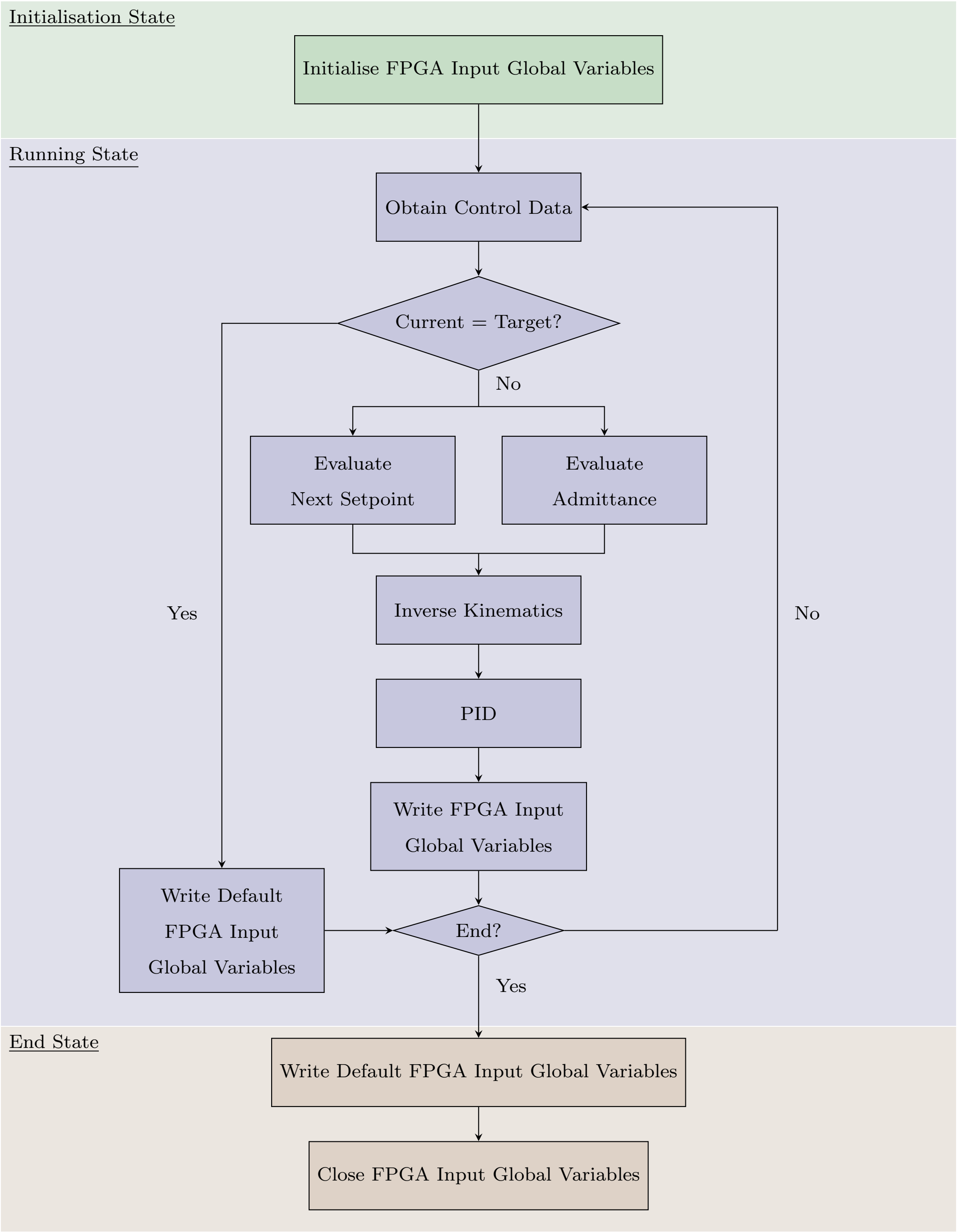
Initialisation State
Initialise FPGA Input Global Variables
Running State
Obtain Control Data
Current = Target?
No
Evaluate Next Setpoint
Evaluate Admittance
Yes
Inverse Kinematics
No
PID
Write FPGA Input Global Variables
Write Default FPGA Input Global Variables
End?
Yes
End State
Write Default FPGA Input Global Variables
Close FPGA Input Global Variables

Figure 8.21: *UML Activity Diagram for the Control Module.*

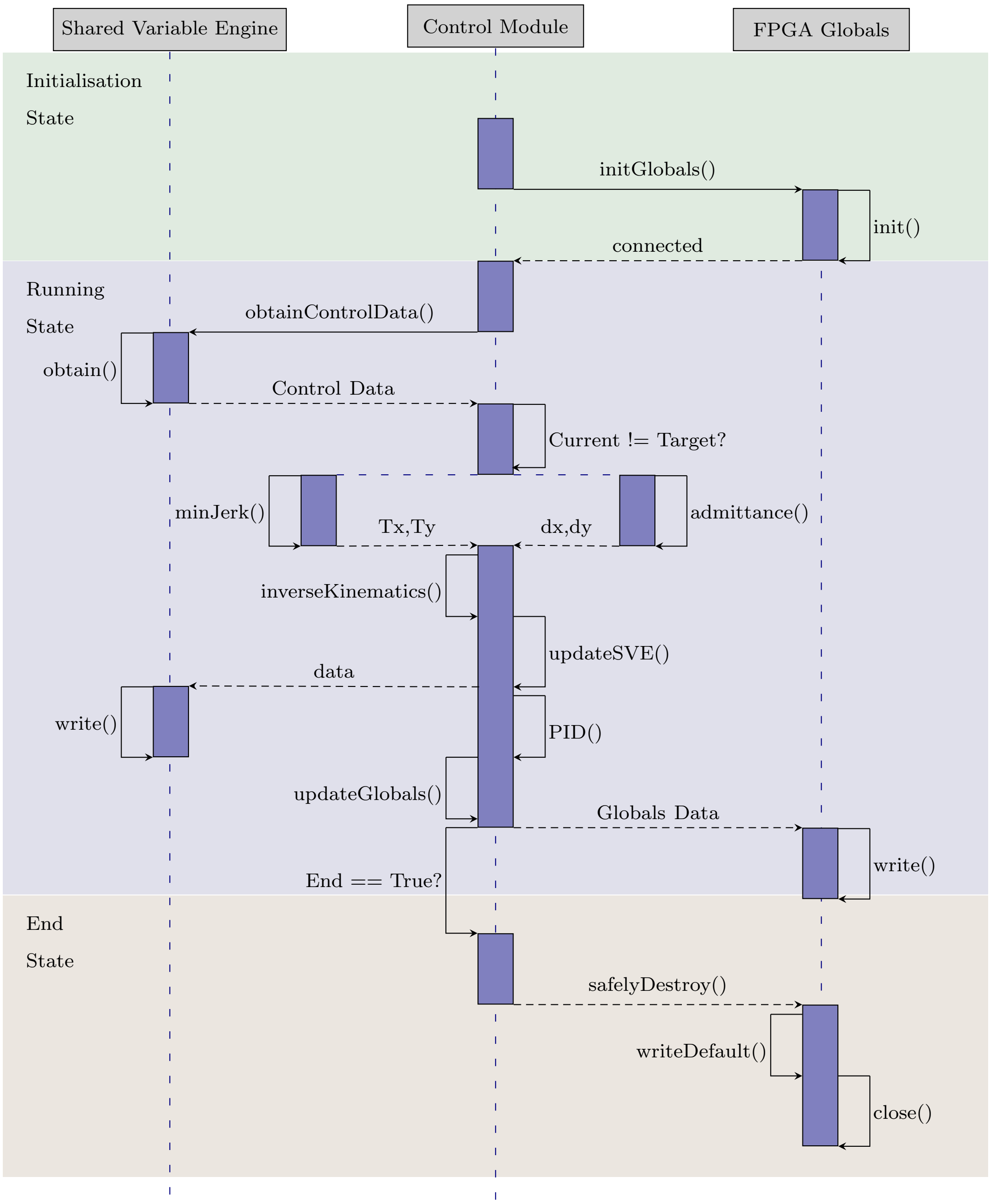

Figure 8.22: *UML Sequence Diagram for the Control Module.*

8.9 Chapter Summary

This chapter presents the software architecture of the Low Level Controller. The role of the LLC in the overall system architecture of the MyPAM is explained. The LLC comprises a number of software modules with distinct tasks. Chapter objective 8.1 is met by the following:

1. The Program Control Module is documented in Section 8.2.
2. The Communications Module is documented in Section 8.3.
3. The Error Module is documented in Section 8.4.
4. The Logging Module is documented in Section 8.5.
5. The Data Acquisition Module is documented in Section 8.6.
6. The FPGA programming is documented in Section 8.7.
7. Control Module is documented in Section 8.8.

In each of the Sections above comprehensive UML diagrams may be found which describe the operations in detail, meeting objective 8.3. Data communication are fully described in each section above, alongside a full justification of data communication procedures in the LLC which may be found in Section 8.1, meeting objective 8.2. The status of the chapter objectives is shown by Table 8.9.

Table 8.9: *Chapter 8 Objectives Status*

Objective	**Description**	**Success?**
8.1	Document each code module developed for the Low Level Controller.	Yes.
8.2	Fully describe and justify the data communication methods used for intermodule communication, intramodule communication, and communication between the Low Level Controller and the Mid Level Controller.	Yes.
8.3	Produce Unified Modelling Language Sequence Diagrams and Activity Diagrams to define processes in each software module.	Yes.

Chapter 9

General Discussion and Conclusion

Chapter Introduction

In this chapter the results of all chapters are discussed in the wider context of upper-limb rehabilitation robotics and the research objectives are assessed. The design and implementation of the low-level controller for this iteration of MyPAM is analysed and future research opportunities are identified.

9.1 Assessment of Research Objectives

The research objectives were presented in Section 1.1 in Chapter 1. This section evaluates the fulfilment of these objectives.

Objective 1: To review current research on upper limb rehabilitation robotic technology and identify opportunities for the development of MyPAM.
In Chapter 2 the system architecture of a rehabilitation robot was identified. Common trajectory generation strategies and low-level control strategies were explored. The societal need for low-cost rehabilitation robotic devices suitable for home use was identified. Patient motivation was identified as a key factor for successful rehabilitation outcome. A paucity was identified in research on the effect that different trajectory generation strategies have on patient outcome.

Objective 2: To develop and validate an appropriate trajectory generation strategy for MyPAM to optimise the rehabilitation outcomes of patients.
In Chapter 3 a discretised Minimum Velocity trajectory generation strategy and a discretised Minimum Jerk trajectory generation strategy were compared using a study with twenty participants. It was found that the Minimum Jerk trajectory promoted greater performance, and there was generally a user preference for the Minimum Jerk trajectory. Functions to adjust the discretised trajectory with points of Attraction and Repulsion were derived and validated, allowing the design of more engaging rehabilitation games. Patient engagement is linked with improved rehabilitation outcomes [42][43].

Objective 3: To develop and validate a detailed dynamic model for MyPAM.
A multidomain dynamic model encompassing the control domain, electro-mechanical domain and mechanical domain including frictional effects was presented in Chapter 4.

Objective 4: To develop, validate and deploy a sensor embedded end-effector/handle for MyPAM which may be used as a control input and a user input for rehabilitation games.
Chapter 5 presented a novel end-effector/handle for MyPAM capable of detecting three levels of input:

1. No user present.
2. User present.
3. Handle Squeezed.

User presence detection is useful as an input to the Low-Level Controller to determine which set of gains to use for the Position Controller. Handle squeeze detection is useful as an input to the High Level Controller as a game input.

Chapter 6 presented the further development of the grip sensor into a novel 2-axis force sensor. The force sensor does not successfully measure interaction force, however, and thus is not useful as a control input.

Objective 5: To develop and validate an appropriate low-level control scheme for MyPAM.
Chapter 7 presents two low-level control schemes. The first is a Position Controller with gains which are adjustable dependent on the presence of a user. The Position Controller is validated by

use throughout Chapters 3 and 4. The second is an Admittance Controller, with the effect of the Admittance filter in MyPAM validated.

Objective 6: To finalise, document, test and deploy a completed low-level control infrastructure for MyPAM.
Chapter 8 presents the design and documentation of the low-level control infrastructure. All data structure design choices and data communication methods are documented and justified. All code units are tested. The completed Low-Level Control system is validated through use in experiments in Chapters 3, 4, 7.

9.2 General Discussion

This section presents a detailed discussion of each chapter and each aspect of the Low-Level controller.

9.2.1 Trajectory Generation

In Chapter 3 the Minimum Jerk trajectory was shown to promote a better user performance than a Minimum Velocity trajectory when tested with twenty participants. Further to this there was a user preference to the Minimum Jerk trajectory. For this reason the Minimum jerk trajectory was selected as the most appropriate discretised trajectory (or task encoding) strategy for MyPAM. In the literature review it was found that there is no evidence that the use of any particular trajectory generation strategy over another did not promote neuroplasticity in the brain of a neurologically impaired user such as a Stroke patient. Indeed, whilst the Minimum Jerk trajectory has been shown to promote better performance, the testing environment was limited. The twenty participants were all able bodied, leaving very little room to improve, and none of them used the MyPAM for an extended period over a number of weeks. A more extensive test would use neurologically impaired participants for a concerted rehabilitation regime, with half selected to use the Minimum Velocity trajectory and half selected to use the Minimum Jerk trajectory.

In Chapter 3 a novel method for adjusting a discretised trajectory with points of Attraction and Repulsion was devised, with the intention that these could be used to create more engaging rehabilitation games on the High-Level Controller. It is clear that the points of Attraction and Repulsion were able to affect the target trajectory, but the effect on the path of the end effector of MyPAM depends on the ability of a patient to track the trajectory. The points of Attraction and Repulsion could instead have been applied directly to the end effector as a virtual force demand, which would have certainly had a greater effect on the response of MyPAM. It was chosen not to do this, however, because this had the potential to add instability to the system should a patient become trapped oscillating around a point of Attraction should they not have the physical ability to move the MyPAM back to the desired trajectory. The effect of the points of Attraction and Repulsion requires user testing to test two things:

1. Whether the behaviour is desirable.
2. Whether the behaviour increases motivation, decreases motivation, or produces no change in motivation.

9.2.2 Dynamic Modelling

The kinematic modelling presented in Chapter 4 is correct and provides necessary maths for implementing control. Likewise the derivation of the Jacobian Matrix is necessary for implementing Force Control, and therefore Impedance Control, though this wasn't implemented in this thesis because there is no way to measure the torque at the joints.

Friction Modelling was performed and an acceptable model was produced, though it is noted that there is a paucity of frictional torque data measured between 0 rad/s and around 1.7 rad/s. Whilst the gathered data was sufficient to have a good understanding of the breakaway frictional torque at 0 rad/s and a reasonable understanding of the effects of viscous frictional torque as the angular velocity increases it is difficult to identify with confidence the value of Coulomb frictional torque at the Stribeck velocity. This is a limitation because it is necessary to have as accurate a friction model as possible when the robot moves slowly. Despite this, the friction model was sufficiently accurate to add value to the Multi-Domain Dynamic model.

Chapter 4 culminates in the development and validation of a Multi-domain Dynamic model for MyPAM, which is a useful outcome. The model was capable of predicting the response of MyPAM to inputs with an acceptable level of accuracy, both in the unloaded condition and also when connected to the human arm proxy. The main limitation in the model is the effect of unmodelled backlash in the gear trains, which is noticeable particularly in joint 1. Further to this, The failure of the models to produce instability in response to a large control input means that this set of software is not appropriate for tuning the control system.

9.2.3 Grip and Force Sensing

The grip sensor presented by chapter 5 worked well, and it was capable of characterising three levels of contact useful as control inputs. The detection of grip/no grip is useful as an input to the low-Level Controller where it is used to determine which set of control gains to apply. The detection of squeeze is useful as an input to the High-Level Controller, where it may be used as an input to a rehabilitation game. This is important because it allows the creation of rehabilitation games which promote greater patient engagement, which has been shown to benefit rehabilitation outcomes.

The grip sensor was designed with the intention for further development into a 2-axis force sensor. This has resulted in a grip sensor which is difficult to manufacture, which demands two iterations of the silicone pouring and curing process. Whilst the sensing methodology is novel in this context, there is little additional benefit gained compared with using low-cost sensitive switches embedded into a handle/end-effector. Further to this it is likely that every grip sensor manufactured must be specifically calibrated due to differences in silicone due to mixing and casting conditions.

The grip sensor was designed with the aim for further development into a low cost 2-axis force sensor. As such the grip sensor is difficult to manufacture, with a sensing methodology which provides no additional benefits benefits over simply using sensitive switches. The force sensor prototypes presented in Chapter 6 did not successfully characterise a force input. The first prototype, based on the use of an array of single axis Hall Effects sensors, was not able to decouple the x and y components of force, which meant that a measurement in one axis affected the measurement in the other. This likely occurred because the magnetic field was too complex to be properly measured by the array of single axis Hall Effects sensors.

The second prototype of the force sensor, based on the use of an array of tri-axis Hall Effects sensors, performed even more poorly. It was not possible to train the Neural Network used to characterise the force beyond a relatively poor Mean Squared Error in the order of magnitude of 1×10^{-3}, with all tested Neural Networks showing overfitting. Overfitting is fundamentally caused by a mismatch between the complexity of the Neural Network and the complexity of the model it is trying to characterise. The design of the force sensor was flawed, in that the movement of of the outer core (embedded with magnets) was not sufficiently constrained to the inner core (embedded with Hall Effects sensors). It was desirable that the relative movement was free in 2 axes and constrained in the others. In reality the movement was apparent in all 6 axes. Coupling this movement with the complexity of the magnetic field caused by the 3 dimensional arrangement of magnets and the non-linear deformation of the silicone separation layer creates a highly complex model. It is likely not practical to create a Neural Network of sufficient complexity to match the model, and it is certainly not clear how complex the Neural Network must be.

9.2.4 Control

The control problem presented by a rehabilitation robot such as MyPAM is not negligible. In a simple linear dynamic system there are some clear performance criteria by which success may be specified and measured. In a linear system with a single setpoint there exist widely accepted modelling and control techniques, and a controller may be designed around performance criteria such as percentage overshoot, rise time, settle time and steady state error. Even in simple systems there is often a balance to be made because it is difficult to optimise performance for all criteria. For example, minimising steady state error often has implications on rise time or settle time. For MyPAM it is difficult to even define the control success criteria.

The first difficulty is presented by the implementation of the trajectory generation strategy. In the example above with a single setpoint, the reference location from which to measure the performance is clear. With a discretised trajectory this is no longer the case. It could be argued that when using a discretised trajectory, an intermediate setpoints may be used to evaluate the performance criteria. This presents, however, different problems:

1. The setpoints in a Minimum Jerk trajectory are not equally spaced, which means that optimising control gains with one intermediate setpoints does not necessarily improve performance for all intermediate setpoints. Indeed, it is likely to decrease performance at other setpoints.

2. The intermediate setpoints are generated live at a rate of 1kHz. In the wider context of the complete trajectory, such criteria as rise time and settle time if optimised, are likely to be meaningless if they are in the microsecond magnitude when the complete trajectory is in the magnitude of tens of seconds.

3. The velocity is not predictable at the start of a movement. In the example above, it is assumed that the velocity is zero (or at least a constant) at start time, allowing a controller to be tuned to optimise the performance criteria. In the case of MyPAM the velocity at the at the start of a movement between setpoints is affected by the velocity at the end of the previous movement between setpoints.

The second difficulty is presented by the non-linear dynamics of MyPAM, which means that the response of MyPAM to an input is different depending on its position. Coupled with this is the poor quality of the gear train in joint 1, which has a significant amount of backlash. The result of this is that the response of MyPAM to a response is difficult to predict, and depends not only on its position (ie the position of the joints relative to each other), but also its previous position. This is because the effects of the backlash come into effect if there has been a change in the direction demand of the motor at joint 1.

The third difficulty is presented by the interaction force at the end effector. In a traditional system this might be considered as a disturbance which needs to be rejected, but in a rehabilitation robot such as MyPAM the interaction force is a useful control input which can be used to protect the patient. The problem presented here is that because MyPAM is a low-cost system and the force sensor developed as part of this work was poor quality. Admittance Control, which uses the interaction force as a control input, requires the measurement of interaction force at the end effector. In this thesis a JR3 force/torque sensor was used and the use of expensive industrial force sensors like this is not uncommon in rehabilitation robotics. This adds significant cost, however, and is not appropriate for a low-cost robot designed for home installation. In the case of MyPAM the sensor cost more than the rest of the robot.

The use of Position Control alone in MyPAM is deemed good enough. Indeed, a similar control scheme was implemented during the successful trials of the first version of the robot. However, the use of Position Control alone limits the use of robots to a subset of patients with a sufficient level of ability, which means that a patient with a large amount if upper limb disability isn't able to safely use the device. This was the case during the trials for the first version of MyPAM where a participant was rejected from trials because their hemiparesis was deemed too severe.

There were limitations to testing of the control system in this thesis, particularly the Admittance Controller. The Position Control system was thoroughly tested, particularly during the trajectory comparison experiment in Chapter 3 and the modelling validation in Chapter 4, the

Admittance Control scheme was only tested with the passive human arm proxy due to the limitations caused by the Covid-19 pandemic. Ideally the Admittance control scheme would have been tested with participants to evaluate its performance with active inputs. Force Control or Impedance Control were not implemented because there is no way to measure torque at the joints of MyPAM currently. This is a limitation because it is suspected by the author that Impedance Control has a different use case to Admittance Control and would therefore extend the use of MyPAM to a wider range of patients. In the literature review is was highlighted that Impedance and Admittance are dual. It was stated that the end-effector should act as an Impedance in an environment which acts as an Admittance and vice versa. Impedance Controllers are better suited to stiff environments and Admittance Controllers are better suited to freer environments. For this reason it is posited that Impedance Control is suited to patients high a high level of spasticity, where the use of an Admittance Controller would be inappropriate.

9.2.5 Low-Level Controller Implementation

The finalised Low-Level Controller presented in Chapter 8 operates as intended, with each code module performing their respective tasks well. The software architecture was carefully considered and all design choices are fully justified. The architecture consisted of Modules, each responsible for a general task such as DAQ or Error Monitoring. Each Module consisted of Code Units, each responsible for a specific task such as Forward Kinematics or decoding information obtained from the MLC.

The strength of the implementation was the testing procedure, which was comprehensive and completed to an industry standard. Each Code unit was periodically unit tested. Integration testing was performed to ensure each Module worked as designed and communicated correctly with each other. System testing was performed to ensure the complete LLC performed as intended. The main challenge encountered during the implementation of the Low-Level Controller was computational resource limitation. This was particularly the case when programming on the FPGA, which operates at almost full capacity with 92% of the LUTs (Look Up Tables) used and 89% of Slices consumed. Ideally, more of the control would have been calculated at source on the FPGA, both in an effort to conserve resources on the RTOS and to react more quickly to the latest datapoints. Resource limitation defined much of the development of the Low-Level Controller, such is the implementation of the Admittance filter and the size of the Neural Network for the force sensor.

9.3 Novel Contributions

This thesis documents a number of novel contributions to the field. The methodology for applying Attractors and Repulsors to a live trajectory differs from previous work (notably, the force field methodology designed to affect the current position of a user proposed by Patton [110]). The main difference is that the Attractors and Repulsors detailed in this thesis affect the next setpoint of the trajectory as it is generated live as opposed to directly affecting the current position of the end effector of the robot as it moves into proximity of the Force field. This is an important distinction because it means that the effect is applied only to users who are performing well, as opposed to the force field which is applied to all users regardless of performance. The key implications of this are:

1. Applying a challenging force to a user who is performing badly may have a demotivating effect, which can have consequences on rehabilitation outcomes.

2. Greater patient safety when using the robot is ensured by preventing the end effector getting 'stuck' in an area of attraction, potentially causing instability of the control system.

This thesis presents a multidomain dynamic model of MyPAM in Chapter 4, which has been published. The model was necessary for creating a baseline against which patient progress can be tracked. Tracking patient progress through a course of robotic tele-rehabilitation requires constant position data logging and comparison, alongside periodic testing with no powered assistance. The test data must be compared with previous test attempts and an ideal baseline, for which a good understanding of the dynamics of the robot is required because the dynamics of the robot make it impossible to track a completely linear trajectory. In short, it is not accurate to track patient performance against a completely linear movement, but instead the performance must be compared to the response of the model.

Chapter 6 presents a novel application of the MagOne methodology for a 2-axis force sensor. The force sensor requires more work, but the application MagOne methodology did result in a tristate grip sensor in Chapter 5 which is useful addition to MyPAM as both a control input and also for integration into the games.

9.4 Future Work

Future work should focus on the development of a low-cost two axis force sensor, which is discussed in further detail in Chapter 6. The main factor in the use of a position control strategy as apposed to Admittance or Impedance control in previous iterations of MyPAM is the prohibitive cost of an industrial 6-axis force/torque sensor, the cost of which is more than the rest of the robot. One potential area of research is the further development of the force sensor presented in this thesis, with the relative motion of the outer and inner rigid cores be sufficiently constrained. Another route in this area is the use of FlexiForce piezoresistive sensors, though this was previously disregarded due to space limitations in the handle assembly.

Further development and testing of the integration of the Low-Level Controller with the Mid-Level Controller and High level controller is desirable, particularly for clinic testing and validation. Indeed, this is the direction that the work presented in this thesis was supposed to go before the Covid 19 Pandemic. In this thesis there are two useful developments designed with patient motivation in mind: The use of Attractors/Deflectors and the input of the grip sensor. Rehabilitation games which use these should be developed and tested for user feedback.

Due to the long lasting effects of COVID there are aspects of the work presented in this thesis which would benefit from more user testing. The Admittance Control scheme is tested only with the passive Human Arm Proxy, which is only able to resist the motion of the MyPAM rather than actively apply force. Two tests should be performed for the Admittance Control Scheme. The first, similar to the trajectory generation strategy comparison presented in Section 3.3 in Chapter 3, should compare the performance of healthy participants using the Admittance Control scheme against the performance of healthy participants using the Position Control Scheme. The second test should compare the rehabilitation outcomes of Stroke patients when using the Admittance Control System compared with the rehabilitation outcomes of Stroke patients when using the Position Control System

References

[1] Rodgers, Helen, Bosomworth, Helen, Krebs, Hermano I., Wijck, Frederike van, Howel, Denise, Wilson, Nina, Aird, Lydia, Alvarado, Natasha, Andole, Sreeman, Cohen, David L., Dawson, Jesse, Fernadez-Garcia, Cristina, Finch, Tracy, Ford, Gary A., Francis, Richard, Hogg, Steven, Hughes, Niall, Price, Christopher I., Ternet, Laura, Turner, Duncan L., vale, Luke, Wilkes, Scott, and Shaw, Lisa. "Robot assisted training for the upper limb after stroke (RATULS): a multicentre randomised controlled trial". In: *The Lancet* 394.10192 (2019), pp. 51–62.

[2] Stroke Association. *UK. State of the nation stroke statistics-February 2018.* 2018.

[3] Van Peppen, R. P.S., Kwakkel, Gert, Wood-Dauphinee, Sharon, Hendriks, H. J.M., Van der Wees, Ph. J., and Dekker, Joost. "The impact of physical therapy on functional outcomes after stroke: what's the evidence?" In: *Clinical rehabilitation* 18.8 (2004), pp. 833–862.

[4] Morreale, Manuela, Marchione, Pasquale, Pili, Antonio, Lauta, Antonella, Castiglia, Stefano F., Spallone, Aldo, Pierelli, Francesco, and Giacomini, Patrizia. "Early versus delayed rehabilitation treatment in hemiplegic patients with ischemic stroke: proprioceptive or cognitive approach". In: *Eur J Phys Rehabil Med* 52.1 (2016), pp. 81–89.

[5] Bayley, Mark T, Hurdowar, Amanda, Richards, Carol L, Korner-Bitensky, Nicol, Wood-Dauphinee, Sharon, Eng, Janice J, McKay-Lyons, Marilyn, Harrison, Edward, Teasell, Robert, Harrison, Margaret, et al. "Barriers to implementation of stroke rehabilitation evidence: findings from a multi-site pilot project". In: *Disability and rehabilitation* 34.19 (2012), pp. 1633–1638.

[6] Jurkiewicz, Michael T, Marzolini, Susan, and Oh, Paul. "Adherence to a home-based exercise program for individuals after stroke". In: *Topics in stroke rehabilitation* 18.3 (2011), pp. 277–284.

[7] Aprile, Irene, Pecchioli, Cristiano, Loreti, Simona, Cruciani, Arianna, Padua, Luca, and Germanotta, Marco. "Improving the efficiency of robot-mediated rehabilitation by using a new organizational model: An observational feasibility study in an Italian rehabilitation center". In: *Applied Sciences* 9.24 (2019), p. 5357.

[8] Stroke Association. *How the Covid-19 pandemic has affected stroke survivors' lives and recoveries.* 2020.

[9] Smith, Eric E., Mountain, Anita, Hill, Michael D., Wein, Theodore H., Blacquiere, Dylan, Casaubon, Leanne K., Linkewich, Elizabeth, Foley, Norine, Gubitz, Gord, and Simard Anne Lindsay, M. Patrice. "Canadian stroke best practice guidance during the COVID-19 pandemic". In: *Canadian Journal of Neurological Sciences* 47.4 (2020), pp. 474–478.

[10] Wang, Chien-Chih, Chao, Jian-Kang, Wang, Mong-Lien, Yang, Yi-Ping, Chien, Chien-Shiu, Lai, Wei-Yi, Yang, Yi-Chiang, Chang, Yu-Hui, Chou, Chen-Liang, and Kao, Chung-Lan. "Care for patients with stroke during the COVID-19 pandemic: physical therapy and rehabilitation suggestions for preventing secondary stroke". In: *Journal of Stroke and Cerebrovascular Diseases* (2020), p. 105182.

[11] Sivan, Manoj, Gallagher, Justin, Makower, Sophie, Keeling, David, Bhakta, Bipin, O'Connor, Rory J., and Levesley, Martin. "Home-based Computer Assisted Arm Rehabilitation (hCAAR) robotic device for upper limb exercise after stroke: results of a feasibility study in home setting". In: *Journal of neuroengineering and rehabilitation* 11.1 (2014), p. 163.

[12] Preston, Nick, Weightman, Andrew, Gallagher, Justin, Holt, Raymond, Clarke, Michael, Mon-Williams, Mark, Levesley, Martin, and Bhakta, Bipinchandra. "Feasibility of school-based computer-assisted robotic gaming technology for upper limb rehabilitation of children with cerebral palsy". In: *Disability and Rehabilitation: Assistive Technology* 11.4 (2016), pp. 281–288.

[13] Marques, Antonio Jose Pereira Silva, Caldas, Helena Maria Martins, Barbosa, Mariana Castro, Soares, Luis Miguel Brazao, Ribeiro, Maria Ines Dias, and Simoes-Silva, Vitor. "Gamification in Stroke Rehabilitation". In: *Digital Therapies in Psychosocial Rehabilitation and Mental Health*. IGI Global, 2022, pp. 187–199.

[14] Moskowitz, Michael A., Lo, Eng H., and Iadecola, Costantino. "The science of stroke: mechanisms in search of treatments". In: *Neuron* 67.2 (2010), pp. 181–198.

[15] Sommerfeld, Disa K., Eek, Elsy U-B., Svensson, Anna-Karin, Holmqvist, Lotta Widén, and Arbin, Magnus H. von. "Spasticity after stroke: its occurrence and association with motor impairments and activity limitations". In: *Stroke* 35.1 (2004), pp. 134–139.

[16] Lawrence, Enas S., Coshall, Catherine, Dundas, Ruth, Stewart, Judy, Rudd, Anthony G., Howard, Robin, and Wolfe, Charles D.A. "Estimates of the prevalence of acute stroke impairments and disability in a multiethnic population". In: *Stroke* 32.6 (2001), pp. 1279–1284.

[17] Xu, Xiang-Ming, Vestesson, Emma, Paley, Lizz, Desikan, Anita, Wonderling, David, Hoffman, Alex, Wolfe, Charles D.A., Rudd, Anthony G., and Bray, Benjamin D. "The economic burden of stroke care in England, Wales and Northern Ireland: Using a national stroke register to estimate and report patient-level health economic outcomes in stroke". In: *European stroke journal* 3.1 (2018), pp. 82–91.

[18] Patel, Anita, Berdunov, Vladislav, King, Derek, Quayyum, Zahidul, Wittenberg, Raphael, and Knapp, Martin. "Current, future and avoidable costs of stroke in the UK". In: *Stroke Association* (2017).

[19] O'Mahony, Paul G., Thomson, Richard G., Dobson, Ruth, Rodgers, Helen, and James, Oliver F.W. "The prevalence of stroke and associated disability". In: *Journal of public health* 21.2 (1999), pp. 166–171.

[20] Office for National Statistics. *Overview of the UK population: November 2018*. Cited 13 November 2018. 2018. URL: https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates/articles/overviewoftheukpopulation/november2018.

[21] Wafa, Hatem A., Wolfe, Charles D.A., Emmett, Eva, Roth, Gregory A., Johnson, Catherine O., and Wang, Yanzhong. "Burden of stroke in Europe: thirty-year projections of incidence, prevalence, deaths, and disability-adjusted life years". In: *Stroke* 51.8 (2020), pp. 2418–2427.

[22] Lee, Ronald Demos and Mason, Andrew. *Population aging and the generational economy: A global perspective*. Edward Elgar Publishing, 2011.

[23] Lynch, Elizabeth, Hillier, Susan, and Cadilhac, Dominique. "When should physical rehabilitation commence after stroke: a systematic review". In: *International Journal of Stroke* 9.4 (2014), pp. 468–478.

[24] Duffau, Hugues. "Brain plasticity: from pathophysiological mechanisms to therapeutic applications". In: *Journal of clinical neuroscience* 13.9 (2006), pp. 885–897.

[25] Kreisel, Stefan H., Hennerici, Michael G., and Bäzner, Hansjörg. "Pathophysiology of stroke rehabilitation: the natural course of clinical recovery, use-dependent plasticity and rehabilitative outcome". In: *Cerebrovascular diseases* 23.4 (2007), pp. 243–255.

[26] Coleman, Elisheva R., Moudgal, Rohitha, Lang, Kathryn, Hyacinth, Hyacinth I., Awosika, Oluwole O., Kissela, Brett M., and Feng, Wuwei. "Early rehabilitation after stroke: a narrative review". In: *Current atherosclerosis reports* 19.12 (2017), p. 59.

[27] Marek, Sarah M., Cramer, Joel T., Fincher, A. Louise, Massey, Laurie L., Dangelmaier, Suzanne M., Purkayastha, Sushmita, Fitz, Kristi A., and Culbertson, Julie Y. "Acute effects of static and proprioceptive neuromuscular facilitation stretching on muscle strength and power output". In: *Journal of athletic training* 40.2 (2005), p. 94.

[28] Lee, Sunghee, Bae, Seahyun, Jeon, Daejung, and Kim, Kyung Yoon. "The effects of cognitive exercise therapy on chronic stroke patients' upper limb functions, activities of daily living and quality of life". In: *Journal of physical therapy science* 27.9 (2015), pp. 2787–2791.

[29] Hatem, Samar M, Saussez, Geoffroy, Della Faille, Margaux, Prist, Vincent, Zhang, Xue, Dispa, Delphine, and Bleyenheuft, Yannick. "Rehabilitation of motor function after stroke: a multiple systematic review focused on techniques to stimulate upper extremity recovery". In: *Frontiers in human neuroscience* 10 (2016), p. 442.

[30] Graham, Julie Vaughan, Eustace, Catherine, Brock, Kim, Swain, Elizabeth, and Irwin-Carruthers, Sheena. "The Bobath concept in contemporary clinical practice". In: *Topics in stroke rehabilitation* 16.1 (2009), pp. 57–68.

[31] Wang, Ray-Yau, Chen, Hsiu-I, Chen, Chen-Yin, and Yang, Yea-Ru. "Efficacy of Bobath versus orthopaedic approach on impairment and function at different motor recovery stages after stroke: a randomized controlled study". In: *Clinical rehabilitation* 19.2 (2005), pp. 155–164.

[32] Paci, Matteo. "Physiotherapy based on the Bobath concept for adults with post-stroke hemiplegia: a review of effectiveness studies". In: *Journal of rehabilitation medicine* 35.1 (2003), pp. 2–7.

[33] Pathak, Abhishek, Gyanpuri, Vyom, Dev, Priya, and Dhiman, Neetu Rani. "The Bobath Concept (NDT) as rehabilitation in stroke patients: A systematic review". In: *Journal of Family Medicine and Primary Care* 10.11 (2021), p. 3983.

[34] Diaz-Arribas, Maria J, Martin-Casas, Patricia, Cano-de-la-Cuerda, Roberto, and Plaza-Manzano, Gustavo. "Effectiveness of the Bobath concept in the treatment of stroke: a systematic review". In: *Disability and Rehabilitation* 42.12 (2020), pp. 1636–1649.

[35] Patten, Carolynn, Lexell, Jan, and Brown, Heather E. "Weakness and strength training in persons with poststroke hemiplegia: rationale, method, and efficacy." In: *Journal of Rehabilitation Research & Development* 41 (2004).

[36] Ridderikhoff, Arne, Peper, C(Lieke) E, and Beek, Peter J. "Unraveling interlimb interactions underlying bimanual coordination". In: *Journal of neurophysiology* 94.5 (2005), pp. 3112–3125.

[37] Mudie, M Heather and Matyas, Thomas A. "Upper extremity retraining following stroke: effects of bilateral practice". In: *Journal of neurologic rehabilitation* 10.3 (1996), pp. 167–184.

[38] Lewis, Gwyn N and Perreault, Eric J. "An assessment of robot-assisted bimanual movements on upper limb motor coordination following stroke". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 17.6 (2009), pp. 595–604.

[39] Burgar, Charles G., Lum, Peter S., Shor, Peggy C., and Van der Loos, Machiel H.F. "Development of robots for rehabilitation therapy: The Palo Alto VA/Stanford experience". In: *Journal of rehabilitation research and development* 37.6 (2000), pp. 663–674.

[40] Carr, Janet H and Shepherd, Roberta B. "A motor learning model for stroke rehabilitation". In: *Physiotherapy* 75.7 (1989), pp. 372–380.

[41] Linder, Susan M, Rosenfeldt, Anson B, Reiss, Aimee, Buchanan, Sharon, Sahu, Komal, Bay, Curtis R, Wolf, Steven L, and Alberts, Jay L. "The home stroke rehabilitation and monitoring system trial: a randomized controlled trial". In: *International journal of stroke* 8.1 (2013), pp. 46–53.

[42] Lohse, Keith R., Boyd, Lara A., and Hodges, Nicola J. "Engaging environments enhance motor skill learning in a computer gaming task". In: *Journal of motor behavior* 48.2 (2016), pp. 172–182.

[43] Novak, Domen. "Promoting motivation during robot-assisted rehabilitation". In: *Rehabilitation Robotics*. Elsevier, 2018, pp. 149–158.

[44] Maciejasz, Paweł, Eschweiler, Jörg, Gerlach-Hahn, Kurt, Jansen-Troy, Arne, and Leonhardt, Steffen. "A survey on robotic devices for upper limb rehabilitation". In: *Journal of neuroengineering and rehabilitation* 11.1 (2014), p. 3.

[45] Culmer, Peter R. "Development of a Cooperative Robot System to Aid Stroke Rehabilitation". PhD thesis. University of Leeds, 2007.

[46] Passon, Arne, Schauer, Thomas, and Seel, Thomas. "Inertial-robotic motion tracking in end-effector-based rehabilitation robots". In: *Frontiers in Robotics and AI* (2020), p. 167.

[47] Sulzer, James S., Peshkin, Michael A., and Patton, James L. "Design of a mobile, inexpensive device for upper extremity rehabilitation at home". In: *2007 IEEE 10th International Conference on Rehabilitation Robotics*. IEEE. 2007, pp. 933–937.

[48] Johnson, Michelle J., Feng, Xin, Johnson, Laura M., and Winters, Jack M. "Potential of a suite of robot/computer-assisted motivating systems for personalized, home-based, stroke rehabilitation". In: *Journal of NeuroEngineering and Rehabilitation* 4.1 (2007), p. 6.

[49] Marchal-Crespo, Laura and Reinkensmeyer, David J. "Review of control strategies for robotic movement training after neurologic injury". In: *Journal of neuroengineering and rehabilitation* 6.1 (2009), p. 20.

[50] Erol, Duygun and Sarkar, Nilanjan. "Design and Implementation of a Control Architecture for Rehabilitation Robotic Systems". In: *Rehabilitation Robotics* (2007), p. 91.

[51] Krebs, Hermano I., Ferraro, Mark, Buerger, Stephen P., Newbery, Miranda J., Makiyama, Antonio, Sandmann, Michael, Lynch, Daniel, Volpe, Bruce T., and Hogan, Neville. "Rehabilitation robotics: pilot trial of a spatial extension for MIT-Manus". In: *Journal of neuroengineering and rehabilitation* 1.1 (2004), p. 5.

[52] Sanchez, R.J., Wolbrecht, Eric, Smith, R., Liu, J., Rao, S., Cramer, S., Rahman, T., Bobrow, James E., and Reinkensmeyer, David J. "A pneumatic robot for re-training arm movement after stroke: Rationale and mechanical design". In: *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* IEEE. 2005, pp. 500–504.

[53] Patton, James L., Stoykov, Mary Ellen, Kovic, Mark, and Mussa-Ivaldi, Ferdinando A. "Evaluation of robotic training forces that either enhance or reduce error in chronic hemiparetic stroke survivors". In: *Experimental brain research* 168.3 (2006), pp. 368–383.

[54] Montagner, Alberto, Frisoli, Antonio, Borelli, Luigi, Procopio, Caterina, Bergamasco, Massimo, Carboncini, Maria C., and Rossi, Bruno. "A pilot clinical study on robotic assisted rehabilitation in VR with an arm exoskeleton device". In: *2007 Virtual Rehabilitation*. IEEE. 2007, pp. 57–64.

[55] Wisneski, Kimberly J. and Johnson, Michelle J. "Quantifying kinematics of purposeful movements to real, imagined, or absent functional objects: implications for modelling trajectories for robot-assisted ADL tasks". In: *Journal of NeuroEngineering and Rehabilitation* 4.1 (2007), pp. 1–14.

[56] Hogan, Neville, Krebs, Hermano Igo, Charnnarong, J., Srikrishna, P., and Sharon, Andre. "MIT-MANUS: a workstation for manual therapy and training. I". In: *[1992] Proceedings IEEE International Workshop on Robot and Human Communication.* IEEE. 1992, pp. 161–165.

[57] Wolbrecht, Eric T., Chan, Vicky, Le, Vu, Cramer, Steven C., Reinkensmeyer, David J., and Bobrow, James E. "Real-time computer modeling of weakness following stroke optimizes robotic assistance for movement therapy". In: *2007 3rd International IEEE/EMBS Conference on Neural Engineering.* IEEE. 2007, pp. 152–158.

[58] Erol, Duygun and Sarkar, Nilanjan. "Intelligent control for robotic rehabilitation after stroke". In: *Journal of Intelligent and Robotic Systems* 50.4 (2007), pp. 341–360.

[59] Buma, Floor E., Kordelaar, Joost van, Raemaekers, Matthijs, Wegen, Erwin E.H. van, Ramsey, Nick F., and Kwakkel, Gert. "Brain activation is related to smoothness of upper limb movements after stroke". In: *Experimental brain research* 234.7 (2016), pp. 2077–2089.

[60] Nef, Tobias, Mihelj, Matjaz, and Riener, Robert. "ARMin: a robot for patient-cooperative arm therapy". In: *Medical & biological engineering & computing* 45.9 (2007), pp. 887–900.

[61] Lum, Peter S., Burgar, Charles G., Shor, Peggy C., Majmundar, Matra, and Van der Loos, Machiel. "Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke". In: *Archives of physical medicine and rehabilitation* 83.7 (2002), pp. 952–959.

[62] Buerger, S.P. and Hogan, N. "Impedance and interaction control". In: *Robotics and Automation Handbook* (2005), pp. 19–1.

[63] Hogan, Neville. "Impedance control: An approach to manipulation". In: *1984 American control conference.* IEEE. 1984, pp. 304–313.

[64] Richardson, Robert. "Actuation and control for robotic physiotherapy". PhD thesis. University of Leeds, 2001.

[65] Ott, Christian, Mukherjee, Ranjan, and Nakamura, Yoshihiko. "Unified impedance and admittance control". In: *2010 IEEE International Conference on Robotics and Automation.* IEEE. 2010, pp. 554–561.

[66] Culmer, Peter R., Jackson, Andrew E., Makower, Sophie, Richardson, Robert, Cozens, J. Alastair, Levesley, Martin C., and Bhakta, Bipin B. "A control strategy for upper limb robotic rehabilitation with a dual robot system". In: *IEEE/ASME Transactions on Mechatronics* 15.4 (2009), pp. 575–585.

[67] Anderson, Robert J. and Spong, Mark W. "Hybrid impedance control of robotic manipulators". In: *IEEE Journal on Robotics and Automation* 4.5 (1988), pp. 549–556.

[68] Richardson, R., Brown, M., Bhakta, B., and Levesley, M.C. "Design and control of a three degree of freedom pneumatic physiotherapy robot". In: *Robotica* 21.6 (2003), pp. 589–604.

[69] Stroke Association. *Using Robotics to help arm, wrist & hand recovery after Stroke.* Cited 26 November 2018. 2018. URL: https://www.stroke.org.uk/news/using-robotics-help-arm-wrist-hand-recovery-after-stroke.

[70] Krebs, H., Hogan, Neville, Aisen, Mindy L., and Volpe, Bruce T. "Robot-aided neurorehabilitation". In: *IEEE transactions on rehabilitation engineering* 6.1 (1998), pp. 75–87.

[71] Volpe, B.T., Krebs, H.I., Hogan, N., Edelsteinn, L., Diels, C.M., and Aisen, M.L. "Robot training enhanced motor outcome in patients with stroke maintained over 3 years". In: *Neurology* 53.8 (1999), pp. 1874–1874.

[72] Micera, Silvestro, Carrozza, Maria Chiara, Guglielmelli, Eugenio, Cappiello, Giovanni, Zaccone, Franco, Freschi, Cinzia, Colombo, Roberto, Mazzone, Alessandra, Delconte, Carmen, Pisano, Fabrizio, Minuco, G., and Dario, Paolo. "A simple robotic system for neurorehabilitation". In: *Autonomous Robots* 19.3 (2005), p. 271.

[73] Lum, Peter S., Burgar, Charles G., Van der Loos, Machiel, Shor, Peggy C., Majmundar, Matra, and Yap, Ruth. "MIME robotic device for upper-limb neurorehabilitation in subacute stroke subjects: A follow-up study". In: *Journal of rehabilitation research and development* 43.5 (2006), p. 631.

[74] Lum, Peter S., Burgar, Charles G., Van der Loos, Machiel, Shor, Peggy C., Majmundar, Matra, and Yap, Ruth. "The MIME robotic system for upper-limb neuro-rehabilitation: results from a clinical trial in subacute stroke". In: *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* IEEE. 2005, pp. 511–514.

[75] Oña, E.D., Cano-de La Cuerda, R., Sánchez-Herrera, P., Balaguer, C., and Jardón, A. "A review of robotics in neurorehabilitation: towards an automated process for upper limb". In: *Journal of healthcare engineering* 2018 (2018).

[76] Reinkensmeyer, David J., Takahashi, Craig D., Timoszyk, Wojciech K., Reinkensmeyer, Andrea N., and Kahn, Leonard E. "Design of robot assistance for arm movement therapy following stroke". In: *Advanced robotics* 14.7 (2001), pp. 625–637.

[77] Kahn, Leonard E., Zygman, Michele L., Rymer, W. Zev, and Reinkensmeyer, David J. "Robot-assisted reaching exercise promotes arm movement recovery in chronic hemiparetic stroke: a randomized controlled pilot study". In: *Journal of neuroengineering and rehabilitation* 3.1 (2006), p. 12.

[78] Liu, Yali, Li, Chong, Ji, Linhong, Bi, Sheng, Zhang, Xuemin, Huo, Jianfei, and Ji, Run. "Development and implementation of an end-effector upper limb rehabilitation robot for hemiplegic patients with line and circle tracking training". In: *Journal of healthcare engineering* 2017 (2017).

[79] Culmer, Peter, Jackson, Andrew, Richardson, Rob, Bhakta, Bipin, Levesley, Martin, and Cozens, Alastair. "Development of a dual robotic system for upper-limb stroke rehabilitation". In: *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* IEEE. 2005, pp. 61–65.

[80] Jackson, Andrew E., Holt, Raymond John, Culmer, Peter R., Makower, Sophie G., Levesley, Martin C., Richardson, R.C., Cozens, J. Alistair, Mon-Williams, M., and Bhakta, Bipin B. "Dual robot system for upper limb rehabilitation after stroke: the design process". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 221.7 (2007), pp. 845–857.

[81] Richardson, R., Jackson, A., Culmer, Peter, Bhakta, B., and Levesley, M.C. "Pneumatic impedance control of a 3-dof physiotherapy robot". In: *Advanced Robotics* 20.12 (2006), pp. 1321–1339.

[82] Jackson, A., Culmer, P., Makower, S., Levesley, M., Richardson, R., Cozens, A., Mon-Williams, M., and Bhakta, B. "Initial patient testing of iPAM-a robotic system for stroke rehabilitation". In: *2007 IEEE 10th International Conference on Rehabilitation Robotics.* IEEE. 2007, pp. 250–256.

[83] Firouzy, Sina. “Control Algorithms to Improve the Dynamic Performance of Robotic Rehabilitation Devices”. MA thesis. University of Leeds (School of Mechanical Engineering), 2011.

[84] Sugar, Thomas G., He, Jiping, Koeneman, Edward J., Koeneman, James B., Herman, Richard, Huang, H., Schultz, Robert S., Herring, D.E., Wanberg, J., Balasubramanian, Sivakumar, Swenson, Pete, and Ward, Jeffrey A. “Design and control of RUPERT: a device for robotic upper extremity repetitive therapy”. In: *IEEE transactions on neural systems and rehabilitation engineering* 15.3 (2007), pp. 336–346.

[85] Malosio, Matteo, Caimmi, Marco, Legnani, Giovanni, and Tosatti, Lorenzo Molinari. “LINarm: a low-cost variable stiffness device for upper-limb rehabilitation”. In: *2014 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2014, pp. 3598–3603.

[86] Malosio, Matteo, Caimmi, Marco, Cotti Cottini, Michele, Crema, Andrea, Dinon, Tito, Mihelj, Matjaz, Molinari Tosatti, Lorenzo, Podobnik, Janez, Prini, Alessio, Seneci, Carlo, et al. “An affordable, adaptable, and hybrid assistive device for upper-limb neurorehabilitation”. In: *Journal of rehabilitation and assistive technologies engineering* 3 (2016), p. 2055668316680980.

[87] Dinon, Tito, Caimmi, Marco, Chiavenna, Andrea, Malosio, Matteo, Prini, Alessio, Scano, Alessandro, Molinari Tosatti, Lorenzo, Currò, Cristian, Lenzi, Bruno, and Megale, Valentino. “DUALarm: An open-source and 3D-printable device for upper limb neurorehabilitation”. In: *Journal of Rehabilitation and Assistive Technologies Engineering* 5 (2018), p. 2055668317749989.

[88] Pezent, Evan, Rose, Chad G, Deshpande, Ashish D, and O’Malley, Marcia K. “Design and characterization of the OpenWrist: A robotic wrist exoskeleton for coordinated hand-wrist rehabilitation”. In: *2017 international conference on rehabilitation robotics (ICORR)*. IEEE. 2017, pp. 720–725.

[89] Pehlivan, Ali Utku, Sergi, Fabrizio, Erwin, Andrew, Yozbatiran, Nuray, Francisco, Gerard E, and O’Malley, Marcia K. “Design and validation of the RiceWrist-S exoskeleton for robotic rehabilitation after incomplete spinal cord injury”. In: *Robotica* 32.8 (2014), pp. 1415–1431.

[90] Metzger, Jean-Claude, Lambercy, Olivier, Califfi, Antonella, Conti, Fabio M, and Gassert, Roger. “Neurocognitive robot-assisted therapy of hand function”. In: *IEEE transactions on haptics* 7.2 (2013), pp. 140–149.

[91] Spencer, SJ, Klein, J, Minakata, K, Le, V, Bobrow, JE, and Reinkensmeyer, DJ. “A low cost parallel robot and trajectory optimization method for wrist and forearm rehabilitation using the Wii”. In: *2008 2nd IEEE RAS & EMBS international conference on biomedical robotics and biomechatronics*. IEEE. 2008, pp. 869–874.

[92] Monardo, Giulia, Pavese, Chiara, Giorgi, Ines, Godi, Marco, and Colombo, Roberto. “Evaluation of patient motivation and satisfaction during technology-assisted rehabilitation: an experiential review”. In: *Games for health journal* 10.1 (2021), pp. 13–27.

[93] Nakamura, Jeanne, Csikszentmihalyi, Mihaly, et al. “The concept of flow”. In: *Handbook of positive psychology* 89 (2002), p. 105.

[94] Ottiger, Beatrice, Van Wegen, Erwin, Keller, Katja, Nef, Tobias, Nyffeler, Thomas, Kwakkel, Gert, and Vanbellingen, Tim. “Getting into a “Flow” state: a systematic review of flow experience in neurological diseases”. In: *Journal of neuroengineering and rehabilitation* 18 (2021), pp. 1–21.

[95] Frisoli, Antonio, Borelli, Luigi, Montagner, Alberto, Marcheschi, Simone, Procopio, Caterina, Salsedo, Fabio, Bergamasco, Massimo, Carboncini, Maria C, Tolaini, Martina, and Rossi, Bruno. “Arm rehabilitation with a robotic exoskeleleton in Virtual Reality”. In: *2007 IEEE 10th International Conference on Rehabilitation Robotics*. IEEE. 2007, pp. 631–642.

[96] Riener, Robert, Wellner, Mathias, Nef, Tobias, Von Zitzewitz, Joachim, Duschau-Wicke, Alexander, Colombo, Gery, and Lunenburger, L. “A view on VR-enhanced rehabilitation robotics”. In: *2006 International Workshop on Virtual Rehabilitation*. IEEE. 2006, pp. 149–154.

[97] Tuah, Nooralisa Mohd, Ahmedy, Fatimah, Gani, Abdullah, and Yong, Lionelson Norbert. “A survey on gamification for health rehabilitation care: Applications, opportunities, and open challenges”. In: *Information* 12.2 (2021), p. 91.

[98] Langhorne, Peter, Bernhardt, Julie, and Kwakkel, Gert. “Stroke rehabilitation”. In: *The Lancet* 377.9778 (2011), pp. 1693–1702.

[99] Robertson, JVG, Jarrassé, N, and Roby-Brami, A. “Rehabilitation robots: a compliment to virtual reality”. In: *Schedae* 1.6 (2010), pp. 77–94.

[100] Rohrer, Brandon, Fasoli, Susan, Krebs, Hermano Igo, Hughes, Richard, Volpe, Bruce, Frontera, Walter R, Stein, Joel, and Hogan, Neville. “Movement smoothness changes during stroke recovery”. In: *Journal of neuroscience* 22.18 (2002), pp. 8297–8304.

[101] Weber, Lynne M and Stein, Joel. “The use of robots in stroke rehabilitation: A narrative review”. In: *NeuroRehabilitation* 43.1 (2018), pp. 99–110.

[102] Li, Lutong, Tyson, Sarah, and Weightman, Andrew. “Professionals’ Views and Experiences of Using Rehabilitation Robotics With Stroke Survivors: A Mixed Methods Survey”. In: *Frontiers in Medical Technology* 3 (2021), p. 780090.

[103] Bishop, Lauri, Khan, Moiz, Martelli, Dario, Quinn, Lori, Stein, Joel, and Agrawal, Sunil. “Exploration of two training paradigms using forced induced weight shifting with the tethered pelvic assist device to reduce asymmetry in individuals after stroke”. In: *American journal of physical medicine & rehabilitation* 96.10 (2017), S135–S140.

[104] Bersano, Anna, Kraemer, Markus, Touzé, Emmanuel, Weber, Ralph, Alamowitch, Sonia, Sibon, Igor, and Pantoni, Leonardo. “Stroke care during the COVID-19 pandemic: experience from three large European countries”. In: *European journal of neurology* 27.9 (2020), pp. 1794–1800.

[105] Kristoffersen, Espen Saxhaug, Jahr, Silje Holt, Faiz, Kashif Waqar, Storstein, Anette Margrethe, Winsvold, Bendik Slagsvold, and Sandset, Else Charlotte. “Acute stroke care during the first phase of COVID-19 pandemic in Norway”. In: *Acta Neurologica Scandinavica* 143.4 (2021), pp. 349–354.

[106] Douiri, Abdel, Muruet, Walter, Bhalla, Ajay, James, Martin, Paley, Lizz, Stanley, Kaili, Rudd, Anthony G., Wolfe, Charles D.A., Bray, Benjamin D., and Collaboration, SSNAP. “Stroke care in the United Kingdom during the COVID-19 pandemic”. In: *Stroke* 52.6 (2021), pp. 2125–2133.

[107] Galan, J.M. Trejo Gabriel y. “Stroke as a complication and prognostic factor of COVID-19”. In: *Neurologia (English Edition)* 35.5 (2020), pp. 318–322.

[108] Richardson, Magnus J.E. and Flash, Tamar. “Comparing smooth arm movements with the two-thirds power law and the related segmented-control hypothesis”. In: *Journal of neuroscience* 22.18 (2002), pp. 8201–8211.

[109] Gallagher, Justin Francis. “Autonomous Robot-assisted Upper Limb Physiotherapy”. PhD thesis. University of Leeds (School of Mechanical Engineering), 2014.

[110] Patton, James L and Mussa-Ivaldi, Ferdinando A. “Robot-assisted adaptive training: custom force fields for teaching movement patterns”. In: *IEEE Transactions on Biomedical Engineering* 51.4 (2004), pp. 636–646.

[111] Khalil, Wisama and Dombre, Etienne. *Modeling identification and control of robots.* CRC Press, 2002.

[112] Heinzmann, Jochen and Zelinsky, Alexander. “A safe-control paradigm for human–robot interaction”. In: *Journal of Intelligent and robotic systems* 25.4 (1999), pp. 295–310.

[113] Bittencourt, André Carvalho and Gunnarsson, Svante. “Static friction in a robot joint—modeling and identification of load and temperature effects”. In: *Journal of Dynamic Systems, Measurement, and Control* 134.5 (2012).

[114] Thurston, Robert Henry. *A Handbook of Engine and Boiler Trials: And of the Indicator and Prony Brake.* Spon, 1890.

[115] Lynch, Kevin M. and Park, Frank C. *Modern Robotics.* Cambridge University Press, 2017.

[116] Erez, Tom, Tassa, Yuval, and Todorov, Emanuel. “Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx”. In: *2015 IEEE international conference on robotics and automation (ICRA).* IEEE. 2015, pp. 4397–4404.

[117] Das, Shuvra. “Modeling and Simulation of Mechatronic Systems Using Simscape”. In: *Synthesis Lectures on Mechanical Engineering* 5.1 (2020), pp. 1–171.

[118] Jackson, A., Levesley, M., Culmer, R., et al. “Development of a mechanical arm model of the human arm for use with an exercise robotic system being developed for people with stroke”. In: *Proceedings of 2nd Cambridge workshop on universal access and assistive technoligies, Cambridge.* 2006, pp. 91–98.

[119] Wang, Hongbo, De Boer, Gregory, Kow, Junwai, Alazmani, Ali, Ghajari, Mazdak, Hewson, Robert, and Culmer, Peter. “Design methodology for magnetic field-based soft tri-axis tactile sensors”. In: *Sensors* 16.9 (2016), p. 1356.

[120] Majidi, C., Kramer, R., and Wood, R.J. “A non-differential elastomer curvature sensor for softer-than-skin electronics”. In: *Smart materials and structures* 20.10 (2011), p. 105017.

[121] Park, Sun-Jun, Kim, Joshua, Chu, Michael, and Khine, Michelle. “Highly flexible wrinkled carbon nanotube thin film strain sensor to monitor human movement”. In: *Advanced Materials Technologies* 1.5 (2016), p. 1600053.

[122] Sparks, Jessica L., Vavalle, Nicholas A., Kasting, Krysten E., Long, Benjamin, Tanaka, Martin L., Sanger, Phillip A., Schnell, Karen, and Conner-Kerr, Teresa A. “Use of silicone materials to simulate tissue biomechanics as related to deep tissue injury”. In: *Advances in skin & wound care* 28.2 (2015), pp. 59–68.

[123] Shi, Xiaoting, Xu, Renxiao, Li, Yuhang, Zhang, Yihui, Ren, Zhigang, Gu, Jianfeng, Rogers, John A, and Huang, Yonggang. “Mechanics design for stretchable, high areal coverage GaAs solar module on an ultrathin substrate”. In: *Journal of Applied Mechanics* 81.12 (2014), p. 124502.

[124] De Boer, Gregory, Raske, Nicholas, Wang, Hongbo, Kow, Junwai, Alazmani, Ali, Ghajari, Mazdak, Culmer, Peter, and Hewson, Robert. “Design optimisation of a magnetic field based soft tactile sensor”. In: *Sensors* 17.11 (2017), p. 2539.

[125] Liu, Z.P. and Castagna, J.P. “Avoiding overfitting caused by noise using a uniform training mode”. In: *IJCNN’99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339).* Vol. 3. IEEE. 1999, pp. 1788–1793.

[126] d’Ascoli, Stéphane, Sagun, Levent, and Biroli, Giulio. “Triple descent and the two kinds of overfitting: Where & why do they appear?” In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 3058–3069.

[127] Bejani, Mohammad Mahdi and Ghatee, Mehdi. “A systematic review on overfitting control in shallow and deep neural networks”. In: *Artificial Intelligence Review* 54.8 (2021), pp. 6391–6438.

[128] Martín-Félez, Raúl and Xiang, Tao. “Uncooperative gait recognition by learning to rank”. In: *Pattern Recognition* 47.12 (2014), pp. 3793–3806.

[129] Doyle, John C., Francis, Bruce A., and Tannenbaum, Allen R. *Feedback control theory.* Courier Corporation, 2013.

[130] Wang, Qing-Guo and Nie, Zhuo-Yun. “PID control for MIMO processes”. In: *PID control in the third millennium.* Springer, 2012, pp. 177–204.

[131] Pooley, Rob and King, Peter. “The unified modelling language and performance engineering”. In: *IEE Proceedings-Software* 146.1 (1999), pp. 2–10.

Appendix A

Minimum Jerk Trajectory Generation

Generating a Smooth Trajectory

Mathematically, generating a smooth trajectory translates to minimising the rate of change of an input where the input corresponds to the order of the system. For example, a 1^{st} order system denotes a kinematic model where velocities may be arbitrarily specified. This is summarised in the Table A.1.

Table A.1: *How the order of a system relates to the input.*

Order of the System	**Input to the System**
1^{st}	Velocity, $\dot{x}$
2^{nd}	Acceleration, $\ddot{x}$
3^{rd}	Jerk, $\dddot{x}$
4^{th}	Snap, $x^{(4)}$
5^{th}	Crackle, $x^{(5)}$
6^{th}	Pop, $x^{(6)}$

The function to determine the trajectory may be found using Calculus of Variations, using the general equation shown by Equation A.1.

$$x^{*}t = argmin_{x(t)} \int_{0}^{T} Ldt \tag{A.1}$$

Where:

$$L = \left(x^{(n)}\right)^{2}$$

Alternatively, the function to determine the trajectory may be found by solving the Euler-Lagrange equation shown by Equation A.2.

$$\frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial L}{\partial \ddot{x}}\right) + ... + (-1)^n \frac{d^n}{dc^n}\left(\frac{\partial L}{\partial x^{(n)}}\right) = 0 \tag{A.2}$$

For example, the shortest distance between 2 points may be found by solving either Equation A.3 or Equation A.4.

$$x^*t = argmin_{x(t)} \int_0^T \left(\dot{(}x)\right) dt \tag{A.3}$$

$$\frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = 0 \tag{A.4}$$

Forming a Minimum Jerk Trajectory by Solving the Euler-Lagrange Equation

A minimum Jerk Trajectory is based on minimising the sum of squared Jerk accross the trajectory, thus:

$$L = (\dddot{x})^2$$

Forming the Euler-Lagrange equation as presented by Equation A.2.

$$\frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial L}{\partial \ddot{x}}\right) - \frac{d^3}{dt^3}\left(\frac{\partial L}{\partial \dddot{x}}\right) = 0 \tag{A.5}$$

$$\frac{\partial L}{\partial x} = 0 \qquad \frac{\partial L}{\partial \dot{x}} = 0 \qquad \frac{\partial L}{\partial \ddot{x}} = 0 \qquad \frac{\partial L}{\partial \dddot{x}} = 2\dddot{x} \tag{A.6}$$

$$\rightarrow \frac{d^3}{dt^3}\left(\frac{\partial L}{\partial \dddot{x}}\right) = \frac{d^3}{dt^3}2\dddot{x} = -2x^{(6)} = 0 \tag{A.7}$$

$$\rightarrow x^{(6)}(t) = 0 \tag{A.8}$$

$$\rightarrow x^{(5)}(t) = \int 0dt = c_5 \tag{A.9}$$

$$\rightarrow x^{(4)}(t) = \int x^{(5)}(t)dt = c_5t + c_4 \tag{A.10}$$

$$\rightarrow \dddot{x}(t) = \int x^{(4)}(t)dt = c_5t^2 + c_4t + c_3 \tag{A.11}$$

$$\rightarrow \ddot{x}(t) = \int \dddot{x}(t)dt = c_5t^3 + c_4t^2 + c_3t + c_2 \tag{A.12}$$

$$\rightarrow \dot{x}(t) = \int \ddot{x}(t)dt = c_5t^4 + c_4t^3 + c_3t^2 + c_2t + c_1 \tag{A.13}$$

$$\rightarrow x(t) = \int \dot{x}(t)dt = c_5t^5 + c_4t^4 + c_3t^3 + c_2t^2 + c_1t + c_0 \tag{A.14}$$

The boundary conditions are shown by Table A.2.

Table A.2: *Boundary conditions.*

	Position	Velocity	Acceleration
t=0	a	0	0
T=t_f	b	0	0

By differentiating Equation A.14 we may find the functions for Velocity and Acceleration.

$$\rightarrow \dot{x}(t) = 5c_5t^4 + 4c_4t^3 + 3c_3t^2 + 2c_2t + c_1 \tag{A.15}$$

$$\rightarrow \ddot{x}(t) = 20c_5t^3 + 12c_4t^2 + 6c_3t + 2c_2 \tag{A.16}$$

Substituting boundary conditions into Equation A.14.

$$x(0) = c_0 = a \tag{A.17}$$

$$x(t_f) = c_5{t_f}^5 + c_4{t_f}^4 + c_3{t_f}^3 + c_2{t_f}^2 + c_1t_f + c_0 = b \tag{A.18}$$

Substituting boundary conditions into Equation A.15.

$$\dot{x}(0) = c_1 = 0 \tag{A.19}$$

$$\dot{x}(t_f) = 5c_5{t_f}^4 + 4c_4{t_f}^3 + 3c_3{t_f}^2 + 2c_2t_f + c_1 = 0 \tag{A.20}$$

Substituting boundary conditions into Equation A.16.

$$\ddot{x}(0) = c_2 = 0 \tag{A.21}$$

$$\ddot{x}(t_f) = 20c_5{t_f}^3 + 12c_4{t_f}^2 + 6c_3t_f + 2c_2 = 0 \tag{A.22}$$

Solving for coefficients:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ {t_f}^5 & {t_f}^4 & {t_f}^3 & {t_f}^2 & t_f & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5{t_f}^4 & 4{t_f}^3 & 3{t_f}^2 & 2t_f & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 20{t_f}^3 & 12{t_f}^2 & 6t_f & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.23}$$

$$\begin{bmatrix} a \\ b \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 6\left(\frac{b-a}{{t_f}^5}\right) \\ -15\left(\frac{b-a}{{t_f}^4}\right) \\ 10\left(\frac{b-a}{{t_f}^3}\right) \\ 0 \\ 0 \\ a \end{bmatrix} \tag{A.24}$$

Giving:

$$x(t) = 6\left(\frac{b-a}{{t_f}^5}\right)t^5 - 15\left(\frac{b-a}{{t_f}^4}\right)t^4 + 10\left(\frac{b-a}{{t_f}^3}\right)t^3 + a \tag{A.25}$$

$$\rightarrow x(t) = x_i + (x_f - x_i)\left(6\left(\frac{t}{t_f}\right)^5 - 15\left(\frac{t}{t_f}\right)^4 + 10\left(\frac{t}{t_f}\right)^3\right) \tag{A.26}$$

Where:

$$\begin{aligned} x_i &= Initial\ Position \\ x_f &= Target\ Position \\ t_f &= Target\ Time \\ t &= Current\ Time \end{aligned}$$

Forming a Minimum Jerk Trajectory by Calculus of Variations

A minimum Jerk Trajectory is based on minimising the sum of squared Jerk accross the trajectory, thus:

$$L = (\dddot{x})^2$$

Substitution into Equation A.1.

$$x^*t = F(x(t)) = argmin_{x(t)} \int_0^T L dt \tag{A.27}$$

$$\rightarrow F(x(t)) = argmin_{x(t)} \int_0^T (\dddot{x})^2 \, dt \tag{A.28}$$

Multiplying through by 1/2 to simplify the function later.

$$F(x(t)) = \frac{1}{2} \int_0^T (\dddot{x})^2 \, dt \tag{A.29}$$

Introducing a small variation $\mu(t)$ with the following properties:

$$\begin{array}{ll} \mu(0) = 0 & \mu(T) = 0 \\ \dot{\mu}(0) = 0 & \dot{\mu}(T) = 0 \\ \ddot{\mu}(0) = 0 & \ddot{\mu}(T) = 0 \end{array}$$

Adding $\mu(t)$ as a variation.

$$x(t) = x(t) + e\mu(t) \tag{A.30}$$

$$F(x + e\mu) = \frac{1}{2} \int_0^T (\dddot{x} + e\dddot{\mu})^2 \, dt \tag{A.31}$$

Differentiating w.r.t. μ.

$$\frac{dF\,(x + e\mu)}{e} = \int_0^T (\dddot{x} + e\dddot{\mu})\, \dddot{\mu} dt \tag{A.32}$$

$$\left. \frac{dF\,(x + e\mu)}{e} \right|_{e=0} = \int_0^T \dddot{x}\dddot{\mu} dt \tag{A.33}$$

Integrating by parts.

$$\int_0^T \dddot{x}\dddot{\mu} dt = \int_0^T u dv = [uv]_0^T - \int_0^T v du \tag{A.34}$$

Where:

$$\begin{array}{ll} u = \dddot{x} & dv = \dddot{\mu} dt \\ du = x^{(4)} dt & v = \ddot{\mu} \end{array}$$

Thus:

$$\int_0^T \dddot{x}\dddot{\mu} dt = [\dddot{x}\ddot{\mu}]_0^T - \int_0^T \ddot{\mu} x^{(4)} dt = - \int_0^T \ddot{\mu} x^{(4)} dt \tag{A.35}$$

Integrating by parts a second time.

$$-\int_0^T \ddot{\mu} x^{(4)} dt = -\int_0^T u dv = -\left[uv\right]_0^T + \int_0^T v du \tag{A.36}$$

Where:

$$\begin{array}{ll} u = x^{(4)} & dv = \ddot{\mu} dt \\ du = x^{(5)} dt & v = \dot{\mu} \end{array}$$

Thus:

$$-\int_0^T \ddot{\mu} x^{(4)} dt = -\left[x^{(4)}\dot{\mu}\right]_0^T + \int_0^T \dot{\mu} x^{(5)} dt = \int_0^T \dot{\mu} x^{(5)} dt \tag{A.37}$$

Integrating by parts a final time.

$$\int_0^T \dot{\mu} x^{(5)} dt = \int_0^T u dv = \left[uv\right]_0^T - \int_0^T v du \tag{A.38}$$

Where:

$$\begin{array}{ll} u = x^{(5)} & dv = \dot{\mu} dt \\ du = x^{(6)} dt & v = \mu \end{array}$$

Thus:

$$\int_0^T \dot{\mu} x^{(5)} dt = \left[x^{(5)}\mu\right]_0^T - \int_0^T \mu x^{(6)} dt = -\int_0^T \mu x^{(6)} dt \tag{A.39}$$

Finally producing:

$$\left.\frac{dF\left(x + e\mu\right)}{e}\right|_{e=0} = -\int_0^T \mu x^{(6)} dt \equiv 0 \tag{A.40}$$

This must hold true for any function of $\mu(t)$ which has the properties specified above, thus Equation A.41 must be true.

$$x^{(6)} = 0 \tag{A.41}$$

The function for a minimum Jerk trajectory may now be solved as was done from Equation A.9 in the previous section.

Appendix B

Preliminary Dynamic Modelling

The Function for the Lagrangian of MyPAM

$$
\begin{aligned}
&\mathcal{L} = 0.0813\dot{\theta_0}^2 - 0.0173\dot{\theta_1}^2 \cos(2\theta_0 + 2\theta_1) - 0.0173\dot{\theta_0}^2 \cos(2\theta_0 + 2\theta_1) + 0.0221\dot{\theta_1}^2 + 0.599L1^2\dot{\theta_0}^2 \\
&+0.0689L2^2\dot{\theta_0}^2+0.0689L2^2\dot{\theta_1}^2+0.0443\dot{\theta}_0\dot{\theta}_1-0.0346\dot{\theta}_0\dot{\theta}_1 \cos(2\theta_0 + 2\theta_1)+10\times10^{-13}\dot{\theta}_0\dot{\theta}_1 \cos(2\theta_0 - 2.0\theta_1) \\
&+ 0.137L2^2\dot{\theta}_0\dot{\theta}_1 - 0.0363L1\dot{\theta_0}^2 \cos(2.0\theta_0 + \theta_1) - 0.0363L1\dot{\theta_0}^2 \sin(2.0\theta_0 + \theta_1) + 0.0363L1\dot{\theta_0}^2 \cos(\theta_1) \\
&- 0.0363L1\dot{\theta_0}^2 \sin(\theta_1) - 0.0689L2^2\dot{\theta_0}^2 \cos(2\theta_0 + 2\theta_1) - 0.0689L2^2\dot{\theta_1}^2 \cos(2\theta_0 + 2\theta_1) \\
&- 0.0363L1\dot{\theta}_0\dot{\theta}_1 \cos(2.0\theta_0 + \theta_1) - 0.0363L1\dot{\theta}_0\dot{\theta}_1 \sin(2.0\theta_0 + \theta_1) + 0.0363L1\dot{\theta}_0\dot{\theta}_1 \cos(\theta_1) \\
&- 0.0363L1\dot{\theta}_0\dot{\theta}_1 \sin(\theta_1) - 0.0689L1L2\dot{\theta_0}^2 \cos(2.0\theta_0 + \theta_1) - 0.137L2^2\dot{\theta}_0\dot{\theta}_1 \cos(2\theta_0 + 2\theta_1) \\
&- 0.0689L1L2\dot{\theta_0}^2 \sin(2.0\theta_0 + \theta_1) + 0.0689L1L2\dot{\theta_0}^2 \cos(\theta_1) - 0.0689L1L2\dot{\theta_0}^2 \sin(\theta_1) \\
&- 0.0689L1L2\dot{\theta}_0\dot{\theta}_1 \cos(2.0\theta_0 + \theta_1) - 0.0689L1L2\dot{\theta}_0\dot{\theta}_1 \sin(2.0\theta_0 + \theta_1) + 0.0689L1L2\dot{\theta}_0\dot{\theta}_1 \cos(\theta_1) \\
&- 0.0689L1L2\dot{\theta}_0\dot{\theta}_1 \sin(\theta_1)
\end{aligned}
$$

The Unprocessed Function for Torque at Joint 0

$$
\begin{aligned}
&\tau_0{=}0.128\ddot{\theta}_0 + 0.00969\ddot{\theta}_1 + 1.19{L_1}^2\ddot{\theta}_0 + 0.0692\cos\left(\theta_0\right)^2\ddot{\theta}_0\\
&+ 0.0692\cos\left(\theta_0\right)^2\ddot{\theta}_1 + 0.0692\cos\left(\theta_1\right)^2\ddot{\theta}_0 + 0.0692\cos\left(\theta_1\right)^2\ddot{\theta}_1\\
&- 10\times10^{-13}\ddot{\theta}_1\cos\left(2\theta_1\right) + 2\times10^{-12}\sin\left(2\theta_1\right){\dot{\theta}_1}^2\\
&+ 0.275{L_2}^2\cos\left(\theta_0\right)^2\ddot{\theta}_0 + 0.275{L_2}^2\cos\left(\theta_0\right)^2\ddot{\theta}_1 + 0.275{L_2}^2\cos\left(\theta_1\right)^2\ddot{\theta}_0\\
&+ 0.275{L_2}^2\cos\left(\theta_1\right)^2\ddot{\theta}_1 - 0.138\cos\left(\theta_1\right)^2\cos\left(\theta_1\right)^2\ddot{\theta}_0 - 0.138\cos\left(\theta_0\right)^2\cos\left(\theta_1\right)^2\ddot{\theta}_1\\
&+ 0.0726L_1\cos\left(\theta_1\right)\ddot{\theta}_0 + 0.0363L_1\cos\left(\theta_1\right)\ddot{\theta}_1\\
&+ 2\times10^{-12}\cos\left(\theta_0\right)^2\ddot{\theta}_1\cos\left(2\theta_1\right) - 0.0726L_1\sin\left(\theta_1\right)\ddot{\theta}_0\\
&- 0.0363L_1\sin\left(\theta_1\right)\ddot{\theta}_1 - 0.0363L_1{\dot{\theta}_1}^2\cos\left(\theta_1\right)\\
&- 0.0363L_1{\dot{\theta}_1}^2\sin\left(\theta_1\right) - 4\times10^{-12}\sin\left(2\theta_1\right){\dot{\theta}_1}^2\cos\left(\theta_0\right)^2\\
&- 0.0692{\dot{\theta}_0}^2\cos\left(\theta_0\right)\sin\left(\theta_0\right) - 0.0692{\dot{\theta}_1}^2\cos\left(\theta_0\right)\sin\left(\theta_0\right)\\
&- 0.0692{\dot{\theta}_0}^2\cos\left(\theta_1\right)\sin\left(\theta_1\right) - 0.0692{\dot{\theta}_1}^2\cos\left(\theta_1\right)\sin\left(\theta_1\right)\\
&- 0.275{L_2}^2{\dot{\theta}_0}^2\cos\left(\theta_0\right)\sin\left(\theta_0\right) - 0.275{L_2}^2{\dot{\theta}_1}^2\cos\left(\theta_0\right)\sin\left(\theta_0\right)\\
&- 0.275{L_2}^2{\dot{\theta}_0}^2\cos\left(\theta_1\right)\sin\left(\theta_1\right) - 0.275{L_2}^2{\dot{\theta}_1}^2\cos\left(\theta_1\right)\sin\left(\theta_1\right)\\
&+ 0.138{\dot{\theta}_0}^2\cos\left(\theta_0\right)\cos\left(\theta_1\right)^2\sin\left(\theta_0\right) + 0.138{\dot{\theta}_0}^2\cos\left(\theta_0\right)^2\cos\left(\theta_1\right)\sin\left(\theta_1\right)\\
&+ 0.138{\dot{\theta}_1}^2\cos\left(\theta_0\right)\cos\left(\theta_1\right)^2\sin\left(\theta_0\right) + 0.138{\dot{\theta}_1}^2\cos\left(\theta_0\right)^2\cos\left(\theta_1\right)\sin\left(\theta_1\right)\\
&- 0.0726L_1\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_1\right) - 0.0726L_1\cos\left(\theta_1\right)\ddot{\theta}_0\cos\left(2\theta_0\right)\\
&- 0.0363L_1\cos\left(\theta_1\right)\ddot{\theta}_1\cos\left(2\theta_0\right) - 0.0726L_1\dot{\theta}_0\dot{\theta}_1\sin\left(\theta_1\right)\\
&- 0.0726L_1\sin\left(2\theta_0\right)\cos\left(\theta_1\right)\ddot{\theta}_0 - 0.0726L_1\sin\left(\theta_1\right)\ddot{\theta}_0\cos\left(2\theta_0\right)\\
&- 0.0363L_1\sin\left(2\theta_0\right)\cos\left(\theta_1\right)\ddot{\theta}_1 - 0.0363L_1\sin\left(\theta_1\right)\ddot{\theta}_1\cos\left(2\theta_0\right)\\
&- 0.138\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_0\right)\sin\left(\theta_0\right) - 0.138\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_1\right)\sin\left(\theta_1\right)\\
&- 0.0726L_1{\dot{\theta}_0}^2\cos\left(\theta_1\right)\cos\left(2\theta_0\right) - 0.0363L_1{\dot{\theta}_1}^2\cos\left(\theta_1\right)\cos\left(2\theta_0\right)\\
&+ 0.0726L_1\sin\left(2\theta_0\right)\sin\left(\theta_1\right)\ddot{\theta}_0 + 0.0363L_1\sin\left(2\theta_0\right)\sin\left(\theta_1\right)\ddot{\theta}_1\\
&+ 2\times10^{-12}\sin\left(2\theta_1\right)\cos\left(\theta_0\right)\sin\left(\theta_0\right)\ddot{\theta}_1 + 0.0726L_1\sin\left(2\theta_0\right){\dot{\theta}_0}^2\cos\left(\theta_1\right)\\
&+ 0.0726L_1{\dot{\theta}_0}^2\sin\left(\theta_1\right)\cos\left(2\theta_0\right) + 0.0363L_1\sin\left(2\theta_0\right){\dot{\theta}_1}^2\cos\left(\theta_1\right)\\
&+ 0.0363L_1{\dot{\theta}_1}^2\sin\left(\theta_1\right)\cos\left(2\theta_0\right) + 4\times10^{-12}{\dot{\theta}_1}^2\cos\left(\theta_0\right)\sin\left(\theta_0\right)\cos\left(2\theta_1\right)\\
&+ 0.0726L_1\sin\left(2\theta_0\right){\dot{\theta}_0}^2\sin\left(\theta_1\right) + 0.0363L_1\sin\left(2\theta_0\right){\dot{\theta}_1}^2\sin\left(\theta_1\right)\\
&+ 0.137L_1L_2\cos\left(\theta_1\right)\ddot{\theta}_0 + 0.0689L_1L_2\cos\left(\theta_1\right)\ddot{\theta}_1 - 0.137L_1L_2\sin\left(\theta_1\right)\ddot{\theta}_0\\
&- 0.0689L_1L_2\sin\left(\theta_1\right)\ddot{\theta}_1 - 0.0689L_1L_2{\dot{\theta}_1}^2\cos\left(\theta_1\right) - 0.551{L_2}^2\cos\left(\theta_0\right)^2\cos\left(\theta_1\right)^2\ddot{\theta}_0\\
&- 0.551{L_2}^2\cos\left(\theta_0\right)^2\cos\left(\theta_1\right)^2\ddot{\theta}_1 - 0.0689L_1L_2{\dot{\theta}_1}^2\sin\left(\theta_1\right)\\
&+ 0.551{L_2}^2{\dot{\theta}_0}^2\cos\left(\theta_0\right)\cos\left(\theta_1\right)^2\sin\left(\theta_0\right) + 0.551{L_2}^2{\dot{\theta}_0}^2\cos\left(\theta_0\right)^2\cos\left(\theta_1\right)\sin\left(\theta_1\right)\\
&+ 0.551{L_2}^2{\dot{\theta}_1}^2\cos\left(\theta_0\right)\cos\left(\theta_1\right)^2\sin\left(\theta_0\right) + 0.551{L_2}^2{\dot{\theta}_1}^2\cos\left(\theta_0\right)^2\cos\left(\theta_1\right)\sin\left(\theta_1\right)\\
&- 0.551{L_2}^2\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_0\right)\sin\left(\theta_0\right) - 0.551{L_2}^2\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_1\right)\sin\left(\theta_1\right)\\
&+ 0.277\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_0\right)\cos\left(\theta_1\right)^2\sin\left(\theta_0\right) + 0.277\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_0\right)^2\cos\left(\theta_1\right)\sin\left(\theta_1\right)\\
&+ 0.138\cos\left(\theta_0\right)\cos\left(\theta_1\right)\sin\left(\theta_0\right)\sin\left(\theta_1\right)\ddot{\theta}_0 + 0.138\cos\left(\theta_0\right)\cos\left(\theta_1\right)\sin\left(\theta_0\right)\sin\left(\theta_1\right)\ddot{\theta}_1\\
&- 0.0726L_1\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_1\right)\cos\left(2\theta_0\right) + 0.0726L_1\sin\left(2\theta_0\right)\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_1\right)\\
&+ 0.0726L_1\dot{\theta}_0\dot{\theta}_1\sin\left(\theta_1\right)\cos\left(2\theta_0\right) + 0.0726L_1\sin\left(2\theta_0\right)\dot{\theta}_0\dot{\theta}_1\sin\left(\theta_1\right)\\
&- 0.137L_1L_2\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_1\right) - 0.137L_1L_2\cos\left(\theta_1\right)\ddot{\theta}_0\cos\left(2\theta_0\right)\\
&- 0.0689L_1L_2\cos\left(\theta_1\right)\ddot{\theta}_1\cos\left(2\theta_0\right) - 0.137L_1L_2\dot{\theta}_0\dot{\theta}_1\sin\left(\theta_1\right)\\
&- 0.137L_1L_2\sin\left(2\theta_0\right)\cos\left(\theta_1\right)\ddot{\theta}_0 - 0.137L_1L_2\sin\left(\theta_1\right)\ddot{\theta}_0\cos\left(2\theta_0\right)\\
&- 0.0689L_1L_2\sin\left(2\theta_0\right)\cos\left(\theta_1\right)\ddot{\theta}_1 - 0.0689L_1L_2\sin\left(\theta_1\right)\ddot{\theta}_1\cos\left(2\theta_0\right)\\
&- 0.137L_1L_2{\dot{\theta}_0}^2\cos\left(\theta_1\right)\cos\left(2\theta_0\right) - 0.0689L_1L_2{\dot{\theta}_1}^2\cos\left(\theta_1\right)\cos\left(2\theta_0\right)\\
&+ 0.137L_1L_2\sin\left(2\theta_0\right)\sin\left(\theta_1\right)\ddot{\theta}_0 + 0.0689L_1L_2\sin\left(2\theta_0\right)\sin\left(\theta_1\right)\ddot{\theta}_1\\
&+ 0.137L_1L_2\sin\left(2\theta_0\right){\dot{\theta}_0}^2\cos\left(\theta_1\right) + 0.137L_1L_2{\dot{\theta}_0}^2\sin\left(\theta_1\right)\cos\left(2\theta_0\right)\\
&+ 0.0689L_1L_2\sin\left(2\theta_0\right){\dot{\theta}_1}^2\cos\left(\theta_1\right) + 0.0689L_1L_2{\dot{\theta}_1}^2\sin\left(\theta_1\right)\cos\left(2\theta_0\right)\\
&+ 0.137L_1L_2\sin\left(2\theta_0\right){\dot{\theta}_0}^2\sin\left(\theta_1\right) + 0.0689L_1L_2\sin\left(2\theta_0\right){\dot{\theta}_1}^2\sin\left(\theta_1\right)\\
&- 0.137L_1L_2\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_1\right)\cos\left(2\theta_0\right) + 0.137L_1L_2\sin\left(2\theta_0\right)\dot{\theta}_0\dot{\theta}_1\cos\left(\theta_1\right)\\
&+ 0.137L_1L_2\dot{\theta}_0\dot{\theta}_1\sin\left(\theta_1\right)\cos\left(2\theta_0\right) + 0.137L_1L_2\sin\left(2\theta_0\right)\dot{\theta}_0\dot{\theta}_1\sin\left(\theta_1\right)
\end{aligned}
$$

$$+ 1.10{L_2}^2\dot{\theta_0}\dot{\theta_1}\cos\left(\theta_0\right)\cos\left(\theta_1\right)^2\sin\left(\theta_0\right) + 1.10{L_2}^2\dot{\theta_0}\dot{\theta_1}\cos\left(\theta_0\right)^2\cos\left(\theta_1\right)\sin\left(\theta_1\right)$$
$$+ 0.551{L_2}^2\cos\left(\theta_0\right)\cos\left(\theta_1\right)\sin\left(\theta_0\right)\sin\left(\theta_1\right)\ddot{\theta_0}$$
$$+ 0.551{L_2}^2\cos\left(\theta_0\right)\cos\left(\theta_1\right)\sin\left(\theta_0\right)\sin\left(\theta_1\right)\ddot{\theta_1}$$

The Unprocessed Function for Torque at Joint 1

$$\tau_1 = 0.00969\ddot{\theta}_0 + 0.00969\ddot{\theta}_1 + 0.0692\cos(\theta_0)^2\ddot{\theta}_0$$
$$+ 0.0692\cos(\theta_0)^2\ddot{\theta}_1 + 0.0692\cos(\theta_1)^2\ddot{\theta}_0 + 0.0692\cos(\theta_1)^2\ddot{\theta}_1$$
$$- 10 \times \times 10^{-13}\ddot{\theta}_0\cos(2\theta_1) - 2 \times 10^{-12}\sin(2\theta_1)\,\dot{\theta}_0^{\,2} + 0.275{L_2}^2\cos(\theta_0)^2\ddot{\theta}_0$$
$$+ 0.275{L_2}^2\cos(\theta_0)^2\ddot{\theta}_1 + 0.275{L_2}^2\cos(\theta_1)^2\ddot{\theta}_0 + 0.275{L_2}^2\cos(\theta_1)^2\ddot{\theta}_1$$
$$- 0.138\cos(\theta_0)^2\cos(\theta_1)^2\ddot{\theta}_0 - 0.138\cos(\theta_0)^2\cos(\theta_1)^2\ddot{\theta}_1 + 0.0363L_1\cos(\theta_1)\,\ddot{\theta}_0$$
$$+ 2 \times 10^{-12}\cos(\theta_0)^2\ddot{\theta}_0\cos(2\theta_1) - 0.0363L_1\sin(\theta_1)\,\ddot{\theta}_0$$
$$+ 0.0363L_1\dot{\theta}_0^{\,2}\cos(\theta_1) + 0.0363L_1\dot{\theta}_0^{\,2}\sin(\theta_1)$$
$$+ 4 \times 10^{-12}\sin(2\theta_1)\,\dot{\theta}_0^{\,2}\cos(\theta_0)^2 - 0.0692\dot{\theta}_0^{\,2}\cos(\theta_0)\sin(\theta_0)$$
$$- 0.0692\dot{\theta}_1^{\,2}\cos(\theta_0)\sin(\theta_0) - 0.0692\dot{\theta}_0^{\,2}\cos(\theta_1)\sin(\theta_1)$$
$$- 0.0692\dot{\theta}_1^{\,2}\cos(\theta_1)\sin(\theta_1) - 0.275{L_2}^2\dot{\theta}_0^{\,2}\cos(\theta_0)\sin(\theta_0)$$
$$- 0.275{L_2}^2\dot{\theta}_1^{\,2}\cos(\theta_0)\sin(\theta_0) - 0.275{L_2}^2\dot{\theta}_0^{\,2}\cos(\theta_1)\sin(\theta_1)$$
$$- 0.275{L_2}^2\dot{\theta}_1^{\,2}\cos(\theta_1)\sin(\theta_1) + 0.138\dot{\theta}_0^{\,2}\cos(\theta_0)\cos(\theta_1)^2\sin(\theta_0)$$
$$+ 0.138\dot{\theta}_0^{\,2}\cos(\theta_0)^2\cos(\theta_1)\sin(\theta_1) + 0.138\dot{\theta}_1^{\,2}\cos(\theta_0)\cos(\theta_1)^2\sin(\theta_0)$$
$$+ 0.138\dot{\theta}_1^{\,2}\cos(\theta_0)^2\cos(\theta_1)\sin(\theta_1) - 0.0363L_1\cos(\theta_1)\,\ddot{\theta}_0\cos(2\theta_0)$$
$$- 0.0363L_1\sin(2\theta_0)\cos(\theta_1)\,\ddot{\theta}_0 - 0.0363L_1\sin(\theta_1)\,\ddot{\theta}_0\cos(2\theta_0)$$
$$- 0.138\dot{\theta}_0\dot{\theta}_1\cos(\theta_0)\sin(\theta_0) - 0.138\dot{\theta}_0\dot{\theta}_1\cos(\theta_1)\sin(\theta_1)$$
$$- 0.0363L_1\dot{\theta}_0^{\,2}\cos(\theta_1)\cos(2\theta_0) + 0.0363L_1\sin(2\theta_0)\sin(\theta_1)\,\ddot{\theta}_0$$
$$+ 2 \times 10^{-12}\sin(2\theta_1)\cos(\theta_0)\sin(\theta_0)\,\ddot{\theta}_0 + 0.0363L_1\sin(2\theta_0)\,\dot{\theta}_0^{\,2}\cos(\theta_1)$$
$$+ 0.0363L_1\dot{\theta}_0^{\,2}\sin(\theta_1)\cos(2\theta_0) - 4 \times 10^{-12}\dot{\theta}_0^{\,2}\cos(\theta_0)\sin(\theta_0)\cos(2\theta_1)$$
$$+ 0.0363L_1\sin(2\theta_0)\,\dot{\theta}_0^{\,2}\sin(\theta_1) + 0.0689L_1L_2\cos(\theta_1)\,\ddot{\theta}_0$$
$$- 0.0689L_1L_2\sin(\theta_1)\,\ddot{\theta}_0 + 0.0689L_1L_2\dot{\theta}_0^{\,2}\cos(\theta_1) - 0.551{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2\ddot{\theta}_0$$
$$- 0.551{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2\ddot{\theta}_1 + 0.0689L_1L_2\dot{\theta}_0^{\,2}\sin(\theta_1)$$
$$+ 0.551{L_2}^2\dot{\theta}_0^{\,2}\cos(\theta_0)\cos(\theta_1)^2\sin(\theta_0) + 0.551{L_2}^2\dot{\theta}_0^{\,2}\cos(\theta_0)^2\cos(\theta_1)\sin(\theta_1)$$
$$+ 0.551{L_2}^2\dot{\theta}_1^{\,2}\cos(\theta_0)\cos(\theta_1)^2\sin(\theta_0) + 0.551{L_2}^2\dot{\theta}_1^{\,2}\cos(\theta_0)^2\cos(\theta_1)\sin(\theta_1)$$
$$- 0.551{L_2}^2\dot{\theta}_0\dot{\theta}_1\cos(\theta_0)\sin(\theta_0) - 0.551{L_2}^2\dot{\theta}_0\dot{\theta}_1\cos(\theta_1)\sin(\theta_1)$$
$$+ 0.277\dot{\theta}_0\dot{\theta}_1\cos(\theta_0)\cos(\theta_1)^2\sin(\theta_0) + 0.277\dot{\theta}_0\dot{\theta}_1\cos(\theta_0)^2\cos(\theta_1)\sin(\theta_1)$$
$$+ 0.138\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\,\ddot{\theta}_0 + 0.138\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\,\ddot{\theta}_1$$
$$- 0.0689L_1L_2\cos(\theta_1)\,\ddot{\theta}_0\cos(2\theta_0) - 0.0689L_1L_2\sin(2\theta_0)\cos(\theta_1)\,\ddot{\theta}_0$$
$$- 0.0689L_1L_2\sin(\theta_1)\,\ddot{\theta}_0\cos(2\theta_0) - 0.0689L_1L_2\dot{\theta}_0^{\,2}\cos(\theta_1)\cos(2\theta_0)$$
$$+ 0.0689L_1L_2\sin(2\theta_0)\sin(\theta_1)\,\ddot{\theta}_0 + 0.0689L_1L_2\sin(2\theta_0)\,\dot{\theta}_0^{\,2}\cos(\theta_1)$$
$$+ 0.0689L_1L_2\dot{\theta}_0^{\,2}\sin(\theta_1)\cos(2\theta_0) + 0.0689L_1L_2\sin(2\theta_0)\,\dot{\theta}_0^{\,2}\sin(\theta_1)$$
$$+ 1.10{L_2}^2\dot{\theta}_0\dot{\theta}_1\cos(\theta_0)\cos(\theta_1)^2\sin(\theta_0) + 1.10{L_2}^2\dot{\theta}_0\dot{\theta}_1\cos(\theta_0)^2\cos(\theta_1)\sin(\theta_1)$$
$$+ 0.551{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\,\ddot{\theta}_0$$
$$+ 0.551{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\,\ddot{\theta}_1$$

The Processed Function for Torque at Joint 0

$$
\begin{aligned}
&\tau_0 = \ddot{\theta_0}\ \Big(0.128 + 1.20{L_1}^2 + 0.0693\cos(\theta_0)^2 + 0.0693\cos(\theta_1) + 0.276{L_2}^2\cos(\theta_0)^2 \\
&+ 0.276{L_2}^2\cos(\theta_1)^2 - 0.139\cos(\theta_1)^2\cos(\theta_1)^2 + 0.0726L_1\cos(\theta_1) - 0.0726L_1\sin(\theta_1) \\
&- 0.0726L_1\cos(\theta_1)\cos(2\theta_0) - 0.0726L_1\sin(2\theta_0)\cos(\theta_1) - 0.0726L_1\sin(\theta_1)\cos(2\theta_0) \\
&+ 0.0726L_1\sin(2\theta_0)\sin(\theta_1) + 0.138L_1L_2\cos(\theta_1) - 0.138L_1L_2\sin(\theta_1) \\
&- 0.552{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2 + 0.139\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1) - 0.138L_1L_2\cos(\theta_1)\cos(2\theta_0) \\
&- 0.138L_1L_2\sin(2\theta_0)\cos(\theta_1) - 0.138L_1L_2\sin(\theta_1)\cos(2\theta_0) + 0.138L_1L_2\sin(2\theta_0)\sin(\theta_1) \\
&+ 0.552{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\Big) \\
&+ \ddot{\theta_1}\Big(0.00969 + 0.0693\cos(\theta_0)^2 + 0.0693\cos(\theta_1)^2 + 0.276{L_2}^2\cos(\theta_0)^2 + 0.276{L_2}^2\cos(\theta_1)^2 \\
&- 0.139\cos(\theta_0)^2\cos(\theta_1)^2 + 0.0363L_1\cos(\theta_1) - 0.0363L_1\sin(\theta_1) - 0.0363L_1\cos(\theta_1)\cos(2\theta_0) \\
&- 0.0363L_1\sin(2\theta_0)\cos(\theta_1) - 0.0363L_1\sin(\theta_1)\cos(2\theta_0) + 0.0363L_1\sin(2\theta_0)\sin(\theta_1) \\
&+ 0.0689L_1L_2\cos(\theta_1) - 0.0689L_1L_2\sin(\theta_1) - 0.552{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2 \\
&+ 0.139\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1) - 0.0689L_1L_2\cos(\theta_1)\cos(2\theta_0) \\
&- 0.0689L_1L_2\sin(2\theta_0)\cos(\theta_1) - 0.0689L_1L_2\sin(\theta_1)\cos(2\theta_0) + 0.0689L_1L_2\sin(2\theta_0)\sin(\theta_1) \\
&+ 0.552{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\Big)
\end{aligned}
$$

The Processed Function for Torque at Joint 1

$$
\begin{aligned}
&\tau_1 = \ddot{\theta_0}\ \Big(0.00969 + 0.0693\cos(\theta_0)^2 + 0.0693\cos(\theta_1)^2 + 0.276{L_2}^2\cos(\theta_0)^2 + 0.276{L_2}^2\cos(\theta_1)^2 \\
&- 0.139\cos(\theta_0)^2\cos(\theta_1)^2 + 0.0363L_1\cos(\theta_1) - 0.0363L_1\sin(\theta_1) - 0.0363L_1\cos(\theta_1)\cos(2\theta_0) \\
&- 0.0363L_1\sin(2\theta_0)\cos(\theta_1) - 0.0363L_1\sin(\theta_1)\cos(2\theta_0) + 0.0363L_1\sin(2\theta_0)\sin(\theta_1) \\
&+ 0.0689L_1L_2\cos(\theta_1) - 0.0689L_1L_2\sin(\theta_1) - 0.552{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2 \\
&+ 0.139\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1) - 0.0689L_1L_2\cos(\theta_1)\cos(2\theta_0) \\
&- 0.0689L_1L_2\sin(2\theta_0)\cos(\theta_1) - 0.0689L_1L_2\sin(\theta_1)\cos(2\theta_0) + 0.0689L_1L_2\sin(2\theta_0)\sin(\theta_1) \\
&+ 0.552{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\Big) \\
&+ \ddot{\theta_1}\Big(0.00969 + 0.0693\cos(\theta_0)^2 + 0.0693\cos(\theta_1)^2 + 0.276{L_2}^2\cos(\theta_0)^2 + 0.276{L_2}^2\cos(\theta_1)^2 \\
&- 0.139\cos(\theta_0)^2\cos(\theta_1)^2 - 0.552{L_2}^2\cos(\theta_0)^2\cos(\theta_1)^2 + 0.139\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1) \\
&+ 0.552{L_2}^2\cos(\theta_0)\cos(\theta_1)\sin(\theta_0)\sin(\theta_1)\ \Big)
\end{aligned}
$$

Appendix C

Unit Tests

Test Summary

0 Test Error(s)
0 Test(s) Failed
0 Test(s) Skipped
55 Test(s) Passed
Time elapsed: 00:06:32

Test Results

Failed

None.

Passed

Module Name	VI Name	Duration [s]	Code Coverage [%]
Program Control Module	sendStartEvent.vi	1.85	100
Program Control Module	startUEFGV.vi	1.80	100
Program Control Module	preRunCTRL.vi	2.03	100
Program Control Module	postRunCTRL.vi	1.89	100
Program Control Module	dequeueMessage.vi	1.84	100
Program Control Module	enqueueShutdownMessage.vi	1.90	100
Program Control Module	initData.vi	1.35	100
Program Control Module	manageQueues.vi	3.04	100
Program Control Module	enqueueMessage.vi	1.84	100
Program Control Module	programControlQueueFGV.vi	1.35	100
Program Control Module	userEventsFGV.vi	2.00	100
Program Control Module	userEventsFGV2.vi	0.88	100

Module Name	VI Name	Duration [s]	Code Coverage [%]
Program Control Module	dataFGV.vi	0.85	100
Communications Module	clustertoRawString.vi	0.77	100
Communications Module	communicateMLCStart.vi	1.90	100
Communications Module	initListenData.vi	1.59	100
Communications Module	scaleMLCToMyPAM.vi	1.01	100
Communications Module	scaleMyPAMToMLC.vi	0.99	100
Communications Module	shutdownReceived.vi	2.05	100
Communications Module	UDPListenFGV.vi	1.16	100
Logging Module	sendErrorLog.vi	1.68	100
Logging Module	dequeueMessage.vi	1.55	100
Logging Module	enqueueMessage.vi	1.57	100
Logging Module	formatFileData.vi	1.21	100
Logging Module	loggingModuleQueueFGV.vi	1.23	100
Logging Module	postRun.vi	1.53	100
Logging Module	preRun.vi	1.54	100
Logging Module	sendCloseLog.vi	1.49	100
Logging Module	sendOpenLog.vi	1.46	100
Logging Module	loggingModuleStart.vi	1.92	100
Error Module	HandleError.vi	1.24	100
Error Module	ErrorPostRun.vi	1.52	100
Error Module	ErrorPreRun.vi	1.52	100
Error Module	dequeueMessage.vi	1.52	100
Error Module	ErrorMonitorQueueFGV.vi	1.23	100
Error Module	enqueueMessage.vi	1.51	100
Error Module	ErrorStart.vi	1.91	100
Data Acquisition Module	countToAngle.vi	0.77	100
Data Acquisition Module	forwardKinematics.vi	1.12	100
Data Acquisition Module	externalAqcuisitionStart.vi	1.93	100
Control Module	AdmittanceFilter.vi	0.97	100
Control Module	InitialConditions FGV.vi	1.39	100
Control Module	InverseKinematics.vi	1.10	100
Control Module	MinimumJerkGenerateNext.vi	1.79	100
Control Module	NewTarget.vi	1.71	100
Control Module	newTarget2.vi	1.41	100
Control Module	PID.vi	1.97	100

Module Name	VI Name	Duration [s]	Code Coverage [%]
Control Module	selectData.vi	1.20	100
Control Module	TargetMet.vi	1.27	100
Control Module	tf FGV.vi	1.39	100
Control Module	Timer FGV.vi	1.57	100
Control Module	RawStringtoCluster.vi	1.27	100
Control Module	lowLevelControllerModuleStart.vi	1.90	100
Control Module	getAllData.vi	1.44	100
Control Module	enableMotors.vi	1.60	100