

Control Design and Implementation of an Active Transtibial Prosthesis

Joseph Klein
Marquette University

Recommended Citation

Klein, Joseph, "Control Design and Implementation of an Active Transtibial Prosthesis" (2018). *Master's Theses (2009 -)*. 481.
https://epublications.marquette.edu/theses_open/481

CONTROL DESIGN AND IMPLEMENTATION OF AN ACTIVE TRANSTIBIAL
PROSTHESIS

by

Joseph G. Klein, B.S.

A Thesis Submitted to the Faculty of the
Graduate School, Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

August 2018

ABSTRACT
CONTROL DESIGN AND IMPLEMENTATION OF AN ACTIVE TRANSTIBIAL
PROSTHESIS

Joseph G. Klein, B.S.

Marquette University, 2018

Prior work at Marquette University developed the Marquette Prosthesis, an active transtibial prosthesis that utilized a torsional spring and a four-bar mechanism. The controls for the Marquette Prosthesis implemented a finite state control algorithm to determine the state of gait of the amputee along with two lower level controllers, a PI moment controller to control the moment during stance and a PID position controller to control the position during stance. The Marquette Prosthesis was successful in mimicking the gait profile presented by Winter. However, after completing human subject testing, the Marquette Prosthesis was insufficient in trying to match the gait profile of those who varied from this textbook stride.

Active transtibial prostheses typically apply finite state control algorithms that struggle with cadence and gait variability of the amputee. Recent work in artificial neural networks (ANN) have shown the possibility to predict the user's intent which can be used as an input signal in an improved controller. The Marquette Prosthesis II was developed that uses a stiffness controller to control the relationship between the position and torque of the ankle. A model of the improved Marquette Prosthesis II was developed in Simulink to ensure that the stiffness controller was robust enough and that this type of control was possible with the limitations of the Marquette Prosthesis, i.e., the link lengths, torsional spring and motor. The mechanical system of the Marquette Prosthesis was then changed such that the spring was in series between the motor and four-bar mechanism to establish a relationship between the motor position, torque of the spring and four-bar mechanism. The control hardware was selected and the stiffness controller was implemented on the Marquette Prosthesis II. The Marquette Prosthesis II control algorithm was tested and validated to show that this approach is feasible.

ACKNOWLEDGMENTS

Joseph G. Klein, B.S.

I would like to thank my Mom, Dad and siblings for their continued support and inspiration. I would like to thank Dr. Philip Voglewede for being an incredible advisor and mentor to me over the past two years, pushing me to be a better person and engineer. I would like to thank Tony Buchta, Ryan Callahan, Jack Alberts and Shaoli Wu for the countless conversations and support in the Wede lab. I would like to thank my friends for helping me to keep my sanity over the past two years and being as excited as I am about the success of this project. I would like to thank Erika Zabre for her patience and help in running the tests with me in the Human Performance Laboratory. I would like to thank Dr. Mark Nagurka for his passion for engineering and pushing me to be a better engineer. I would like to thank Dr. Aderiano daSilva for pushing me to pursue a career in controls and teaching me how to be a better controls engineer. Finally, I would like to thank Marquette University and all of those at Marquette who have helped me develop into the person and engineer I am.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	i
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER 1 Introduction	1
1.1 Motivation	1
1.2 Fundamental Causes of Transtibial Amputation	1
1.3 Development of Lower Limb Protheses	3
1.3.1 Passive Protheses	4
1.3.2 Active Protheses	4
1.4 Development of Artificial Neural Network	5
1.5 Motivation for this Research	6
1.6 Scope	7
1.7 Research Contributions	8
CHAPTER 2 Literature Review	10
2.1 Myoelectric Control of Powered Upper Limb Protheses	10
2.1.1 Weighting Synergistic Muscle Movements - Temple University	12
2.1.2 Endpoint Control - University of California, Los Angeles . . .	14

2.1.3	Transient Myoelectric Control - University of New Brunswick in Canada	15
2.2	Mechanical and Control Systems of Lower Limb Prostheses	17
2.2.1	Össur Propio Foot with Evo	19
2.2.2	BioM - MIT Powered Ankle-Foot Prosthesis	19
2.2.3	SPARKy 3 - ASU Powered Ankle-Foot Prosthesis	25
2.2.4	Vanderbilt University Powered Ankle and Knee Prosthesis	31
2.2.5	System Hardware	34
2.3	Summary	35
2.4	Marquette Prosthesis	35
	CHAPTER 3 Dynamic Model and Simulation	37
3.1	Four Bar Mechanism Analysis	38
3.1.1	Kinematics	39
3.1.2	Quasi-statics	42
3.2	Simulink Model	43
3.2.1	Dynamic Simulation and PI Tuning	44
3.2.2	Investigation of Linear Spiral Spring	46
3.2.3	Effect of Spring Constant on Output Torque	47
3.2.4	Motor Torque and Speed Capabilities	49
3.2.5	Final Spring Selection	50

CHAPTER 4 Mechanical System Redesign	51
4.1 Redesign of Link 3 to move spring in Series	53
4.1.1 Press Fit Analysis	56
CHAPTER 5 Control System Architecture and Implementation	58
5.1 Overall Control System Architecture	58
5.2 Controller Implementation	59
5.2.1 Computer System Overview of Other Successful Prostheses	60
5.2.2 Selection of Control Hardware for Marquette Prosthesis II	64
5.2.3 Selection of Sensor Hardware for Marquette Prosthesis II	65
5.2.4 Marquette Prosthesis II Hardware compared to Marquette Prosthesis Hardware	66
CHAPTER 6 Bench Testing	71
6.1 Stiffness Control Testing	71
6.2 Bench Testing	72
6.2.1 Hanging Hand Test Procedure	72
6.2.2 Hanging Hand Test Results	74
6.2.3 Force Hand Test Procedure	76
6.2.4 Force Hand Test Results	79
CHAPTER 7 Conclusion and Future Work	88
7.1 Conclusions	88

7.2	Future Work	91
7.2.1	Implementation of ANN	92
7.2.2	Redesign of the Four-Bar Mechanism	92
7.2.3	Reselection of Linear Spiral Torsional Spring	93
7.2.4	Reselection of Motor	94
7.2.5	Higher Level Controller	95
	References	96
	APPENDIX A Four Bar Kinematics	101
	APPENDIX B Four Bar Quasi-statics	105
	APPENDIX C Simulink Model	110
	APPENDIX D Control Block Diagram	116
	APPENDIX E Calculating Ankle Moment from Force Plate and Kinematic Data	119

LIST OF TABLES

3.1	Final Parameters for Tuned Controller	46
5.1	Mass of Components for Marquette Prosthesis	68
5.2	Mass of Components for Marquette Prosthesis II	68
5.3	Size of Components for Marquette Prosthesis	70
5.4	Size of Components for Marquette Prosthesis II	70

LIST OF FIGURES

2.1	Block Diagram of Myoelectric Controlled Prosthesis [13]	11
2.2	Block Diagram of UNB Control System [13]	17
2.3	Schematic of the BioM Ankle-Foot Prosthesis [4]	20
2.4	BioM Controller Architecture [4]	22
2.5	Computer System of BioM [4]	25
2.6	Robotic Tendon Model: Motor and Spring in Series [23]	27
2.7	Two Degrees-of-Freedom SPARKy Model [21]	27
2.8	CAD Model of SPARKy 3 [22]	28
2.9	Tibia Angle Profile vs Human Gait. Each curve represents a different stride length [22]	29
2.10	SPARKy 3 Control Hardware [22]	30
2.11	Slider Crank Configuration [41]	32
2.12	Knee and Ankle Prosthesis with Major Components [41]	32
2.13	Powered Ankle and Knee Prosthesis Control Structure [43]	33
2.14	Embedded System Framework [41]	34
3.1	Model of the Marquette Prosthesis II	39
3.2	Sketch of the Four-Bar Mechanism	40
3.3	Published Position Versus Time According to Winter [46]	44

3.4	Desired Ankle Angle versus Actual Ankle Angle	46
3.5	Desired Ankle Torque versus Actual Ankle Torque with Original Spring .	47
3.6	Effect of Spring Constant on Motor Torque	48
3.7	Required Speed Torque of Motor Compared to Motor Speed Torque Curve	49
4.1	Model of Marquette Prosthesis with Spring in Parallel with the Motor Adopted from [40]	52
4.2	Model of Marquette Prosthesis II with Spring in Series with the Motor .	52
4.3	Mount that Connects the Spiral Torsion Spring to Link 0	53
4.4	Redesign of Link 3	54
4.5	Linear Spiral Torsional Spring	55
4.6	Side View of Assembly of Link 3 and Linear Spiral Torsional Spring . . .	55
4.7	ANSYS Deformation Results	57
4.8	ANSYS Stress Results	57
5.1	Block Diagram Representation of Control System	59
5.2	Massachusetts Institute of Technology Hardware Flow	62
5.3	Arizona State University Hardware Flow	62
5.4	Vanderbilt University Hardware Flow	63
5.5	Marquette Prosthesis Hardware Flow	63
5.6	Marquette Prosthesis II Hardware Flow	66
6.1	Block Diagram of Control System for Testing (Identical to Fig. 5.1) . . .	72

6.2	Side View of Marquette Prosthesis II with Hardware Attached	73
6.3	Back View of Marquette Prosthesis II with Hardware Attached	74
6.4	Motor Position Results, Hanging Hand Test	75
6.5	Ankle Angle Results, Hanging Hand Test	75
6.6	Force Hand Test, Prosthesis at Heel Strike	77
6.7	Force Hand Test, Prosthesis at Flat Foot	78
6.8	Force Hand Test, Prosthesis at Plantar Flexion	78
6.9	Published Ankle Reaction Force in x-direction [46]	79
6.10	Published Ankle Reaction Force in y-direction [46]	80
6.11	Reaction force in x-direction - Test 1	82
6.12	Reaction force in x-direction - Test 2	82
6.13	Reaction force in x-direction - Test 3	83
6.14	Reaction force in y-direction - Test 1	83
6.15	Reaction force in y-direction - Test 2	84
6.16	Reaction force in y-direction - Test 3	84
6.17	Motor Position Results, Force Hand Test	86
A.1	Sketch of the Four-Bar Mechanism (Identical to Fig. 3.2)	102
B.1	Sketch of the Four-Bar Mechanism (Identical to Fig. 3.2)	106
B.2	Free Body Diagram of Link 1	106
B.3	Free Body Diagram of Link 2	107

B.4	Free Body Diagram of Link 3	107
C.1	Simulink Model of System	111
C.2	Simscape Multibody Simulation	112
C.3	Current Angle to Motor Position Subsystem	114
D.1	Simulink Control Block Diagram for Stiffness Control	117
E.1	Free-body diagram of foot during weight bearing with ground reaction forces and example values [46]	119

CHAPTER 1

Introduction

1.1 Motivation

Lower limb amputations alter an individual's life, both mentally and physically. Unfortunately, the number of amputations is continuously increasing. Complications with diabetes, war, and accidents (e.g., car accidents, motorcycle accidents, etc.) are the leading causes of lower limb amputation. Due to current trends, it is presumed that each one of these will rise in the coming years.

1.2 Fundamental Causes of Transtibial Amputation

About 60% of non-traumatic lower-limb amputations among people aged 20 years or older occur in those with diagnosed diabetes [3]. Unfortunately, due to the obesity epidemic in the United States, it is presumed that there will be a steady, if not increasing, rate of individuals with Type 2 diabetes. Since the early 1960s, the prevalence of obesity among adults has more than doubled, increasing from 13.4 to 35.7 percent in United States adults age 20 and older [33]. Obesity prevalence remained mostly stable from 1999 to 2010, but still has slightly increased [33]. The

overweight and obese population accounts for more than 90% of individuals with Type 2 diabetes [17]. Those with Type 2 diabetes have a significant chance of losing a lower limb; in the U.S., the rate of amputees of the diabetic population was 3.3% from 2005-2009 [16]. In 2010, about 73,000 non-traumatic lower-limb amputations were performed in adults aged 20 years or older with diagnosed diabetes [3]. Considering diabetes accounts for more than half of the nontraumatic lower-limb amputations, the number of lower-limb amputees tends to continue to increase.

War is gruesome, and many times inevitable, and is the second leading cause of lower limb amputations. Through April 2011, 2.4% of all soldiers Wounded in Action (WIA) in Iraq were amputees and 3.2% of all WIA in Afghanistan were amputees [19]. These are twice the percentage of both WWI and WWII. Furthermore, in 2005, limb loss secondary to trauma accounted for 45% of all amputations [49]. These men and women live a much different life than those with lower-limb amputations due to diabetes. Troops go from a physically and mentally strong lifestyle to losing a limb and struggling with daily tasks. Unfortunately, war appears to never cease to exist, thus keeping the number of lower limb amputees due to war steady and requiring an improved design to current prostheses to return the WIA back to fully functional lives.

Aside from war and diabetes, automobile accidents are a leading cause of amputations each year. There are millions of accidents each year, with 2.35 million

individuals being injured or disabled in the United States alone [7]. In a study examining the epidemiology and outcomes of post-traumatic upper extremity amputations (UEA) and lower extremity amputations (LEA), motor collisions accounted for 51% of the amputation group studied [8]. Considering 87% of the United States driving-age population has a license and the number of drivers has increased from 112 million in 1970 to 210 million in 2009 [9], automobile accidents, including those that lead to amputations, are a reality.

1.3 Development of Lower Limb Prostheses

Prostheses have been developed over the years to restore natural human function. Unfortunately, current prostheses are unable to meet the needs of lower limb amputees. Prostheses cannot physically restore amputees back to a normal gait, particularly when in transition between different gait tasks. Many amputees with lower limb prostheses must swing their leg outside of the sagittal plane to avoid dragging their prostheses on the ground or tripping over their prostheses during the swing phase. Furthermore, prostheses do not supply the full, natural energy that one's leg can supply, making amputees expend excessive energy compared to a non-amputee. Ultimately, prostheses have failed to naturally get amputees back to their normal lives with two fully functional legs.

1.3.1 Passive Prostheses

Current lower limb prostheses can be classified into two categories: passive and active. A passive prosthesis is a prosthesis that uses passive elements, i.e., springs and dampers, to propel the amputee forward in an attempt to restore natural gait. Passive lower limb prostheses help restore some of amputee's walking ability; however, they do not restore full, natural walking. Whether it's walking on level ground, climbing stairs or walking on a slight incline, users with passive prostheses expend unnecessary energy to avoid unsteady gait or walking with difficulty. Walking abnormalities cause much discomfort for amputees, but passive prostheses are a cost-effective solution that allow amputees to make use of their residual limb.

1.3.2 Active Prostheses

Due to the limitations of passive prostheses, active (i.e., powered) lower limb prostheses have been developed over the years. Active prostheses restore the walking ability of amputees by incorporating an actuator in their design to drive the prostheses. By having an actuator drive the prosthesis, the actuator fulfills the amputee's need for energy expenditure and aids the amputee to move more naturally. The actuator makes up for the lack of muscles and attempts to mimic the sound limb ankle motion.

The control systems on the active prostheses use a variety of means to mimic normal ankle motion. Current active prosthetic controllers typically use Finite State Control (FSC) [40] to identify what state of gait an amputee is in. These controllers typically break gait into phases based upon heel strike, flat foot, push off and leg swing. These distinct gait phases result in current active prostheses struggling when the user is in transition to or from different types of gait, i.e., flat ground to climbing stairs, walking to running, etc. The controller does not anticipate an amputee's next step to be any different than the previous step. Due to this type of control, it cannot adjust to a different terrain until the amputee has taken one or two steps on the new terrain or with an external signal from the amputee. The controllers do not actively sense user intent continuously to drive their prostheses continuously.

1.4 Development of Artificial Neural Network

Electromyography (EMG) is widely used to access the physiological processes that cause the muscle to generate force and produce movements [28]. However, it is extremely difficult to get a clean signal to be used for controls. Over recent years, many have succeeded in providing a continuous, high quality signal from EMG activity that can be sent to a controller on the state of a prosthesis to be used for control [1]. This drove Farmer et al. [14] to investigate the contribution EMG activity can make to controlling active, lower limb prostheses. Farmer et al. came to

the conclusion that EMG activity can predict intervals up to 150 ms with less than 6° of error [14], which leads to the goal of creating an Artificial Neural Network (ANN) for this predictive capability. At the time of this research, the ANN was not fully developed for implementation.

1.5 Motivation for this Research

Previous work by Sun [40] developed a control method for the Marquette Prosthesis; however, it needed to be improved. The control algorithm for the first generation was an FSC and mimicked the gait profile presented in Winters [46]. As every amputee has a different stride, the FSC was unable to compensate for gait profiles that vary from this profile. It was unable to detect changes in terrain and make appropriate adjustments like other active prostheses. The inability to predict a user's gait variance and terrain variances can be improved by using the ANN developed by Farmer et al. as discussed previously [14].

There are also issues with the compactness of the hardware used to control the Marquette Prosthesis. The control system components were laid out on a backpack that weighs about 10 lbs. Other active prostheses have been successful in miniaturizing many of the electrical components by mounting them around the active prosthesis, the residual limb or on a belt fitted around the amputee. Doing so makes these mounting systems much lighter than a backpack.

The purpose of this research was to use published data [46] in place of the ANN as described in [14] to develop a predictive stiffness controller to control an active transtibial prosthesis continuously. This research utilized published data to discern the desired stiffness of the ankle as a function of time within a set prediction window. Through the kinematics and quasi-statics [8,32], the motor position was controlled such that the desired stiffness of the ankle was met. Once the ANN is fully developed, the continuous stiffness controller will allow an amputee to avoid awkward transitions in gait. Furthermore, a more compact control system that can be contained around the residual limb was created to make the Marquette Prosthesis II more user friendly.

1.6 Scope

In Chapter 2, the success of upper limb prostheses integrating myoelectric control [13] is first described. Myoelectric control of upper limbs has been implemented since the 1970s, proving that controlling via these signals is feasible. The mechanical systems, control systems and control hardware of other active lower limb prostheses are then outlined. In Chapter 3, a dynamic model built in MATLAB and Simulink to simulate the dynamic process, ensure stiffness control is feasible with the limitations of the Marquette Prosthesis and adjust the PI gains of the controller is presented. In Chapter 4, the mechanical system design changes

required to implement stiffness control are described in detail. In Chapter 5, the implementation and architecture of the control system are discussed. The theory behind how stiffness control was implemented on the Marquette Prosthesis II along with the hardware, software, sensors and power supplies required to implement stiffness control are described. In Chapter 6, the methods and results of a hanging hand test to verify the dynamic model completed in Chapter 3 are presented. The methods and results of a force hand test that was developed in an attempt to validate stiffness control are also presented in this chapter. In Chapter 7, conclusions are made and potential future work to enhance the Marquette Prosthesis II is explored.

1.7 Research Contributions

There are several contributions from the research completed for this thesis. First, a dynamic model was successfully created that models the Marquette Prosthesis II, placing the linear spiral torsional spring in series between the motor and four-bar mechanism. This model provides insight on how the motor parameters, link lengths and linear spiral torsional spring effect the performance of the prosthesis. Second, the mechanical system was altered to make stiffness control possible. Third, the hardware weight was reduced by 39% and the footprint was reduced by 65%. Fourth, stiffness control was implemented on the Marquette

Prosthesis II and the prosthesis proved to be stable when interacting with the ground. Lastly, the groundwork to implement the ANN and test the Marquette Prosthesis II on an amputee was completed such that future researchers can fully optimize stiffness control with the Marquette Prosthesis II to make amputees' lives normal again.

CHAPTER 2

Literature Review

From the first attempt of a powered ankle-foot prosthesis by Klute [26] to the development of the first thought-controlled active leg prosthesis in 2013 [4], many academic institutions as well as companies have made major strides in the development of active lower limb prostheses. The powered prosthesis industry has expanded from research and bench testing completed in research labs to making active prostheses available to the public. The work in this thesis was focused around enhancing the mechanical and control systems of the Marquette Prosthesis. This being the case, the mechanical and control systems of current active upper and lower limb prostheses were investigated.

2.1 Myoelectric Control of Powered Upper Limb Prostheses

In the late 1940s, it was proposed that the myoelectric signals (MES) detected in the residual limb of upper limb amputees could be used for the control of a mechanical hand [37]. However, it wasn't until the late 1960s and 1970s when myoelectric control of powered, upper limb prostheses became an intensive goal for researchers. Researchers realized that in order to get a powered prosthesis to

perform similar to a physical arm, elbow and hand, the prosthesis should interface with the neuromuscular system and mimic the closed-loop control non-amputees possess [13]. In an active upper limb myoelectric controlled prosthesis, motor command signals from the central nervous system are used in the forward path. These neuromuscular signals are then utilized to move the prosthesis as the user desires. After completing a desired move, visual, auditory and residual limb vibration and pressure are used in the feedback signal. These signals feedback to the neuromuscular system to complete the feedback loop. Fig. 2.1 shows a block diagram of the closed loop control of powered, upper limb prosthesis using myoelectric control [13]. The shaded blocks in Fig. 2.1 indicate problematic areas in the control system that must be perfected for a more synergetic relationship between an amputee and the prosthesis.

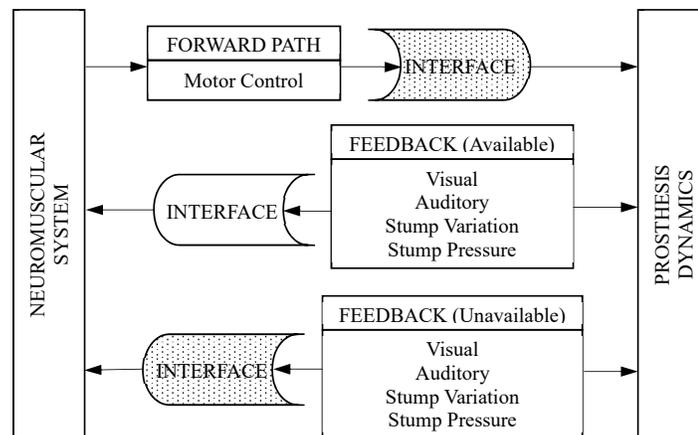


Figure 2.1: Block Diagram of Myoelectric Controlled Prosthesis [13]

Many have thought improving the quality of feedback to the central nervous system would be desirable for a prosthesis to be successful. It has been attempted to use proprioceptive and sensory feedback but it has not been accepted due to difficulties in developing a proper interface [38]. However, experimental investigations demonstrated that this feedback is not necessary for the control of upper limb prostheses [35]. Considering the difficulties in man-machine interface, simple myoelectric control is used to actuate a single device such as a hand or elbow. Using one or two myoelectric channels, the system receives the MES and derives the control signal either as a measure of its amplitude [12] or of the rate of change of its amplitude [10]. Despite the challenges presented in the interfaces, there have been several institutions to successfully implement multifunction prosthesis control.

2.1.1 Weighting Synergistic Muscle Movements - Temple University

Engineers at Temple University with the Moss Rehabilitation Hospital in Philadelphia [15] were one of the first to attempt multifunction control of upper limb prostheses. The team looked to implement control based on myoelectric statistical pattern recognition techniques. They recognized that in able-bodied individuals, elbow movements were completed by several muscle groups acting together. Because movements are a synergistic muscle movement and not a

one-muscle-for-one-muscle control, the goal of their work was to characterize muscle synergies of the arm using pattern recognition.

Their initial work amplified and quantified six channels of MES [15]. The myoelectric activity was monitored during different elbow movements. This activity was analyzed using multivariate pattern recognition techniques in an attempt to optimally distinguish between different desired movements. With normally-limbed subjects who received moderate training producing isometric contractions, the authors reported 92% accuracy for elbow flexion and 97% accuracy for forearm pronation and supination [13].

Further work considered multiple-axis control for an above-elbow amputee where 14 myoelectric sites were used instead of six. After recording data from able-bodied individuals, patterns were classified according to the parameters of their distribution. From these patterns, weighting coefficients were computed to provide maximum separation of the classes. Weighting coefficients were computed for each of the six desired motions and applied to the magnitude of the MES. From these weighing coefficients, it was determined that each movement is distinguishable using a summation threshold. However, the results were not as spectacular as before. Although elbow flexion had a performance accuracy of 97%, humeral rotation had an accuracy of 43% [47]. Unfortunately, the project ultimately demonstrated that

the prosthesis was unable to be controlled properly because the control approach needed a sensory feedback system to be truly effective.

2.1.2 Endpoint Control - University of California, Los Angeles

Engineers at the University of California, Los Angeles (UCLA) sought to refine multifunction control of upper-limb prostheses. Similarly to the work developed at Temple University, the group at UCLA looked to implement control based on myoelectric statistical pattern recognition techniques. This work originated from studies done by one of the engineers' previous work with the U.S. Air Force [39] mapping synergistic muscle activity during select arm movements. Instead of mimicking muscle synergies with multiple axes working in unison like the research team at Temple, the UCLA team saw the need for endpoint-based control.

The objective of endpoint control is to relieve the operator of part of the control responsibility. Additionally, because human beings cognitive task space tends to be in Cartesian endpoint coordinates, this type of control provides a more natural control interface. The goal of UCLA's research was to produce a control algorithm that would allow the user to specify direction and speed of the endpoint along some natural set of coordinates [13]. The team concluded there is a basic limitation on the ability of a user to supply the control information required for adequate performance and no attempt was made to use the MES as the input to the

endpoint control. Instead, a three degree of freedom joystick input was mapped to the appropriate three degree of freedom joint angle vector needed to emulate upper-limb positioning. The control system used the same pattern recognition approach as Temple University, however they looked to provide adaptive mapping to optimize the pattern recognition system for individual users myoelectric patterns. In the end, the team claimed their results were excellent as their prosthesis completed the proper movement in under 0.5 seconds with two or less erroneous decisions made [48].

2.1.3 Transient Myoelectric Control - University of New Brunswick in Canada

The University of New Brunswick in Canada (UNB) control approach used patterns in the instantaneous MES to define a signature for a particular limb function. More specifically, they looked to implement control based on pattern recognition using the transient MES [13]. The research team at UNB recognized that the complexity of the MES is the root cause for issues with these control systems. Instead of dismissing the idea of using the MES as the input, as the research team at UCLA did in concluding there is not enough control information provided by the user, the team at UNB decided to explore the application of MES as a control input.

After investigation, the team at UNB came to the conclusion that most myoelectric control systems use the steady state myoelectric signals as the control input. The issue with using the steady state signals is that they have little temporal structure due to the active modification of recruitment and firing patterns needed to sustain a contraction [11]. Instead, Hudgins, one of the researchers on the team, investigated the information in the transient burst of myoelectric activity accompanying the onset of sudden muscular effort. In doing so, Hudgins discovered that the patterns of transient MES exhibit differences in temporal waveforms [24]. From this work, the research group at UNB designed a control system based on the transient response instead of mapping synergistic muscle activity during select arm movements.

The UNB control system used a multilayer perceptron (MLP) ANN to classify values of the time-domain feature set extracted from the single channel MES. Signals from the biceps and triceps were measured, allowing the controller to identify four types of muscle contractions. The block diagram of UNB's control system can be seen in Fig. 2.2. Utilizing an ANN proves three major benefits: multifunction control from a single site, the control signals can be derived from natural contractions, and conscious effort of the user is minimized [13]. The control system is trained on the distinct MES patterns of the amputee. This information is

then downloaded into the control system which can be used to either control a three degree of freedom prosthetic arm or a virtual arm simulator on a computer.

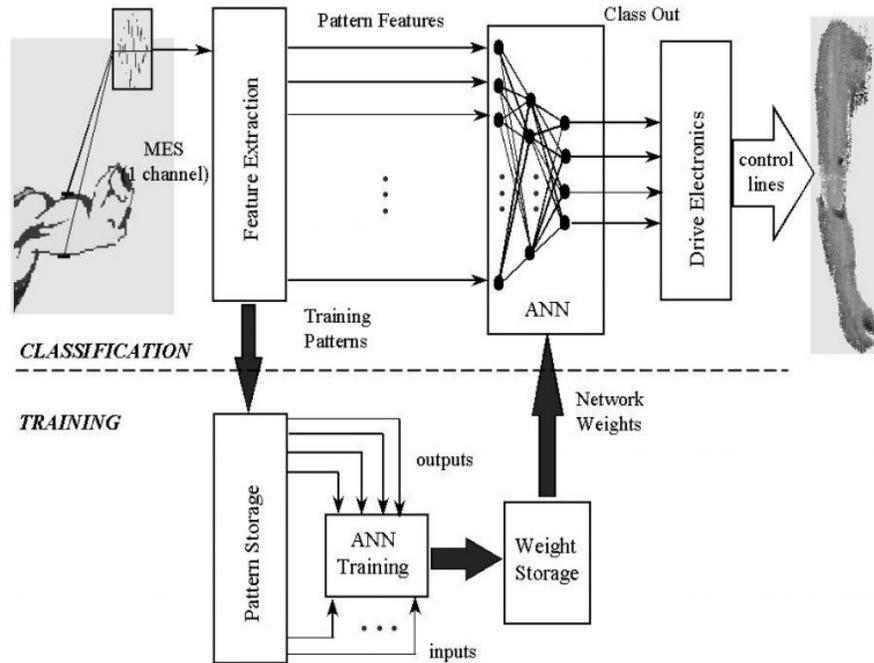


Figure 2.2: Block Diagram of UNB Control System [13]

2.2 Mechanical and Control Systems of Lower Limb Prostheses

The three universities discussed above looked to implement control based on myoelectric statistical pattern recognition techniques in different ways. Temple University characterized muscle synergies of the arm using MES pattern recognition. UCLA produced a control algorithm that would allow the user to specify direction and speed of the endpoint along some natural set of coordinates. UCLA abandoned using MES as the inputs and concluded that there is a basic

limitation on the ability of a user to supply the control information required for adequate performance. UNB developed a control system based on the transient response instead of mapping synergistic muscle activity during select arm movements. Using an ANN, they classified values of the time-domain feature set extracted from the single channel MES.

Despite the success of utilizing EMG signals measured from the residual limb as control commands for active upper limb prostheses, these signals have not been incorporated in the control of active lower limb prostheses. Members of Herr's laboratory of MIT proposed two EMG-controllers for an active ankle-foot prosthesis, one using a biomimetic muscle model approach and the other using a neural network approach. They concluded that both controllers demonstrate the ability to predict desired ankle movement patterns qualitatively, with the biomimetic controller having a smoother movement pattern than the neural network control [5]. Despite these conclusions, these methods for implementing EMG based controllers were not exploited in active lower limb prostheses. The mechanical design, control system and hardware of three academic active lower limb prostheses along with a commercially available passive prosthesis are explored below.

2.2.1 Össur Proprio Foot with Evo

The Össur Proprio Foot with Evo is an active prosthetic device for low to moderately active transtibial amputees. Motor-powered ankle motion increases ground clearance and reduces the risk of tripping and falling. Due to the motor-powered motion, users are able to navigate different kinds of terrain in a natural and secure way. However, the Proprio Foot is not a fully active transtibial prosthesis. The active components of the prosthesis engage during the swing phase until the ankle reaches a state of maximum dorsiflexion so that the foot has enough clearance. Unlike the Marquette Prosthesis, as well as other fully active prostheses, the joint is locked during the stance phase of swing, not providing any external assistance to the user, thus making it a passive transtibial prosthesis during stance.

2.2.2 BioM - MIT Powered Ankle-Foot Prosthesis

The BioM, first designed in Herr's laboratory, is an active prosthetic designed to improve amputee walking by decreasing the users metabolic cost of transport. The prosthesis decreased the amputee's metabolic cost of transport by developing a control system that mimics normal human ambulator stance phase dynamics. The BioM exploits series and parallel elasticity with an actuator to match human-ankle motion with a control system that allows the prosthesis to mimic the target stance phase behavior.

Mechanical Design

The overall mechanical design of the BioM utilizes a parallel spring with a force-controllable actuator with series elasticity to actuate an ankle-foot mechanism. There are five main mechanical elements in the system as shown in Fig. 2.3: a high power output D.C. motor, a transmission, a series spring, a unidirectional parallel spring, and a carbon composite leaf spring prosthetic foot. The first three components are combined to form a force-controllable actuator, called Series-Elastic Actuator (SEA). The SEA provides force control by regulating the extent to which the series spring is compressed. Using a linear potentiometer, the BioM can obtain the force applied to the load by measuring the deflection of the series spring.

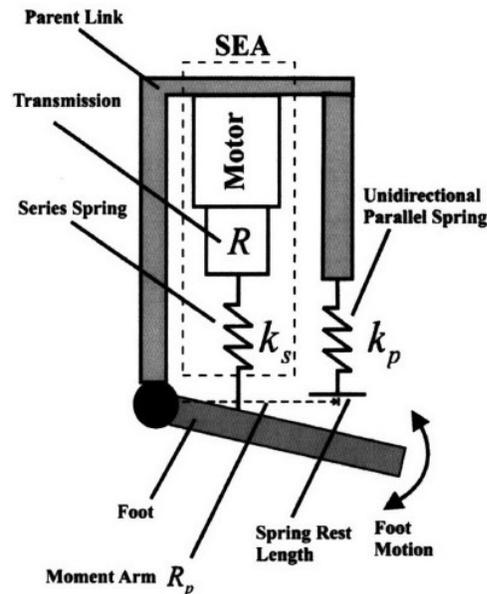


Figure 2.3: Schematic of the BioM Ankle-Foot Prosthesis [4]

The BioM requires a high mechanical power output as well as a large peak torque. Due to the demanding output torque and power to provide the active push-off during Powered Plantarflexion (PP), the parallel spring shares the payload with the force-controllable actuator. The reduction of the payload experienced by the SEA also allows for a smaller transmission ratio to be used. The elastic leaf spring foot is used to emulate the function of a human foot that provides shock absorption during foot strike, energy storage during the early stance period, and energy return in the late stance period [4].

Control System

The control system of the BioM mimics a normal human beings stance phase behavior. The control system consists of an FSC to replicate the target stance behavior and three types of low-level servo controllers to support basic ankle behavior. The overall architecture of the control system is shown in Fig. 2.4.

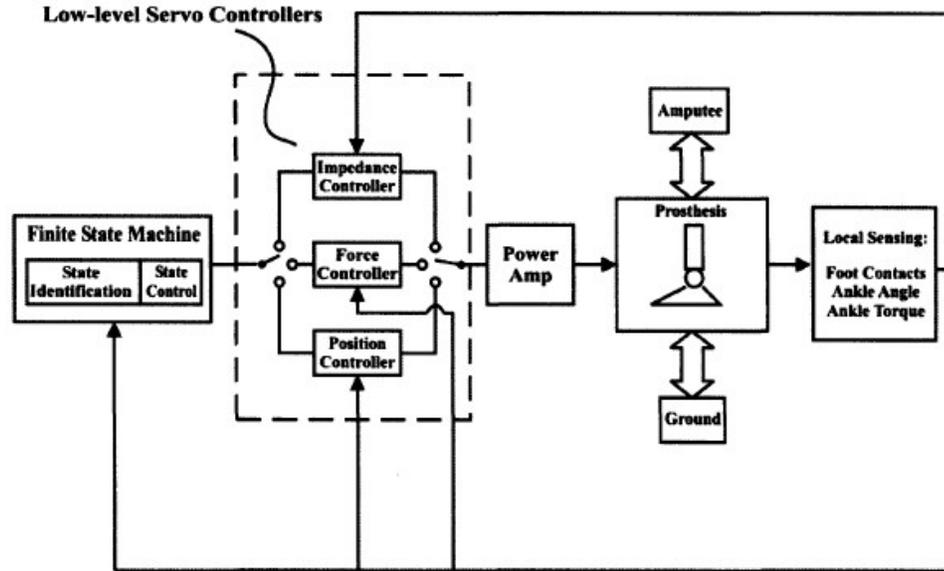


Figure 2.4: BioM Controller Architecture [4]

The FSC was implemented to replicate the target ankle behavior. It is comprised of two parts: stance phase control and swing phase control. For stance phase control, three states were designed to mimic natural walking [4]. The first state, Controlled Plantarflexion (CP), begins at heel strike and ends at mid-stance. The second state, Controlled Dorsiflexion (CD), begins where CP left off and ends right before Powered Plantar (PP) or toe-off. The final state, PP, begins only if the measured total ankle torque is larger than the predefined torque threshold. If this is not satisfied, the prosthesis remains in CD until the foot is off the ground. For swing phase control, three states were also designed to ensure foot clearance and preparation for the next gait cycle. The first state, Swing Phase 1 (SW1), begins at

toe-off and ends after a specified time. The second state, Swing Phase 2 (SW2), begins right after SW1 and finishes when the foot reaches zero degree. The final state, Swing Phase 3 (SW3), runs for a specified time period determined experimentally to ensure the system is ready for heel-strike [6].

As shown in Fig 2.4, the three types of low-level servo controllers are an impedance controller, a force controller and a position controller. The torque controller was designed to provide the offset torque and facilitate the stiffness modulation. This controller uses force feedback, estimated from the series spring deflection, to control the output joint torque of the SEA. The force controller utilizes a PD controller to regulate the input voltage to the motor amplifier and allow the prosthesis to bounce back if it hits a hard boundary due to large impact force seen in the spring.

The impedance controller modulates the joint stiffness of the SEA. Impedance control regulates a dynamic relationship between manipulator variables such as the ratio of position and force rather than just control these variables alone. In the BioM impedance controller, a dynamic relationship is established between the output torque and the ankle joint angle. The impedance controller uses the ankle angle, fed back via a position encoder, to increase the output joint impedance. Due to the complexity of the ankle joint, it is crucial to take into account major components of impedance, such as the effective mass, inertia, damping and stiffness,

for specific tasks and movements to be completed as desired [30]. In order to avoid the effect of intrinsic impedance, the torque controller was incorporated into the impedance controller. The actual output consists of desired output impedance due to the controller plus that due to the mechanism.

A standard PD controller was implemented as a position controller. Its purpose is to control the equilibrium position of the foot during swing phase. This control is similar to that of the Össur Proprio Foot with Evo and ensures the foot clears the ground to avoid tripping or falling. With PD control, however, the BioM is able to prepare for Controlled Plantarflexion (CP) and the next stance phase.

System Hardware

The BioM electronic system is shown in Fig. 2.5. It contains an onboard computer, PC/104, with a data acquisition card, power supply and motor amplifiers. The system is powered by a 48V, 4000mAh Li-Polymer battery pack. A custom breakout board interfaced the sensors to the D/A board on the PC/104 as well as provided power to the signal conditioning boards. The PC used was a MSMP3XEG PC/104. The PC/104 systems are intended for applications where a small, rugged computer system is required [34]. It was fitted with a PENTIUM III 700 MHz processor. A PC/104 multifunctional I/O Board from Sensory Co. was connected to the PC/104, containing 8 analog inputs, 2 analog outputs and 4 quadrature encoder

counters. The system runs in real-time using the MATLAB xPC real-time kernel and Simulink. The Simulink model was compiled on the host PC using MATLAB RealTime Workshop and a C++ compiler created executable code. The system uses a linear potentiometer to measure the displacement of the series springs, a quadrature encoder to measure the joint angle of the prosthesis ankle and six capacitive force transducers, two beneath the heel and four beneath the forefoot.

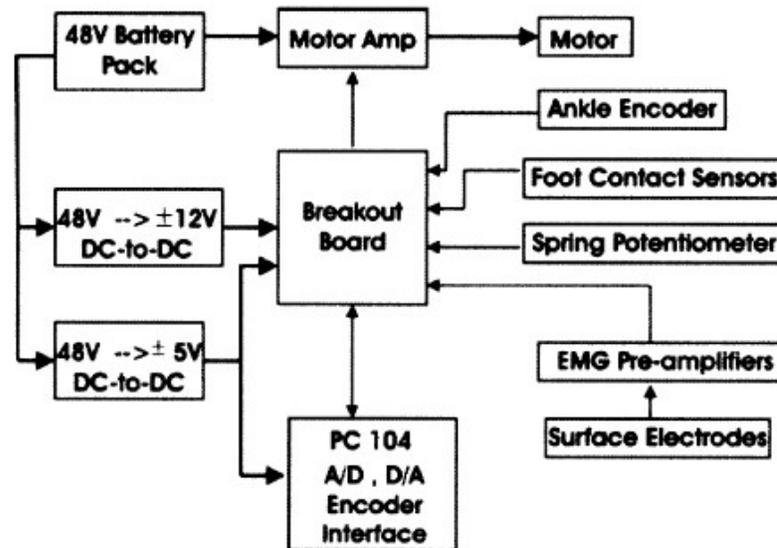


Figure 2.5: Computer System of BioM [4]

2.2.3 SPARKy 3 - ASU Powered Ankle-Foot Prosthesis

SPARKy 3, short for Spring Ankle with Regenerative Kinetics, was designed and implemented by Sugar's laboratory at Arizona State University. This active prosthesis attempts to bring full able-bodied ankle function to transtibial amputees.

The first generation, SPARKy 1 [20], used regenerative kinetics to accurately and efficiently reproduce the human gait cycle. The second generation, SPARKy 2 [7], enhanced the mechanical and electrical system of the first generation. The chief goal of the third phase of SPARKy 3 was to build a more dynamic prosthesis, i.e., one that moves in two directions instead of one, without sacrificing size, weight or performance compared to the previous generations [7].

Mechanical Design

In order to minimize the required peak power supplied by the motor, SPARKy utilizes a robotic tendon shown in Fig. 2.6 [23]. This device includes a linear actuator in series with a spring and a small, lightweight, low energy motor that is used to adjust the position of the helical spring. The robotic tendon design was developed to mimic the elastic nature of the human muscular system in minimizing the work and peak power. The simple inclusion of a spring to a linear actuator can provide energy and power savings to the prosthesis. The spring releases its stored energy to provide most of the peak power required during the powered plantar phase of gait. In the simple series model of SPARKy, the keel and the robotic tendon springs are in series. The two degrees-of-freedom model is shown in Fig. 2.7.

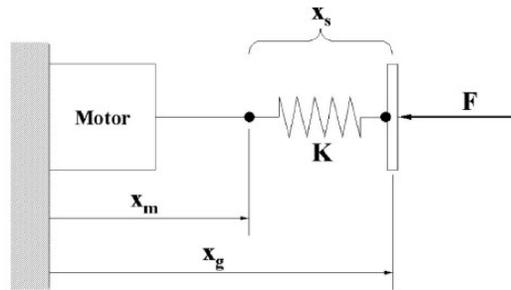


Figure 2.6: Robotic Tendon Model: Motor and Spring in Series [23]

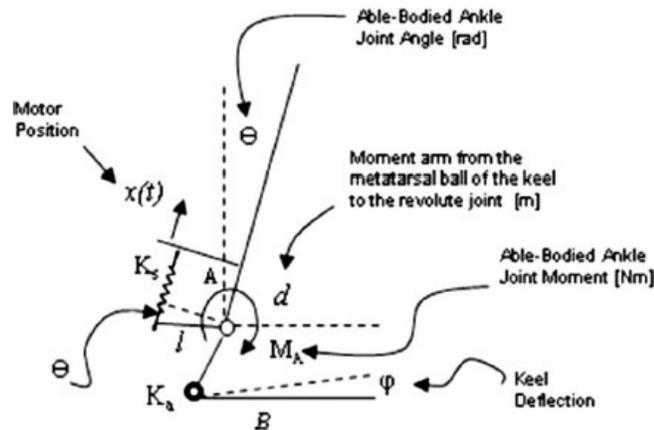


Figure 2.7: Two Degrees-of-Freedom SPARKy Model [21]

The goal of SPARKY 3 was to achieve running and jumping. Thus a second DC Powermax 30 motor had to be introduced to the system in order to increase power capacity. The SPARKY 3 CAD model is shown in Fig. 2.8. With regards to walking, the two motors work together either compressing or extending the two helical springs. For example, during powered push-off, the motors move together releasing the energy in the springs [7].

Control System

In order to control the system shown in Fig. 2.8, Arizona State uses the tibia angle profile for able bodied human gait. This relationship between tibia angle and percent gait cycle analyzed for different stride lengths. These profiles are seen in Fig. 2.9. From this relationship, it was observed that each different stride length produces an almost identical curve, only scaled by some function of stride length. They created a control system that sought to find a measurable variable to determine a mathematical relationship between the tibia angle and ankle angle. Ultimately, the tibia global angle was chosen due to this relationship and how easy it is to measure this angle by attaching a sensor to the prosthetic device [22, 23].

The angular velocity of the shank is measured using a rate gyro. The ankle

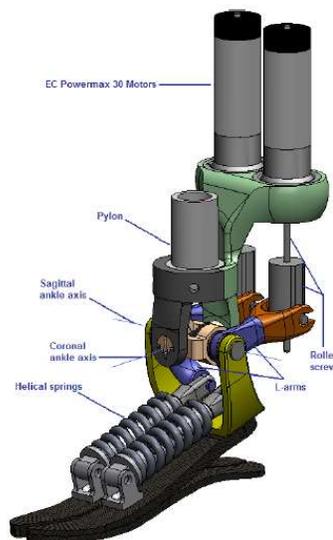


Figure 2.8: CAD Model of SPARKy 3 [22]

angle and ankle moment is calculated via published kinematic data as shown in Fig. 2.9. With the moment and angle, they are then able to calculate the nut reference for the robot [23]. The major advantage the control algorithm for ASU has is that it uses a vast amount of published data of ankle angles, moments, tibia angular velocity, etc. such that the prosthesis is adaptive to its user despite of height or weight.

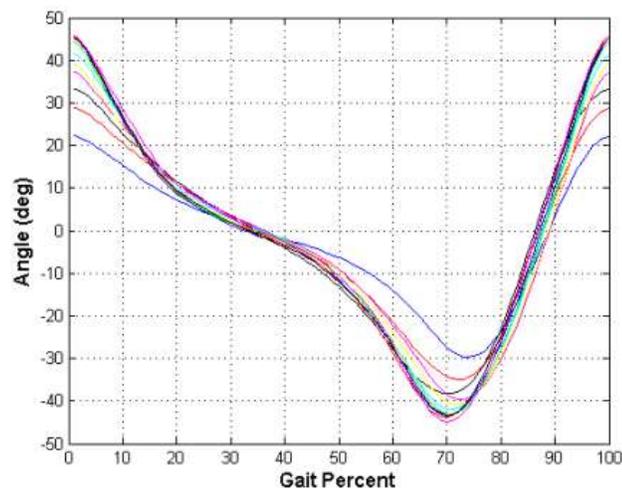


Figure 2.9: Tibia Angle Profile vs Human Gait. Each curve represents a different stride length [22]

System Hardware

The SPARKy 3 electronic system can be seen in Fig. 2.10. SPARKy 3 utilizes a simple series model with the Maxon RE 40 motor with a planetary gear box, Maxon GP42, having a 4.3:1 gear ratio. From Fig. 2.8, one can see the mechanical system utilizes two EC Powermax 30 motors, roller screws, a pylon,

L-arms and two helical springs. It is controlled in real time using Real Time Workshop and Simulink. The model is compiled onto the embedded target PC running the xPC target operating system. Advantech's 650 MHz PC/104 with 512 MB on board memory is selected to run the system. A multifunctional I/O board from Sensoray Co., Model 526, is connected to the PC/104 via ISA Bus, controls the motors with encoders as the feedback devices. There are three sensors used on the system: one encoder at the motor, another encoder at the ankle joint, and an optical switch embedded at the heel. As stated previously, the controller has a predetermined gait pattern based on able-bodied gait data and kinetic analysis expressed as a time-based function embedded in the controller [21].

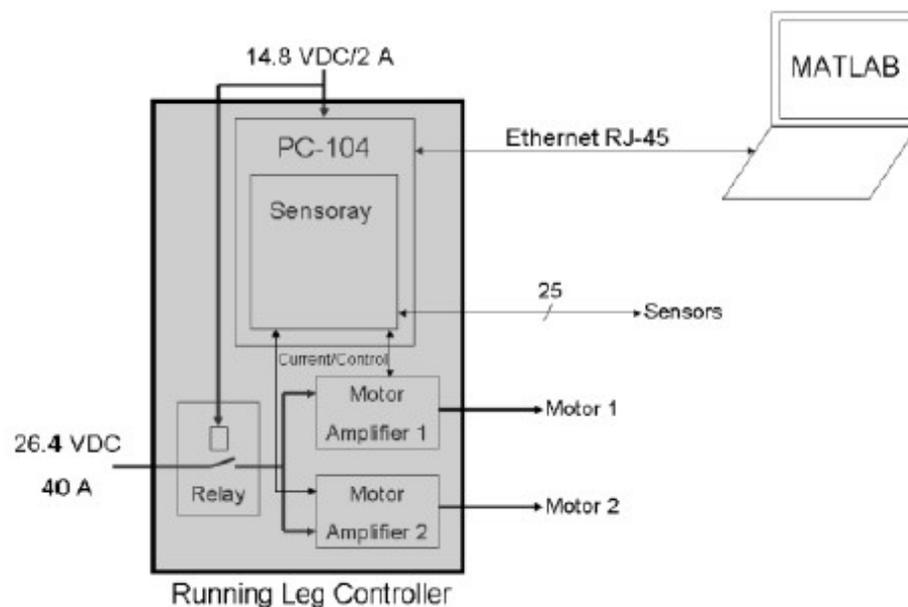


Figure 2.10: SPARKy 3 Control Hardware [22]

2.2.4 Vanderbilt University Powered Ankle and Knee Prosthesis

A transfemoral prosthesis was developed and implemented by Goldfarb's lab at Vanderbilt University. Unlike the work done by MIT, ASU and in this thesis, Vanderbilt designed a prosthesis that consists of both a powered knee and ankle. Prior to the work of Goldfarb and Sup [45], a powered knee transfemoral prostheses and powered ankle transtibial prostheses did not exist as one entity, only powered knee or powered ankle prostheses. The goal was to create a prosthesis with the capability to deliver power at the knee and ankle joints to enable the restoration of biomechanically normal locomotion [42].

Mechanical Design

The driving mechanisms of the powered ankle and knee prosthesis are two slider-crank mechanisms, as shown in Fig. 2.11 and Fig. 2.12: one located at the knee and one locating at the ankle. The first generation of the prosthesis utilized two pneumatic actuators to drive the system. However, due to the expense as well as practicality of a pneumatic actuator, the second generation looked to move from a pneumatic system to an electrical system. The second-generation device consists of a Maxon motor for each actuation unit. The motor is connected to a ball screw assembly that actuates the prosthesis. The kinematic configuration of the actuators was selected via a design optimization to minimize the volume of the actuators. The

active joint torque specifications for the prosthesis were based on the torque/angle phase space required for an average user for fast walking and stair climbing, as derived from body-mass-normalized data from Winter et al. [41,42].

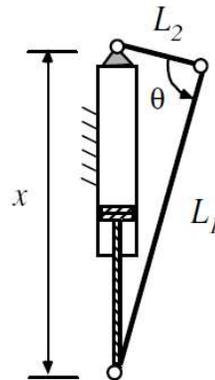


Figure 2.11: Slider Crank Configuration [41]



Figure 2.12: Knee and Ankle Prosthesis with Major Components [41]

Control System

The powered knee and ankle prosthesis has three activity modes: walking, standing and sitting. It has control algorithms for transition from one activity mode to another as well as control algorithms within each activity mode. The control structure is shown in Fig. 2.13 [43]. It consists of high-level controllers, middle-level controllers and low-level controllers. The highest level consists of three parts: the activity-mode intent recognizer, the cadence estimator and the slope estimator. The latter two estimate the slope and cadence during walking to adjust the parameters of the walking-mode controller. The activity mode recognizer recognizes one of several activity modes, whether its walking, running, sitting, standing, climbing, etc. and switches to the appropriate middle level controller.

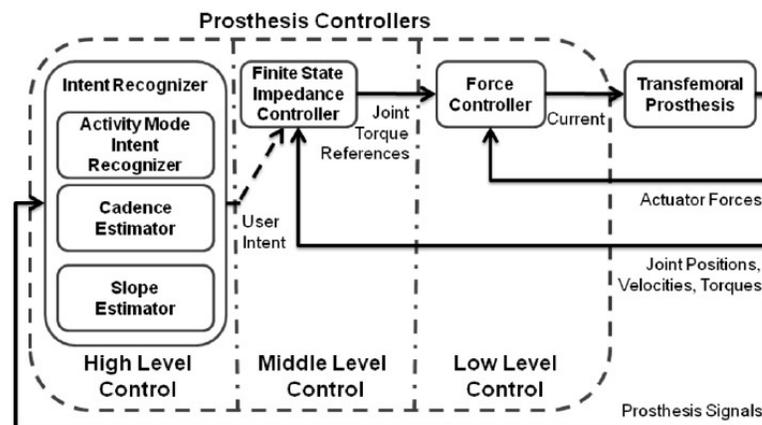


Figure 2.13: Powered Ankle and Knee Prosthesis Control Structure [43]

2.2.5 System Hardware

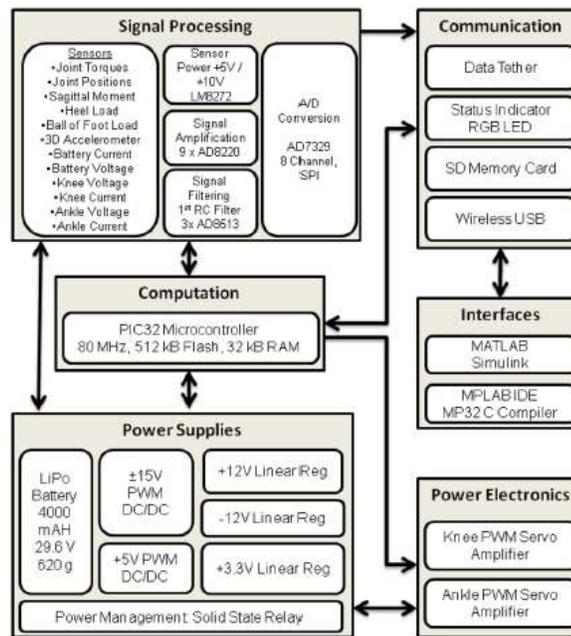


Figure 2.14: Embedded System Framework [41]

The self-contained version of the knee and ankle prosthesis includes an embedded system which can be controlled from a laptop via MATLAB Simulink or untethered using an 80MHz, 512 kB Flash, 32 kB RAM PIC32 onboard microcontroller. In the tethered version, the prosthesis is controlled via MATLAB Simulink RealTime Workshop. In the untethered version, the microcontroller performs the servo and activity controllers of the prosthesis. It is actuated with two Maxon EC30 Powermax brushless motors, powered by a lithium polymer battery [44]. The prosthesis integrates a custom load cell to measure the sagittal-socket interface moment above the knee joint, a custom foot with strain

gages to measure the ground reaction force at the heel and ball of the foot, and precision potentiometers and load cells to measure joint positions and torques, respectively [43]. The embedded system framework is shown in Fig. 2.14.

2.3 Summary

While the prostheses mentioned in this chapter look to restore natural walking of amputees, their control systems fall short when the terrain changes abruptly. These prostheses struggle when amputees transfer from flat ground to climbing a flight of stairs or going down a flight of stairs. This is because these control systems are unable to properly predict when a change in the amputees walking environment is about to occur. By using EMG information measured at the residual limb, this research looks to implement a neural network control approach to predict ankle movement, ultimately making a more natural, smooth stride for normal and changing terrains.

2.4 Marquette Prosthesis

The Marquette Prosthesis was able to meet the published ankle angle and torque when tested on human subjects. However, the Marquette Prosthesis exploited the issue with an FSC - it is not robust enough to consider more normal or abnormal situations such as a transition from walking on an incline or from walking

to running [40]. Unfortunately, due to the reactionary nature of an FSC, an FSC is not robust enough in these transition phases. The three schools discussed above, MIT, ASU and Vanderbilt, were able to get their prostheses to run and walk up inclines, however the initial step to transition to running or climbing were awkward for the amputee due to the FSC. Because the controller proposed in this thesis received the predicted stiffness of the ankle from an ANN [14] and controls the stiffness of the ankle continuously, the issue experienced by an FSC is eliminated.

CHAPTER 3

Dynamic Model and Simulation

In this chapter, the updated dynamic model of the Marquette Prosthesis II will be presented and discussed followed by the simulation results. Creating a dynamic model of any system is critical to understand how a system will behave and to assess the validity of a system. A model offers insight and saves time and money by ensuring a system performs as expected and avoids potential unseen problems. With this goal in mind, Sun created a two-phase model of the Marquette Prosthesis and simulated it using Simulink and SimMechanics before implementing its controller [40]. Unfortunately, this model is insufficient for this thesis due to changes in the mechanical design as well as changes in the type of controller implemented.

Because the Marquette Prosthesis controller in [40] was an FSC which changed controller type based on whether the amputee was in stance or swing mode, two separate models were created: one for swing and one for stance [40]. With the Marquette Prosthesis II, the controller drives the ankle continuously regardless of the stage or type of gait. The ANN will provide a desired torque and position regardless of the stage of gait [14]. Thus, the model needed only to have one stage.

As with any model, the goal of the model should drive its creation and

fidelity. There were two major outcomes of this model and simulation. The first was to determine the PI parameters for the motor position to ensure the relationship between the ankle position and torque was satisfied. This simulation provided a benchmark for the final PI parameters used in the physical control system. The second was to determine an appropriate spring constant for the linear spiral torsional spring. In order to implement stiffness control, the linear spiral torsional spring was redesigned to be in series with the four-bar mechanism. With the spring in series with the four-bar mechanism, the desired spring constant to meet the torque requirements of the ankle needed to be investigated.

3.1 Four Bar Mechanism Analysis

A model of the Marquette Prosthesis II can be seen in Fig. 3.1. In this model, the motor moves independently of the mechanism. The ANN will receive EMG activity and the current position of the ankle and output the desired position, θ , and torque of the ankle, τ_{ankle} [14]. For the dynamic model, published data was used to determine the desired position and desired torque of the ankle [46]. The angular position of the motor shaft, ψ , was determined from the ratio of the desired position of the ankle, θ , and desired torque of the ankle, τ_{ankle} . However, the motor shaft angular position, ψ , relies heavily upon the input angle of the four-bar mechanism, ϕ , and the torque provided by the spring, τ_{spring} . Therefore, the angle,

ϕ , was derived from the desired angle of the ankle, θ , using kinematic relationships.

Furthermore, the torque provided by the spring, τ_{spring} was derived from the

published torque at the ankle, τ_{ankle} , using the quasi-static relationship.

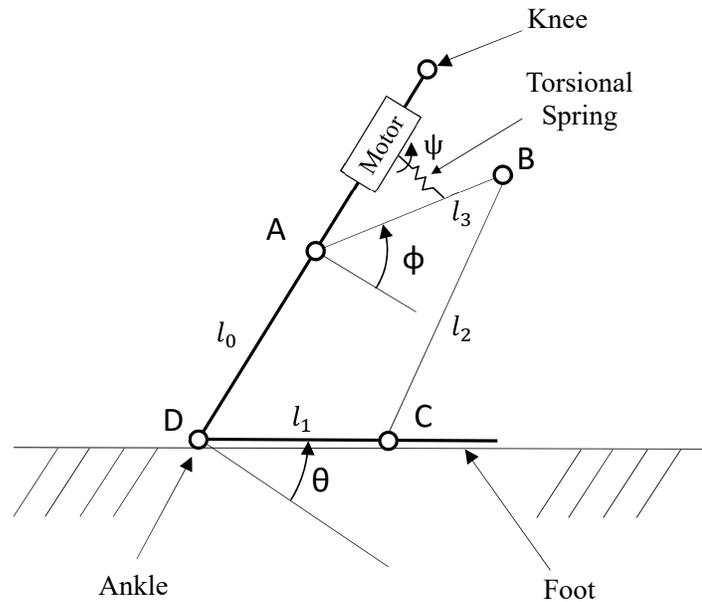


Figure 3.1: Model of the Marquette Prosthesis II

3.1.1 Kinematics

The first step in creating the model was to determine the kinematic relationship between the angles of the four-bar mechanism. A sketch of the four-bar mechanism with the angles labeled and the coordinate system defined is shown in Fig. 3.2 and is repeated in Fig. A.1 and Fig. B.1. Because the position of the motor is independent of the mechanism, a classical vector loop approach was used to determine the input angle of the four-bar mechanism, ϕ , based on the position of

the ankle, θ . As such, the derivation was standard and follows the approach found in Norton's Design of Machinery and is repeated in this thesis in Appendix A for completeness [32].¹

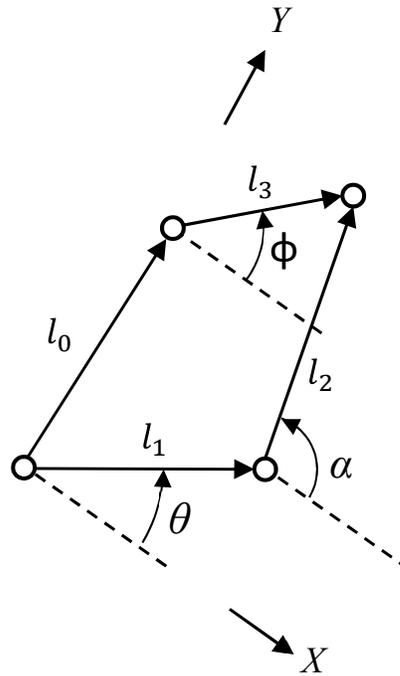


Figure 3.2: Sketch of the Four-Bar Mechanism

¹This derivation is based on applying half-angle identities with Freudenstein's equation

From Appendix A, the solution for the input angle of the four-bar mechanism, ϕ as a function of the ankle angle, θ , was found by using the quadratic equation:

$$\phi_{1,2} = 2 \arctan\left(\frac{-B \pm \sqrt{B^2 - 4AC}}{2A}\right) \quad (3.1)$$

where:

$$A = \cos \theta - K_1 - K_2 \cos \theta + K_3$$

$$B = -2 \sin \theta$$

$$C = K_1 - (K_2 + 1) \cos \theta + K_3$$

Equation 3.1 has two solutions that may be of three types: real and equal, real and unequal, or complex conjugates. If the solutions are real and equal, then the mechanism is in a singular configuration [1]. For this work, the singular position is to be avoided in the workspace. If the solutions are complex, then the desired angle is impossible to reach. In other words, the link lengths are not capable of being connected at that angle. If the solutions are real and unequal, then these results are two different solutions for the two assembly modes. Thus, the result from Equation 3.1 should be real and unequal. From these two possibilities, the appropriate assembly solution was selected depending on whether or not the configuration of the linkages is closed or open [32].

From Equation 3.1, the desired input angle of the four-bar mechanism, ϕ , was determined for a given ankle angle, θ . Because the spring was moved in series with the motor and the four-bar mechanism, a relationship was established between

the position of the motor, ψ , input angle of the four-bar mechanism, ϕ , spring constant, K_{spring} , and the torque provided by the spring, τ_{spring} . This relationship is shown in Equation 3.2. Therefore, before controlling the position of the motor, ψ , the torque provided by the spring was determined using quasi-static relationships.

$$\psi = \frac{\tau_{spring}}{K_{spring}} + \phi \quad (3.2)$$

3.1.2 Quasi-statics

The desired torque provided by the spring was determined using quasi-static relationships based on the desired output torque at the ankle and current pose of the prosthesis. Quasi-statics is the ability to analyze a dynamic system statically as the inertial force is very small compared to the torque exhibited [1]. In other words, the dynamic forces are small compared to the static forces. According to Bergelin [8] and further shown by Malak [29], the dynamic forces and gravitational forces in this application are small and the system can be modeled quasi-statically. With this approach, a static analysis was performed at different poses of the mechanism. The quasi-static analysis can be found in Appendix B.

The set of 13 equations and 13 unknowns found in Appendix B fully describe the statics of the mechanism when used in conjunction with the kinematic equations, Equations A.1-A.9. They were solved simultaneously to calculate the

desired torque provided by the spring, τ_{spring} . The desired position of the motor, ψ , was determined using the relationship shown in Equation B.13.

The software must compute the desired position of the motor based on the desired position of the ankle in real time. Due to this, the simplified model shown via the quasi-statics was utilized in the controller to quickly calculate the desired angle.

3.2 Simulink Model

With the dynamic relationships established, the full dynamics of the plant were modeled and simulated. This model, while not applicable to the real time implementation, was used to validate the controller for robustness. Due to their ability to accurately model systems with many degrees of freedom due to their ability to solve several ODEs simultaneously, MATLAB and Simulink were used to create a higher fidelity model. In this thesis, the primary purpose of using a Simulink model was to design a robust controller. Although it has been proven that the dynamic forces and gravitational forces are small in this application [8, 29], simulating the system with these forces ensured the actual stiffness met the desired stiffness.

The Simulink model and the process through which it was developed is found

in Appendix C. This desired ankle position with respect to time is shown in Fig. 3.3 and is based on published data [18, 46].

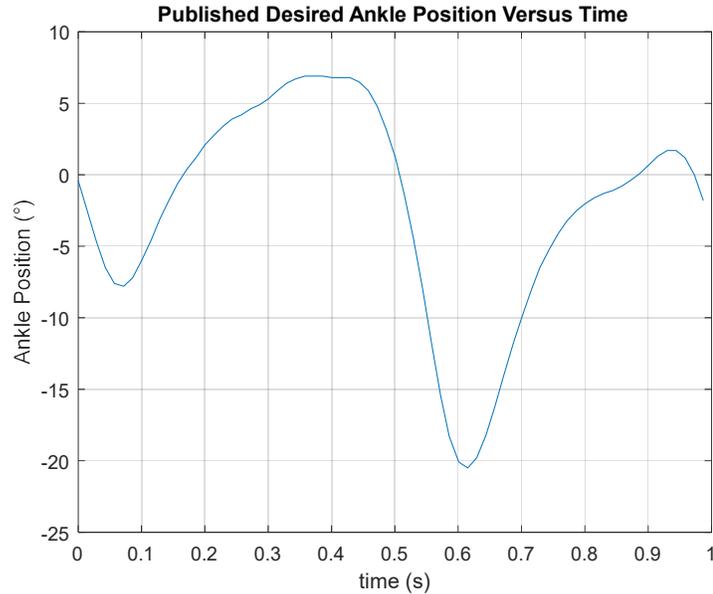


Figure 3.3: Published Position Versus Time According to Winter [46]

3.2.1 Dynamic Simulation and PI Tuning

With the system modeled as shown in Fig. C.1, the PI gains of the motor controller were determined such that the desired stiffness of the ankle was met. Note that because the mechanism is a four-bar mechanism, the input/output relationship is nonlinear. Typically, the PI parameters are tuned by linearizing the model from the plant and then determining the parameters from the derived model. Due to the nonlinearity of the four-bar mechanism, a modified Ziegler Nichols approach was used on the model to obtain the PI parameters.

There are typically two Ziegler-Nichols methods used to tune PI controllers: the Ultimate Cycle Method and the Process Reaction Method. The Process Reaction requires a step change of the controller's output that alters the controlled variable. Based on the shape and magnitude of the controlled variable's reaction curve in reference to the step change, the response is linearized to obtain values to be used in mathematical formulas to calculate the control gains [27]. As the four-bar mechanism is nonlinear, the Ultimate Cycle Method was implemented on the Simulink model. The Ultimate Cycle Method requires one to set up a proportional controller and slowly increase the gain until the system has sustained oscillations. Once sustained oscillations have been achieved, the proportional gain used to obtain sustained oscillations is used to calculate approximate gains for a PI controller. From here, the gains are fine-tuned until the actual position correlates to the desired position. The final PI gains that yielded the position response shown in Fig. 3.4 are seen in Table 3.1.

Table 3.1: Final Parameters for Tuned Controller

Parameters	P (unitless)	I (s^{-1})
Tuned Value	416	159

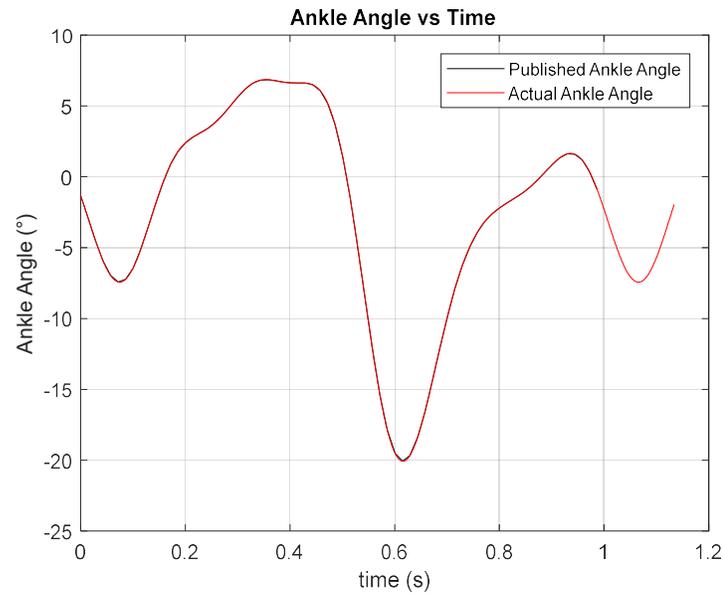


Figure 3.4: Desired Ankle Angle versus Actual Ankle Angle

3.2.2 Investigation of Linear Spiral Spring

The Marquette Prosthesis was developed and optimized such that the spring would provide the required ankle torque with the assumption that the spring was fixed to the grounded link [8], i.e., in parallel with Joint A. Due to this assumption, previous models with parallel springs are not valid when trying to implement stiffness control. Stiffness control requires the linear spiral torsional spring to be in series with the motor and four-bar linkage as shown in the Simulink model in

Appendix B. This new model along with the limitations of the mechanism, motor, and gearbox can be used to select a spring constant such that the torque of the ankle meets the desired torque of the ankle. Fig. 3.5 shows the desired ankle torque compared to the actual ankle torque with the old spring, now in series, as determined by the Simulink model.

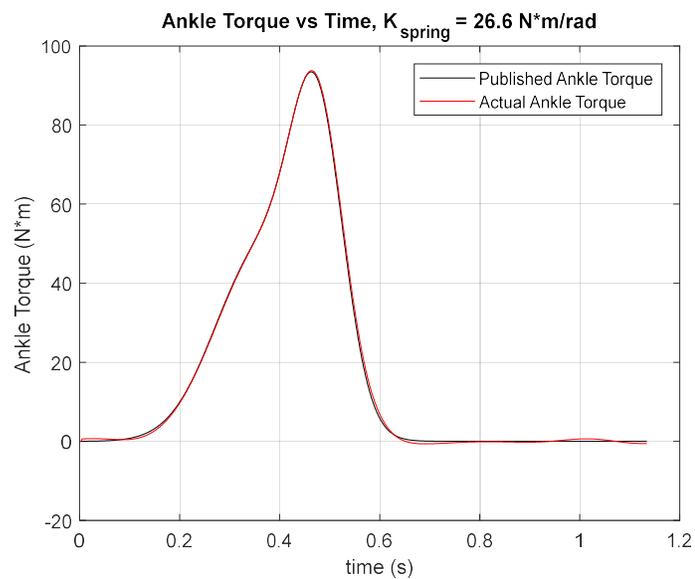


Figure 3.5: Desired Ankle Torque versus Actual Ankle Torque with Original Spring

3.2.3 Effect of Spring Constant on Output Torque

As an attempt in finding an adequate spring constant, a few limiting cases of spring constants were attempted. Fig. 3.6 shows the ankle torque with a spring constant of 26.6 N*m/rad compared to the motor torque with a spring constant value of 13.3 N*m/rad, 6.7 N*m/rad and 2.66 N*m/rad, half, quarter and a tenth of

the original spring constant value, respectively, to investigate the effect of the spring constant on the output torque.

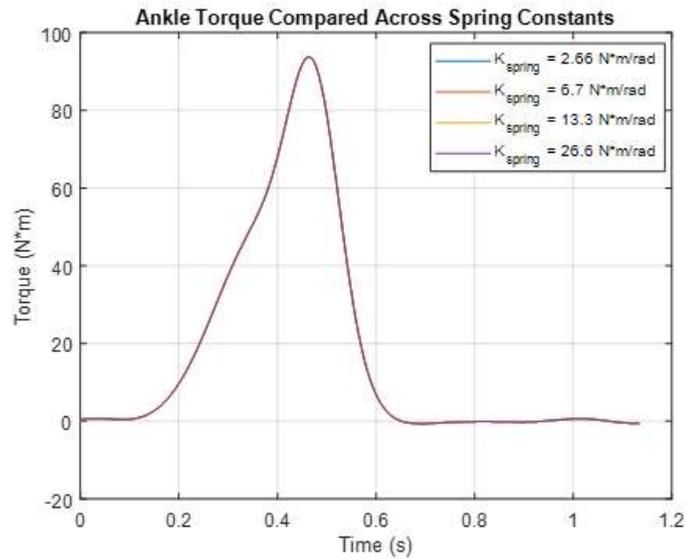


Figure 3.6: Effect of Spring Constant on Motor Torque

There is little to no effect of the spring constant on the motor torque because the torques are consistent regardless of the spring constant as shown in Fig. 3.6. However, the selection of a spring is vital and an engineering judgement should be made in selecting the appropriate spring constant. For instance, a spring with a lower spring constant will cost less, will weigh less and will reduce the noise, i.e., sound from the motor. However, there is a limitation of a spring with a low spring constant. A lower spring constant will require the motor to move faster so that the desired ankle torque and position are met.

3.2.4 Motor Torque and Speed Capabilities

With the effect of the spring constant on the output torque investigated, it was important to ensure the required torque and speed curves fall under the speed torque curve for this motor. Fig. 3.7 shows the motor speed torque curve with a spring constant of $26.6 \text{ N}\cdot\text{m}/\text{rad}$ from Simulink, where the Maxon RE-40 Brushed DC motor has a no-load speed of 7590 rpm and a stall torque of $2.56 \text{ N}\cdot\text{m}$. From Fig. 3.7, it is clear that the motor should have no issue meeting the speed torque requirements of the prosthesis. Unfortunately, due to the motor's speed and torque, its efficiency drops below 50%. When the motor is running below 2000 rpm or above $0.2 \text{ N}\cdot\text{m}$, its efficiency drops off [21].

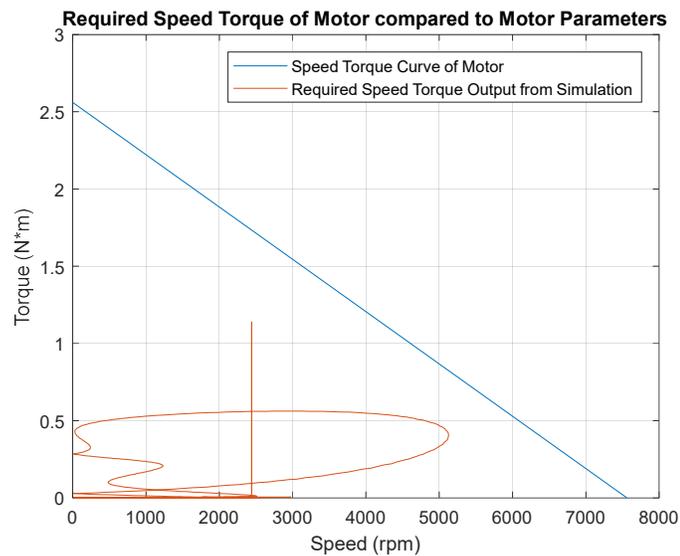


Figure 3.7: Required Speed Torque of Motor Compared to Motor Speed Torque Curve

3.2.5 Final Spring Selection

The model shows that the spring does not have much of an effect on the motor and the motor can meet the speed and torque requirements of the system. As with any model, these results should be validated with the physical system. The effect of changing the spring constant on the physical system is something that will be revisited in the future work of this thesis. For this work however, the original spring with a spring constant of $26.6 \text{ N}\cdot\text{m}/\text{rad}$ was selected due to the constraints of this project.

CHAPTER 4

Mechanical System Redesign

The Simulink model of the Marquette Prosthesis II shows that stiffness control is feasible, but some mechanical changes were made before implementation. The design of the Marquette Prosthesis utilized a linear spiral torsional spring in parallel with a four-bar mechanism to mimic the complex nonlinear response of the foot and ankle as shown in Fig. 4. Because stiffness control requires a relationship between the motor position, ψ , the torque of spring, τ_{spring} , and the input angle of the four-bar mechanism, ϕ , the linear spiral torsional spring was moved in series with the motor and link 3 as shown in Fig. 4. Therefore, to move the spring from being in parallel to series with link 3 and the motor, link 3 was redesigned. Due to cost and manufacturing constraints, the Marquette Prosthesis II was constrained by the linear spiral torsional spring and link lengths chosen for the Marquette Prosthesis.

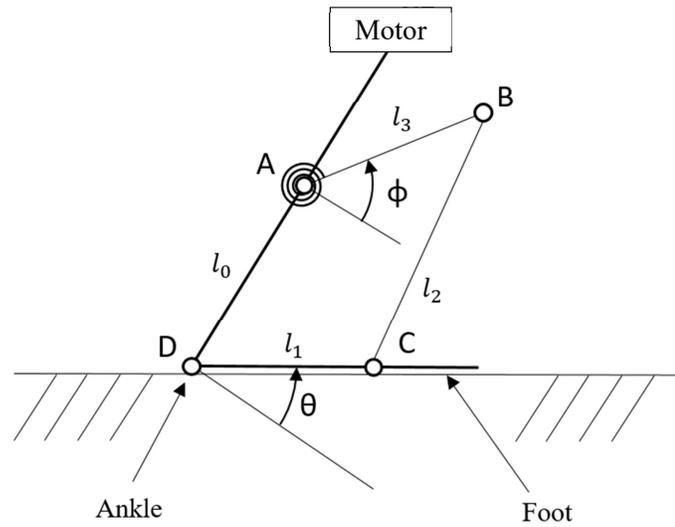


Figure 4.1: Model of Marquette Prosthesis with Spring in Parallel with the Motor Adopted from [40]

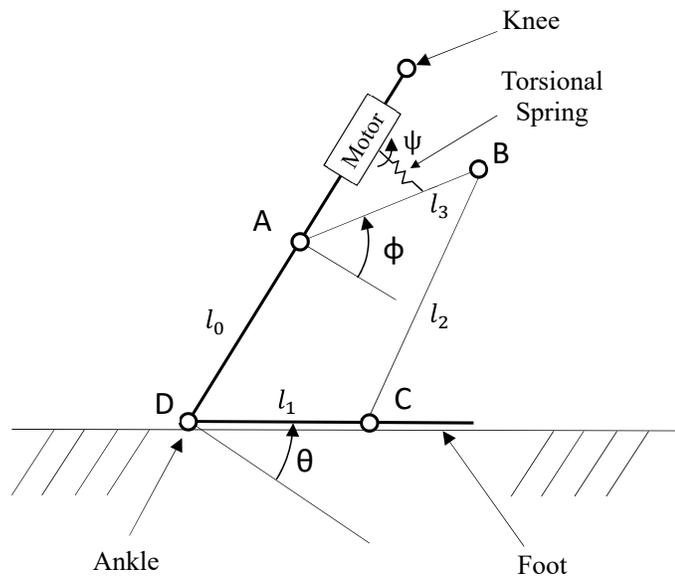


Figure 4.2: Model of Marquette Prosthesis II with Spring in Series with the Motor

4.1 Redesign of Link 3 to move spring in Series

The Marquette Prosthesis had one end of link 3 fixed to the motor shaft and the center of the spring slotted into the motor shaft. A mount, shown in Fig. 4.3, connected the outer hook of the spring to a bolt which threaded into the mount. The two slots of the block were fit around the ground link and were secured via a bolt; therefore, the spring was grounded. To move the spring in series with the motor and link 3, this mount was removed.

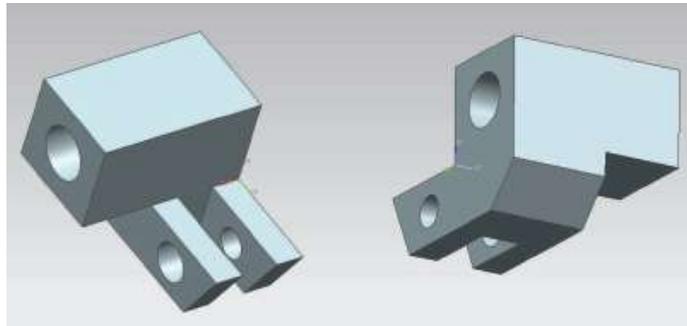


Figure 4.3: Mount that Connects the Spiral Torsion Spring to Link 0

The new design of link 3, shown in Fig. 4.4, was a rectangular bar with rounded edges containing three holes, two thru holes and one threaded hole, for attachment to the motor, spring and link 2. The new design required a bearing to connect link 3 to the motor shaft to rotate freely about the motor shaft. The center of link 3 was designed to press fit a bearing into a thru hole. Because the redesign of link 3 was constrained by the size of the spring and a bolt was used to connect the

spring to the grounded mount shown in Fig. 4.3, the distance from the bearing hole to the threaded hole was designed to be equal to the distance from the center of the spring to the center of the outer loop of the spring, shown in Fig. 4.5. The center of the spring was then slotted into the motor shaft and a bolt connected the outer loop of the spring to link 3. A nut was then threaded onto the bolt to secure the connection between the spring and link 3. A side view of this assembly is shown in Fig. 4.6, where the linear spiral torsional spring is black and the link is white. Because the link length was constrained by the Marquette Prosthesis design, the distance from the center of the bearing to the center of the other thru hole was equal to the length of link 3. A dowel pin was then press fit into this second thru hole to connect link 3 to link 2.

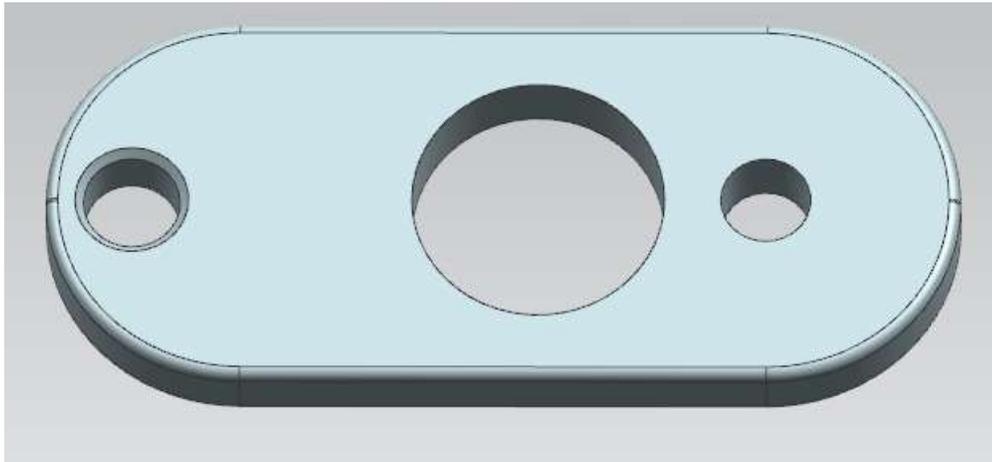


Figure 4.4: Redesign of Link 3

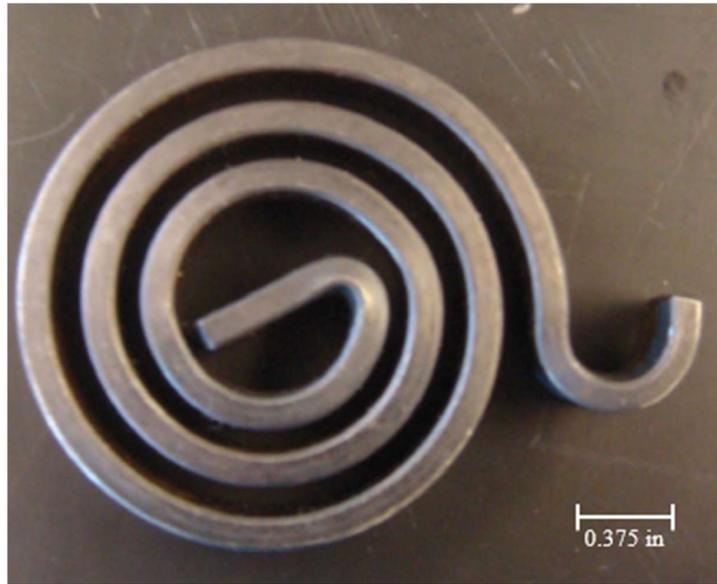


Figure 4.5: Linear Spiral Torsional Spring

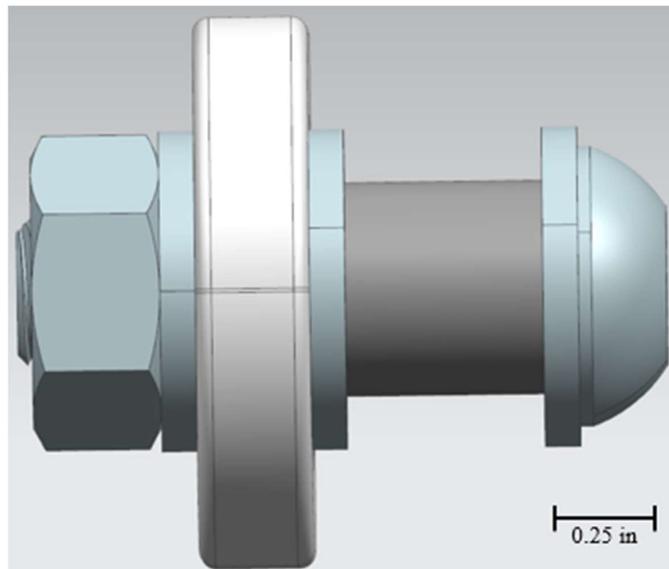


Figure 4.6: Side View of Assembly of Link 3 and Linear Spiral Torsional Spring

4.1.1 Press Fit Analysis

ANSYS 19.0 was used to ensure the stress and deformation due to the press fit would not cause the part to fail. After the dowel pin and link were imported from NX 10.0 as an assembly, the appropriate material was assigned to the pin and link. The pin was assigned alloy steel as its material and the link was assigned Aluminum 7076-T6 as its material. A frictional contact was established between the surface of the dowel and the surface of the hole of the press fit. In the settings of the frictional contact, the normal stiffness factor was changed manually from 1.0 to 0.1 and the stiffness was updated on each iteration since the stiffness will change as the dowel pin is pressed into the hole. In the geometric modification section of the contact, the interface treatment setting was changed to “Add Offset Ramped Effects” as this appropriately models the expansion of the hole as the dowel is press fit. The mesh was original sized to have an element size of 1.0 mm and was further refined to 0.5 mm to validate the results. The sides of the link were then fixed and the analysis settings were altered such that the solution would converge, outputting the deformation and equivalent stress.

The deformation and equivalent stress due to the press fit are shown in Figs. 4.7 and 4.8, respectively. From the Finite Element Analysis, the maximum deformation was 0.0216 mm and the max von Mises stress was 415 MPa compared with a yield strength of 470 MPa. Although the resulting stress is close to the yield

strength of Aluminum 7076, is it still less than the yield strength. This result, along with only 0.0216 mm of deformation, inferred that the press fit can be completed without failure.

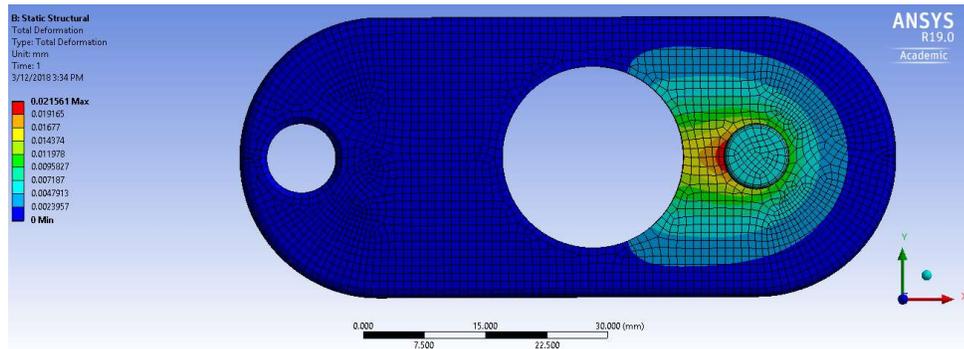


Figure 4.7: ANSYS Deformation Results

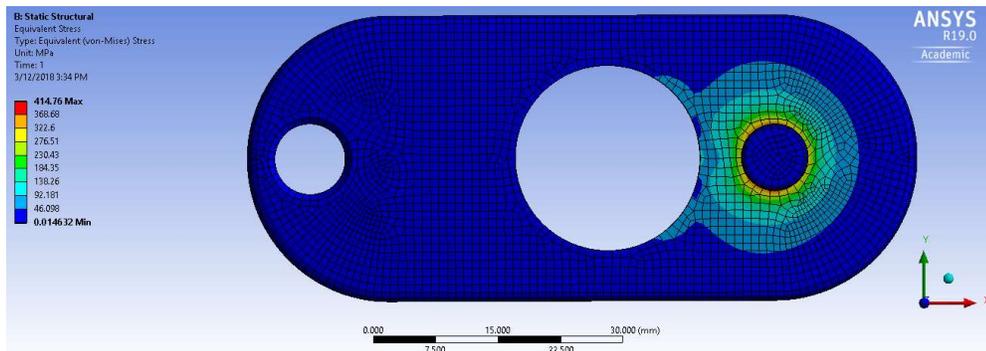


Figure 4.8: ANSYS Stress Results

CHAPTER 5

Control System Architecture and Implementation

With the appropriate mechanical changes made on the Marquette Prosthesis as described in Chapter 4, the controller was implemented. In this chapter, the architecture of the control system is outlined and details of the implementation are provided.

5.1 Overall Control System Architecture

There were two main components to the architecture of the control system developed for the Marquette Prosthesis II: the ANN and the stiffness controller. The ANN will provide the desired ankle position and torque while the stiffness controller controlled this stiffness via the motor. Because the ANN was not fully developed at the time of this work, published data was used to provide the desired position, θ , and torque of the ankle, τ_{ankle} . From the desired position and torque of the ankle, the kinematics and quasi-statics as discussed in Chapter 3 and Appendices A and B were computed to determine the desired motor position.

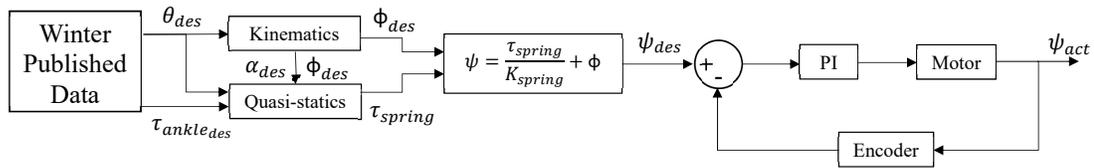


Figure 5.1: Block Diagram Representation of Control System

Fig. 5.1 illustrates the overall control architecture. Published data provided the desired ankle position, θ , and torque of the ankle, τ_{ankle} . The desired input angle of the four-bar mechanism, ϕ , was then determined via the vector loop method and the desired spring from the torque, τ_{spring} , was determined via the quasi-statics. The input angle of the four-bar mechanism, ϕ , and torque of the spring, τ_{spring} were then input into the relationship relating the position of the motor, ψ , to the spring torque, τ_{spring} , input angle of the four-bar mechanism, ϕ , and spring constant of the spring, K_{spring} , as shown in Equation 3.2. From this relationship, the desired position of the motor, ψ , was input into the control system, where the PI controller regulates the motor position as desired.

5.2 Controller Implementation

The goal of the Marquette Prosthesis II controller and hardware was to be lighter and more compact than the Marquette Prosthesis. The hardware platform also needed to have the computing power to be able to complete the control

calculations within a 150 ms window [14]. Due to cost constraints, the Marquette Prosthesis II utilized the motor and encoder chosen for the Marquette Prosthesis.

5.2.1 Computer System Overview of Other Successful Prostheses

The selection of hardware was pivotal to ensure it was lighter compared to previous work and capable of meeting the control requirements. For comparison, the hardware utilized at MIT, ASU and Vanderbilt University are shown in Fig. 5.2-5.4. The systems used by MIT and ASU were similar, both using a PC/104 Single Board Computer with a Sensoray 526 DAQ as their computing platform and Advanced Motion Control (AMC) motor drives. The PC/104 board was programmed using Simulink Real-Time Explorer, in which the PC/104 board outputs to the Sensoray 526 DAQ via ISA bus. Vanderbilt University used a PIC32 microcontroller as their computing platform with two AMC motor drives because their system has two motors. The PIC32 microcontroller was programmed in C using MPLAB IDE and MP32 C Compiler [43].

The Marquette Prosthesis had a drastically different configuration compared to MIT, ASU and Vanderbilt (see Figure 5.5). Because the Marquette Prosthesis was the first generation of an active transtibial prosthesis developed at Marquette University, the hardware selected was manufactured by Maxon to mitigate the amount of communication issues. This hardware system was much bulkier than the

hardware used by MIT, ASU and Vanderbilt. The hardware consisted of an EPOS2 50/5 motor controller powered by a 14.8V lithium-ion polymer battery, an eZdsp F28335 control board powered by a 5V lithium-ion battery and an FSR circuit powered by three 1.5V standard AAA batteries connected in series. The portable testing platform for the Marquette Prosthesis housed the hardware on a backpack that was bulky and heavy [40]. The Marquette Prosthesis was successful in choosing hardware that was reasonably light and compact, as the EPOS2 50/5 motor controller only has a mass of 240g and is 120 mm by 93.5 mm [31] and the eZdsp control board has a mass of 137.9g and is 76 mm by 136 mm [25]. However, this system required three power supplies, which made this hardware bulky.

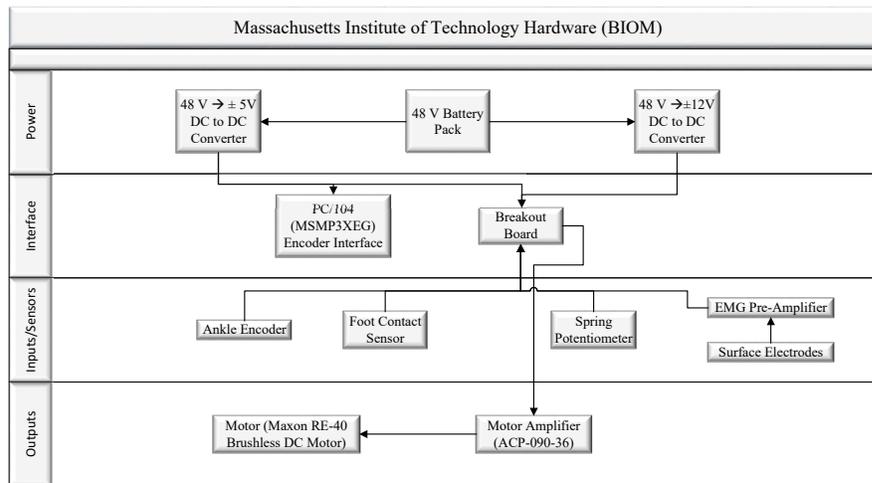


Figure 5.2: Massachusetts Institute of Technology Hardware Flow

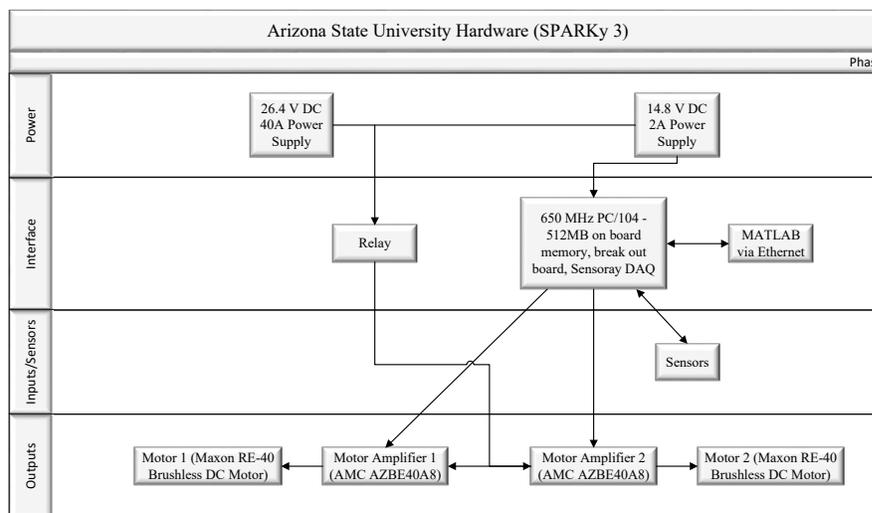


Figure 5.3: Arizona State University Hardware Flow

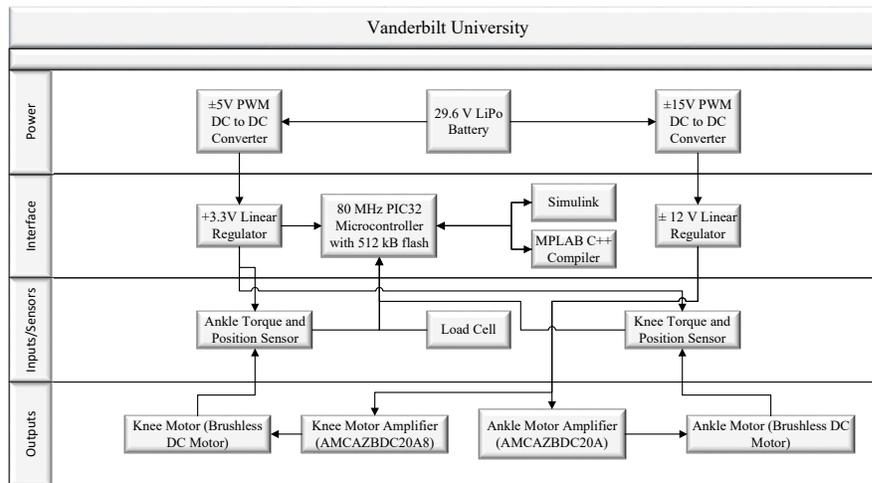


Figure 5.4: Vanderbilt University Hardware Flow

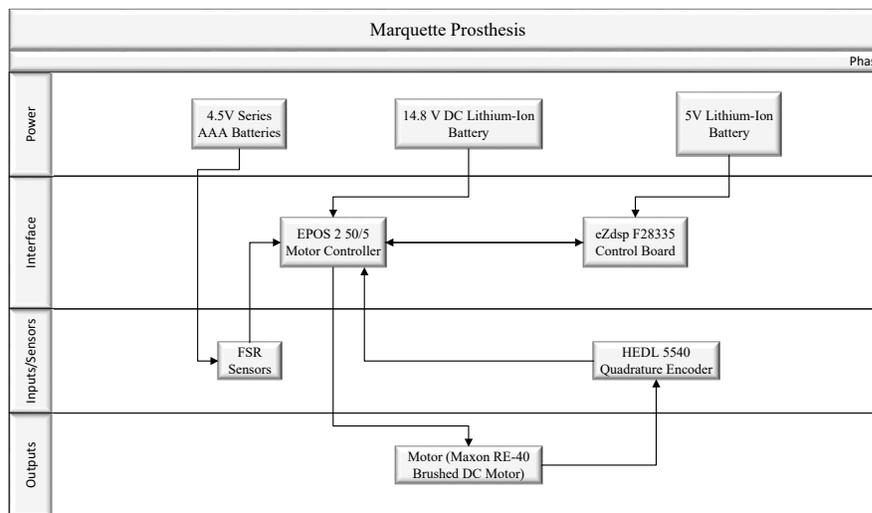


Figure 5.5: Marquette Prosthesis Hardware Flow

5.2.2 Selection of Control Hardware for Marquette Prosthesis II

In selecting the computing platform for the Marquette Prosthesis II, the PC/104 platform was initially investigated due to the success of both MIT and ASU. Due to compatibility, support and communication issues with the PC/104 hardware platform, it was not utilized in this work. The Arduino platform was utilized as it is widely used at Marquette and is compatible with MATLAB and Simulink. The Arduino Mega 2560 was ultimately chosen as the microcontroller for the control system due to its level of compatibility and small footprint.

To control the motor continuously, a motor drive was required as the Arduino Mega 2560 only outputs 20 mA of current per I/O pin and the motor is rated with a nominal current of 3.17A and a peak current of 6A. Advanced Motion Controller (AMC) drives were investigated initially (similar to MIT, ASU and Vanderbilt), but these drives require input signals with a frequency in the range of 10kHz to 25 kHz. The Arduino Mega 2560 can output PWM signals at 490 Hz, 980 Hz, 3.9kHz or 32kHz, as configured by the user, thus the AMC motor drives were dismissed as possible drivers. Instead, an LMD18201T/NOPB H-Bridge was selected as the motor driver for two reasons: first, it can output signals 3A continuously, 6A max, meeting the specifications of the motor; secondly, it has a much smaller footprint than the AMC drives.

5.2.3 Selection of Sensor Hardware for Marquette Prosthesis II

The encoder for the Marquette Prosthesis II was the HEDL 5540 Quadrature Encoder used for the Marquette Prosthesis. This encoder has a high channel A, low channel A, high channel B and low channel B. For proper quadrature encoder feedback to the Arduino Mega, an MC3486 line receiver was used with the HEDL 5540 Quadrature Encoders to convert the differential A and B channels into TTL values to be decoded by the Arduino board. The output signals from the MC3486 line receiver were connected to the inputs of the Arduino that measured the angular position of the motor shaft.

The ANN will require EMG activity of the amputee as well as the current position of the ankle to predict the desired position and torque of the ankle. Therefore, an encoder was mounted at the ankle joint. A second HEDL 5540 Quadrature Encoder was selected to be mounted at the ankle joint. Similar to the motor encoder, the ankle encoder signal was input to the MC3486 line receiver which output the ankle encoder signal to the Arduino. The hardware selected for the Marquette Prosthesis II is outlined in Fig. 5.6.

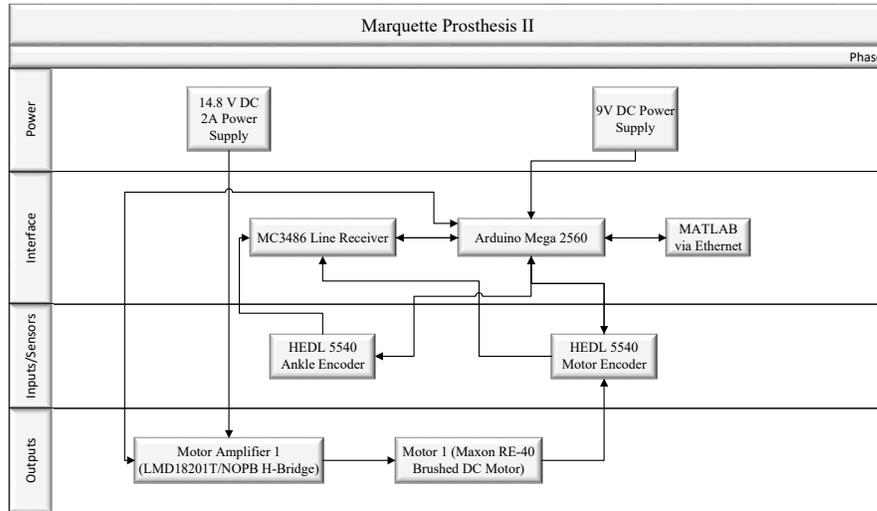


Figure 5.6: Marquette Prosthesis II Hardware Flow

5.2.4 Marquette Prosthesis II Hardware compared to Marquette Prosthesis Hardware

One of the goals of the hardware for the Marquette Prosthesis II was to make it more compact and lighter than the Marquette Prosthesis hardware. Compared to the Marquette Prosthesis hardware, the hardware for the Marquette Prosthesis II was much lighter as shown in Tables 5.1 and 5.2. The Arduino Mega 2560 has a mass of 37g [2] and the LMD18201T/NOPB H-Bridge has a mass of 5.4g. The Arduino Mega 2560 and the H-bridge replaced the eZdsp control board, which has a mass of 137.9g, and the EPOS2 motor controller, which has a mass of 240g. To wire all the components appropriately, a small breadboard was mounted to the gearbox that has a mass of 50g. A breadboard was also used for wiring components on the

Marquette Prosthesis that has a mass of about 50g. Note that the ANN requires the actual ankle position, a second HEDL 5540 quadrature encoder was mounted at the ankle, adding another 15g to the total mass of the hardware for the Marquette Prosthesis II. The Arduino Mega also contains three 5V power supplies, which were used to power the two encoders and the MC3486 line receiver, thus eliminating one of the 5V power supplies which has a mass of 155g. Therefore, including the same 14.8V LiPo battery that has a mass of 465g, the total mass of the hardware for the Marquette Prosthesis was 1077.9g while the mass of the hardware for the Marquette Prosthesis II was 662.4g, a 39% reduction in mass.

Table 5.1: Mass of Components for Marquette Prosthesis

Component	Mass(g)
eZdsp Control Board	137.9
EPOS2 Motor Controller	240
Breadboard	50
HEDL 5540 Quadrature Encoder	30
LiPo 5V Power Supply	155
LiPo 14.8V Power Supply	465
Total Mass	1077.9

Table 5.2: Mass of Components for Marquette Prosthesis II

Component	Mass(g)
Arduino Mega 2560	37
LMD 18201T/NOPB H-Bridge	5.4
Breadboard	50
2x HEDL 5540 Quadrature Encoder	60
LiPo 14.8V Power Supply	465
Total Mass	662.4

The hardware for the Marquette Prosthesis II was much more compact than Marquette Prosthesis hardware as shown in Tables 5.3 and 5.4. The Arduino Mega 2560 is 101.52 mm by 53.3 mm [2], and was mounted to one side of the gearbox, and the LMD18201T/NOPB H-Bridge is 20.02 mm by 17.02 mm and was mounted to the gearbox. The H-Bridge was mounted to the gearbox not only to save space but also to use the gearbox as a heat sink. Along with these two pieces of hardware, the breadboard mounted on the other side of the gearbox is 55.12 mm by 82.80 mm. Meanwhile, for the Marquette Prosthesis, the eZdsp control board is 76.2 mm by 135.89 mm and the EPOS2 50/5 is 119.89 mm by 93.47 mm. The breadboard used for the Marquette Prosthesis is 165.1 mm by 114.3 mm. Each HEDL 5540 encoder is 30 mm by 41.1 mm. Furthermore, the power supply eliminated is 88.9 mm by 57.15 mm. Therefore, including the same 14.8V LiPo battery being 38.1 mm by 146.05 mm, the total footprint for the Marquette Prosthesis was 18350 square mm while the total footprint for the Marquette Prosthesis II was 52310 square mm, reducing the footprint by 65%.

Table 5.3: Size of Components for Marquette Prosthesis

Component	Size (sq mm)
eZdsp Control Board	10350
EPOS2 Motor Controller	11210
Breadboard	18870
HEDL 5540 Quadrature Encoder	1233
LiPo 5V Power Supply	5080
LiPo 14.8V Power Supply	5565
Total Size	52310

Table 5.4: Size of Components for Marquette Prosthesis II

Component	Size (sq mm)
Arduino Mega 2560	5413
LMD 18201T/NOPB H-Bridge	340.7
Breadboard	4564
2x HEDL 5540 Quadrature Encoder	2466
LiPo 14.8V Power Supply	5565
Total Size	18350

CHAPTER 6

Bench Testing

To validate the Marquette Prosthesis II model, mechanical system redesign and control system, bench testing was performed. The hanging hand test was first completed where the prosthesis was not in contact with the ground to verify the theory discussed in Chapter 3. The force hand test was conducted where the prosthesis was moved by hand across a force plate in an attempt to validate stiffness control. The testing methodologies and procedures are first discussed, followed by an analysis of the results.

6.1 Stiffness Control Testing

The block diagram representing the control system used to validate the Marquette Prosthesis II is shown in Fig. 5.1 and is repeated in Fig. 6.1. As explained in Chapter 3 and 5, the ankle moment and ankle angle as published by Winter [46] were used as the inputs to the controller. The controls were then implemented as explained in Section 5.1.

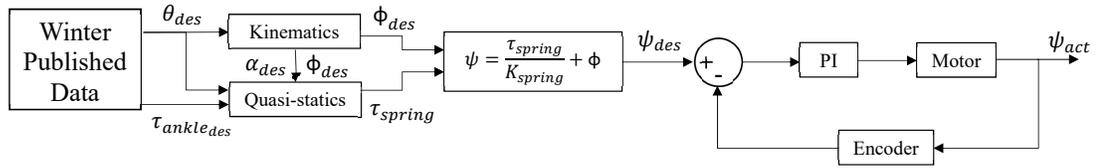


Figure 6.1: Block Diagram of Control System for Testing (Identical to Fig. 5.1)

6.2 Bench Testing

With the control system developed, the Marquette Prosthesis II was tested.

Two different tests were completed: a hanging hand test and a force hand test.

6.2.1 Hanging Hand Test Procedure

The hanging hand testing was completed to validate the dynamic theory discussed in Chapter 3. In this test, the Marquette Prosthesis II was held above the ground such that it would complete a desired ankle profile. Because the Marquette Prosthesis II was not touching the ground, an ankle torque of 0 N*m was input into the quasi-statics instead of the published torque from Winter [46], while the ankle angle from Winter [46] was input into the vector loop equations. Because the ankle torque was equal to zero, the torque provided by the spring was equal to zero, thus the angle of the motor, ψ , was equal the input angle of the four-bar mechanism, ϕ .

During testing, the Simulink control model as shown in Fig. D.1 was run in

external mode on the Arduino Mega 2560. The HEDL 5540 encoder attached to the motor measured the motor angle while the HEDL 5540 encoder attached to the ankle measured the ankle angle. A power supply was used to regulate 12V to the LMD18201T/NOPB H-Bridge. The hardware was attached to the Marquette Prosthesis II and is shown in Figs. 6.2 and 6.3. By measuring the motor position as the prosthesis completed the desired move, the robustness of the controller was validated. Furthermore, by measuring the position at the ankle, the dynamic theory in Chapter 3 was validated.



Figure 6.2: Side View of Marquette Prosthesis II with Hardware Attached

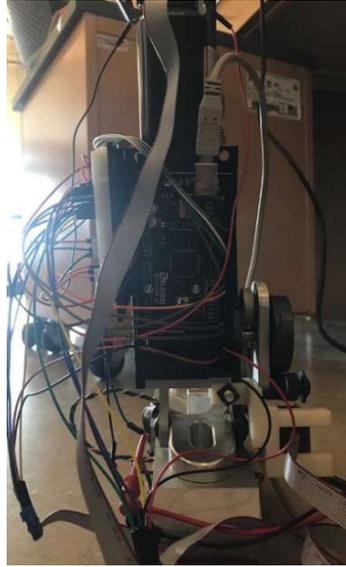


Figure 6.3: Back View of Marquette Prosthesis II with Hardware Attached

6.2.2 Hanging Hand Test Results

Fig. 6.4 shows the motor position results from the hanging hand test. Fig. 6.4 shows that the controller was tuned such that the motor matched the desired position curve for all three tests. There was some error at the peaks and valleys of the profile, where the motor tended to oscillate about the desired position. This error can be attributed to the inability of the PI controller to adapt to the fast-changing ankle signal. The way to counteract this effect would be to implement a feed-forward controller that can anticipate the quick changes in the position of the ankle.

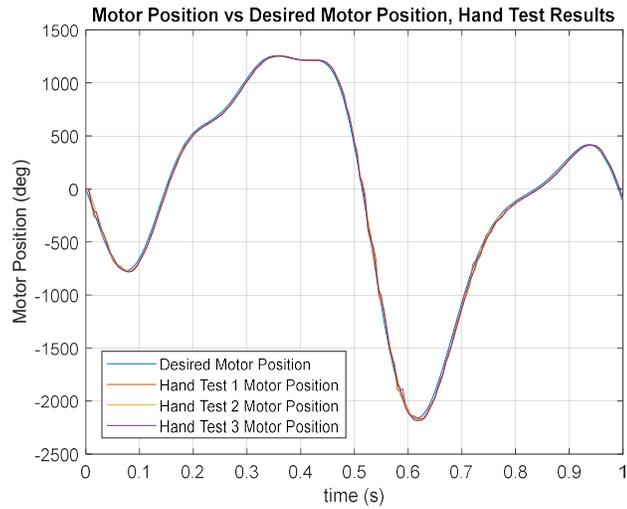


Figure 6.4: Motor Position Results, Hanging Hand Test

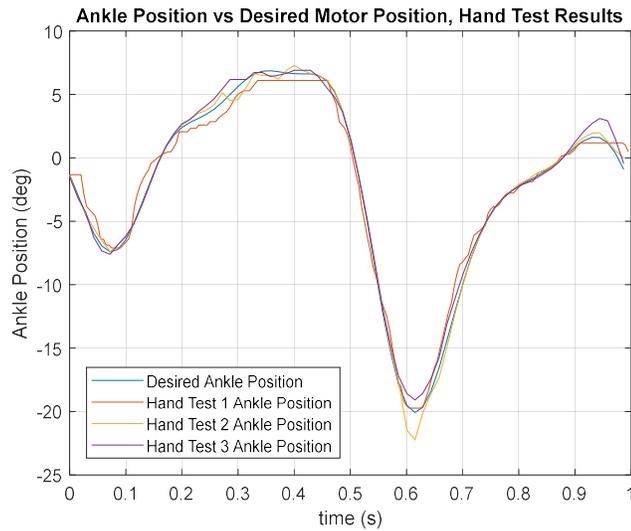


Figure 6.5: Ankle Angle Results, Hanging Hand Test

Fig. 6.5 shows the ankle position results from the hanging hand test. Fig. 6.5 shows that the ankle position for all three tests was close to the desired ankle

position profile, but there was some error in all three tests. There are two reasons for this error. First, some of the error can also be attributed to noise in the encoder at the ankle from the sensor not being perfectly lined up; any slight misalignment of the encoder will cause an incorrect reading. Because the Marquette Prosthesis was not developed to have an encoder at the ankle, the encoder was mounted via a 3D printed part that attached to the ground link. A way to reduce this effect is to create a mount for this encoder that is built into the ground link of the four-bar mechanism. Second, some error could be attributed to some delay in recording the position from the encoder at the ankle. The Arduino Mega 2560 is capable of taking in two quadrature encoder readings; however, it may not be fast enough to keep up with both simultaneously for gait cycles that differ from published data. Ultimately, by controlling the motor and measuring both the position of the motor and position of the ankle, the hanging hand test validated the success of the controller and the vector loop method.

6.2.3 Force Hand Test Procedure

In an attempt to validate the stiffness controller, force hand tests were completed in the Human Performance Laboratory (HPL) at Marquette University's Engineering Hall.

Figs. 6.6, 6.7 and 6.8 show how this test was conducted. The Marquette

Prosthesis II was held by the pylon and moved in the sagittal plane across a force plate built into the floor of the HPL as the Marquette Prosthesis II completed the desired motion profile. The motion profile for the ankle was recorded by the encoder at the ankle, the motor position was recorded by the encoder on the motor and the force experienced by the force plates from the Marquette Prosthesis II was recorded in LabView. The motion of the Marquette Prosthesis II was captured on video. A total of approximately 15 runs were conducted, while the results from three are discussed due to the nature of this testing procedure.

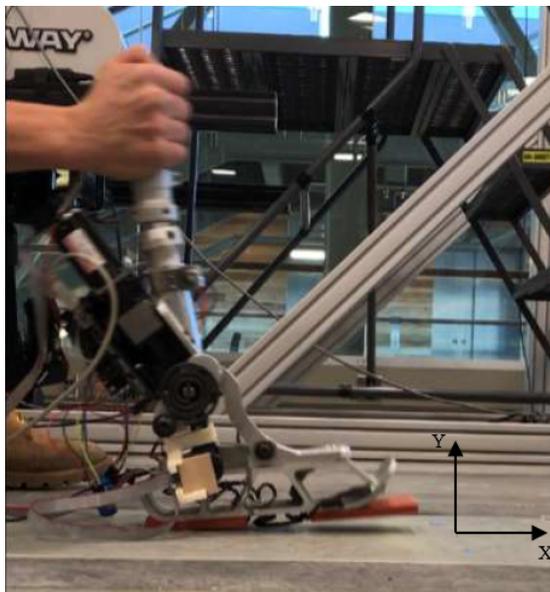


Figure 6.6: Force Hand Test, Prosthesis at Heel Strike

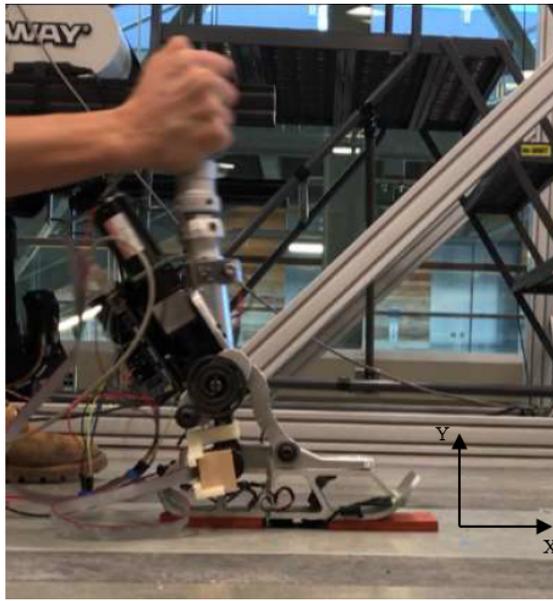


Figure 6.7: Force Hand Test, Prosthesis at Flat Foot

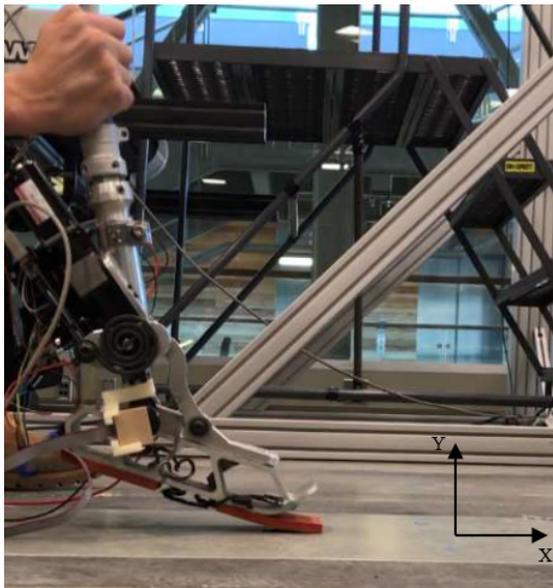


Figure 6.8: Force Hand Test, Prosthesis at Plantar Flexion

6.2.4 Force Hand Test Results

The force hand test was run to test the stability of the Marquette Prosthesis II and to ensure the force curves follow the same profile as the published data as shown in Figs. 6.9 and 6.10. It is possible to calculate the moment generated at the ankle from the forces generated in the X and Y direction and from kinematic data. Doing so was unnecessary because the forces generated by the prosthesis for this test were known to be incorrect. Because of the inconsistent nature of this test, markers were not attached to the prosthesis during this testing to collect kinematic data to help calculate the moment generated by the ankle. The method to do so follows the procedure described by Winter [46] as shown in Appendix E.

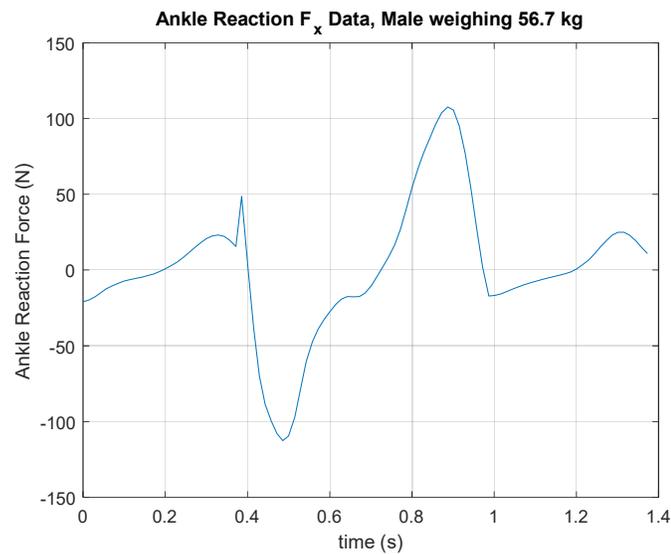


Figure 6.9: Published Ankle Reaction Force in x-direction [46]

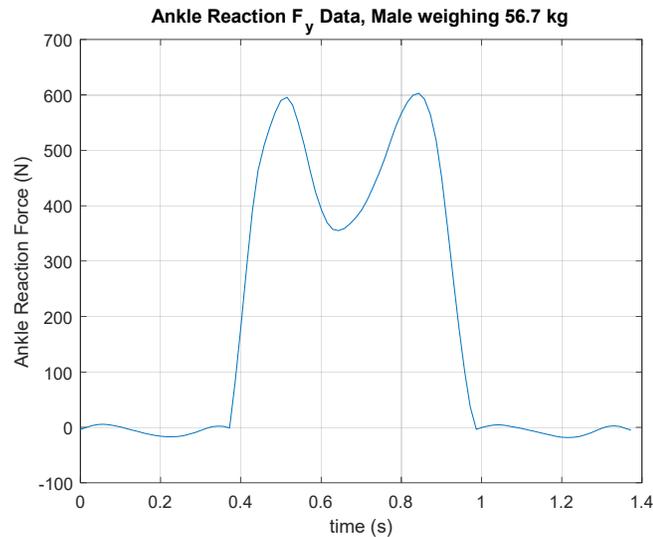


Figure 6.10: Published Ankle Reaction Force in y-direction [46]

Out of the 15 force hand tests that were run, only three are discussed in this chapter due to the ability of the tester's hand to produce a consistent movement in the sagittal plane. The forces recorded in the x-direction for the three tests are shown in Figs. 6.11, 6.12 and 6.13 and the forces recording in the y-direction for the three tests are shown in Figs. 6.14, 6.15 and 6.16. The reaction force in the x-direction mimics the first half of the published data, spiking down upon heel strike and then returning to zero as the ankle shifts to flat foot. However, once the ankle reached flat foot after dorsiflexion, all three tests yielded a force of approximately 0 N. The reason for the force being zero after flat foot was due to this force being recorded in the same direction that the tester was moving the prosthesis, thus being highly influenced by the input from the tester. Regardless, the magnitude of the

initial spike was incorrect due to the nature of this test. The reaction force in the y-direction for all three tests follows the profile from published data; however, the magnitude for these recorded forces deviate from the published data was incorrect as well. This error once again was caused by the nature of the test. The force test depends on the tester's external force, which was minimized by the tester because the force exerted by the tester cannot be properly measured.

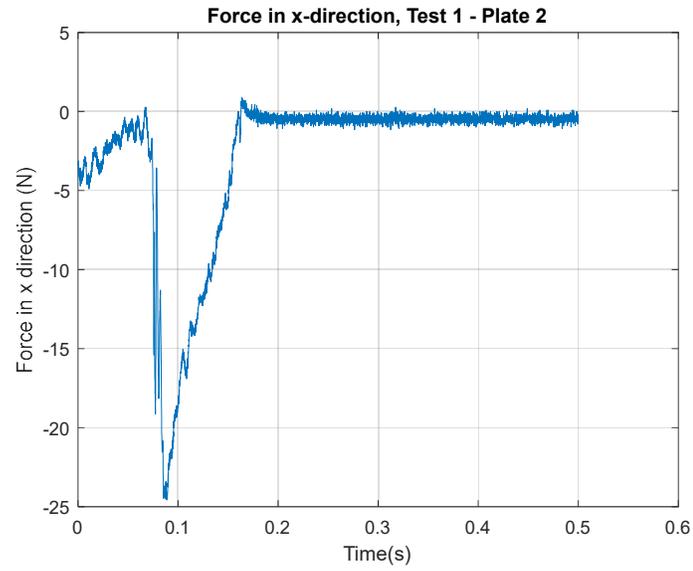


Figure 6.11: Reaction force in x-direction - Test 1

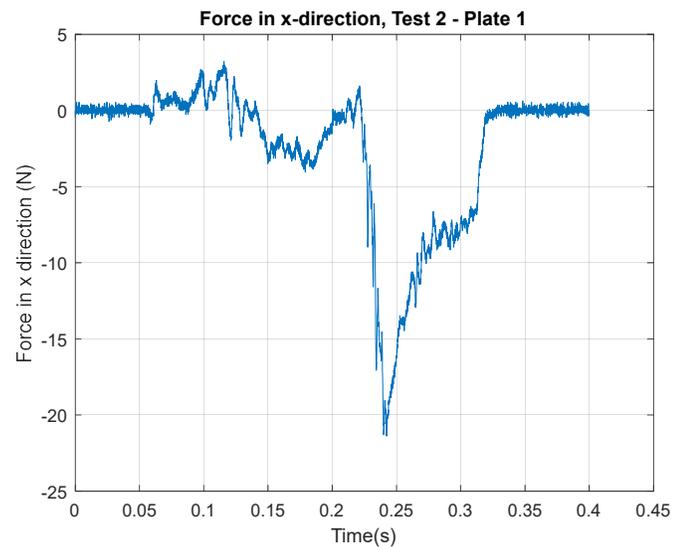


Figure 6.12: Reaction force in x-direction - Test 2

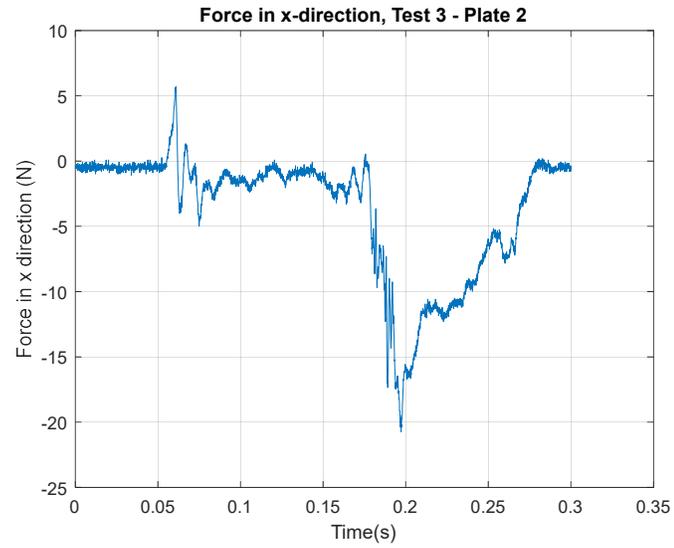


Figure 6.13: Reaction force in x-direction - Test 3

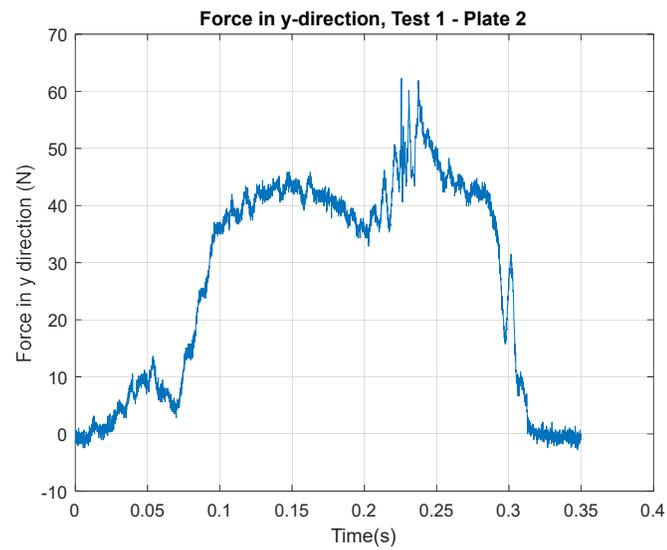


Figure 6.14: Reaction force in y-direction - Test 1

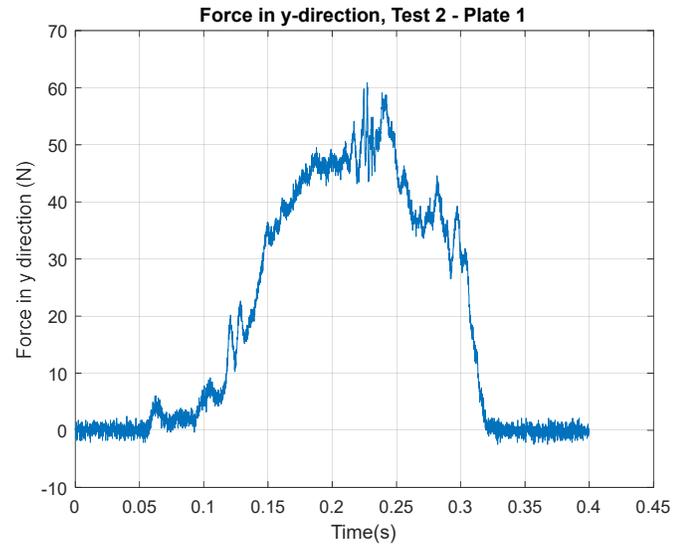


Figure 6.15: Reaction force in y-direction - Test 2

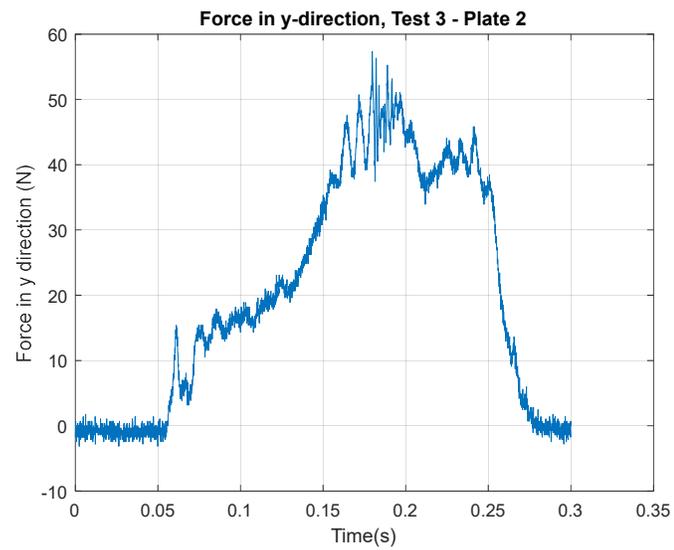


Figure 6.16: Reaction force in y-direction - Test 3

Due to the forces recorded by the force plate deviating significantly from the published data in magnitude due to the nature of the test, the ankle angle recorded by the encoder located at the ankle also contained a large amount of error. Most of this error could be attributed to the torque at the ankle being incorrect, thus the stiffness of the ankle was incorrect. However, some error could be attributed to the mounting of the encoder as discussed for the hanging hand test. For this test, there were times, i.e., during heel strike and toe off, where the encoder reading would experience spikes or the encoder would lag. This error could have been caused by the vibrations experienced by the encoder mounting. In making contact with the ground, the force test proved that a plastic mount is not the ideal holster for the encoder at the ankle.

Due to the amount of error in the ankle angle, the motor position was examined to validate the robustness of the controller. The motor position for the three force tests were plotted against the desired motor position and is shown in Fig. 6.17. Because the Marquette Prosthesis II was in contact with the ground for this test, the motor position was assumed to not be as smooth as it was for the hanging hand testing. Fig. 6.17 proves this assumption to be true at the beginning of the graph, when the foot makes contact with the ground, and at transition points in stride represented by a peak or valley in the figure. There are two reasons for these oscillations. The first is due to the nature of a PI controller with an analog signal,

as discussed earlier. The second is due to the back torque from the mechanism.

When an external disturbance is introduced, such as the back torque from the ankle to the motor, the controller tries to overcorrect the error, causing some oscillations at transition points when there is a brief change in position and torque.



Figure 6.17: Motor Position Results, Force Hand Test

The force results from the hand force testing does not prove that stiffness control was implemented on the Marquette Prosthesis II perfectly. To validate the stiffness controller would require human subject testing. However, this was not possible at the time of this work because the ANN was not fully developed. The hand hanging test proves the vector loop theory discussed in Chapter 3 to be correct. The force hand testing procedure was extremely sensitive to the tester's influence, thus the results from this test were analyzed qualitatively. The hand force

testing does prove however that the Marquette Prosthesis II is stable when it makes contact with the ground. Furthermore, it proves that the Marquette Prosthesis II can generate the same force profile as published data, thus the same moment profile.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusions

Active lower limb prostheses have been available for several years, but active transtibial prostheses do not fully restore normal functions. The major challenge in developing active transtibial prostheses is being able to function during a transition in gait, i.e., climbing up and down stairs, walking to running, etc. While current active prostheses restore most of an amputee's gait, they struggle in the transition primarily due to the type of controller currently implemented.

This thesis overcomes this challenge by creating an ankle stiffness controller instead of using an FSC ankle position controller. This type of control is believed to be the most appropriate approach for active transtibial prostheses. However, it has been proven that the timing of an FSC can be incorrect during transitions in gait [36]. A stiffness controller overcomes this challenge by actively controlling the relationship between the position and torque of the ankle. By receiving a desired ankle position and ankle torque continuously from an ANN, the amputee is able to signal seamlessly the transition from walking to climbing of stairs. The stiffness

controller does not have to take the time to figure out what state of gait the amputee is in because it controls the prosthesis from the continuous signal provided by the ANN.

In order to prove the approach, several tasks were completed without the ANN stiffness controller on the Marquette Prosthesis II. The first task was to implement stiffness control. The system was first modeled in Simulink to prove that this type of control is possible. Once the model was developed, stiffness control was implemented on the physical prosthesis by moving the spring from being in parallel with the motor and link 3 to being in series between the motor and link 2. A Simulink model was used to implement the stiffness controller on the hardware, which controlled the position of the motor such that the desired published ankle position and torque were met. The ANN was not been fully developed yet, thus published data was used to test the prosthesis instead of a continuous signal from the ANN. The controller was tested via the hanging hand test and the force hand test and the results from these tests are shown in Figs. 6.4-6.5 and Figs. 6.11-6.17, respectively.

The second task was to create a control system that was much more compact than the hardware used for the Marquette Prosthesis. By eliminating a power supply and bulky motor controller, the weight was reduced by 39% and the footprint

was reduced by 65%. With this reduction in weight and size, it was possible to mount most, if not all, of the hardware on the residual limb and prosthesis.

This thesis makes a few major contributions in the development of the Marquette Prosthesis II towards the ultimate goal of implementing the prosthesis on a human subject with a working ANN. First, this thesis shows that stiffness control is a feasible control algorithm for controlling the relationship between the ankle torque and position. Stiffness type control is able to overcome the challenge FSC face with unique styles of gait. Second, although the Marquette Prosthesis was not created for this type of control, this thesis shows that stiffness control is possible to implement with the Marquette Prosthesis limitations, i.e., the motor, the link lengths and the spring optimized for the Marquette Prosthesis. Third, this thesis shows that stiffness control can be implemented in a hardware package that is compact enough to be mounted on the residual limb. With a much more compact hardware package, this thesis has provided the project with an extremely mobile and light package that is more comfortable for an amputee.

The Arduino Mega 2560 used as the microcontroller has two major advantageous over the PC/104 system. First, it is much more compact and affordable. Second, it is easily programmed with Simulink and MATLAB. However, it also has two major disadvantages. First, it is not as powerful compared to the PC/104 system. Second, the Arduino Mega 2560 outputs PWM signals at a

frequency of 3.9kHz, while the PC/104 system can output analog signals up to 50kHz. The lower frequency output of the Arduino Mega 2560 can create some lag in the system.

Before the Arduino Mega 2560 was selected, the PC/104 system was attempted and tested but was ultimately dropped due to several issues. The first issue was that the system is no longer compatible with MATLAB and Simulink. The PC/104 system can be programmed via C compiler and MPLAB IDE, however these programs are foreign to the individual working on the Marquette Prosthesis II. The second issue was that there were numerous communication issues between the different boards that were part of the PC/104 stack. Ultimately, the Arduino Mega 2560 was moved forward with due to the familiarity with Simulink and success with this hardware.

7.2 Future Work

The bench testing of the prosthesis showed that this type of controller is possible to implement. Because the Marquette Prosthesis II could not be mounted to an amputee to ensure the desired stiffness of the ankle was met, the Marquette Prosthesis II does not fully validate stiffness control. Although this work does not develop the complete active transtibial prosthesis, it has developed the project significantly to fully use stiffness control with the ANN in the future. Because the

prosthesis is not fully developed, there are several actions that can be completed in the future which involve implementing the ANN, reoptimization of different components of the prosthesis and a higher-level controller.

7.2.1 Implementation of ANN

The end goal for the Marquette Prosthesis II is to use an ANN to provide the desired stiffness of the ankle within 150ms. To date, the ANN has not been completed but should be developed soon. Once the ANN is completed, it will be integrated with the hardware of the Marquette Prosthesis II to control the stiffness of the ankle on a human subject. When integrating the ANN, it is pivotal to ensure that the Arduino Mega 2560 has the processing capabilities to complete the required computing the ANN requires and control the stiffness within the 150ms window. If the system seems to be lagging, the control hardware used for the Marquette Prosthesis II should be revisited. This lagging could be fixed by selecting a device with a higher sampling rate that could operate at higher frequencies.

7.2.2 Redesign of the Four-Bar Mechanism

The link lengths of the four-bar mechanism for the Marquette Prosthesis II were not revisited as it was out of the scope of this project. The original design was optimized to match an average moment versus time curve with a four-bar

mechanism and parallel linear spiral torsional spring [8]. With the new stiffness controller and a series linear spiral torsional spring, the design of the four-bar should be revisited. A possible optimization of the mechanism can maximize the range of motion of the ankle for applications that require a greater range of motion. Optimizing the link lengths to maximize the range of motion would also utilize the nonlinearity of the mechanism. Due to the nonlinearity of the mechanism, it can provide a larger range of motion for the motor and unique torque profiles that would allow for unique types of gait to be completed by an amputee wearing the prosthesis. The four-bar mechanism can also be redesigned to better incorporate the secondary encoder at the ankle instead of having a 3D printed part to mount this encoder.

7.2.3 Reselection of Linear Spiral Torsional Spring

The linear spiral torsional spring was originally selected to be in parallel such that the Marquette Prosthesis met the torque requirement of the ankle. With it now in series, the selection of the stiffness of the spring can be revisited. A spring with a smaller spring constant could be selected as it would not be as bulky as the current spring; however, a spring with a smaller spring constant has some positive and negative consequences. A smaller spring constant would require the motor to move further and faster to meet the desired stiffness of the ankle. This effect could be beneficial, as the motor's efficiency drops below 50% when it runs below 2000

rpm [21] and the current configuration requires the motor to run below 2000 rpm. However, this effect could also be detrimental as the controller may not be able to keep up with a higher speed. Another consequence is that there are limitations in making a linear spiral torsional spring with a low spring constant. There comes a point when the spacing between the coils of the spring is too small for the spring to function. However, it could also be designed to be smaller and lighter. A possible optimization would optimize the spring so that it is lighter and requires the motor to move above 50% of its maximum speed to increase the efficiency of the motor. Another possible optimization would look to optimize the linear spiral torsional spring with the four-bar mechanism to optimize the range of motion of the ankle as stated in the previous section.

7.2.4 Reselection of Motor

Similar to the four-bar link lengths and the linear spiral torsional spring, the motor used for the Marquette Prosthesis II was originally selected for the Marquette Prosthesis. The motor was originally selected to be a brushed DC motor over a brushless DC motor as it is more efficient, produces a greater nominal and stall torque, and simplifies the control. The particular brushed DC motor was selected based on the overall size and weight. However, the efficiency at low speeds is a downfall to this motor for the Marquette Prosthesis. With the Simulink model

developed for the Marquette Prosthesis II, the characteristics of the motor can be investigated to see how it affects the performance of the system. Further, the range of the speed the motor has to move in order to meet the stiffness of the ankle is known due to this work. Possible optimization of the motor would rely heavily on the four-bar link lengths and the linear spiral torsional spring, as these two components would determine the speed at which the motor would have to travel. To simplify the total optimization, it would be best to constrain the motor first while optimizing the spring and four-bar link lengths and then optimize the motor once those two components have been fully optimized.

7.2.5 Higher Level Controller

The controller used on the motor for the Marquette Prosthesis II is a PI controller, which is a low level type of controller and has its disadvantages. The actual motor position follows the general desired position, however it has a stepping profile due to its reactionary nature. This result is characteristic of a PI controller and is difficult to avoid with this type of controller with an analog signal such as the profile an ankle angle follows. Thus, a higher-level controller, such as a feedforward controller or model predictive controller, would work better to not only control the Marquette Prosthesis II but be able to increase the accuracy of the Marquette Prosthesis II when the amputee has an abrupt transition in gait.

References

- [1] J. Angeles. *Kinetostatics of Serial Robots*. Springer, Boston, MA, 2007.
- [2] Arduino. Arduino Mega 2560 Technical Specifications.
<https://store.arduino.cc/usa/arduino-mega-2560-rev3>, March 2018.
- [3] American Diabetes Association. Statistics about Diabetes.
http://diabetes.org/diabetes-basics/statistics/?utm_source=Offline&utm_medium=Print&utm_content=statistics&utm_campaign=CON&s_src=vanity&s_subsrc=statistics.
- [4] S. Au. *Powered Ankle-Foot Prosthesis for the Improvement of Amputee Walking Economy*. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, June 2007.
- [5] S. Au, P. Bonato, and H. Herr. An EMG-position controlled system for an active ankle-foot prosthesis: an initial experimental study. In *9th International Conference on Rehabilitation Robotics*, pages 375–379, June 2005.
- [6] S. Au, J. Weber, and H. Herr. Powered Ankle-Foot Prosthesis Improves Walking Metabolic Economy. volume 25, pages 51–66, February 2009.
- [7] R. Bellman, M. Holgate, and T. Sugar. SPARKy 3: Design of an Active Robotic Ankle Prosthesis with Two Actuated Degrees of Freedom using Regenerative Kinetics. In *2008 2nd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 511–516, October 2008.
- [8] B. Bergelin. Design of a Transtibial Prosthesis Utilizing Active and Passive Components in Conjunction with a Four-Bar Mechanism. Master’s Thesis, Marquette University, Milwaukee, WI, August 2009.
- [9] Giampiero Campa and MathWorks. Device Drivers. <https://www.mathworks.com/matlabcentral/fileexchange/39354-device-drivers>, September 2016.
- [10] D. Childress. A Myoelectric Three-State Controller using Rate Sensitivity. *Proceedings of the 8th International Conference on Medical and Biological Engineering*, 1969.

- [11] C. DeLuca. Control of upper-limb prostheses: A case for neuroelectric control. *Journal of Medical Engineering & Technology*, 2(2):57–61, March 1978.
- [12] D. Dorcs and R. Scott. A three-state myo-electric control. *Journal of Medical & Biological Engineering*, 4(4):367–370, August 1966.
- [13] K. Englehart, B. Hudgins, and P. Parker. Multifunction Control of Prostheses Using the Myoelectric Signal. In *Intelligent Systems and Technologies in Rehabilitation Engineering*, pages 153–208. CRC Press, Inc., 2000.
- [14] S. Farmer, B. Silver-Thorn, P. Voglewede, and S. Beardsley. Within-socket myoelectric prediction of continuous ankle kinematics for control of a powered transtibial prosthesis. *Journal of Neural Engineering*, 11(5):056027, 2014.
- [15] F. Finley and R. Wirta. Myocoder studies of multiple myopotential response. *Archives of Physical Medicine and Rehabilitation*, 48(11):598–601, November 1967.
- [16] Centers for Disease Control and Prevention. Diabetes Home. <https://www.cdc.gov/diabetes/basics/diabetes.html>, July 2017.
- [17] American Society for Metabolic and Bariatric Surgery. Type 2 Diabetes and Obesity: Twin Epidemics. <https://asmbs.org/wp/uploads/2009/03/Type-2-Diabetes-Fact-Sheet.pdf>, December 2014.
- [18] F. Freudenstein. *Design of Four-link Mechanisms*. Ph. D Dissertation, Columbia University, New York, NY, June 1954.
- [19] M. Goldberg. Death and Injury Rates of U.S. Military Personnel in Iraq. Technical Report 4, 2010.
- [20] J. Hitt, R. Bellman, M. Holgate, T. Sugar, and K. Hollander. The SPARKy (Spring Ankle with Regenerative Kinetics) Project: Design and Analysis of a Robotic Transtibial Prosthesis with Regenerative Kinetics. September 2007.
- [21] J. Hitt, T. Sugar, M. Holgate, and R. Bellman. An Active Foot-Ankle Prosthesis with Biomechanical Energy Regeneration. *Journal of Medical Devices, Transactions of the ASME*, 4(1):011003, March 2010.

- [22] M. Holgate, A. Bohler, and T. Sugar. Control Algorithms for Ankle Robots: A Reflection on the State-of-the-Art and Presentation of Two Novel Algorithms. pages 97–102, October 2008.
- [23] K. Hollander, R. Ilg, T. Sugar, and D. Herring. An Efficient Robotic Tendon for Gait Assistance. *Journal of Biomechanical Engineering*, 128(5):788–791, October 2006.
- [24] B. Hudgins. *A Novel Approach to Multifunction Myoelectric Control of Prostheses*. Ph.D. Dissertation, University of New Brunswick, Brunswick, Canada, 1991.
- [25] Spectrum Digital Incorporated. eZdsp F28335 Technical Reference. http://c2000.spectrumdigital.com/ezf28335/docs/ezdspf28335c_techref.pdf, November 2007.
- [26] G. Klute, J. Czerniecki, and B. Hannaford. Development of powered prosthetic lower limb. *Proceedings of the 1st National Meeting*, October 1998.
- [27] M. Leite and P. Araujo. Relay Methods and Process Reaction Curves: Practical Applications. Technical report, Aracaju, Brazil, February 2012.
- [28] C. De Luca. The Use of Surface Electromyography in Biomechanics. *Journal of Applied Biomechanics*, 13(2):135–163, May 1997.
- [29] P. Malak. Analysis and Synthesis of a Planar Reconfigurable Mechanism with a Variable Joint. Master’s Thesis, Marquette University, Milwaukee, WI, August 2016.
- [30] J. Mizrahi. Mechanical Impedance and Its Relations to Motor Control, Limb Dynamics, and Motion Biomechanics. *Journal of Medical and Biological Engineering*, 35(1):1–20, January 2015.
- [31] Maxon Motor. Epos2 Positioning Controllers Summary. https://www.servo.com.sg/sites/default/files/2018-01/EPOS_Program_2017-18.pdf, May 2017.
- [32] R. Norton. *Design of Machinery: an Introduction to Synthesis and Analysis of Mechanisms and Machines*. McGraw-Hill, New York, NY, 2012.

- [33] National Institute of Diabetes, Digestive, and Kidney Diseases. Overweight & Obesity Statistics. <https://www.niddk.nih.gov/health-information/health-statistics/overweight-obesity>, August 2017.
- [34] PC/104 Consortium. Purpose. <https://pc104.org/consortium/purpose/>, December 2017.
- [35] A. Pollit and E. Bizzi. Characteristics of motor programs underlying arm movements in monkeys. *Journal of Neurophysiology*, 42(1):183–194, January 1979.
- [36] D. Quintero, D. Villarreal, D. Lambert, S. Kapp, and R. Gregg. Continuous-Phase Control of a Powered Knee-Ankle Prosthesis: Amputee Experiments Across Speeds and Inclines. *IEEE Transactions on Robotics*, 34(3):686–701, June 2018.
- [37] R. Reiter. Eine neu elektrokunstant. *Grenzgebiete der Medicin*, 1(4):133–135, 1948.
- [38] R. Scott, R. Brittain, R. Caldwell, A. Cameron, and V. Dunfield. Sensory-feedback system compatible with myoelectric control. *Medical and Biological Engineering and Computing*, 18(1):65–69, January 1980.
- [39] G. Sullivan, M. Mitchell, J. Lyman, and F. DeDiasto. Myoelectric servo control. Technical report, March 1964.
- [40] J. Sun. Powered Transtibial Prosthetic Device Control System Design, Implementation and Testing. Master’s Thesis, Marquette University, Milwaukee, WI, August 2012.
- [41] F. Sup. Design and Control of a Powered Transfemoral Prosthesis. Master’s Thesis, Vanderbilt University, Nashville, TN, December 2006.
- [42] F. Sup, A. Bohara, and M. Goldfarb. Design and Control of a Powered Transfemoral Prosthesis. *The International Journal of Robotics Research*, 27(2):263–273, February 2008.
- [43] F. Sup, H. Varol, J. Mitchell, T. Withrow, and M. Goldfarb. Preliminary Evaluations of a Self-Contained Anthropomorphic Transemoral Prosthesis.

- IEEE/ASME Transactions on Mechatronics*, 14(6):667–676, December 2009.
- [44] F. Sup, H. Varol, J. Mitchell, T. Withrow, and M. Goldfarb. Self-Contained Powered Knee and Ankle Prosthesis: Initial Evaluation on a Transfemoral Amputee. pages 638–644, June 2009.
- [45] H. Varol, F. Sup, and M. Goldfarb. Multiclass Real-Time Intent Recognition of a Powered Lower Limb Prosthesis. *IEEE Transactions on Biomedical Engineering*, 57(3):542–551, March 2010.
- [46] D. Winter. *Biomechanics and Motor Control of Human Movement*. Wiley-Interscience, Hoboken, NJ, 1990.
- [47] R. Wirta, D. Taylor, and F. Finley. Pattern-recognition arm prosthesis: A historical perspective - a final report. *Bulletin of Prosthetic Research*, pages 8–35, Fall 1978.
- [48] H. Zadaca, J. Lyman, and A. Freedy. Studies and Development of Heuristic Endpoint Control for Artificial Upper Limbs. Biotechnology Laboratory Technical Report 54, University of California, Los Angeles, School of Engineering and Applied Science Report UCLA-ENG-74-79, 1974.
- [49] K. Ziegler-Graham, E. MacKenzie, P. Ephraim, T. Trivison, and R. Brookmeyer. Estimating the Prevalence of Limb Loss in the United States: 2005 to 2050. *Archives of Physical Medicine and Rehabilitation*, 89(3):422–429, March 2008.

APPENDIX A

Four Bar Kinematics

The Vector Loop Method is used to analyze the motion of mechanisms by expressing links as vectors, summing the links and breaking the summation into the appropriate components to yield the relationship between the angles of the mechanism. This appendix shows the derivation of the relationship between the angles of the four-bar mechanism based on the vector loop method. This derivation was completed such that the input angle of the four-bar mechanism, ϕ , is determined based on the desired ankle angle, θ . This derivation follows the derivation found in Norton [32].

Fig. A.1 shows a sketch of the four-bar mechanism with the appropriate angles identified and the moving coordinate system defined. Each link is considered to be a vector whose direction is indicated by an arrowhead.

The summation of the linkage vectors should be zero. Therefore,

$$\vec{l}_0 + \vec{l}_3 - \vec{l}_1 - \vec{l}_2 = 0 \quad (\text{A.1})$$

Based upon the XY coordinate system and the angle notation, Equation A.1 is broken into X and Y components, respectively:

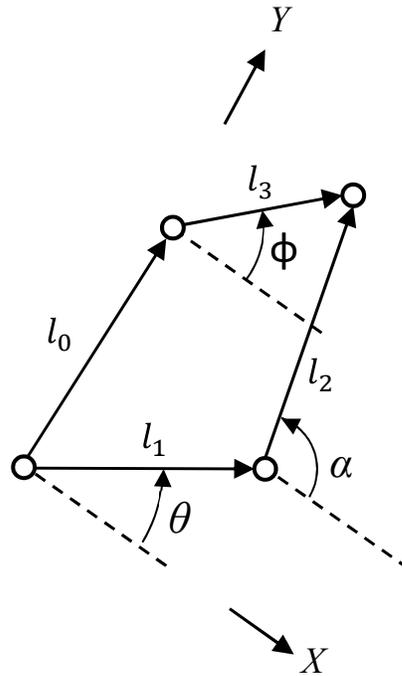


Figure A.1: Sketch of the Four-Bar Mechanism (Identical to Fig. 3.2)

$$l_1 \cos \theta + l_2 \cos \alpha - l_3 \cos \phi = 0 \quad (\text{A.2})$$

$$l_1 \sin \theta + l_2 \sin \alpha - l_3 \sin \phi - l_0 = 0 \quad (\text{A.3})$$

Isolating the α terms in both equations, squaring and then adding the equations yields:

$$l_2^2 = l_1^2 + l_3^2 + l_0^2 - 2l_1l_3(\cos \theta \cos \phi + \sin \theta \sin \phi) - 2l_1l_0 \sin \theta + 2l_3l_0 \sin \phi \sin \theta + l_2 \sin \alpha - l_3 \sin \phi - l_0 \quad (\text{A.4})$$

To further simplify Equation A.4, the constants K_1 , K_2 and K_3 are defined in terms of the constant link lengths:

$$K_1 = \frac{l_0}{l_1}, \quad K_2 = \frac{l_0}{l_3}, \quad K_3 = \frac{l_0^2 + l_1^2 - l_2^2 + l_3^2}{2l_1l_3}$$

With these lengths defined, Equation A.4 can be rewritten as:

$$K_1 \sin \phi - K_2 \sin \theta + K_3 = \sin \theta \sin \phi + \cos \theta \cos \phi \quad (\text{A.5})$$

In order to eliminate the number of trigonometric functions, the trigonometric identity:

$$\cos \theta - \phi = \sin \theta \sin \phi + \cos \theta \cos \phi$$

is substituted into Equation A.5 and yields the form known as Freudenstein's equation [18]:

$$K_1 \cos \phi - K_2 \cos \theta + K_3 = \cos(\theta - \phi) \quad (\text{A.6})$$

In order to reduce Equation A.6 to a more tractable form, it is useful to substitute the half angle identities which converts the $\sin \phi$ and $\cos \phi$ terms to $\tan \phi$ terms:

$$\sin \phi = \frac{2 \tan(\frac{\phi}{2})}{1 + \tan^2(\frac{\phi}{2})}, \quad \cos \phi = \frac{1 - \tan^2(\frac{\phi}{2})}{1 + \tan^2(\frac{\phi}{2})} \quad (\text{A.7})$$

This substitution results in the following simplified form, where the link lengths and known input value θ terms have been collected as constants A, B and C:

$$A \tan^2\left(\frac{\phi}{2}\right) + B \tan\left(\frac{\phi}{2}\right) + C = 0 \quad (\text{A.8})$$

where:

$$A = \cos \theta - K_1 - K_2 \cos \theta + K_3$$

$$B = -2 \sin \theta$$

$$C = K_1 - (K_2 + 1) \cos \theta + K_3$$

Equation A.8 has a quadratic form and the solution for the angle can be found by using the quadratic equation:

$$\begin{aligned} \tan\left(\frac{\phi}{2}\right) &= \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \\ \phi_{1,2} &= 2 \arctan\left(\frac{-B \pm \sqrt{B^2 - 4AC}}{2A}\right). \end{aligned} \quad (\text{A.9})$$

APPENDIX B

Four Bar Quasi-statics

With the input angle of the four-bar mechanism derived in Appendix A, the motor position was determined via quasi-static relationships. This Appendix determines the quasi-static torque required by the spring to meet the required ankle torque. The desired motor position was determined from the derivations in Appendices A and B such that the required stiffness of the ankle was met.

Fig. B.1 shows a sketch of the four-bar mechanism with the appropriate angles identified and the coordinate system defined. The free body diagram of each link was drawn at a generic position to sum the forces and moments on links 1, 2 and 3. The free body diagrams are seen in Figs. B.2, B.3 and B.4.

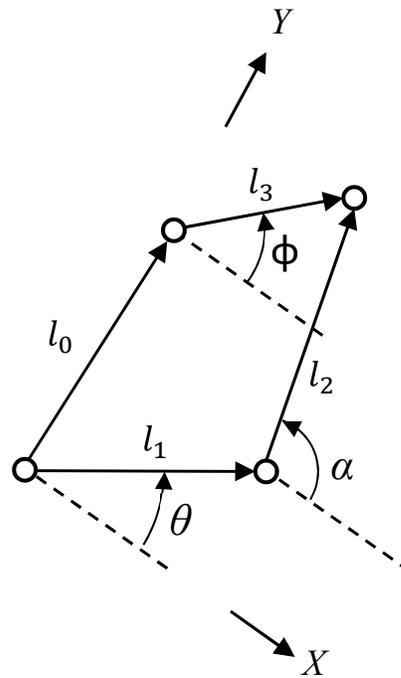


Figure B.1: Sketch of the Four-Bar Mechanism (Identical to Fig. 3.2)

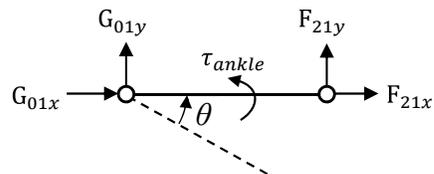


Figure B.2: Free Body Diagram of Link 1

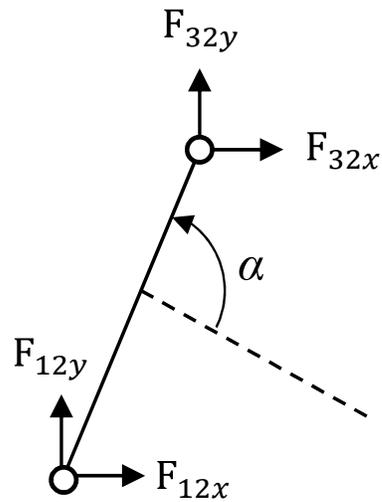


Figure B.3: Free Body Diagram of Link 2

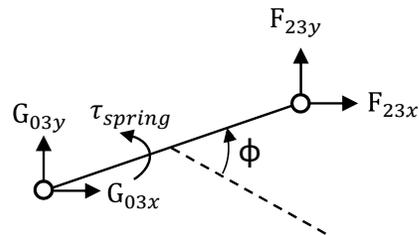


Figure B.4: Free Body Diagram of Link 3

With the free body diagram drawn for each linkage, a static analysis was completed by summing the forces in the X and Y directions for each link and setting these equations equal to zero as well as by summing the moments about a point on each linkage and setting these equations equal to zero. From the statics, Equations B.1-B.9 were obtained. From Newton's third law, Equations B.10-B.13 were

obtained. These thirteen equations could be solved simultaneously such that the required spring torque was determined.

$$G_{01x} + F_{21x} = 0 \quad (\text{B.1})$$

$$G_{01y} + F_{21y} = 0 \quad (\text{B.2})$$

$$F_{21x}l_1 \sin \theta + F_{21y}l_1 \cos \theta + \tau_{ankle} = 0 \quad (\text{B.3})$$

$$F_{32x} + F_{12x} = 0 \quad (\text{B.4})$$

$$F_{32y} + F_{12y} = 0 \quad (\text{B.5})$$

$$F_{32x}l_2 \sin \alpha + F_{32y}l_2 \cos \alpha = 0 \quad (\text{B.6})$$

$$F_{23x} + G_{03x} = 0 \quad (\text{B.7})$$

$$F_{23y} + G_{03y} = 0 \quad (\text{B.8})$$

$$F_{23x}l_3 \sin \phi + F_{23y}l_3 \cos \phi + \tau_{spring} = 0 \quad (\text{B.9})$$

$$F_{12x} = -F_{21x} \quad (\text{B.10})$$

$$F_{12y} = -F_{21y} \quad (\text{B.11})$$

$$F_{23x} = -F_{32x} \quad (\text{B.12})$$

$$F_{32y} = -F_{23y} \tag{B.13}$$

APPENDIX C

Simulink Model

This Appendix explains the creation of the Simulink Model. This Appendix breaks down the model into its three major sections: the input to the four-bar mechanism, the four-bar mechanism, and the connection between the motor and the spring. It then discusses the connection between the motor and spring, explaining how the vector loop method and quasi-static analysis were implemented in the model as well as the complete model of the brushed DC motor.

With the static and dynamic relationships established, the system was modeled in Simulink to ensure the required position and torque of the ankle were satisfied. The Simulink model is seen in Fig. C.1. The figure is broken up into three sections, A, B and C.

The first is section A labeled “Desired Position Input.” To actuate the four-bar linkage, a desired position, velocity and acceleration were input into the ankle joint based on published data [46]. The velocity and acceleration were determined by taking the derivative with respect to time of the position and velocity, respectively.

The second section is section B labeled “Four Bar Mechanism.” The

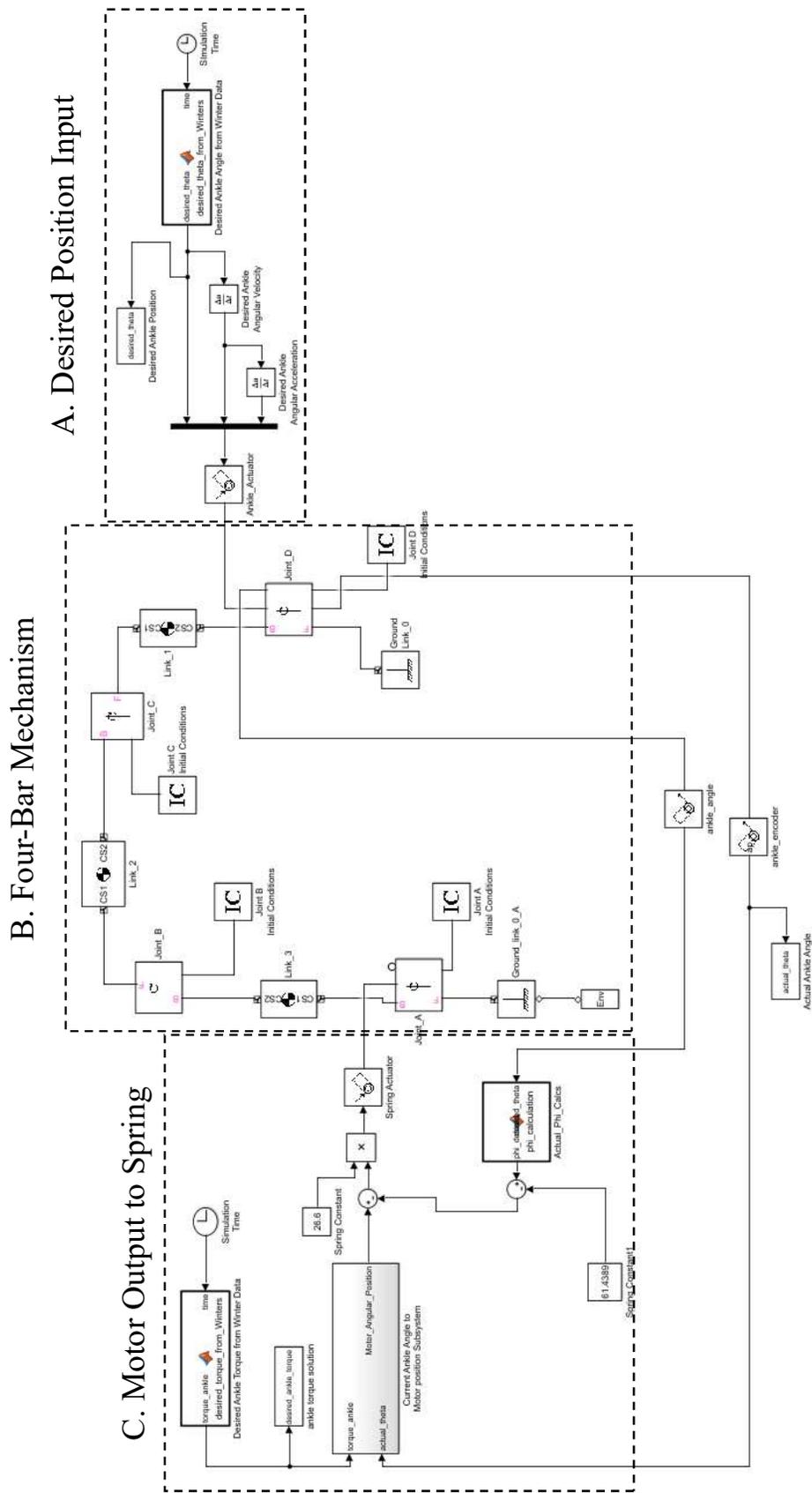


Figure C.1: Simulink Model of System

SimScape Multibody animation is seen in Fig. C.2. The four-bar mechanism was built using SimMechanics toolbox. Note that link 0 is not shown, however two of the joints are grounded, representing the grounded link, l_0 .

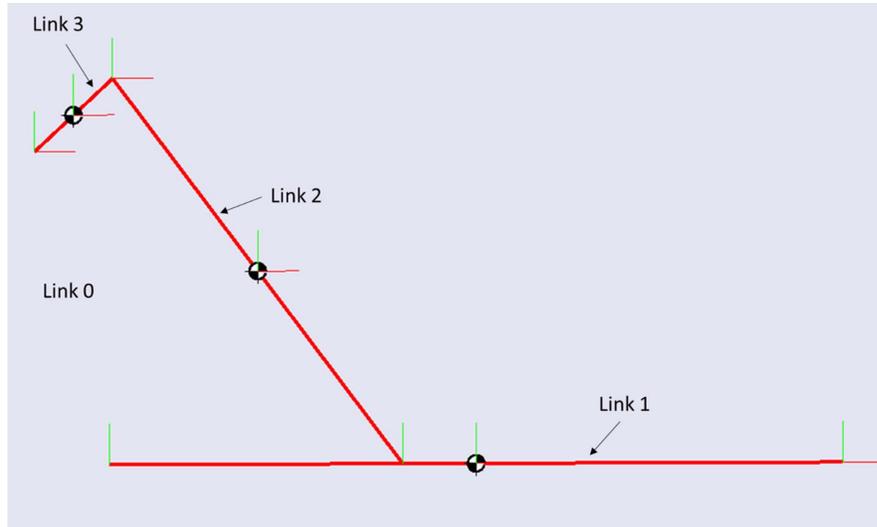


Figure C.2: Simscape Multibody Simulation

The last section is section C labeled “Motor Output to Spring.” This section incorporates much of the dynamics discussed earlier and outputs the actual motor position, ψ , to the spring, which actuates Joint A. There is a subsection called “Current Ankle Angle to Motor Position Subsystem” that has two inputs: the position and torque at the ankle joint. The blocks that make up the subsystem can be seen in Fig. C.3, which has also been broken down into four sections, i, ii, iii and iv. The first boxed section of this subsection is section i labeled “Angle Calculations via Vector Loop.” This section calculates the input angle of the four-bar

mechanism, ϕ , and the angle from the x-axis to link 2, α , based on the relationships established by the vector loop method. The next boxed section is section ii labeled “Quasi-statics.” This section calculates what the desired motor position, ψ , should be based on the calculated angles from the vector loop method using the quasi-static relationships. The desired motor position, ψ , is then fed into the final boxed section, which is section iii labeled “Motor Model and Controller.” This section contains the generic motor block diagram while the load torque due to the gearbox and spring. From this motor model, the subsection outputs the actual motor angle, ψ . Furthermore, the motor torque was obtained from the motor model and multiplied by the gear ratio to calculate the actual spring torque, τ_{spring} . The three angles and the spring torque were then used to calculate the ankle torque, τ_{ankle} . Joint A was actuated using a Simmechanics Spring Actuator based on the relationship shown in Equation C.1. Furthermore, the P and I gains of the controller were tuned via a modified Ziegler-Nichols method such that the actual motor position matched the desired motor position.

$$\psi = \frac{\tau_{spring}}{K_{spring}} + \phi \quad (C.1)$$

In boxed Section C in Fig. C.1, the MATLAB function block “Actual_Phi_Calcs” calculated the current input angle of the four-bar mechanism, ϕ , based on the current angle of the ankle, θ , using the vector loop method. With

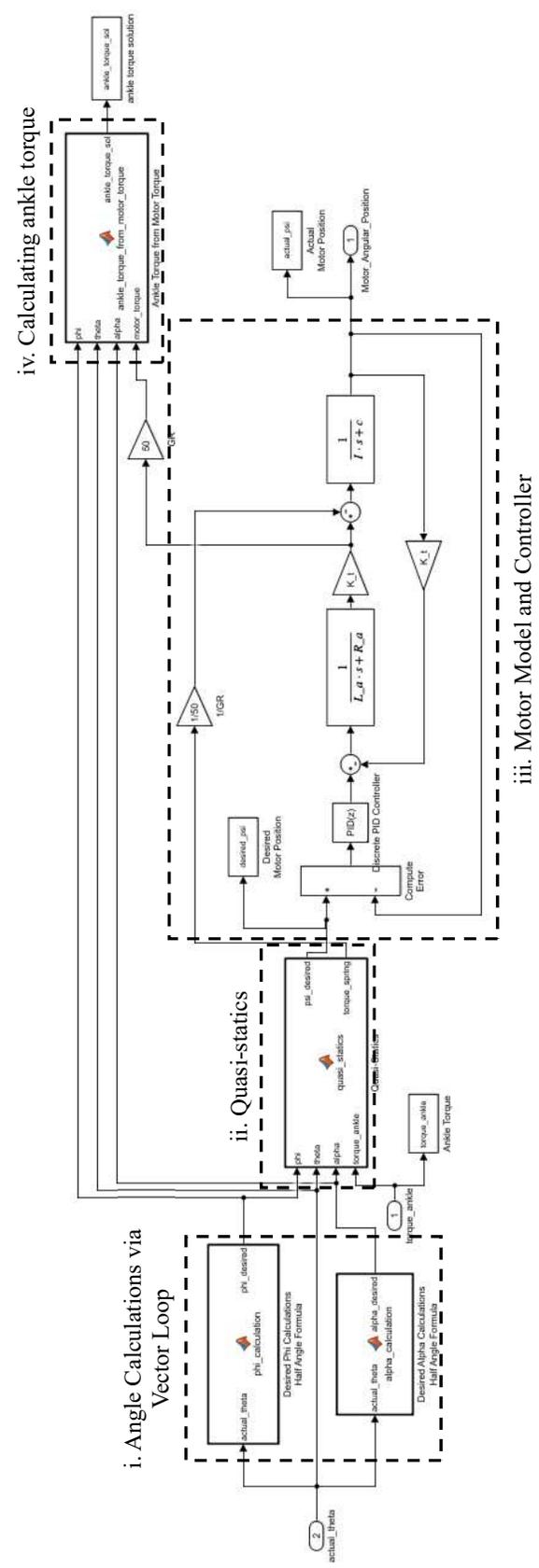


Figure C.3: Current Angle to Motor Position Subsystem

the position of the motor, ψ , input angle of the four-bar mechanism, ϕ , and the spring constant, K_{spring} , defined, the actual torque of the ankle was determined according to Equation C.1 to actuate Joint A where the spring is located.

APPENDIX D

Control Block Diagram

To generate a control signal with an Arduino, a block diagram was created in Simulink. Simulink contains support packages that allow users to interact with the different ports of the Arduino, i.e., PWM pins, Digital Input pins, Digital Output pins, etc. Simulink also allows the user to interact with the control system in real-time using dashboard features such as edit boxes and switches. Finally, real-time data can be viewed on via a scope in Simulink and this scope can output the data to MATLAB's workspace.

The control block diagram for the stiffness controller is shown in Fig. D.1. As with any control system, there were four major features to the controller: the input, the feedback, the controller and the output. The input to the system was published data [46]. The desired position according to Winter was input into the positive side of the addition block in Fig. D.1. For implementation with the ANN, this signal can be replaced with an "Artificial Neural Network" found in Simulink's library, making the implementation fairly simple. To get a reading from the HEDL 5540 encoder, an s-function was created to output the position of the encoder based on the A and B channel readings [9]. From here, the signal was converted and input

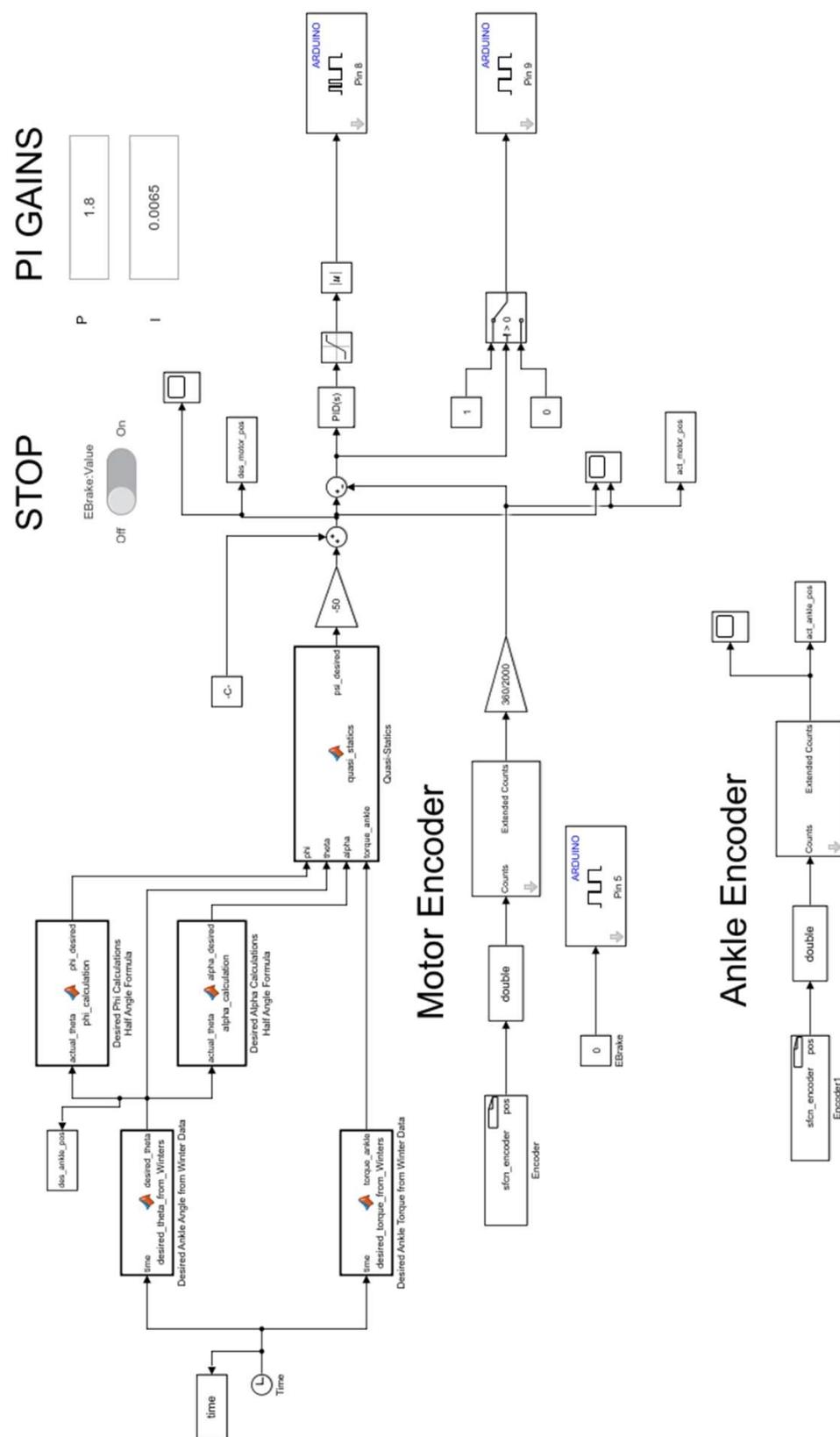


Figure D.1: Simulink Control Block Diagram for Stiffness Control

into an “extended counts” block which allowed Simulink to record more data points from the encoder. Finally, the reading from the encoder was converted to the position via a gain block. The PI controller was implemented via a PID block. The difference between the actual and desired position, or the error, was fed into the PID block, which corrects the error. To tune the PI gains at a faster rate, edit blocks were added to the block diagram to change the gains on the fly. The control signal was then input into a saturation block which has a maximum of 255 and a minimum of -255. The Arduino’s digital output is rated at 8-bits, thus it can receive data from 0 to 255, a total of 256 data points or two raised to the eighth power. This control signal was then input into an absolute value block and outputs the appropriate PWM signal. To determine direction, the sign of the error was determined. The error was input into a switch block which output a HIGH or a LOW signal to the H-Bridge depending on the sign of the error. As a safety feature, an emergency brake was included in the control system. By clicking a slider switch, the user can quickly stop the system.

APPENDIX E

Calculating Ankle Moment from Force Plate and Kinematic Data

The moment generated at the ankle can be determined from the forces generated in the X and Y direction on the force plate and from kinematic data. By putting point trackers on a foot, the acceleration of the ankle in the X and Y direction could be determined. Summing the forces in the X and Y direction yields the ankle reaction forces in these directions, respectively. With the ankle reaction forces, the moment can be summed around the center of mass of the foot to determine the moment generated by the ankle. This derivation follows the derivation found in [46].

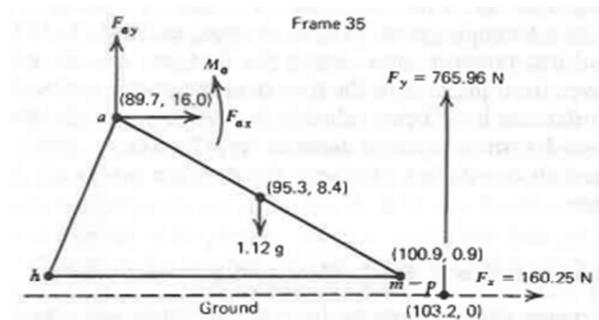


Figure E.1: Free-body diagram of foot during weight bearing with ground reaction forces and example values [46]

Summing forces in the x-direction,

$$\begin{aligned}\sum F_x &= ma_x \\ F_{ax} + F_x &= ma_x \\ F_{ax} &= ma_x - F_x\end{aligned}$$

Summing forces in the y-direction,

$$\begin{aligned}\sum F_y &= ma_y \\ F_{ay} + F_y &= ma_y \\ F_{ay} &= ma_y - F_y - mg\end{aligned}$$

Summing the moments about the center of the mass of the foot,

$$\begin{aligned}\sum M &= I\alpha \\ M_a + F_x(p_y - c_y) + F_y(p_x - c_x) - F_{ax}(a_y - c_y) - F_{ay}(a_x - c_x) &= I\alpha \\ M_a &= I\alpha + F_{ax}(a_y - c_y) + F_{ay}(a_x - c_x) - F_x(p_y - c_y) - F_y(p_x - c_x)\end{aligned}$$