2023-05-01

Development of a Cost-Constrained Intelligent Prosthetic Knee with Real-Time Machine Learning, Predictive Stumble Control

Lucas Jonathan Galey
University of Texas at El Paso

DEVELOPMENT OF A COST-CONSTRAINED INTELLIGENT PROSTHETIC KNEE WITH REAL-TIME MACHINE LEARNING, PREDICTIVE STUMBLE CONTROL

LUCAS JONATHAN GALEY

Doctoral Program in Biomedical Engineering

APPROVED:

Roger V. Gonzalez, Ph.D., Chair

Olac Fuentes, Ph.D.

Hector Erives-Contreras, Ph.D.

Stephen L. Crites, Jr., Ph.D.
Dean of the Graduate School

Dedication

This work is dedicated to my parents, Tom and Gabi Galey; my wife, Gina Galey; and my Lord, Jesus Christ. My parents have never ceased to encourage my work and have given me examples in both word and deed of how to lead lives of excellence and hard work. Gina has been a support emotionally, physically, financially, and technically throughout this journey. She has been by my side for almost 8 years of marriage, and now I'll be done with school finally. Lastly, none of this work would have been possible without the sustaining power of God. I dedicate this and all future work in service to Him and His glory.

DEVELOPMENT OF A COST-CONSTRAINED INTELLIGENT PROSTHETIC KNEE WITH REAL-TIME MACHINE LEARNING, PREDICTIVE STUMBLE CONTROL

by

LUCAS JONATHAN GALEY, MS

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Department of Metallurgical, Materials and Biomedical Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

May 2023

Acknowledgements

This work is the culmination of many years with input from many people. I would never have completed this program without all the support that I have received.

My committee has been incredibly helpful. They helped shape the vision of this research, and their expertise was instrumental in completing it. Foremost, I thank Dr. Roger Gonzalez, the chair of my committee and my academic adviser. He has inspired and mentored me through this academic journey, and though his standards have been high, he has also been kind. Dr. Olac Fuentes opened my eyes to the world of machine learning through his classes and feedback. Dr. Hector Erives-Contreras shed light on the confusing world of control systems through our individual studies course. Finally, Dr. Thompson Sarkodie-Gyan helped shape the biomechatronic aspects through the beginning of this work, though he retired before its completion.

The research community at UTEP was influential and provided much insight and feedback. Dr. Joshua Green and Dr. J Slager both preceded me in finishing their doctorates but were present and helpful throughout it. Large portions of this work were supported by undergraduate students, and it would not have been possible without them. Guillermo Beckman, Ethan Ramos, Luis Aranda, and Luis Armenta were instrumental in manufacturing and testing the prototype knee systems. Special thanks also to Bob Schulz and Tim Perrault for machining prototype components.

Testing the prosthetic knee would not have been possible without the support of prosthetists, clinics, labs, and patients. I thank prosthetists Mundo Corchado CP and Sherie Ford CP, from Hanger Clinics for their assistance with the initial data collection. I also thank Dr. Jason Boyle for allowing me to use his lab many times for this purpose. During the deployed trials of Aim 4, I thank UTSouthwestern's Tiffany Graham CP and Jan Petric CP, and UTSouthwestern for the use of the rehabilitation space. I thank all the patients who have participated in my studies over

the years. This work would not be possible without them, and I hope that they will someday benefit from it.

My wife, Gina, has not only been a source of strength and support, but also a fantastic technical editor. Her expertise in machine learning was the spark that started this research, and her continued feedback and edits helped round this work into something worthy.

Finally, I would like to thank my God and Savior, Jesus Christ, for only by Him and through Him did I find the strength to finish this dissertation.

"Oh, the depth of the riches both of the wisdom and knowledge of God! How unsearchable are His judgments and unfathomable His ways! For who has known the mind of the Lord, or who became His counselor? Or who has first given to Him that it might be paid back to him again? For from Him and through Him and to Him are all things. To Him be the glory forever. Amen." (Romans 11:33-36 NASB95)

Abstract

The field of biomechatronics is evolving quickly with advances in computer science, biology, and electrical and mechanical engineering. Coupled with increased interests in machine learning (ML) across all industry sectors, there are opportunities to leverage advanced analytics in uniquely complex problems. This study aimed to deploy real-time ML predictions in a novel microprocessor-controlled prosthetic knee (MPK) device capable of identifying and responding to stumble-events to reduce amputee fall prevalence. Innately, stumbling is a chaotic event. Current MPKs operate by detecting gait characteristics and reacting to preprogrammed states. While these systems are beneficial in significant ways, such as energy expenditure and stability, chaotic events can mislead traditional gait interpretation methods. A novel method was designed to implement an ML model capable of predicting amputee stumble occurrences using Long-Short Term Memory (LSTM) architecture on three-dimensional acceleration and velocity obtained from inertial measurement units (IMUs). This innovative approach had four main aims: (1) develop a cost-constrained prototype MPK, the GKnee; (2) collect *in vivo* stumble-induction data and train an ML model for the unique use-case; (3) create a control system to use the ML model to incorporate into the GKnee; and (4) combine and test components in a comprehensive stumble-induction study against a passive mechanical knee and industry standard MPKs.

For the first aim, the prototype GKnee was designed to bear a 100 kg patient during heavy loading activities with an additional safety factor (three times body weight). The developed prosthetic knee does not limit flexion and dampening does not prevent normal leg swing. It was equipped with accelerometers, gyroscopes, and a servo to allow data collection and microprocessor control. The final system cost was $1,392, which was approximately $900 above the hypothesized cost.

Towards the second aim, an initial patient trial was conducted with three subjects and each was induced to stumble under three different methods: bungee (n=42), obstacle (n=43), and uneven surface (n=65). With an IMU-equipped knee, the data from these trials was recorded and labeled for ML training. An LSTM ML model was then developed to classify the data and achieved an average step accuracy of 66.9%.

Building on the ML model, the third aim involved developing a novel control system that switched between "walking" and "stumbling" conditions in real-time to control the GKnee with an electronically actuated hydraulic system. The control system was able to change between true states in less than 0.15 seconds and switch out of false states in less than 0.034 seconds. Additionally, the system interpreted and improved the classification rates of the ML model and increased the step accuracy of the system to 91.4%.

As part of the fourth aim, a final five-subject *in vivo* trial was conducted that compared the GKnee to industry standard MPK systems and a mechanical passive system during stumble inductions. A total of 500 stumbles were induced across all patients, knees, and stumble induction modes. With a statistical significance of $p = 0.0422$, the GKnee was found to have a large effect improvement over both the MPK (Cohen's $d = 1.63$) and M3 (Cohen's $d = 1.91$) in stumble recovery rate. An error during data collection made it clear that this improvement resulted purely from the mechanical portion of the GKnee with no active control methods. Thus, while the ML model and control system performed according to their objectives, they ultimately had no effect on the improved recovery rate during testing.

Overall, this research contributes an in-depth, transfemoral-amputee stumble dataset with an effective ML model network while also presenting a cost-constrained prosthetic knee prototype

that effectively reduces amputee fall incidence rate during treadmill induced stumbles compared to industry standard devices.

Table of Contents

List of Tables

List of Figures

List of Equations

Chapter 1: Introduction

Limb loss affects 36 to 58 million people globally, with transfemoral (TF) amputations affecting approximately 25% of that population and growing by 44,400 per year within the United States (Amputee Coalition, 2012; Center for Orthotic & Prosthetic Care, 2008; Michael, 2001; WHO, 2011). The current commercial prosthetic knee market ranges from low-cost, purely mechanical devices to high-end electronic prosthetics. However, an affordable, technology-rich prosthetic knee does not currently exist. Studies have shown that microprocessor-controlled prosthetics are more adaptable to individual walking patterns, require significantly lower patient exertion, and reduce complications when walking compared to passive mechanical systems (Hafner et al., 2007; Johansson et al., 2005). Despite these significant advantages over simple passive knees, the cost of electronic systems keeps this technology out of reach for the majority of amputees. To the subject, microprocessor knees start around $30,000, while more advanced models easily reach $80,000. With a US median income of $67,521 in 2020, such systems cost 44% and 118% of a yearly wage, respectively (Shrider et al., 2021). Health insurance provides these systems to many, but for the less fortunate in the US and other countries, no affordable advanced system exists.

Nevertheless, even subjects wearing electronic systems experience periodic stumbles and falls. Traditional electronic systems use state programming to determine changes in activity. With the chaotic nature of stumbles and variability of normal gait, false negatives and false positives can occur and result in a fall. The following research proposes a combination of machine learning, control systems, and in vivo testing to create a low-cost microprocessor prosthetic knee that will identify and prevent stumbles, adapt to patient gait, and be comparable to high-tech systems at less than 5% of the cost.

The purpose of prosthetic knees is to restore functionality to patients by mimicking the articulation of natural limb joints. Current knee systems accomplish this goal admirably, but more work needs to be done, especially in fall prevention. In the United States, a hospitalizing fall can have an average direct medical cost of $25,600 (Mundell et al., 2017). Fear of falling affects approximately half of lower limb amputees (Miller et al., 2001). Statistically, even though active knees show an improvement over passive knees in falling, almost all patients still fall (Hafner et al., 2007; Kahle et al., 2008), and more than 50% of lower limb patients report falling at least once a year (Highsmith et al., 2010). Current programming of active knees (like MPKs) moves between the discrete states of gait based on the system inputs (Torrealba & Fonseca-Rojas, 2019; Wen et al., 2017). However, falling is a naturally chaotic event. Therefore, a system is needed that will more accurately predict stumbles to prevent falls. One solution is to use non-discrete programming by drawing conclusions from sensor data in conjunction with machine learning methods.

1.1.1 Stumble Literature

Stumbles have been studied for many years with various levels of intensity. To-date, there are eight stumble studies involving able-bodied subjects (Cordero et al., 2004a; Forner-Cordero et al., 2011; Grabiner et al., 1993; Hajj Chehade et al., 2012; King et al., 2019; Lawson et al., 2010a; Schillings et al., 1996a; Yoo et al., 2019). Though some numbers were difficult to ascertain from methods, the number of subjects in these studies ranged from 4 to 18 (avg: 8 ±4), and each subject stumbled between 1 and 27 times (avg: 11 ±9). The most common methods of stumble induction were an obstacle, a line, or a treadmill perturbation. For the most part, these were set at either a preprogrammed position or gait phase. Other than Eveld et al., which proposed a method for inducing stumbles, the studies conflated perturbation with stumbling. Eveld et al. indicated attempted versus induced stumble perturbations (Eveld et al., 2021).

Among TF amputees, there are more variables to consider, like knee type and stumble side (intact or prosthetic). Fewer stumble studies were found for TF amputees, and the methodology was often not clearly delineated. While there are some assumptions with the able-bodies subject trials, several of the amputee subject trials have stumbling time or summative stumbles. The studies had on average nine subjects (±6) but were often mixed between TF and able-bodies subjects (Bellmann et al., 2019; Blumentritt et al., 2009a; Crenshaw et al., 2013; Hak et al., 2013; Highsmith et al., 2014; K. Kaufman et al., 2008; Sessoms et al., 2014; Shirota et al., 2015). Outside of two studies, the average amount of stumbles was six (±4), and repeatability is further reduced by the stumbles being split across leg sides and types of knees. Those two studies, though both still with mixed subject pools (including transtibial), performed a large quantity of induced stumbles (Shawen et al., 2017; Shirota et al., 2015). With the study by Shirota et al. having high levels of repeatability of 36 stumbles per amputee per side with only one knee tested. The study by Shawen et al. was much less repeatable with stumble induction methods ranging from self-induced tripping to poorly defined pushes from the side, but still inducing stumbles 36 times per amputee. These two studies offer a large pool of stumble inductions but are both limited in variance of controlled induction type and in scope of knee systems compared repeatably.

1.1.2 Research Scope

To address both the cost of MPKs and the potential weaknesses of state-based programming, the overarching objective of this research was to develop a stumble prediction machine learning (ML) algorithm that would be deployed in a low-cost prosthetic knee. The project was thus broken into four sections: development of a knee, development of an ML algorithm to detect stumbles, integration of both knee and ML model in a control system, and testing and evaluation of the completed system. Specific aims are given in Chapter 2.

Further, each section required its own development phases, data collection, and evaluation. The knee development, Aim 1, includes mathematical modeling and analysis, experimental evaluation, material sourcing, and prototype evaluation. Novel data was collected to develop the ML algorithm for Aim 2. This aim includes data collection, multiple ML models, and model evaluations. In Aim 3, both Aim 1 and 2 were brought together in a control system. This aim focuses on the implementation methods and subsequent effect on the stumble classification and reaction. The final evaluation of all previous aims working in conjunction are discussed in Aim 4. This aim evaluates a passive system, an MPK, and the developed prototype in a comparative stumble induction study to ascertain the rate of recovery compared with falls with each system.

Therefore, the data collection trials with TF amputees can be split into three separate trials: initial data collection (Aim 2), knee evaluation and comparison (Aim 4), and GKnee control system recollection (Aim 4). The subject pool for the Aim 2 trial is separate from that of Aim 4. Throughout this paper, subjects from the Aim 2 trial will have the prefix “S”, e.g., S1 as the first subject in the Aim 2 trials; and those of Aim 4 will have the prefix “P”, e.g., P1 as the first patient in the Aim 4 trials. “S” and “P” were arbitrarily derived from “Subject” and “Patient”. In both sets of trials, a patient consented, but was unable to complete the trial. Thus, both trials have a subject/patient number that is skipped (S3 and P5).

Chapter 2: Aims and Objectives

The purpose of the proposed research is to develop and test an affordable intelligent prosthetic knee with patient adaptation and stumble control through the use of machine learning software and microprocessor-controls. Interconnectivity of the aims can be seen in Figure 2.1 below.

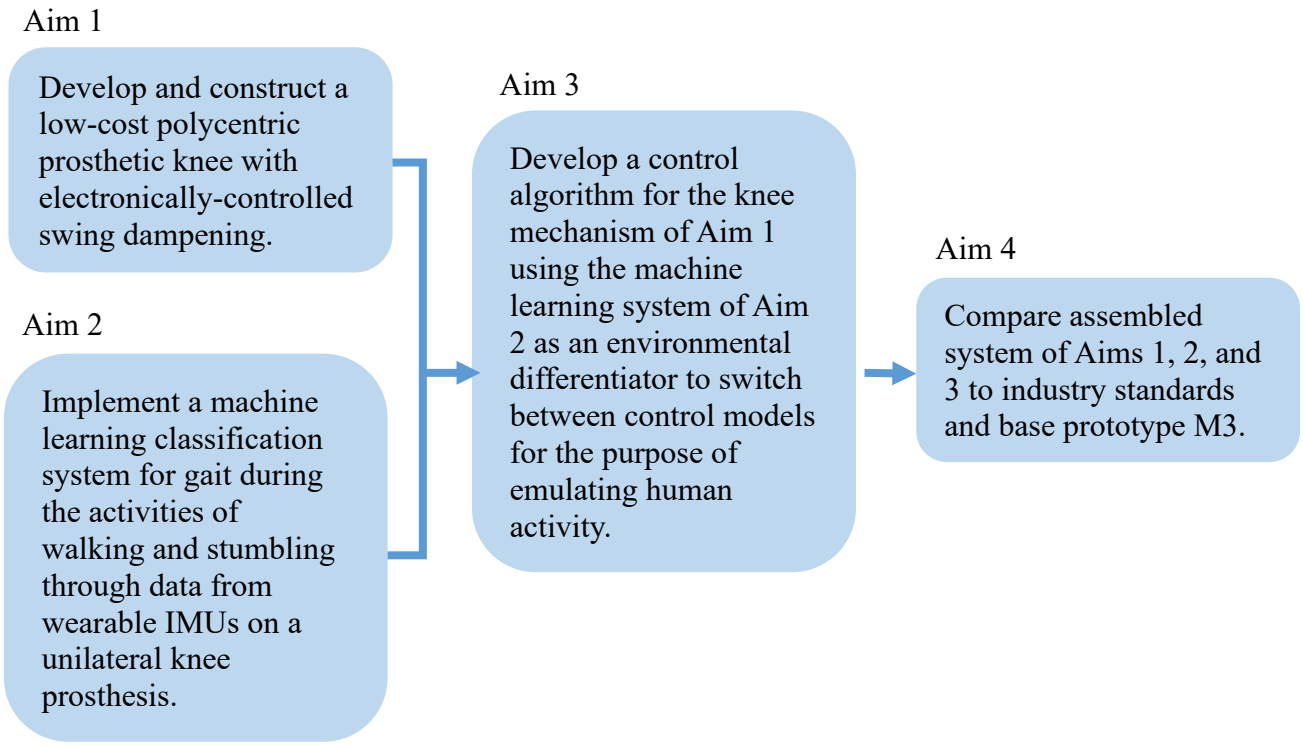

Figure 2.1: Pictographical layout of aims.

Aim 1: Develop and construct a low-cost polycentric prosthetic knee with electronically-controlled swing dampening.

For this aim, it is hypothesized that the developed prosthetic knee will support a 100 kg patient, allow knee swing up to 500 degrees per second, and have a component cost less than $500. The primary objectives are to optimize and construct an electronic damper that interfaces with existing knee design in the most efficient manner, so that it may be used to control the knee by limiting flexion.

Aim 2: Implement a machine learning classification system for gait during the activities of walking and stumbling through data from wearable IMUs on a unilateral knee prosthesis.

It is hypothesized that the system will achieve an activity classification accuracy greater than 90% and detect stumble with a precision greater than 75% and a recall greater than 60%. The output of this aim will be a program trained to classify gait activity for integration with the electronic control system of the knee. This will provide the basis of adaptive software control for the device.

Aim 3: Develop a control algorithm for the knee mechanism of Aim 1 using the machine learning system of Aim 2 as an environmental differentiator to switch between control models for the purpose of emulating human activity.

The control algorithm will switch between true changes in state within 0.15 seconds and will switch out of a false change of state within 0.30 seconds. Using embedded sensor information from the device, the control algorithm will adapt to current gait speed and transition to alternative activity models based on the activity classifier. This will use the additional sensor information along with the predicted activity from Aim 2 to control the physical knee device from Aim 1 – achieving predictive, reactive, and adaptive control for each wearer.

Aim 4: Compare assembled system of Aims 1, 2, and 3 to industry standards and base prototype M3.

It is hypothesized that the system will have a smaller rate of falls per stumble event than the M3 and will have no statistical difference in rate of falls per stumble compared with the subject's MPK. It is also hypothesized that the system will exhibit increased gait symmetry compared to the M3 and decreased gait symmetry compared with the subject's MPK. This aim seeks to evaluate the prototype knee system (GKnee) when compared to current passive and active

knee systems in the categories of falling prevalence and walking mechanics. Performance in this aim will ultimately be a summative performance of the overall research project.

Chapter 3: Aim 1 - Develop and construct a low-cost polycentric prosthetic knee with electronically-controlled swing dampening.

3.1 OBJECTIVE

For this aim, it is hypothesized that the developed prosthetic knee will support a 100 kg patient, allow knee swing up to 500 degrees per second, and have a component cost less than $500. The primary objectives are to optimize and construct an electronic damper that interfaces with existing knee design in the most efficient manner, so that it may be used to control the knee by limiting flexion.

3.2 INTRODUCTION

Transfemoral (TF), or above knee (AK), amputees utilize a full-leg prosthetic system consisting of a socket, knee, pylon, foot, and necessary connective hardware (Sagawa et al., 2011). While there is room for improvement in all systems, the operational joints, such as the knee and ankle, have received the most attention by researchers due to their complexity. Most prosthetic ankle-joint movements are represented in a flexible foot prosthesis, though there is ongoing research for powered ankle mechanisms housed separately from the foot (Shepherd & Rouse, 2017; Sup, Varol, et al., 2008). Because of its vital role in ambulation, the knee has drawn attention in major research efforts over the years. While passive knees (which have limited support phases and are not dynamically controlled) have existed for many years with reasonable success, emergence of dynamically adjusting microprocessor knees (MPKs) within the last thirty years has greatly improved the gait for patients worldwide. MPKs can adjust to patient gait, even in mid-stride, and have shown reduced metabolic energy expenditure (Johansson et al., 2005). Approximately 82% of amputees prefer MPKs to the passive alternatives (Hafner et al., 2007). While the benefits of the MPKs include increased stability, biomechanical symmetry, and other functional benefits, these knees currently cost, on average, $20,000 to the prosthetist and nearly

twice as much to the patient. This price point is unreachable for 80% of the world's population that earns less than $10 per day (Chen & Ravallion, 2010). An affordable solution with the features of a high-end system is needed for the majority of amputees in the world.

The design work of this aim focused on improving a validated system from previous results while incorporating major changes (Galey, 2016). Components and systems were evaluated and validated on an ongoing basis. Two major prototype iterations were completed, and the systems were evaluated for load bearing capacity, pressure ratings, hydraulic fluid flow, dampening capabilities, and component cost.

3.2.1 Knee Geometry

Prosthetic knees are designed to provide optimal support and swing. While the possible geometric arrangements are endless, there are three main categories most knees fall under: single axis, four-bar, and six-bar. Four-bar and six-bar are both polycentric mechanisms, meaning that the center of rotation moves as the knee flexes. Through mathematical optimization, polycentric knees are able to offer increased stance stability and toe clearance (Gard et al., 1996).

Single Axis

Within the field of actively controlled knees, single axis devices are the most common, most likely due to the single center of rotation and the simple application of torque. The three most popular MPKs (Ottobock C-Leg®, Össur Rheo Knee®, and Freedom Innovation Plié® 3) use different hardware mechanisms, but all utilize a single axis (Freedom Innovations, 2015; Össur, 2016; Ottobock, 2016). Actively controlled knee mechanisms have the advantage of applied torque for the purpose of stability, which is why passive knees often avoid single axis designs. A single axis without an adaptive system is difficult to design for both body weight support and free swing during gait.

Four-Bar Mechanism

Four-bar mechanisms are the most common form of polycentric knees (Chauhan & Bhaduri, 2011). Seen in Figure 3.1, the basic governing principle of this system is that at full extension, the system rests against itself during vertical loading, providing stability to the stance phase. This allows the resistance of the system to determine the leg swing speed but is not responsible for supporting the weight of the patient. An advantage to this system is the shortening of the functional prosthesis length during flexion, which allows the system to give patients greater toe clearance (Tang et al., 2008). Increased ground clearance helps patients walk more naturally by avoiding lateral sway during gait, decreasing lateral hip rotation, and reducing risk of secondary injury (Gailey et al., 2008; Gard et al., 1996).

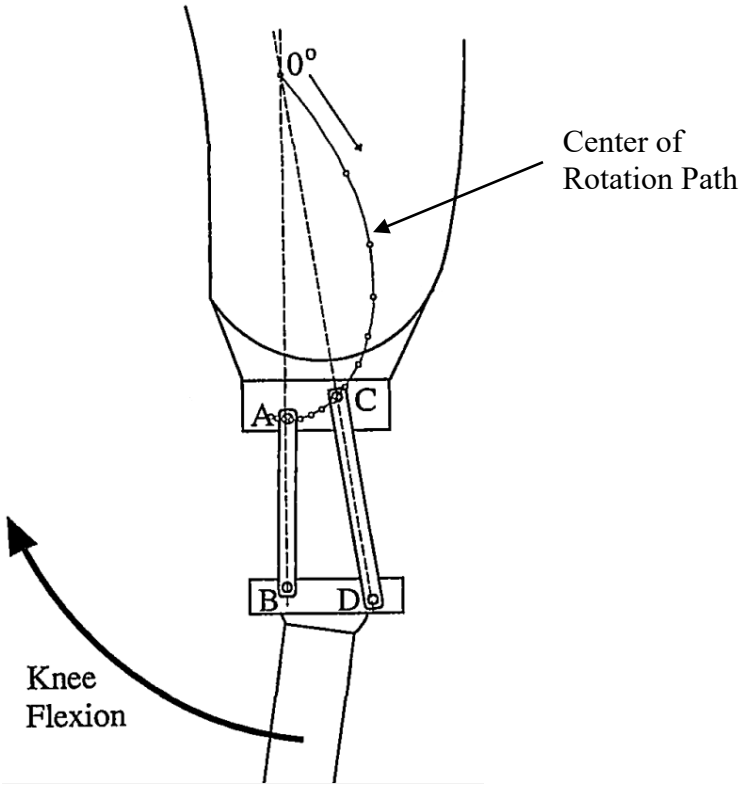

Figure 3.1: Four-bar knee mechanism. Dotted lines indicate the moving center of rotation. Path of center of rotation throughout flexion is given. From (Gard et al., 1996).

Six-bar Mechanism

As used by the Össur Total Knee system, six-bar mechanisms augment the innate stability of a four-bar system. The function of the two additional linkages depends on the designer, but the most common application is to allow the knee to effectively lock during full-extension and vertical loading. While four-bar mechanisms achieve mechanical stability at full extension under vertical loading, six-bar mechanisms subjected to this situation will also be resistant to horizontal forces, such as an impact to the knee from the side (Jin et al., 2003). This system still allows the knees to swing freely while flexing but maintains greater stability during knee-extended loading.

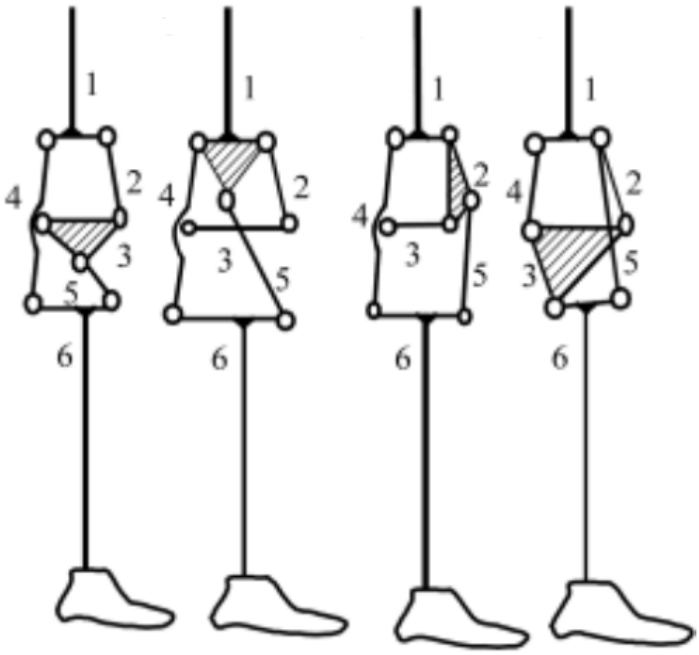

Figure 3.2: Example configurations of six-bar mechanism. From (Jin et al., 2003).

3.2.2 Gait Data

Many studies have documented gait patterns over the years. As discussed in later chapters, this data has been used to model and to predict gait characteristics and outcomes. Gait data has been shown to be repeatable with known variance bounds between subjects (Riley et al., 2007;

Winter, 1990). Ground Reaction Forces (GRFs) are the measured forces of a person's gait upon the ground. These follow established patterns that correspond to different gait events such as heel strike, toe off, and stance phase. Seen below in Figure 3.3 are knee angle and GRFs as measured by Winter. Because such forces and angles are repeatable and known, they can be used for force loading of prosthetic systems during various angles of knee flexion to simulate gait.

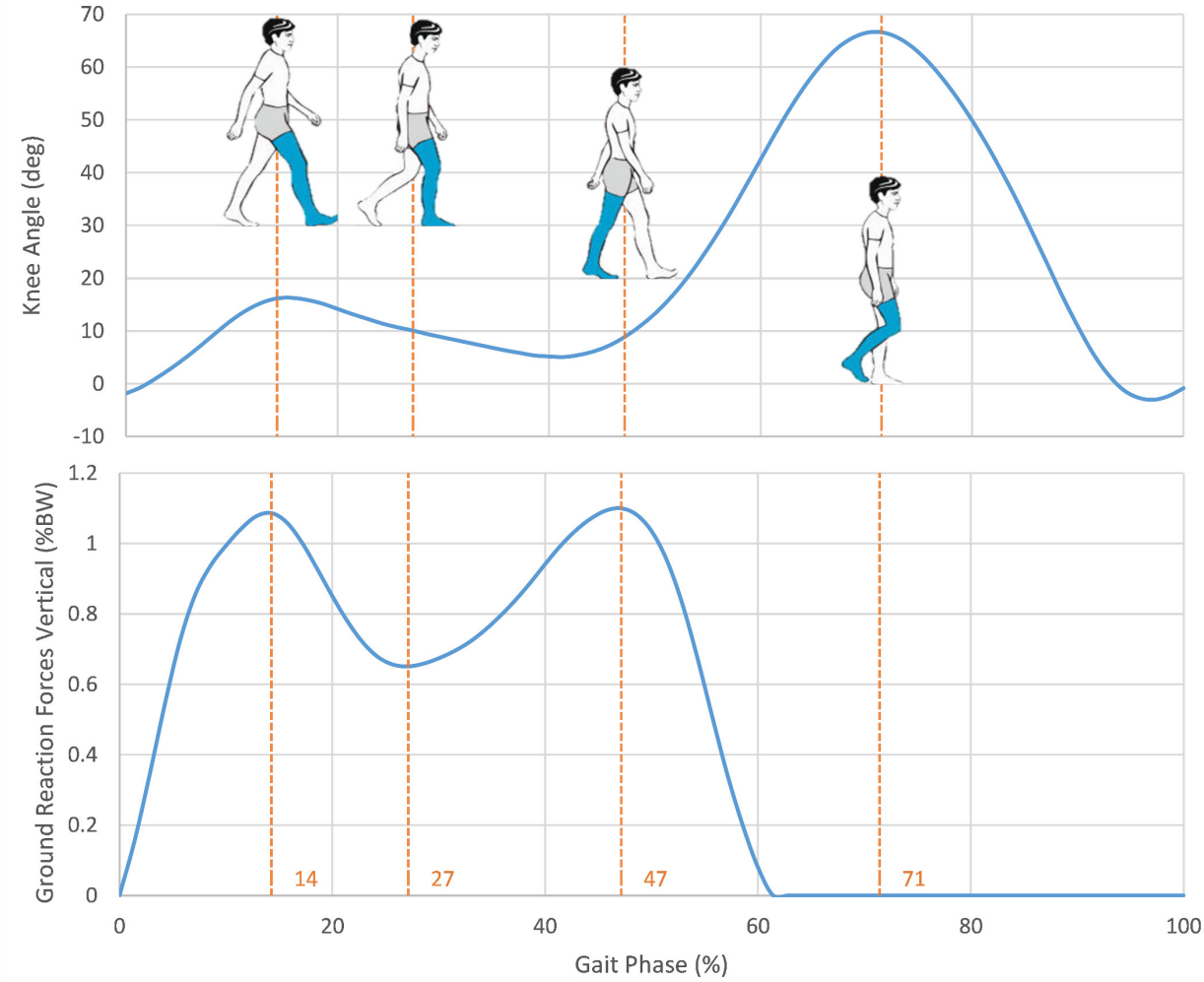

Figure 3.3: Knee angle and Ground Reaction Forces in knee during healthy gait. Adapted data from (Winter, 1990).

3.2.3 Control Methods

Though the topic of control methods is explored more extensively in the introduction to Aim 3, a basic understanding is needed to discuss Aim 1 as well. Prosthetic knees generally come

in two variants: passive control or active control (microprocessor control). The fundamental difference is that passive systems are non-adaptive to environmental changes. They have resistances to flexion and extension that are either set or adjusted manually. Microprocessor knees have sensors that detect knee activity and thus can dampen the knee in a corresponding manner automatically. Common methods for controlling the joint movement of prosthetic knees include the following: mechanical (K. R. Kaufman et al., 2008; Sup, Varol, et al., 2008), pneumatic (Radcliffe & Lamoreux, 1968; Sup, Bohara, et al., 2008; Tang et al., 2008), hydraulic (Bellmann et al., 2010; Johansson et al., 2005; Tang et al., 2008), and magnetorheological (Herr & Wilkenfeld, 2003a; Johansson et al., 2005; Ochoa-Diaz et al., 2014).

3.2.4 Previous Work

Previous work conducted in 2016 established the first iteration of a feasible low-cost, microprocessor-controlled knee device (E-Knee) for the current study to build upon (Galey, 2016; Galey & Gonzalez, 2022). The E-Knee, shown in Figure 3.4, was found to be a viable prototype that met low-cost, electronic-control design specifications and had comparable stability to commercial systems in qualitative and quantitative analysis. Preliminary testing of the E-Knee prototype suggested that it performed more closely to the Ottobock C-Leg than the LIMBS M3 passive knee. Though the E-Knee had reduced functionality compared to currently available commercial MPKs, it was shown to have comparable stability to the C-Leg while its components cost 2.4% the C-Leg's commercial cost.

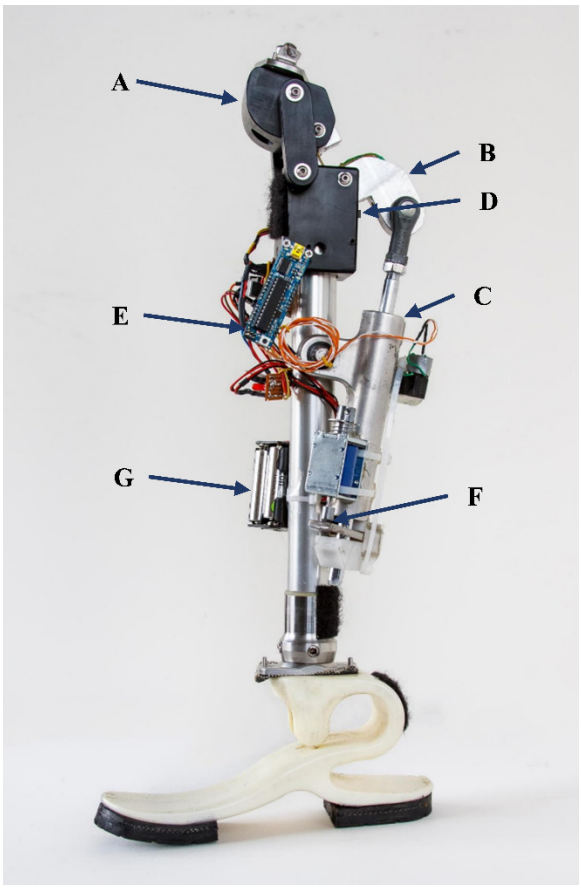

Figure 3.4: Fully assembled 2016 prototype (E-Knee). Knee is flexed and Niagara foot is attached. (A. LIMBS M3 Knee, B. Modified Backlink, C. Hydraulic Damper, D. Angle Sensors, E. Microprocessor, F. Locking Mechanism, G. Power Supply). From (Galey & Gonzalez, 2022).

3.3 METHODS

This research aimed to combine the mechanical advantages presented by four-bar knee systems with an electro-hydraulic damper so that the prototype knee system (GKnee) could provide active control. The primary geometric design was based on the linkage from the LIMBS International M3 and was used in the previous E-Knee device discussed above (Galey, 2016). The LIMBS M3 has innate stability up to six degrees of flexion. This research addressed the following limitations of the 2016 E-Knee system: restricted flexion range (90 degrees max), subpar dampening capabilities (could not arrest motion), and elevated power requirements (mainly due to

the solenoid). Additionally, the flexion stopping mechanism and the damper were separate units and should be united.

Based on the previous E-Knee prototype, the damper remained an electro-hydraulic system. Components were previously selected based on the following criteria: stroke length, flow rate, and pressure response (Galey, 2016). Using the force model from the geometric analysis in section 3.3.2, the system was optimized so that the damper could fully arrest flexion while the subject was running. The damper restricted flow resulting in stance support that could hold up to three times the body weight of a 100 kg patient, ISO P4. From the literature, running puts up to two and a half times the body weight on the knee, and a safety factor of one-half the body weight was added (Keller et al., 1996).

Based on the previous work and limitations, Table 3.1 outlines design specifications for the overall capabilities of the GKnee.

Table 3.1: Design Specifications of the prototype knee system (GKnee)

Criteria	**Specification**
Stability	Provide mechanism to arrest flexion.
Weight Limit	Support patient of 100 kg.
Maintenance ease	Use retail parts commonly available.
Cost	Cost less than $500.
Variable cadence	Include mechanism for variable swing control.
Immediate support on step initiation	Must be able to detect step.
Supportive yield for sitting down	Include mechanism for variable knee resistance.
Knee locking	Must have immediate effect from stability mechanism.
Redesign amount	Minimally redesign IM knee.
Degree of flexion	Minimum 120 degrees of flexion.
Weight	Weigh less than 2.27 kg (5 lb).
Endurance	12-hour electric capacity.

3.3.2 Geometry Kinetic and Kinematic Analysis

The piston attachment point was a key point goal of the redesign. The previous work attached externally to maximize the linkage moment arm, but this configuration limited the flexion

capabilities of the system (to 90 degrees). Therefore, a mathematical model was developed to find appropriate internal attachment points and maximize the ratio between damper stroke and maximum force. The geometry of the GKnee was mathematically modeled in Python using a linkage fixing model to calculate angles and positions of all links throughout the gait cycle. A model of forces and moments acting through each link member was created using kinetic and kinematic gait data from Godest et al. and Winter (Godest et al., 2002; Winter, 1990). A pseudo-static system was used to calculate the forces moving through the components at every time-increment.

Free Body Diagram

The first step in analyzing the forces acting on the various linkages was establishing the free body diagram (FBD) and defining the various components in effect. Figure 3.5, below, shows the basic configuration of the forces. The analysis of the system was done during static locking, therefore, the forces acting on the system from gait were two-dimensional vectors that were translated through the tibia from the GRFs. The tibia angle was used to translate the vertical GRF into a tibia vector. For this study, the analysis focused primarily on the force vector acting on the piston. This was done to appropriately select hydraulic components that could withstand gait. The posterior axle on the lower block, denoted by a black dot in the figure below, was the point from which the forces were considered.

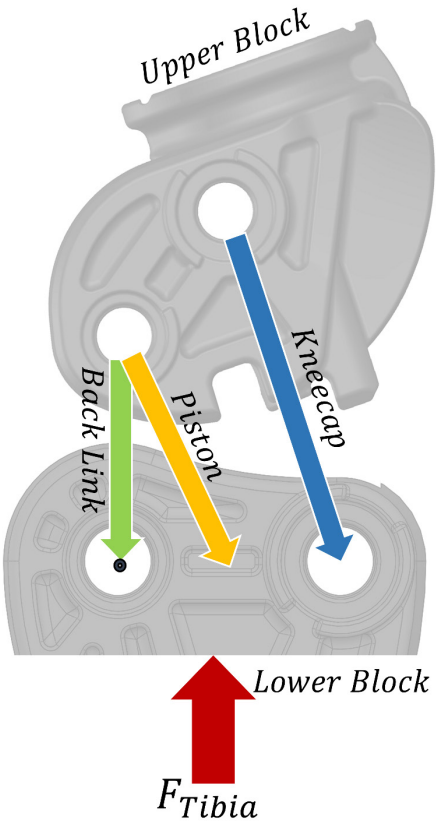

Figure 3.5: Free body diagram (FBD) for the four-bar IM knee mechanism. Shown are the forces acting upon the linkages relevant to the point of calculation (black dot in axle hole). The external force is the force exerted by the ground through the tibia on the bottom block (B). This force is not uniaxial but can have perpendicular components as well. The forces acting through the back link (L), piston (P), and kneecap (C) are uniaxial because they are attached to rotating axles. The piston attached to the lower block on a rotating axle lower than the image shows. The lower link was considered unmoving for force analysis.

Force Calculations

An objective of the knee design was to build a system that could resist loading during gait and stumble. Because the piston (P) prevented further movement when it was locked into position during activities requiring it, the forces acting through the system could be assumed to be static. Though a static analysis simplified conditions, the system is a four-bar mechanism, which made the forces acting through the various linkages more difficult to determine initially. The first step to analyzing the forces through each component was to solve the angles of each linkage corresponding to the flexion of the knee system. The lower block and back link intersection was

considered the reference point in both angle and force calculations. For a four-bar analysis, the back link (*L*; 36 mm), kneecap (*C*; 50 mm), upper block (*T*; 25 mm), and lower block (*B*; 30 mm) are considered. Because the system is a double-rocker linkage, meaning that both the back link and the kneecap do not complete full rotations and have constrained angles, continuous motion is not possible. Therefore, the angles were calculated by trigonometric geometry using the set link lengths. These angle formulas were validated with CAD calculations on the linkage.

From the reference point, three equations were established: a sum of the moments, a sum of the forces in x, and a sum of the forces in y. Generalized equations can be seen below in Equation 3.1, Equation 3.2, and Equation 3.3. These equations were calculated for a given load for all angles of the gait cycle.

$$\text{Equation 3.1} \quad \sum M = \begin{array}{l} P * \sin\theta_{\mathrm{p}} * Attach_x - P * \cos\theta_{\mathrm{p}} * Attach_y + \\ C * \sin\theta_{\mathrm{c}} * B_l - Tibia_y * \frac{B_l}{2} + Tibia_x * Tibia_l \end{array}$$

$$\text{Equation 3.2} \quad \sum F_x = P * \cos\theta_{\mathrm{p}} + L * \cos\theta_{\mathrm{L}} + C * \cos\theta_{\mathrm{c}} - Tibia_x$$

$$\text{Equation 3.3} \quad \sum F_y = P * \sin\theta_{\mathrm{p}} + L * \sin\theta_{\mathrm{L}} + C * \sin\theta_{\mathrm{c}} - Tibia_y$$

In these formulas, the letters representing the linkages are being used to display the force in each respective link. The primary force components were piston (*P*), *Tibia*, kneecap (*C*), and back link (*L*). The angle values are given by "*θ"* with the respective linkage subscript. Linkage lengths were given by the subscript "*l"*. Finally, the subscripts "*x"* and "*y"* were used to give reference frame force and distance information. "*Attach"* was the coordinates of the piston attachment to the lower block.

Validation

The mathematical model was validated by constructing a knee frame with the same established dimensions and inserting an s-beam load cell in place of the piston. The knee frame was then inserted into a testing rig that had a pneumatic piston at the bottom and a load cell at the top. Because the GRF data was the known quantity, the knee was inserted upside down so that the force on the simulated tibia of the system would be read directly. The femur of the system would be pressed on by the pneumatic cylinder. At the start of both the femur and the tibia were axle joints that allowed the forces exerted to attempt to flex the knee.

A threaded bolt was used to adjust the faux piston length to set the knee to several different knee angles. At each set angle, the system was subjected to a loading cycle. The pneumatic cylinder applied force at a rate of 100 N/s and would hold at the set load for 30 seconds. During this time, the s-beam and the rig load cell were collecting data at 10 and 100 Hz, respectively. An angle finder (accuracy: ±0.1 degrees) was used to collect the following angles: tibia, femur, knee flexion. These were needed so that the mathematical model would apply the simulated GRF forces to the knee at the appropriate angles.

At each set angle, the first load cycle was 200 N to settle the frame and remove any motion in the system. This angle value was recorded and became the set angle label. Loading cycles were conducted at 7, 12, 19, 24, 29, and 39 degrees. At each set angle, the true load cycles started at 1,000 N and increased in steps of 200 N. The load cycles were discontinued after the s-beam read approximately 8,000 N or the load angle reached 3,000 N, which reflected the s-beam's maximum load (1 imperial ton) and more than the maximum running weight of a 100 kg patient (~3,000 N when including a factor of safety). The data was then used to validate both the accuracy of the mathematical model and also quantitatively measure the forces that would be acting on the piston

throughout gait. The validity of the mathematical model allows it to be used for future attachment optimization and material selection.

3.3.3 Hydraulic Components

The force calculations were used to establish two parameters for the hydraulic system – the piston load and the piston attachment points. A balance had to be struck between the cylinder size, the maximum fluid pressure, and the flowrate of fluid. A larger cylinder would decrease the fluid pressure but would increase the fluid flow rate. Additionally, the points of connection between the components could limit the flowrate and thus necessitate a higher-pressure system, which decreased the options for component selection. Therefore, these three variables were carefully balanced during the selection and design process.

Cylinder Selection

Due to the design specifications for cost and maintenance ease, component selection was restricted largely to off-the-shelf components without additional customization costs. Electric linear dampers were considered as well but were ultimately disregarded; though they can apply great force, it comes at a trade for speed. Based on preliminary data of knee angular velocity during gait, it was determined that the stroke speed of the cylinder would be at least four cm/s. Additionally, even if such systems had both the speed and load capabilities necessary, they would need to be powered during every gait cycle. This was considered an unacceptable power constraint.

Fundamentally, two types of cylinders were left to consider: single acting and double acting. Because the primary function of the hydraulic system was to provide dampening, there was no practical advantage to the double acting cylinder category. Therefore, the more affordable single acting cylinder was chosen. From there, the selection criteria were dependent on balancing

the force and fluid flow calculations. Systems of various specifications were readily available from commercial vendors.

Valve Selection

Fluid flowrate and valve actuation method were the primary considerations in selecting an appropriate valve for the system. Early in the process of the mathematical model, an error led to a force specification increased by an order of magnitude. To mitigate this, larger cylinders were selected. These large cylinders came with the condition of needing a large amount of fluid as well, and a flowrate of 7.11 liters per minute (lpm). Therefore, flow testing for several valves for Prototype 1 was conducted and ultimately proven to be unnecessary. However, despite the flowrate calculations being found incorrect, flowrate control remained necessary. To increase or decrease dampening of the hydraulic system, the valve needed to have proportional control.

Valve flow methods range from ball mechanisms to very small orifices, and actuation methods include mechanical and electrical inputs. Methods that allow large flow, such as ball valves, allow more fluid but would be more difficult to actuate. Electronic proportional valves, such as the SV08-25 cartridge by HydraForce, exist, but most function through activation of a solenoid. As determined by the previous work, a solenoid was not ideal for a portable knee due to the power consumption required of continuous electromagnet operation. Methods that use motors to actuate valves do exist, but none were found preassembled within the pressure, flow, and size requirements. Therefore, mechanical proportional valves were explored with the intent of electronically innervating them with a small positional motor.

Mechanical valves were evaluated on the following criteria: pressure rating (>20 MPa), flow rate (>7.11 lpm), and operating torque (<25 kg/cm). The operating torque limitation was established by finding a compact servo with the appropriate speed and voltage requirements,

presented in the results below. While pressure rating was a product specification, flow rate and operating torque often were not disclosed. Therefore, tests were conducted on the systems considered to validate their capability.

Fluid Flow and Reservoir

Noncompressible oils come in many viscosities, densities, and prices—they were evaluated accordingly for use in this system. By calculating the head loss between the various sections of Prototype 1, the total pressure drop for the system could be calculated. The characteristics of each oil changed the pressure drop and was used to evaluate its performance. The oils considered for this system were the following: olive oil, canola oil, brake fluid, motor oil, power steering fluid, industrial hydraulic fluid, and transmission fluid. Although more advanced oils are available, these were analyzed because they were also low-cost. The pressure drop calculations were also in consideration during design or selection phases of any connections or manifolds because the change in diameter would affect the head loss of that section.

The hydraulic system, when acting as a damper, was not applying force toward moving, but rather restricting flow. As the fluid flowed out of the cylinder, it needed to be stored so that when the cylinder was extended again, the negative pressure would refill the cylinder. Therefore, a reservoir system was explored. Its primary specification was that it could hold at least the volume of fluid equal to that of the stroke of the cylinder during use.

Validation

A key component to the hydraulic system selection was identifying the optimal fluid flow. Too much restriction would establish a base dampening force that would not allow the knee to move freely. Therefore, if a candidate valve did not have a manufacturer provided datasheet, a flow test was conducted. The flow test was conducted with water from a garden hose. A flow

sensor collected data at the valve's output. The flow test established two specifications: the flow rate and flow control profile of the valve.

To evaluate the torque required to turn the valve and thus aid in selecting a servo with enough capacity to do so, the valves were tested with a simple movement test. A bar was attached to the valve and weight was added until the valve began to turn (called break-away torque). This was done with a 7 cm and 11 cm lever arm, and the results were averaged.

The hydraulic assembly of the system was tested within the assembled knee. Because the system was designed and components selected to meet pressure parameters, the primary validation of this system was to confirm that the sealing of the system had been successful. Body weight (99.8 kg) was applied to the locked system at various angles. The researcher would balance on the system so that it bore the whole weight. The vertical weight was applied up to a 0.30-meter offset, resulting in a 30.4 kg.m moment when the knee was flexed to 30 degrees.

Additionally, the mathematical model was used to calculate the force on the piston, and therefore pressure on the hydraulic system, for the three-times body weight test of a 100 kg patient in recovery. Recovery in this case was set to a 20-degree knee angle, which is approximately 26% of a gait cycle before regular heel strike, and an 85-degree tibia angle, which is higher than typical during gait. It should be noted that this is a worst-case scenario simulation where not only is the subject running (2.5 times body weight on the system), but the safety factor is considered (0.5 times body weight on the system), and the subject is also stumbling at drastic knee angle.

During intact-leg gait, subjects show an average peak angular velocity of 250 deg/s (Mentiplay et al., 2018). From preliminary data collected of the M3 Knee, 400 deg/s were regularly reached during gait. Therefore, the objective was to exceed both with a measure of safety of 100 deg/s. Baring a severe flow restriction, the hydraulic system could be forced to move quickly

enough with enough pressure. However, this would not test the angular velocity capabilities of the prototype in a realistic manner. To assess whether the knee was able to swing at 500 deg/s, walking data of the trials in Aim 4 were evaluated. These trials used the GKnee, and knee angular velocity was calculated by finding the difference between the recorded sensor data for the shank and thigh.

3.3.4 Electronic Components

Components for the electronic system were chosen around the microprocessor unit. With a restriction to Python-based machine learning systems, several options were considered such as the Google Coral or the Raspberry Pi. Ultimately, the Google Coral was the only system, at the time of design, that was a ready-built, machine-learning-capable microprocessor unit. Therefore, components were selected that worked within its Mendel Linux distribution used by the processor. The compatibility of electronic components with the Python and Linux environment was the largest deciding factor when choosing sensors.

Servos were selected based on speed, torque capabilities, and size. Because the servo was to actuate the valve, the selection of this component occurred after initial valve break-away torque testing. Smaller servos were desirable to keep the system compact, but smaller servos often operate in the grams/cm range at any reasonable speed. The objective was to have a servo that was powerful enough to actuate the valve, but minimal in size.

3.4 RESULTS

Adhering to the methods described in Section 3.3, the following force calculation models were created (shown below in Figure 3.6). From these calculations, it was determined that the piston needed to support a force up to 14,800 N. The forces shown are a theoretical simulation from a static evaluation.

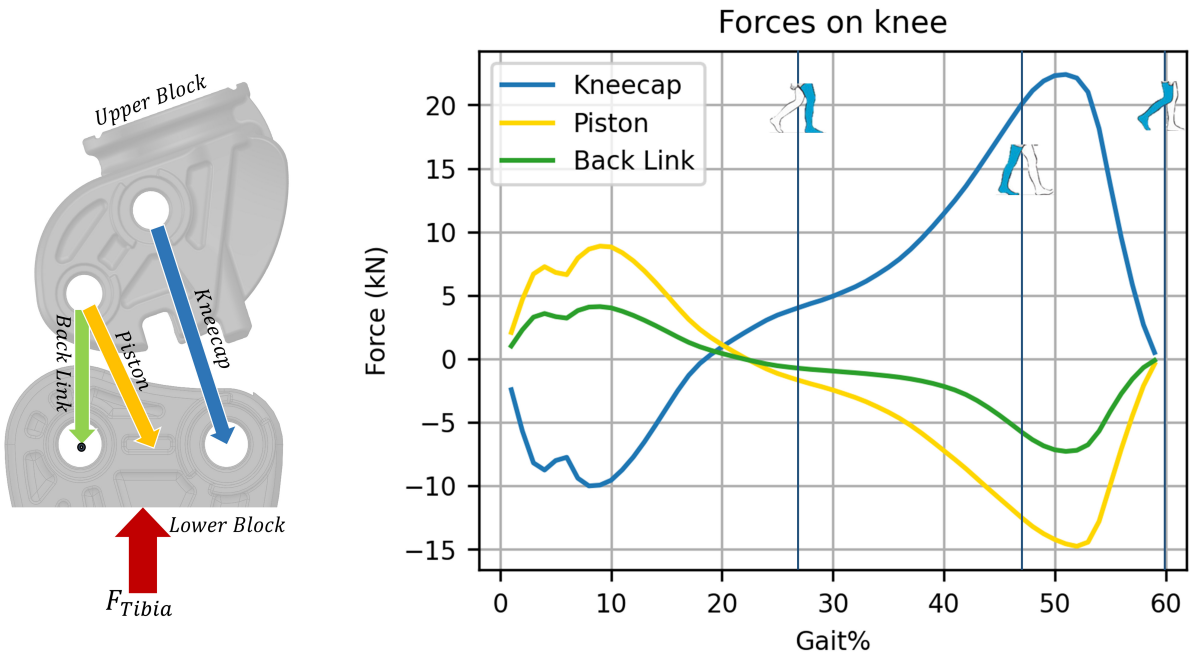

Figure 3.6: Results of the force calculations for the damper attachment point. Left shows the basic configuration of forces on the prosthetic knee system (shaded components) and right shows simulated forces if piston had to perform a complete stop. Force calculations were done with patient's weight (100 kg) with the calculated maximum forces during running.

The mathematical model was first used to establish attachment points for the two prototypes, and then was validated by force measurement. Figure 3.7 highlights both the validation of the mathematical model and a summary of the forces acting on the piston at various loads and angles. Note that angle 39 has been excluded from the figure below. Angles 29 and 39 began at 300 N and increased in 100 N steps until the limits had been reached because the normal 1,000 N starting load already exceeded the measurement limitations. Despite this, angle 39 was at such flexion that the test only had two datapoints (not shown).

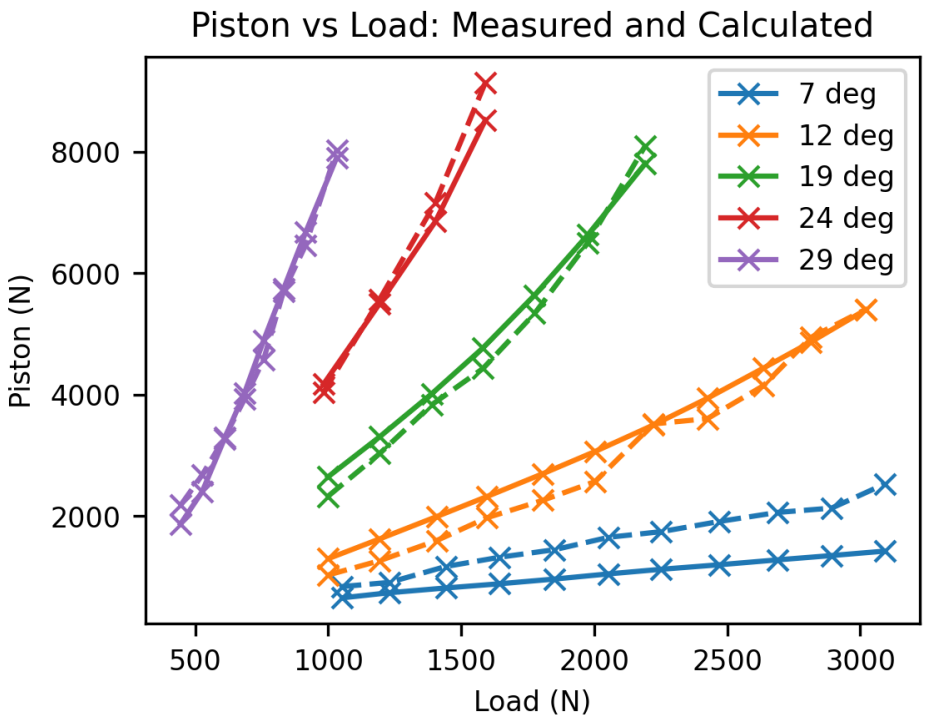

Figure 3.7: Measured (solid line) and calculated (dashed line) forces acting on the piston during loading at various set angles. The "x" markers indicate actual datapoints. The degree given is the settling (200 N) angle of that test run.

Table 3.2: Measured and calculated forces acting on the piston difference error percentage at the various set angles.

	7 deg	12 deg	19 deg	24 deg	29 deg
Average Error	51%	12%	6%	4%	5%

3.4.2 Prototype 1

Built to exceed the force requirements of the mathematical model, Prototype 1 was overengineered, but served as a first fitting for all components. The frame was made of 6062 aluminum and clamped to the pylon at the distal end.

Hydraulic Assembly

The major components of Prototype 1's hydraulic assembly were as follows: VekTek 1-5/16-Inch threaded cylinder, HAWE AC 13-1/4 diaphragm accumulator, 316 stainless steel 1/8-

inch ball valve, and a custom 6061 aluminum manifold. The component with the smallest maximum pressure was the ball valve, which limited the pressure to 20.7 MPa (10,382 N).

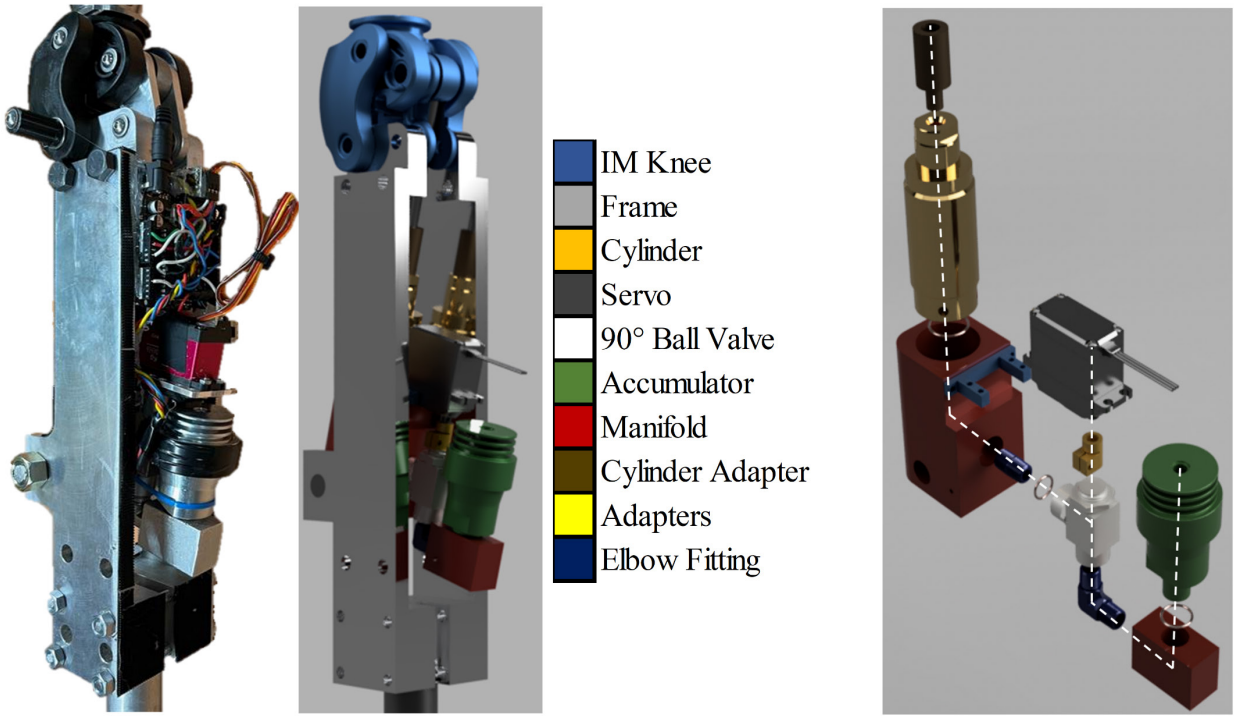

Figure 3.8: Assembled prototype 1 knee system. Left shows actual assembly. Middle and right show rendering of the full assembly and the hydraulic system in detail, respectively. A legend is included to label the various components.

Testing

The ball valve was tested for flow performance and its flowrate was six lpm. Figure 3.9, below, shows the flow profile of the valve. While it is not perfectly linear, the shape was repeatable and consistent across opening and closing of the valve. The torque needed to open the ball valve while dry was 3.73 kg.cm.

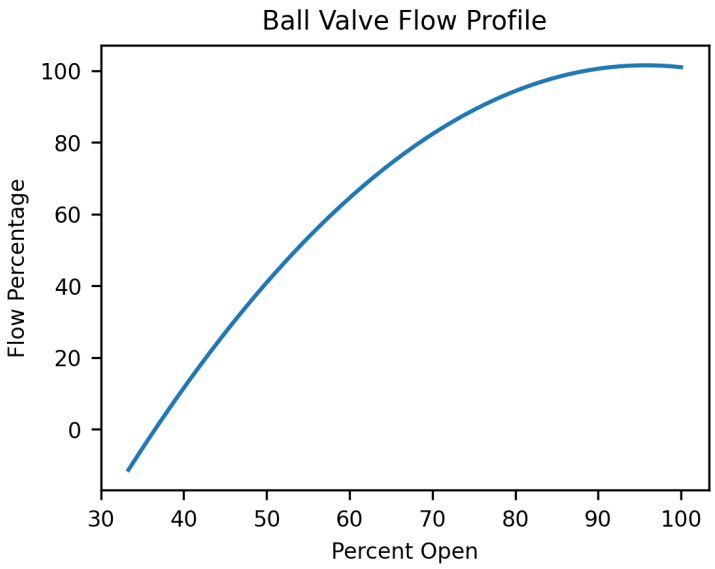

Figure 3.9: Flow profile of the ball valve. At fully open, the flowrate was six lpm.

3.4.3 Prototype 2 (GKnee)

The changes between Prototypes 1 and 2 were relatively simple adjustments with two notable exceptions. While Prototype 1 had a servo and an Arduino to control it, these were used for testing purposes only. The GKnee, on the other hand, had a full electronic system with microprocessor and input/output system, which is described in greater detail in the Electronic System, below. The second primary change was the adoption of a rotary valve. Because the valve was shaped as a cartridge both machined manifold pieces were combined into a singular, more compact manifold. It should be noted that in this aim, the designation “GKnee” refers to the purely physical construct, but in later aims also includes the control system of Aim 3.

Hydraulic Assembly

The rotary valve was the Eaton Vickers MRV2-10V flow control valve. Similar to Prototype 1, this system was limited by the valve to a pressure of 21 MPa (10,635 N). The other hydraulic components remained the same. 3D printing was used to construct the gears to interface the servo with the valve, and to create mounting hardware for the various low-force components.

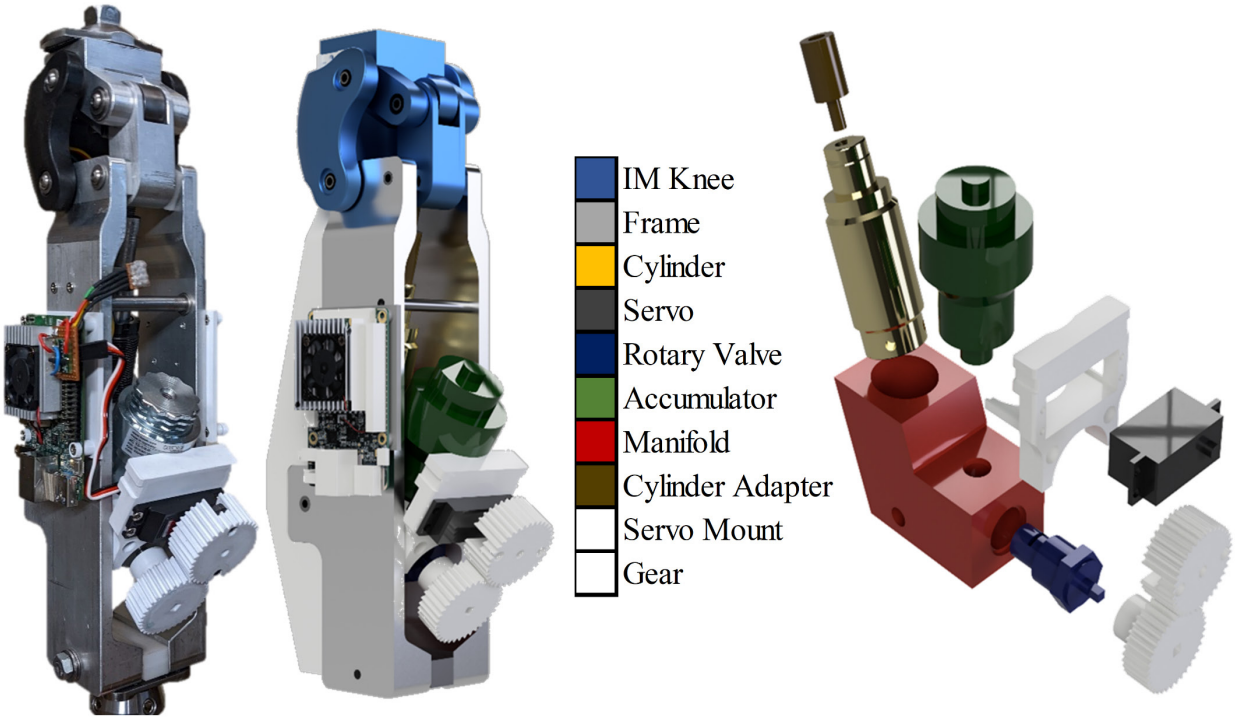

Figure 3.10: Assembled prototype 2 (GKnee) system. Left shows actual assembly. Middle and right show rendering of the full assembly and the hydraulic system in detail, respectively. A legend is included to label the various components.

Electronic System

The electronic components of the GKnee prototype were relatively simple in design and nature. For computation, the system needed a microcontroller; for input, the system needed a set of inertial measurement units (IMUs) and for output, the system needed a servo. Figure 3.11, below, shows all the components and their wiring diagram.

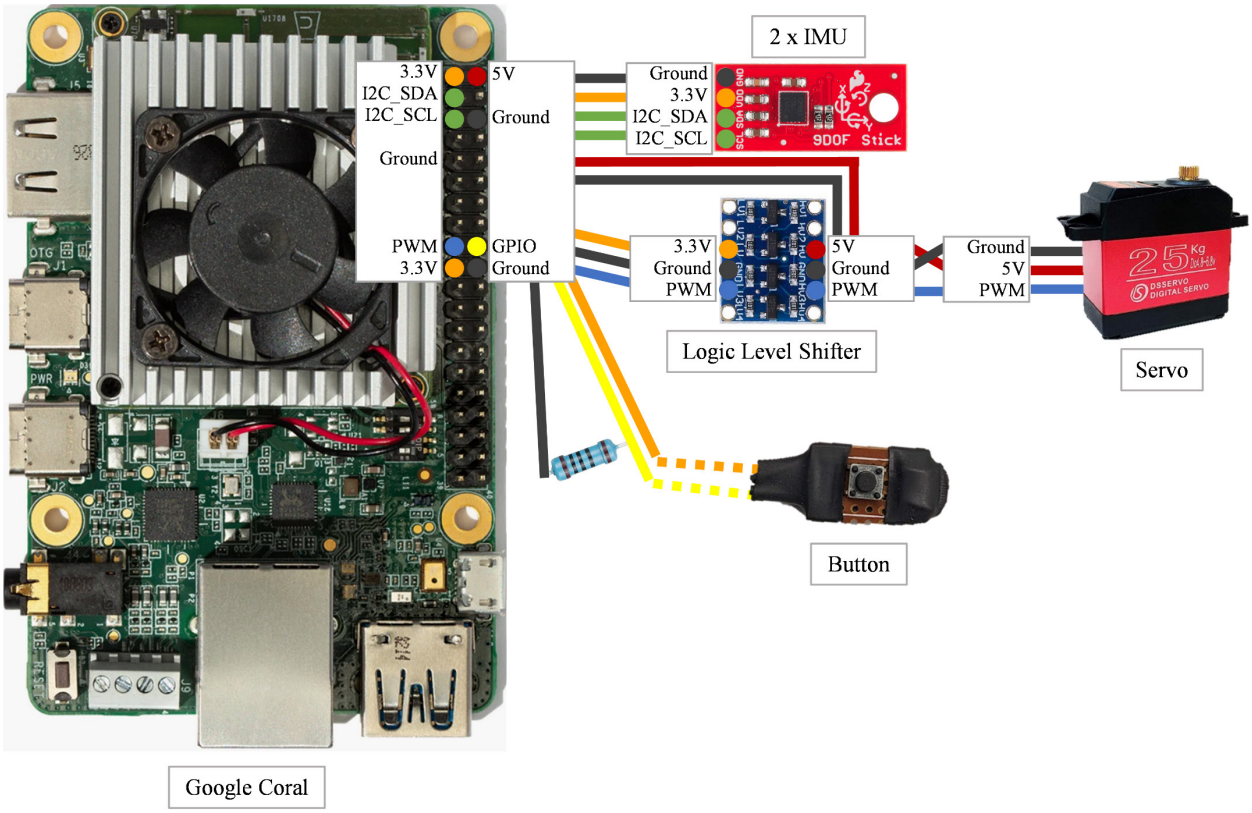

Figure 3.11: Component and connection diagram of the electronic system of the GKnee. Major components are the Google Coral, 2 IMUs, Logic Level Shifter, Button, and Servo. The dotted lines between the Button and the Google Coral represent a long wire. There are two IMUs, but since they are connected via I2C, they are connected to the same wires, and thus shown grouped.

The Google Coral was chosen as the system's microcontroller because it was designed for deployable machine learning models. It is a single-board computer capable of performing four trillion operations per second and operates on a Linux derivative. Its key features are the following: 1 GB of RAM, 8 GB of flash memory, WiFi, LAN, 3.3V GPIO rail, and 5V input power. The RAM and flash memory seemed very suited for a portable model, and the WiFi meant that the system could be controlled untethered.

The inputs to the system are a button and two IMUs. The button was designed to be activated from an extended distance and was therefore tethered by a 10-foot wire. The button was a simple through-hole tactile push button. It was mounted on a piece of prototyping board for grip

and connected to the two leads. The button was a normally open device that closed the circuit when pressed. The two leads were connected to the 3.3V source and a GPIO on the Google Coral. Therefore, when the button was pressed, the GPIO was connected to the 3.3V, and would read HIGH. Additionally, a 100 Ohm resistor pulled the GPIO to ground. Thus, when the button was not pressed, the GPIO would be connected to ground, and read LOW. The IMU units that were selected and used interchangeably were the LSM9DS1 and the LSM6DS3. Both featured accelerometers and gyroscopes and were integrated into the CircuitPython libraries. Originally, it was planned to use the sensors at 120 Hz, but only the LSM9DS1 was able to perform at this level with the CircuitPython and Google Coral. The LSM6DS3 peaked out at approximately 80 Hz. However, in practice, the system ran at 60 Hz; thus, both devices were functional. Their software libraries were similar enough that a toggle portion of code could enable one or the other sensor. Both operate over I2C protocol and thus when a set of IMUs, one for shank and one for thigh, was connected, they attached to the same pins (3.3V, ground, SDA, and SCL). Therefore, one set of wires connected to the relevant pins on the Google Coral, and later split to attach to both sensors.

The only physical output to the electronics system was the servo motor. Though the logic of the PWM output of the Google Coral is 3.3V, the servo required 5V logic. Therefore, a logic level shifter was used to step the voltage from 3.3V to 5V. The servo was a DS3225MG RC motor rated for 4.8-6.8V operating voltage. At 5V, its operating speed was 0.15 sec/60 degrees. Its range was 180 degrees, and it had a stall torque of 21 kg.cm at that voltage. The servo interfaced with the hydraulic system valve, allowing the Google Coral to control the dampening of the prosthetic knee effectively.

Cost

Displayed in Table 3.3 is the cost breakdown by component for the GKnee. The total cost of the system was $1,392. The costs shown include machining and material costs.

Table 3.3: Cost breakdown by component of Prototype 2 (GKnee). Costs include raw materials and machining and are shown in US Dollars ($).

Component	**Cost (USD)**
Kneecap	$15
Top Block	$50
Frame	$80
Cylinder	$249
Rotary Valve	$45
Accumulator	$220
Manifold	$300
Cylinder Adapter	$80
Back Link	$100
Miscellaneous	$32
Google Coral	$180
IMU (2)	$15
Servo	$25
Wiring	$2
Total	**$1,392**

Weight

Shown below in Table 3.4 is the weight breakdown of the GKnee. Smaller components were summed together into overarching categories. The total weight of the system is 3.33 kg (7.34 lbs).

Table 3.4: Weight breakdown by component/category of Prototype 2 (GKnee) in kilogram. Weight does not include external adapters.

Component	Weight (kg)
Kneecap	0.11
Top Block	0.17
Frame	0.68
Cylinder	0.50
MR2 Rotary Valve	0.08
Accumulator	0.30
Manifold	0.44
Cylinder Adapter	0.07
Back Link	0.04
Miscellaneous	0.65
Electronics	0.15
Printed Components	0.14
Automatic Transmission Fluid	0.01
Total	**3.33**

Component Testing

For the rotary valve, the manufacturer provided flow characteristics therefore flow testing was unnecessary. The valve has a max rated flow of 18.9 L/min, which exceeds the minimum requirement, and its flow profile is linear to the valve openness. The tested break-away torque for the valve was approximately 0.08 kg.cm, which is significantly less than the capabilities of the servo.

System Testing

Using the mathematical model, the position that optimized piston force, stroke length, and hydraulic system pressure was found to be 54 cm forward and 157 cm downward when measured from the back link bottom axle. At this attachment point, for a 100 kg patient, the stroke would be equal to 1.67 cm with the recovery piston force of 9,896 N and recovery hydraulic pressure of 19.4

MPa. For the max pressure of the rotary valve, which is the limiting component of the system, the GKnee should be able to bear a 10,635 N force.

As stated in the methods, the system was tested up to a 30.4 kg.m moment to evaluate the piston's seal. No leakage was found, and the system resisted flexion through the moment. During the trials of Aim 4, the knee system routinely exceeded 500 deg/s, which validates minimum dampening capacity.

Design Specifications

Table 3.5 shows the design specifications from the methods along with the feature that accomplishes the specification. The primary design criteria that were not achieved were the following: cost, redesign amount, and weight.

Table 3.5: Design Specifications of the prototype knee system (GKnee) with the resulting feature.

Specification	Feature
Provide mechanism to arrest flexion.	Hydraulic system
Support patient of 100 kg.	Able
Use retail parts commonly available.	Partial success
Cost less than $500.	$1,392
Include mechanism for variable swing control.	Rotary valve with servo
Must be able to detect step.	Microcontroller
Include mechanism for variable knee resistance.	Rotary valve with servo
Must have immediate effect from stability mechanism.	Valve can lock within 0.15 seconds
Minimally redesign IM knee.	Not achieved
Minimum 120 degrees of flexion.	Flexion of 146 degrees
Weigh less than 2.27 kg (5 lb).	3.33 kg
12-hour electric capacity.	Low-power components selected

The final assembled prototype can be seen in Figure 3.12, below. This was the functional system used during the trails of Aim 4. Any changes and adaptations mentioned in Aim 4 were changes to software and not physical design or assembly. In this GKnee, the electronic components were coated in a protective layer of hot-melt adhesive.

Figure 3.12: Final assembled GKnee prototype. Seen are the electronic system and the hydraulic system together within the frame as a functional prototype. The foot pictured is a modified Niagara Foot v1.

3.5 DISCUSSION

The overarching components of this aim were the mathematical model and the GKnee prototype. To evaluate the forces the GKnee is subjected to, the mathematical model had to first be developed and verified.

Mathematical Model

Though early mathematical models developed had considerable error, the final version of the model showed a consistent performance when compared to the s-beam force measurement. The highest error was 12 percent with exception for the seven-degree run (51%). A large portion of the force calculations depend on the angle of the input forces and the flexion of the system. With an angle measurement accuracy of 0.1 degrees, the forces were expected to have offset errors; however, using a safety factor of 0.5 body weight (~500 N) controls for this error. A high error in the seven-degree run is to be expected as well. As mentioned previously, the four-bar mechanism of the system has innate stability up to approximately six degrees. At seven degrees, with error, the system wavers between stable and unstable: not fully unstable, but no longer stable. Much would depend on small amounts of ductility in the joints that would allow the knee to deform in either direction: flexion or extension. Therefore, it is expected that the model is less accurate at this angle. Figure 3.7 also shows the large effect of knee angle on the forces transmitted to the piston demonstrating similar phenomena.

Figure 3.6 highlights the maximum forces during gait on the piston. At first glance, it seems that the 14,800 N clearly exceeds the limitation of the hydraulic valve (10,635 N). However, this force is seen at toe-off when intact gait already has an elevation of knee angle, which amputee data has not demonstrated. This is a limitation of using healthy gait and GRF data to model the forces acting through the piston during gait. Future work will include an amputee specific GRF and kinematic standard on which to model the forces expected. The limiting factor of the GKnee system is the rotary valve. It is unsure what its failure mode would be at such high forces, but it is likely that it would result in greater leakage instead of catastrophic failure.

The largest multiplier on piston force loading is the knee angle during load. As the Figure 3.7 demonstrates by the slight curvature of each line, the knee deformation past the set angle

continues to increase the piston force. Based on the geometry of the four-bar mechanism, it was estimated that near 23 degrees of flexion the piston would become less effective at arresting flexion. This occurs because past this angle, minute angle increments of the back link, and the piston attached, cause substantially larger changes in overall knee flexion. Therefore, a limitation of the system is the ductility and tolerances. Further testing would be needed to isolate the sources of the ductility completely, but likely culprits are the kneecap and the axle joints. The kneecap is made from Delrin 100p, which has worked well for the M3 relief knee, but its flexibility in this case may be detrimental to arresting flexion in certain, specific situations. The axle joints were precision machined into the frame, but the anterior axles do not go through the whole system, but rather only pair the kneecap and the frame. A short axle has a much greater likelihood of movement because the contact points are spaced more closely spaced.

GKnee

Several design specifications were established early on as ideals to pursue. However, with the complexity of the novel systems, it became apparent that a legitimate, working prototype achieved more than a within-specifications, unreliable prototype. Therefore, several specifications were put on hold to establish a system that functioned dependably and correctly. While a battery system was an initial specification, it became apparent that during sensitive data collection, power interruptions would vastly complicate trials and require recollections. A minimal redesign of the IM knee was simply not possible once the magnitude of the forces became apparent. Essentially, the cost of the system is triple the specification. While this is not ideal, many of the associated costs were that of prototyping, not of final production.

The same principle applies to the system's weight; the GKnee is over the specification by more than 1 kg. Though frame and manifold redesign between Prototypes 1 and 2 may not appear

to be a large difference, the weight difference in the frame alone is 0.45 kg. As can been seen in both designs, the prototype frames are thicker than they need to be to support patient weight.

Limitations

The current GKnee system is functional and was used successfully in the latter aims. However, this study is limited to the tests conducted at several loading conditions—ISO 10328 tests were not performed. In that ISO, there are established loading conditions (force vectors, durations, etc.) for a simulated 100 kg patient with a fully extended knee system. Because such tests apply maximum loads and can take more than a month to complete, the time delay and potential destruction to the system were not considered worth the benefit.

Likewise, the electronic system was limited to hobbyist components and easily removable connections. This warrants further testing to be fully evaluated. They were neither expensive nor guaranteed to be consistently accurate. While such considerations may not influence the performance in the short-term, these components are unlikely to be suited to long-term usage in their current configuration. Additionally, the electronics were selected to be used with the ML components; therefore, hardware and software limitations of the Google Coral restricted the components to specific sensors. Others were tested, but most had incompatible libraries for quicker refresh rates.

Lastly, the current weight of the system is limiting because it is close to double that of commercial systems. Anecdotally, patients did not have a problem with the weight, but studies have demonstrated that heavier systems increase patient fatigue and discomfort. This can also lead to poor gait and habits in the long run. Therefore, it would not be recommended to use such a heavy system for extended periods beyond evaluation. Largely, this is due to the constraint of using commercially available hydraulic components.

Future Work

While the prototype knee addresses many of the concerns of the previous system, there is still room for improvement. The primary aims for future work are the total cost ($1,392), weight (3.33 kg), and the lack of portable power.

The modular components of this system allowed for simplicity but came at a cost for both price and weight. The manifold was not optimized for weight or reduction of material. At $300 and 0.44 kg, the simplification of this component would have a large impact. Because it fits to the cylinder via threading, using a cylinder with built-in attachments would greatly reduce cost and weight. The current frame was designed during an earlier iteration of the mathematical model, which means that it was designed for much larger forces than are acting through the system. It is estimated that the frame and the manifold can each be reduced to half of their size and still be overengineered for the same loads—reducing the weight by 0.56 kg. Additionally, though the reservoir fulfills a vital fluid storage function, it became clear that a machined system with backpressure functionality was unnecessary in this design. The accumulator is never subjected to pressures because the pressures acting on the system do so on the piston-side of the valve. A simple bladder system would save up to $200 and 0.3 kg.

To include portable power in the system, the power consumption would need to be measured. Components were selected to be low power, but analysis would be needed to develop a battery system with enough capacity. To that end, components such as the servo should be re-evaluated. With the application of the rotary valve, which has a much lower torque than expected, the large size, weight, and strength of the servo is no longer necessary. A smaller servo would improve these specifications. However, the frame does have the spatial capacity to include portable batteries with a 25 Ah capacity. Because the system uses a servo instead of solenoid valve, such a power supply should be adequate for a day-long usage.

Many parts of this system can be redesigned, and current work seeks to address them. However, the overall findings of this aim do support the hypotheses that the knee will support a 100 kg patient and that the knee can swing at 500 deg/s.

Chapter 4: Aim 2 - Implement a machine learning classification system for gait during the activities of walking and stumbling through data from wearable IMUs on a unilateral knee prosthesis.

4.1 OBJECTIVE

It is hypothesized that the system will achieve an activity classification accuracy greater than 90% and detect stumble with a precision greater than 75% and a recall greater than 60%. The output of this aim will be a program trained to classify gait activity for integration with the electronic control system of the knee. This will provide the basis of adaptive software control for the device.

4.2 INTRODUCTION

Gait characteristics have been well documented by the literature. Both in individuals with healthy gait and pathological diagnoses, such as those with cerebral palsy or muscular dystrophy, key defining gait characteristics have been found (Benedetti et al., 1999; Gard, 2006; Kelleher et al., 2010; White et al., 1999; Wise et al., 2004). The field of gait classification has seen growth in the area of applied machine learning (ML), specifically in differentiation between healthy and pathological gait (Alaqtash et al., 2011; Mannini et al., 2016; Tahir & Manap, 2012). Likewise, human gait activity, such as walking, running, and standing, has been documented by datasets, such as HuGaDB, and classified by machine learning algorithms with high levels of success (Badawi et al.: accuracy: 98.6%; Keçeci et al.: accuracy: >99% with many networks) (Badawi et al., 2019; Chereshnev & Kertész-Farkas, 2017; Keçeci et al., 2018).

Additionally, the literature shows several implementations of real-time gait event detections using sensors, and many more gait activity classifications with potential for real-time applications (Hanlon & Anderson, 2009; Lambrecht et al., 2017; Maqbool et al., 2017; Marayong et al., 2017; Rueterbories et al., 2010). However, with the limitation of studies that measure human

stumble characteristics, especially constrained to patients with amputations, no studies have been found that attempt to classify stumbles in gait in real-time. Even most gait activity studies are limited to postliminary processing. The work presented in this chapter seeks to fill that gap, both by beginning to provide the scientific community with gait data pertaining to amputees during stumble-events and by presenting a first approach to classifying those stumbles and gait events using ML.

4.2.1 Gait Data

Some of the most complete research done in gait analysis was by Sagawa et al. who reviewed the preferred biomechanical and physiological parameters that are most often used in determining gait characteristics. While their results were inconclusive, the most common parameters pertaining to the knee joint were oxygen consumption, knee flexion angle, gait velocity, cadence, stride length, stance time, and maximum vertical ground reaction force (Sagawa et al., 2011). While some of these parameters are not feasible to measure within a prosthetic knee system, the following sensors are commonly used to determine knee-joint parameters: accelerometers (Lawson et al., 2010b; Seel et al., 2014; Torrealba et al., 2010), gyroscopes (Miyazaki, 1997; Seel et al., 2014), goniometers (Boonstra et al., 1993; Kumar et al., 2010), magnetometers (Chelius et al., 2011), and linear sensors (i.e. hall pass or linear potentiometer). Many lower-limb prosthetic systems use hybrid arrays with a combination of different sensors to contribute to the knee's sensing capabilities. These hybrid arrays are often called inertial measurement units (IMUs).

While most gait movement happens on the sagittal plane (forward and backward movements), the transverse (up and down) and frontal (side-to-side) planes do also show significant movement, especially in female gait due to physiological differences in hip joint

anatomy. The planes and related axes of the lower limb skeleton are shown in Figure 4.1. In practical application, the center of this coordinate system can be shifted to apply directly to each body segment, such as the thigh and shank. The vertical axis of such shifted systems would then lie parallel to the long axis of the segment, but the sagittal plane would remain the same.

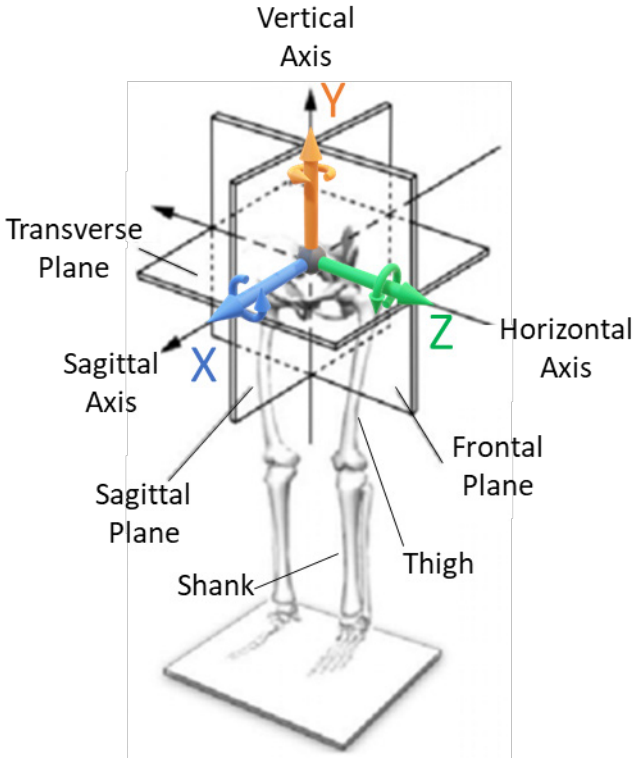

Figure 4.1: Planes and axes of the lower limb skeleton. Figure adapted from (Ren et al., 2019).

IMUs attached to body segments can be oriented along the planes and axes described above. Figure 4.2 is an example of an IMU that contains a three-axes accelerometer, gyroscope, and magnetometer. Both the accelerometer and magnetometer measure linear movement, while the gyroscope (second image in the trio) measures rotational movement along the axes.

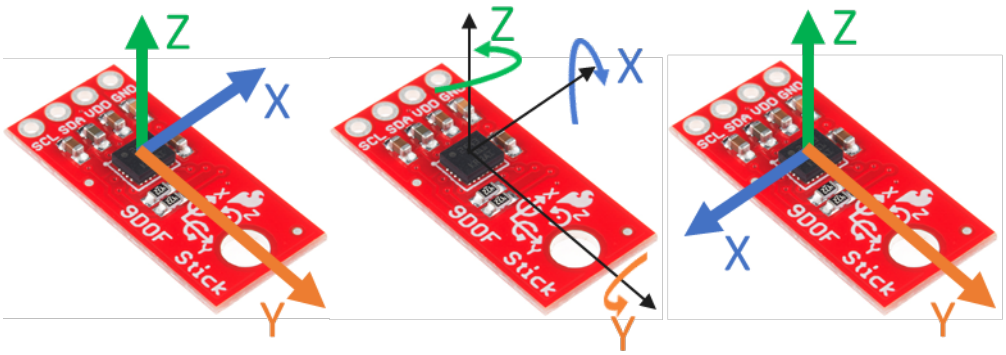

Figure 4.2: SparkFun 9-DoF Sensor Sticks with axes orientations for accelerometer (left), gyroscope (middle), and magnetometer (right). Image from sparkfun.com/products/13944.

4.2.2 Machine Learning Networks

As stated previously, ML networks have been successful in classifying a variety of gait types and conditions and differentiating between gait activities. The types of networks used have varied between Nearest Neighbor, Random Forest, Neural Networks, and combinations of those and more. Fundamentally, the classifications of those networks have focused on gait mode classification (walking, running, etc.). Though the input data is time-dependent, gait mode classification is not considered a forecasting task.

However, the current research does involve forecasting and gait classification due to the machine's requirement to use a continuous time-dependent data stream to predict gait mode. To do this, forecasting on a time series requires different configurations or types of ML networks that retain information from previous predictions known as recurrent neural networks (RNN). A popular RNN is Long Short-Term Memory (LSTM), see Figure 4.3 below (Hochreiter & Schmidhuber, 1997).

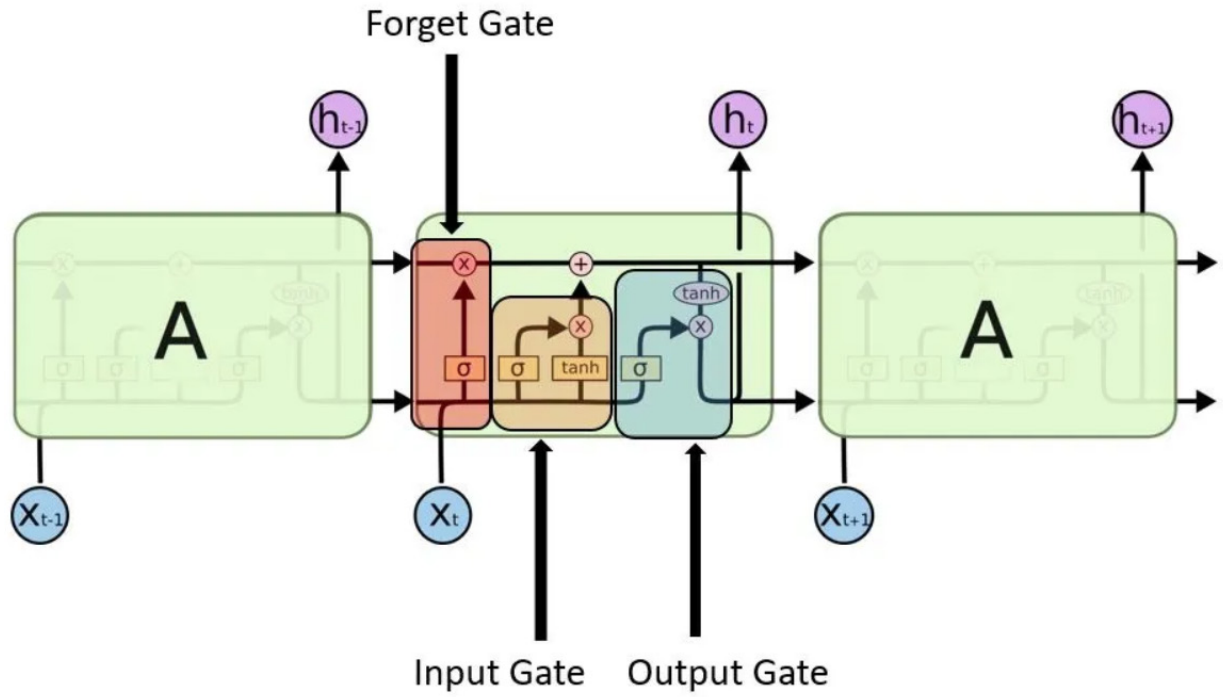

Figure 4.3: Visual representation of a Long Short-Term Memory (LSTM) network block. Image from (Mittal, 2019).

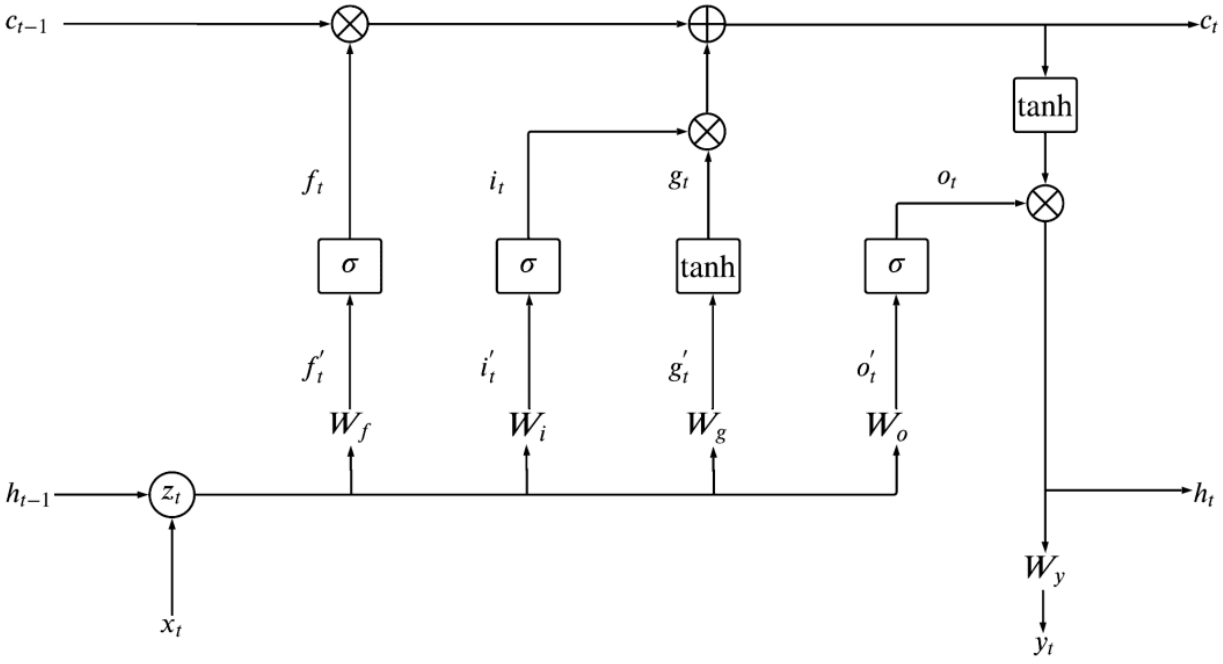

Figure 4.4: More detailed visual representation of a LSTM. Image from (Gosthipaty et al., 2022).

LSTMs function by using three types of data gates within each memory block: forget, remember/input, and output. As time progresses, the network retains previous information via two recurrent states, the memory state (c_t) and the hidden state (h_t) , seen in detail in Figure 4.4. The blocks of the network receive the network input (x_t), the hidden output of the previous block (h_{t-1}), and the memory data stream from the previous block (c_{t-1}). Learning layers (memory blocks between the initial input and final output of the network) decide what information to forget from the data stream, what information to add to the data stream, and what prediction to make (y_t). The output is a combination of the current block's data stream and the block's prediction. The dependency on past predictions coupled with the ability to discard old information, allows LSTMs to function in time series where the subsequent output depends on the previous output.

LSTMs use a non-linear sigmoid function (σ) which outputs either 0 or 1 and is multiplied to the data stream to either open or close a given gate. The input and output gates also pass data through a tanh function to determine how much of the stream is passed to the memory state (c_t) and the hidden or prediction state (h_t or y_t, respectively). Coefficients or weights (W) are determined through model training and tuning for each gate and output by minimizing an error function.

The weights for the network are determined using a training data set of previously collected instances where the gait phase (classification) is already known. This ground-truth data allows the network to learn the weights as it uses historic input data to predict the gait phase class. The network is then tested against a hold-out test sample of historic gait data and its performance on the test set is measured against the known truth.

The combination of different classifications models into one final prediction is called ensembling. With each model tuned to observe slightly different patterns or occurrences,

ensembles improve overall prediction accuracy. Transfer learning is an additional method to improve prediction accuracy. It is done when networks that have been trained for one application are used in a similar, but separate context (transferred to a new context). This is used regularly when larger, more general datasets exist to train networks for broad classification tasks using comparable forms of data (Weiss et al., 2016). Those general networks are then more accurate when used with similar data in a more specific application. For example, using large amounts of healthy patient gait data to train a network that will learn to identify amputee walking and stumbling events. The weights and biases of these pretrained models are then used to initiate the network for the specific data and predictions of interest.

4.2.3 Evaluation Metrics

While error functions are used to optimize an ML network, accuracy is often the metric used to evaluate the performance of the network. Standard evaluation of networks is done by presenting the network with a new set of inputs and comparing the predicted outputs to the known ground truth. Table 4.1 shows a standard confusion matrix that compares the predicted classifications to the actual, ground truth classifications (Chawla et al., 2002).

Table 4.1: Standard confusion matrix comparing predicted class to actual class.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

In addition to accuracy, other metrics, such as precision and recall, can be calculated depending on use-case requirements given that each metric focuses on different performance goals. Formulas for each of these metrics are given below (Olson & Delen, 2008).

Equation 4.1 $$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Equation 4.2 $$Precision = \frac{TP}{TP + FP}$$

Equation 4.3 $$Recall = \frac{TP}{TP + FN}$$

Equation 4.4 $$FPR = \frac{FP}{FP + TN}$$

Equation 4.5 $$F\text{-}Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Accuracy measures the ratio of all true (correct) predictions, positive or negative, over the sum of all predictions made. Precision measures the ratio of correctly classified positives to the total number of predicted positives (i.e. of all positive predictions made, the ratio classified correctly). Recall, also known as True Positive Rate (TPR), measures the ratio of correctly classified positives to the total number of actual positives (i.e. of the total instances in the positive class, the ratio of instances identified correctly). False Positive Rate (FPR) measures the ratio of incorrectly classified positives to the total number of actual negatives (i.e. of the total instances of the negative class, the ratio of instances identified incorrectly). Finally, the F-Score is a combination of precision and recall that balances both measures against each other. It measures the model's ability to predict the positive case both completely (recall) and accurately (precision).

4.3 METHODS

To train the ML network, appropriate data first needed to be collected and classification labels added manually. Amputee gait and stumble data were collected using a sensor system

embedded in the LIMBS M3 prosthetic knee. The knee had the same rotational geometry of the intended and completed design for Aim 1, and the sensor array was made of two, nine degrees-of-freedom (DoF) sensors—one attached to the thigh and one to the shank segment of the knee (pictured in Figure 4.5). Each sensor recorded data from a unique perspective. From a position-time perspective, the data progressed from position (magnetometer) to change in position (gyroscope) to the rate of change of position (accelerometer). Furthermore, the sensors did this from different reference frames. The magnetometer, which measured magnetic force, displayed information about the sensor's orientation with respect to the earth's magnetic field, assuming there was no other large magnetic force nearby. The gyroscope measured angular velocity, allowing analysis of the rotational components of the system. Finally, the accelerometer measured linear acceleration, which added a final dimension to the data.

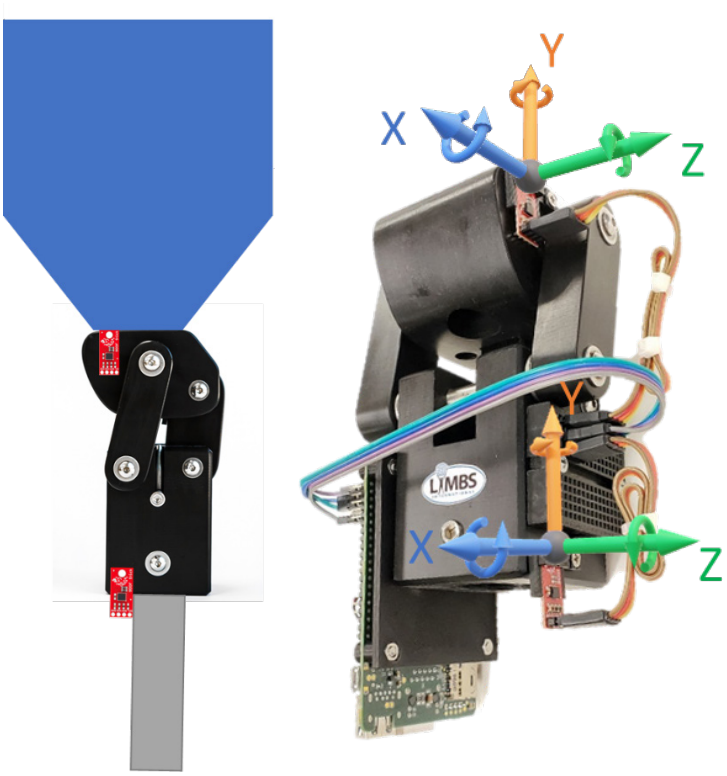

Figure 4.5: Data collection system. Left shows model with sensors, pylon, and socket. Right shows device with coordinate systems for the sensors.

Subjects were asked to walk at selected speeds on a treadmill while in a safety harness, and stumbles were induced three different ways: (1) to simulate inadequate leg momentum during a step, a bungee attached to the foot applied a breaking force to their prosthesis during gait; (2) to simulate tripping over an obstacle, a bumper was inserted in front of their prosthesis during gait; (3) to simulate uneven surfaces, an object was dropped onto the treadmill while the subject was stepping. These three events cover the primary forms of stumble during gait as described by the literature and aided in collecting a variety of stumble simulations to train the ML network to predict gait activity modes. For the purpose of classification, the data was split into four events: walking, bungee stumble, obstacle stumble, and uneven stumble.

4.3.2 Gait Trials

The study was reviewed and approved by The University of Texas at El Paso's (UTEPs) IRB for study 1329153-3. Four subjects consented to the study. Due to a technical error, one subject was rescheduled after consenting, but not collecting data. The study was permanently put on hold due to Covid-19 before data could be collected for that subject. Therefore, the training study had data from three subjects, but are labeled S1, S2, and S4. Subject qualifying criteria were as follows: age between 21 and 60, unilateral transfemoral amputation, no secondary health conditions or neurological disorders, activity level of at least K3, and general good health.

The trials were conducted at UTEP in the Virtual Reality and Motor Control (VRMC) lab, located in Room 114 in the Campbell Building, using the data collection system, Biodex RTM600™ Rehabilitation Treadmill, and Biodex Unweighing Harness. Sherie Ford, CPO, from Hanger Clinic performed the prosthetic fitting. The data collection system was made up of two SparkFun nine-DoF Sensor Sticks and a Raspberry Pi 3 (or Google Coral).

The unweighing harness was rated up to 300 pounds and used to support patients when the prosthesis did not support their weight. It featured three shared points of pressure through the hip straps, the gluteal straps, and the lumbar-thoracic harness. Harnesses similar to this have been used throughout the literature to prevent injury during trials that induced stumbling (Cordero et al., 2004b; Schillings et al., 1996b; Zhang et al., 2011). To prevent chances of injury, the harness was adjusted so that in the event of a fall, neither the patient's arms nor knees touched the ground.

The data collected was: three-axis linear acceleration, three-axis rotational velocity, and three-axis magnetic fields for both the shank and thigh of the subject's prosthetic leg, along with the corresponding timestamps and labeled stumble instances (example in Table 4.2). Data was collected during walking and stumbling activities using a treadmill and unweighing harness to prevent injury risk. The induced stumbles were also timed and recorded. Subjects were aware of stumble induction types, but not the precise timing of each event.

Table 4.2: Example data collected for one sensor (nine values per sensor). "g" is gyroscope, "a" is accelerometer, and "m" is magnetometer. Each is represented on the x, y, and z planes. Actual data collection contained two sensors for 18 total values plus the stumble marker column of zero or one for walking or stumbling classes, respectively.

time	**g2_x**	**g2_y**	**g2_z**	**a2_x**	**a2_y**	**a2_z**	**m2_x**	**m2_y**	**m2_z**	**stumble**
$\mathbf{x}_t$	32.12	-61.2	-148	0.73	0.43	0.907	-0.3	-0.46	0.117	0

The subjects were fitted with a LIMBS M3 knee that contained the sensor array. The array in no way impeded the normal movement of the prosthetic knee. A prosthetist fitted the prosthetic appropriately and restored the original prosthesis at the end of the trial. The foot on the knee system was the Niagara Foot v1 with a rubber sole. Subjects were fit with the harness before walking on the treadmill.

Subjects were asked to walk at three self-selected speeds for five minutes, and a final five minutes of walking at speeds that were varied within the self-selected speed ranges. This walk was without induced stumbles. After this acclimation period, the patients were asked to wear a pair of glasses that restricted their vision of the ground. These were worn to prevent the subjects from reflexively predicting stumbles as they saw them approaching. Patients were asked to walk normally at a speed they selected between 0.8 m/s and 1.5 m/s. Stumbles were induced during that time. The patients were asked to allow themselves to fall naturally until caught by the harness. This process continued until a minimum of 30 stumbles had been recorded. Finally, subjects were asked to stand for one minute while shifting their weight on and between legs.

Recorded Data

Each stumble-event was recorded and tracked with the time of the data stream. The frequency of the data recording was 60 Hz. Data recordings from each sensor included three-dimensional sensor data of acceleration, velocity, and magnetic field for the prosthetic knee. The data collection system had one sensor attached to the thigh and one sensor attached to the shank. Each sensor was oriented for acceleration and magnetic field to be orthogonal to and velocity to be rotational on the Frontal (X), Transverse (Y), and Sagittal (Z) planes with respect to the body segment, Figure 4.6. If the subjects stumbled, a researcher pressed a button that tracked stumble alongside the sensor data.

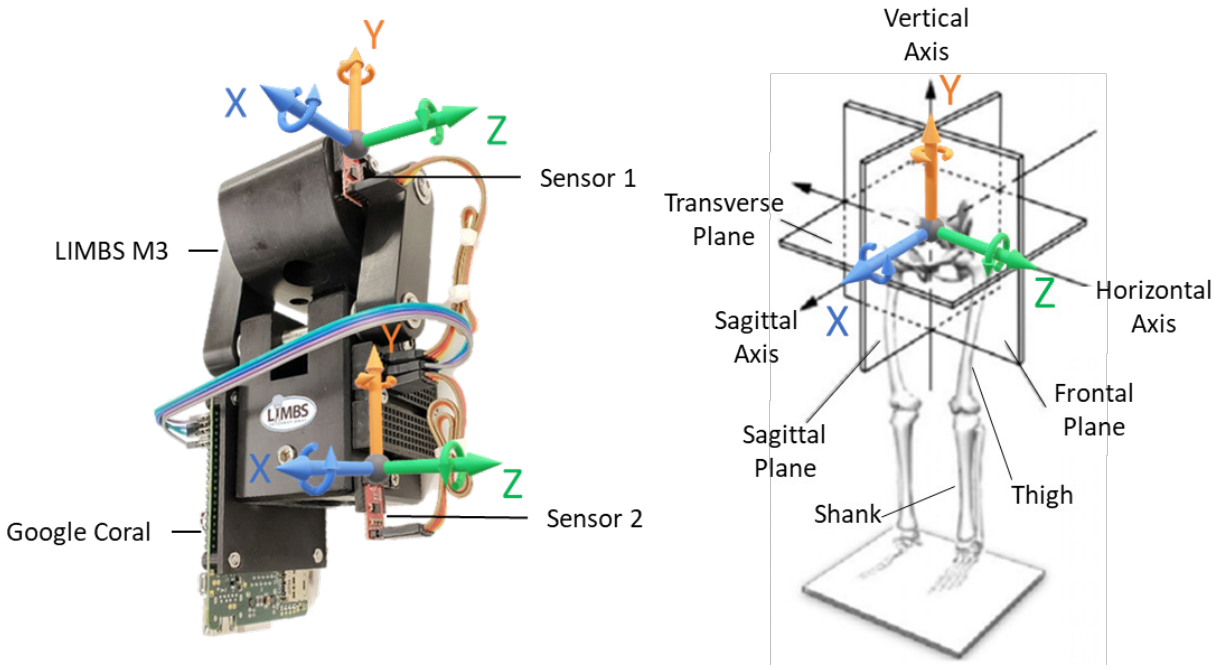

Figure 4.6: Data collection system (left) with anatomical orientation (right). Skeletal figure adapted from (Ren et al., 2019).

During data evaluation and preparation, it was observed that several of the tracked stumbles were not aligned with actual stumble induction. By observing all channels of sensor data, the stumbles were manually re-marked, and several were deleted for being indiscernible from normal gait. The distribution of the cleaned stumbles can be seen in the results section.

4.3.3 Machine Learning

The overarching method behind training the ML model was incremental testing with increasingly complex mechanisms. The ML architecture was implemented through Keras and TensorFlow (Abadi et al., 2015; Chollet & others, 2015) Described below are the methods used with specifics being found in the “Models” section.

Classes

Fundamentally, the data had four classes: walking, stumbling by bungee, stumbling by obstacle, and stumbling by uneven. However, the classes were also at times combined into simply walking and stumbling. In this case, the separate stumble classes were summed into one.

To train the networks, the data was split into training and testing data sets. Training sets were used to identify data stream patterns that lead to successful predictions; test sets were used to evaluate the model's performance. Because of the time-series-dependent nature of the data and prediction sequences (motions leading up to a stumble or non-stumble are sequential and time-dependent), samples could not simply be randomized and split for training and testing. Rather, to avoid compromising network integrity, the train/test split used a leave-one-out (LOO) methodology. LOO views each subject as one complete case, so that the sampling will iterate through all subjects as the test set. This eliminated the overlap of training and testing data and allowed the network to be trained on two subjects while predicting the events of the third iteratively. To offset data class imbalances, artificial data points were created during stumbling events based on raw data curve fitting techniques.

Additionally, the classes of standing and sitting were deemed outside the scope of this project due to lack of data, proper evaluation methods, and impact on control methods. In the end, the remaining classes were a summative "stumble" and "walking."

Ensembling and Optimizing

Ensembling several networks was done by training networks with variations and choosing the most likely prediction from their combined results. A few methods were applied to increase accuracy and decrease the computational load. First, the features were evaluated, and it was determined that magnetometer data was not highly reproducible. Though easy to interpret and

understand, as demonstrated in section 0, magnetometer data is dependent on the sensor's orientation relative to the magnetic field of the earth. Therefore, the lab setup made the data for all three initial subjects comparable, but there was no guarantee that future tests would have similar and consistent orientations. Therefore, all six magnetometer features were eliminated.

Transfer Learning

Using a larger more general dataset to pretrain a model that is then fine-tuned with a more specific dataset is a common method called transfer learning. The justification is that when using similar types of data, the model can learn the shape of the data type, and that knowledge will then be used in the specific model even if the output classes are dissimilar. The methods used for the transfer learning in this study were as follows: develop a base network with reasonable success on the HuGaDB, freeze learning on some layers of this network, and add components of the stumble network as unfrozen layers. The base network for the HuGaDB dataset was the same as being tuned for stumble prediction, and it performed remarkably well, achieving an accuracy of 99.1% and similarly high scores on all precisions and recalls for four classes: walking, running, sitting, and standing. Therefore, all but the last layer (output layer) in the HuGaDB trained model were alternatively frozen, unfrozen, included, and excluded with various parts of stumble model grafted on. This process was repeated exhaustively while using the standard LOO practices.

Hyperparameter Tuning

The final stage of network configuration was hyperparameter tuning within the bounds established by the earlier testing. The table below highlights the parameters tuned, the ranges, and the steps. Training batch size was also evaluated separately within 60, 80, 120, 200, 400, and 600. Hyperparameter evaluation was set to maximize accuracy and F-Score. This was implemented through the KerasTuner (O'Malley et al., 2019)

Table 4.3: Hyperparameter tuning parameters, minimum and maximum range, and step size.

Parameter	Min	Max	Step
LSTM 1 Units	25	500	25
LSTM 2 Units	25	500	25
Dense Units	25	500	25
Dropout Rate	0	0.9	0.1

Various combinations of the different layers were also tried with the layer orders being switched, disabled, or tuned individually. The hyperparameter output was evaluated for the top three combinations of values for each LOO dataset. The combination that was the most common across the different datasets was chosen as the final output.

Models

Classification of the data was done in two major stages: initial and deployed. Neither of these stages represents one network, but rather the result of many iterations. A difference is made between them because the initial network was developed after an exploration of different ML algorithms and basic data cleaning methods. The deployed network refined the basic methodology set by the initial network and implemented a more robust algorithm application.

Before the gait trials were conducted, the gait data from HuGaDB was explored and classified with a simple LSTM network. Because HuGaDB included similar data as intended to be used, namely thigh and shank accelerometer and gyroscope data, the principles learned were applied to this study. While the LSTM network was preselected for its success with the HuGaDB, Gradient Recurrent Unit (GRU) and Simple Recurrent Neural Network (SimpleRNN) were still tested in comparison by evaluating all three networks' performances with different networks sizes (50, 100, 200, and 400) and output layer activation functions (softmax, sigmoid, or none). This was done at the beginning with just an input and output layer model. For the recurrent networks,

the data was transformed to include 10 past data points along with the current sensor sample (example below in Figure 4.7).

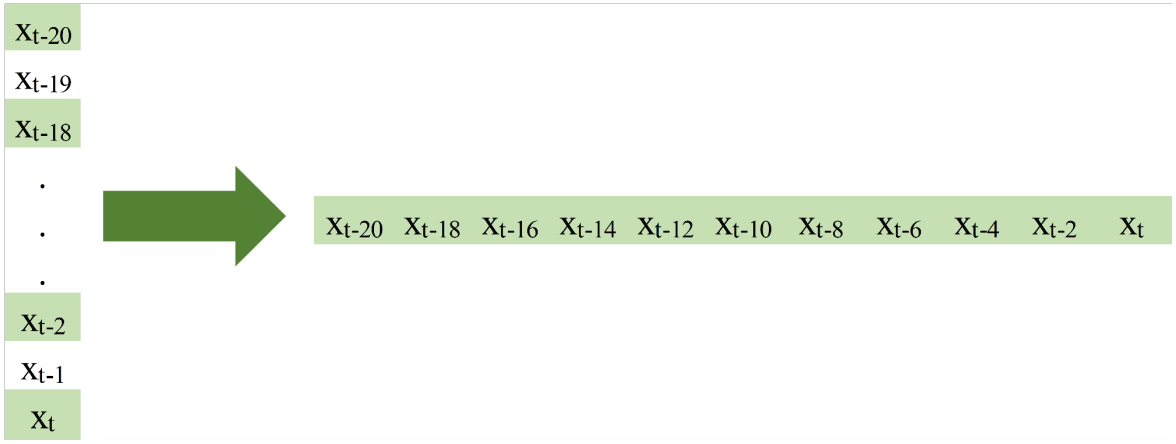

Figure 4.7: Illustration of how the historic data was compiled into one row of data for the network. The data on the right represents one "sample," and each data point "x" included all the sensor data shown previously (12 values per data point).

Additionally, to model the rule-based method of commercial prosthetic knees, a simple rule classification model was developed. This model used a calculated knee angle (>25 deg) with knee angular velocity (>100 deg/s; extension) obtained from the sensors to determine step occurrences, particularly the leg extension phase. During each such extension, the model checked for three rules: bungee, obstacle, and uneven detection. Each used repeatable sensor information to establish whether that type of stumble occurred. When a stumble was detected, it would continue for 0.5 seconds after the last detection. Further details on this rule system can be found in Appendix 9.1. This "Simple Rule" model was used as a baseline comparison for the ML developed through this aim.

Thereafter, data was cleaned with a MinMax scaler that adjusted each data channel to fall between 0 and 1. It was fit to training data and applied to transform validation and test data. Training files were augmented with artificial data and manually marked stumble data was used.

Various learning rates and optimizers were attempted, but RMSprop with its default learning rate of 0.0001 performed well most consistently.

The initial machine learning algorithm applied to the collected data was an LSTM network with one input layer, three hidden layers, and one output layer. The layers were a 300-node input LSTM, a 200-node LSTM hidden layer, a 100-node dense hidden layer, a 0.9-dropout hidden layer, and a two-class output with softmax activation layer. The train length was 20 epochs with a categorical cross entropy loss function using the RMSprop optimizer. Training involved the full 18 features from the sensors and included 10 historical datapoints. The three different forms of stumble were trained as one general "stumble" class. Because there were only three subjects, LOO was used and two were used for training and one for testing. For the purpose of training, the data collected during the induced stumbles was used. This data included both gait and stumble-events at a roughly 9:1 ratio, though artificial data shifted that ratio to 4:1. The validation dataset was 15% of the training data. At this stage, one minute of non-stumble walking data from the test subject was added to the training data. This was thought to keep the algorithm from overfitting and to help the system be aware of the subject's specific gait.

After hyperparameter tuning, the deployed network retained all three hidden layers. The layer parameters chosen during tuning were a 200-node input LSTM, a 50-node LSTM hidden layer, a 300-node dense hidden layer, a 0.1-dropout hidden layer, and a two-class output layer with softmax activation. The train length was 30 epochs with a categorical cross entropy loss function using the RMSprop optimizer at a learning rate of 0.001. The data included 12 features from the two sets of 3D accelerometers and gyroscopes, and the current data was given with 10 historic datapoints equally distributed across the last 20 samples. As before, the three different forms of stumbles were combined into one "stumble" class, and artificial stumble data was used to balance

the data. Different from before, the cleaned stumble data was used. As before, LOO was used in conjunction with a small batch of non-stumble walking data from the test subject. The sum of changes between the initial and deployed networks are the following: hyperparameter tuning, feature reduction, manual data marking. Each LOO procedure was run three times and the results averaged.

4.3.4 Result Evaluation

Traditionally, ML algorithms are evaluated by various metrics, such as accuracy, precision, recall, and F-score, which all depend on the prediction to ground-truth comparison. The results given in the Data Classification section of the results highlight this principle. Yet, it became clear that for this application the ground-truth comparisons were more difficult to ascertain. The marked stumbles were often not initiated at the most accurate time, nor did they end exactly when the stumbles were over. This phenomenon was one of the reasons why the stumble data used for training was cleaned manually. However, even with a cleaned stumble, the true positive prediction would often not coincide perfectly with a marker. This led to errors in accuracy. Since an improper reaction to a few false positive predictions could cause an artificial stumble, a different metric had to be applied. Therefore, instead of predictions evaluated against the ground-truth on a rolling data-sample basis (as described in Figure 4.19 above), the predictions and ground-truths were used to evaluate individual steps instead. Steps were identified using the cyclic angular velocity peaks from the data to identify the gait cycle. Within each step, the rolling samples were evaluated, and a small six sample buffer was applied to eliminate spikes of FPs. If 80% of the six samples were positive, then the ML model was predicting a stumble for the step. This was essentially a first step toward the control system.

4.4 RESULTS

The data shown in the subsequent subsections are the results of the three-patient trial discussed in the methods. While data from the subsequent trial in Aim 4 would greatly enhance the results, those are outside of the objective of this aim. The positive prediction class is "stumble."

4.4.1 Data Collection

The data collection methodology described in Section 4.3 was successfully completed with three subjects successfully. Subjects 1 and 4 were male, and subject 2 was female. All amputations were the result of trauma, but with no secondary gait complications. Height ranged from approximately 1.63 to 1.85 m, and weight from approximately 54 to 95 kg. For each subject, approximately 900 total steps were collected. The distribution of the different stumble-events can be seen in Table 4.4, below. Measurements and data recording all happened on the side of the prosthetic knee.

Table 4.4: Breakdown of stumble-event occurrences per subject.

	Bungee	**Obstacle**	**Uneven**	**TOTAL**
Subject 1	11	5	12	28
Subject 2	12	18	29	59
Subject 4	19	20	24	63
Total	**42**	**43**	**65**	**150**

Stumble-events were induced until a minimum of ten of each event had been collected per subject. Exceptions occurred when the subject did not wish to proceed with a specific stumble-event. As mentioned in the methods, the stumbles were manually validated and re-marked, and some were eliminated. The updated table is shown below, Table 4.5.

Table 4.5: Breakdown of stumble-event occurrences per subject manually re-marked. Some stumbles were removed when there was no observable effect. Removed occurrences are marked in parentheses (-n). Total is summed after removing ignored cases.

	Bungee	**Obstacle**	**Uneven**	**TOTAL**
Subject 1	11 (-2)	5 (-0)	12 (-2)	24
Subject 2	12 (-1)	18 (-1)	29 (-1)	56
Subject 4	19 (-3)	20 (-0)	24 (-10)	50
Total	**42 (-6)**	**43 (-1)**	**65 (-13)**	**130**

Examples of the different stumble induction methods can be seen in Figure 4.8, below.

Figure 4.8: Examples of stumble induction for Subject 4 for stumble with bungee (left), obstacle (middle), and stumble with uneven ground (right). Leftmost image, demonstrating stumble with a bungee, is a simulated image showing the attachment point.

An example of sensor data during walking can be seen in Figure 4.9. Examples of the sensor data during the different stumble-events can be seen in Figure 4.10, Figure 4.11, and Figure 4.12. Each figure displays the full sensor array data, grouped by sensor type and body segment, and the tracked stumble (horizontal red line). Additionally, approximate gait events (heel strike, mid-stance, and toe-off) and beginning of stumble (vertical red line) were added. These additions were estimates but reflect historical measurements and specific sensor indications.

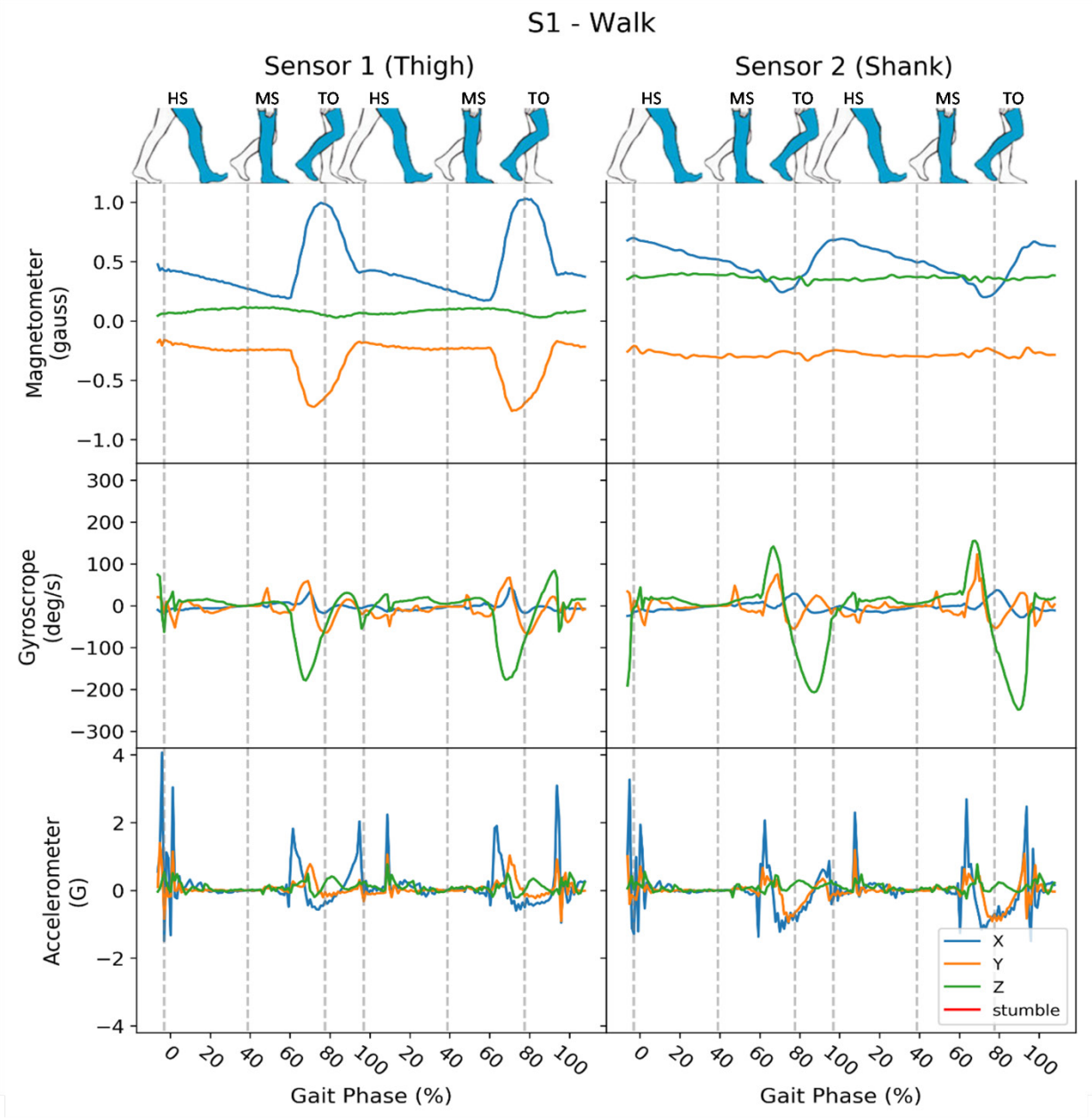

Figure 4.9: Sample of Subject 1 prosthetic knee sensor data for walking. No stumble occurred. Each column of graphs represents one sensor, and each row of graphs represents a sensor type. X, Y, and Z represent vectors orthogonal to (accelerometer and magnetometer) and rotational on (gyroscope) to the frontal (X), transverse (Y), and sagittal (Z) planes respective to the two body segments (thigh and shank). The horizontal axis shows the data respective to gait percentage and is cyclical. Vertical (grey) dotted lines represent key gait phase events of the prosthetic knee (blue). Gait events shown are heel strike (HS), mid-stance (MS), and toe-off (TO). Gait phase figures adapted from (Pirker & Katzenschlager, 2017).

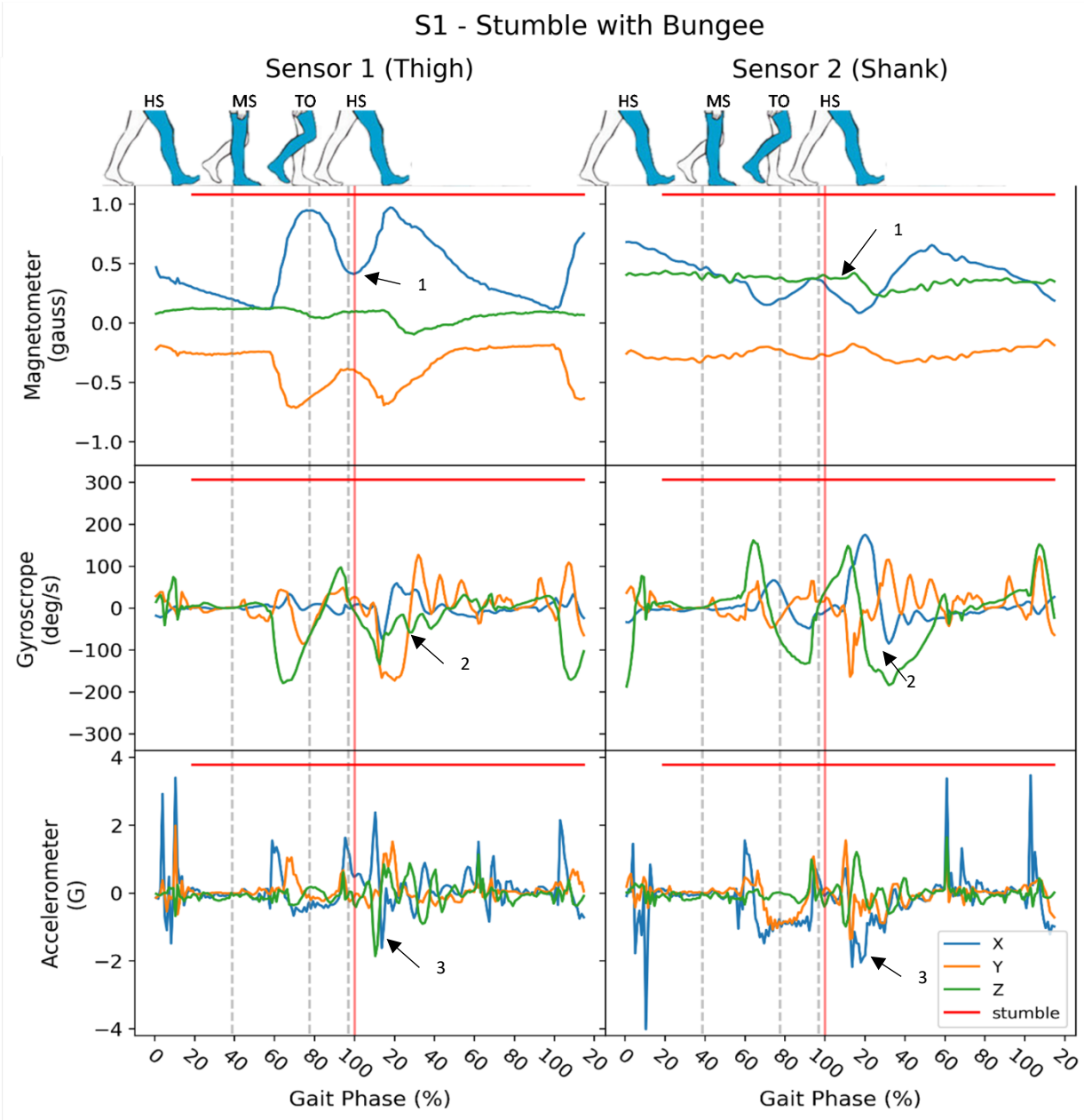

Figure 4.10: Sample of Subject 1 prosthetic knee sensor data for stumble by bungee. Stumble data recording indicated by red horizontal line. Approximate begin of actual stumble marked by vertical red line. Each column of graphs represents one sensor, and each row of graphs represents a sensor type. X, Y, and Z represent vectors orthogonal (accelerometer and magnetometer) and parallel (gyroscope) to the frontal (X), transverse (Y), and sagittal (Z) planes respective to the two body segments (thigh and shank). The horizontal axis shows the data respective to gait percentage and is cyclical. Vertical (grey) dotted lines represent key gait phase events of the prosthetic knee (blue). Gait events shown are heel strike (HS), mid-stance (MS), and toe-off (TO). Arrow one highlights magnetometer troughing, Arrow two highlights angular velocity spikes, and Arrow three highlights acceleration spikes. Gait phase figures adapted from (Pirker & Katzenschlager, 2017).

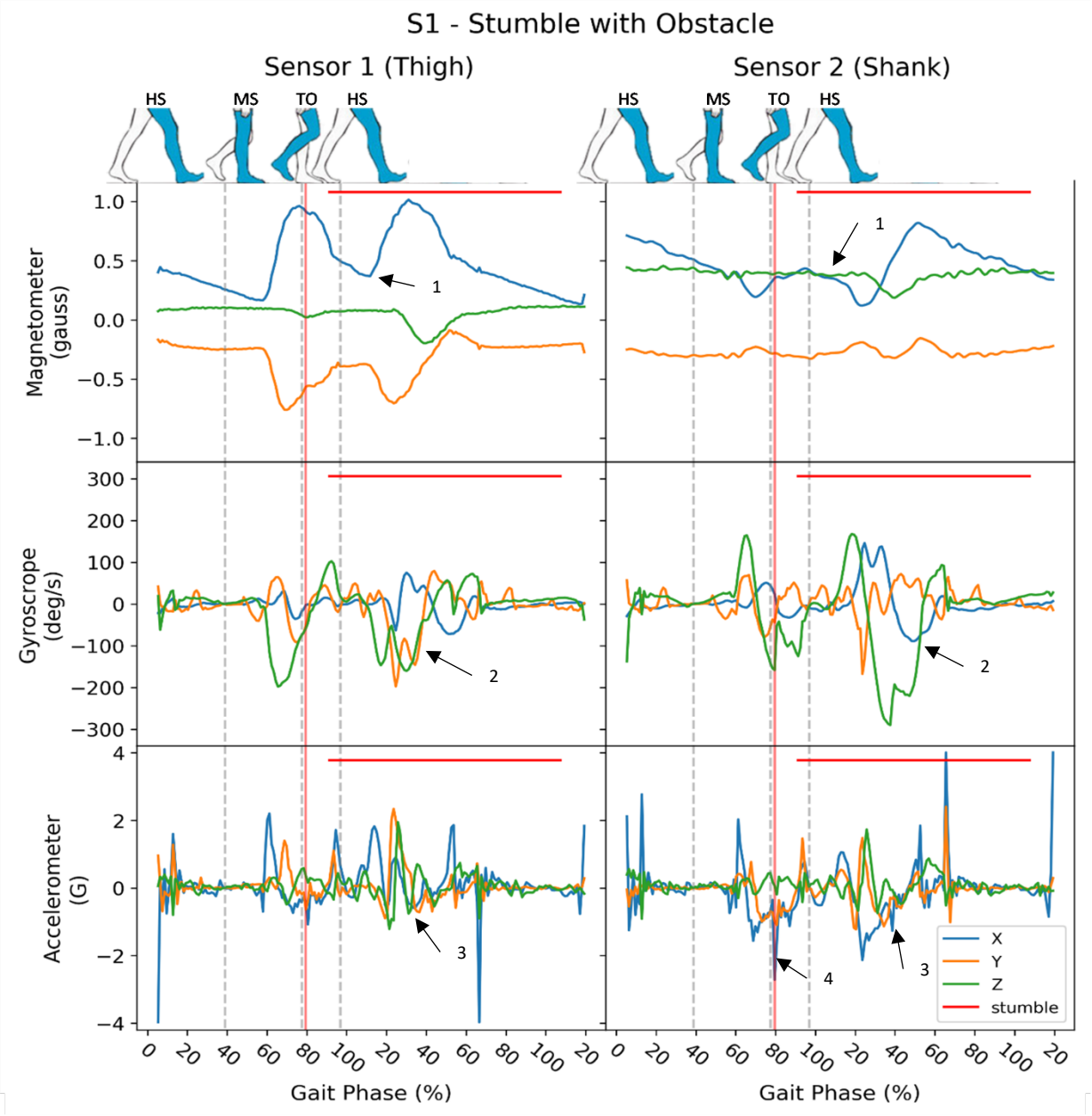

Figure 4.11: Sample of Subject 1 prosthetic knee sensor data for stumble by obstacle. Stumble data recording indicated by red horizontal line. Approximate beginning of actual stumble marked by vertical red line. Each column of graphs represents one sensor, and each row of graphs represents a sensor type. X, Y, and Z represent vectors orthogonal (accelerometer and magnetometer) and parallel (gyroscope) to the frontal (X), transverse (Y), and sagittal (Z) planes respective to the two body segments (thigh and shank). The horizontal axis shows the data respective to gait percentage and is cyclical. Vertical (grey) dotted lines represent key gait phase events of the prosthetic knee (blue). Arrow one highlights magnetometer troughing, Arrow two highlights angular velocity spikes, Arrow three highlights acceleration spikes, and Arrow four shows the obstacle hitting the shank. Gait events shown are heel strike (HS), mid-stance (MS), and toe-off (TO). Gait phase figures adapted from (Pirker & Katzenschlager, 2017).

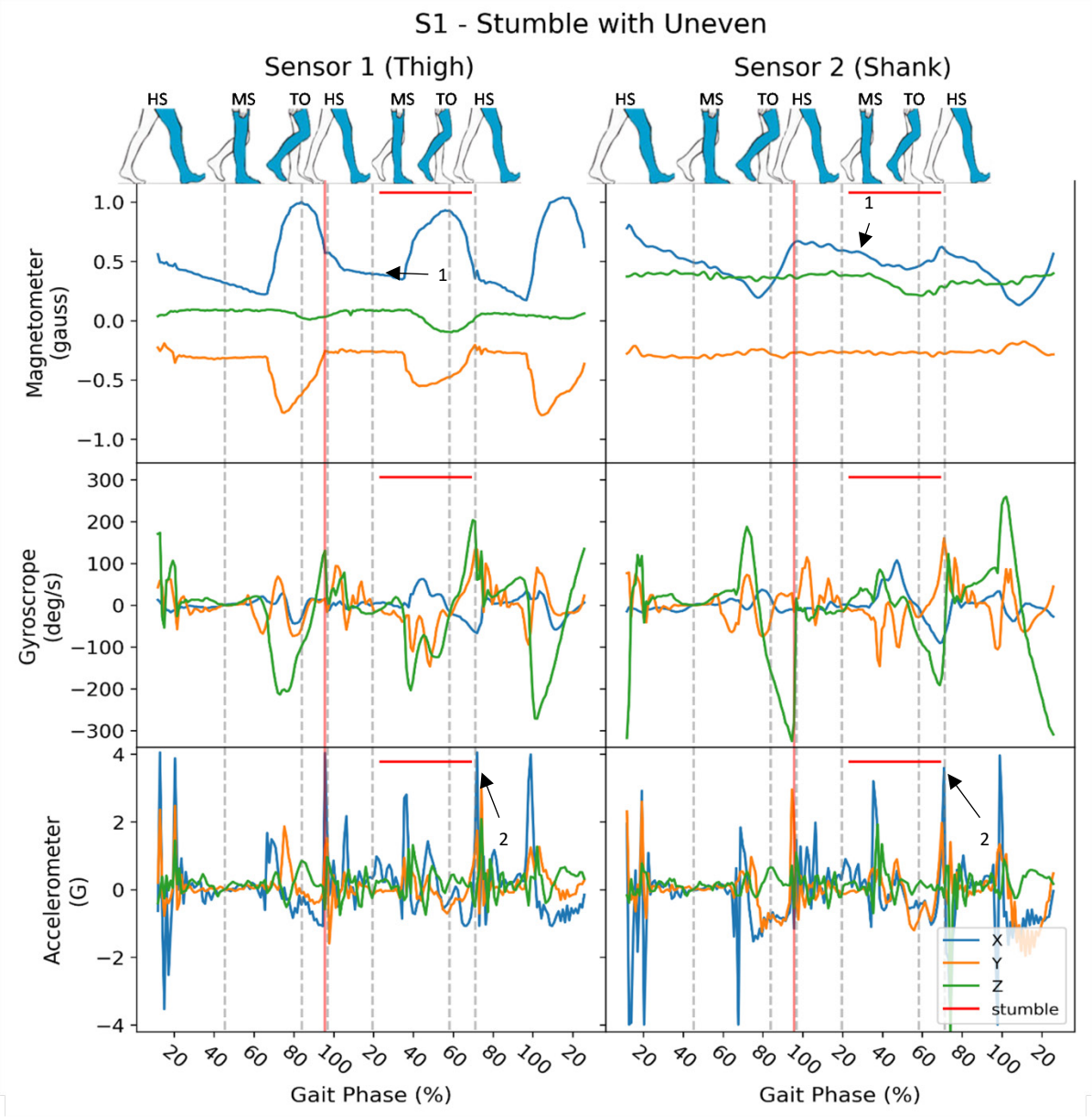

Figure 4.12: Sample of Subject 1 prosthetic knee sensor data for stumble by uneven surface. Stumble data recording indicated by red horizontal line. Approximate beginning of actual stumble marked by vertical red line. Each column of graphs represents one sensor, and each row of graphs represents a sensor type. X, Y, and Z represent vectors orthogonal (accelerometer and magnetometer) and parallel (gyroscope) to the frontal (X), transverse (Y), and sagittal (Z) planes respective to the two body segments (thigh and shank). The horizontal axis shows the data respective to gait percentage and is cyclical. Vertical (grey) dotted lines represent key gait phase events of the prosthetic knee (blue). Gait events shown are heel strike (HS), mid-stance (MS), and toe-off (TO). Arrow one highlights magnetometer troughing and Arrow two highlights acceleration spikes from HS. Gait phase figures adapted from (Pirker & Katzenschlager, 2017).

4.4.2 Machine Learning Algorithm

This section describes the results of the initial data exploration, transfer learning, and ensembling methods implemented to improve the performance of the machine learning model.

Data Exploration

The data below, Table 4.6, represents the first foray into different recurrent networks. SimpleRNN appears to not perform well despite the variances of structure. Though the results are very similar for GRU and LSTM, the previous experience with LSTM made it the more logical choice to continue on with.

Table 4.6: Accuracy output of simple recurrent models using one recurrent layer (variable "Network") and one Dense for class output to walking and all three stumble modes. Data is filtered to show only accuracies above 0.50 (and thus excludes all "None" activations). Configurations with accuracies of 0.70 or greater are bolded. The "Simple Rule" results for all three subjects given last row.

Network	Net Size	Activation	Accuracy
LSTM	**50**	**softmax**	**0.70**
LSTM	**100**	**softmax**	**0.71**
LSTM	**200**	**softmax**	**0.76**
LSTM	400	softmax	0.67
LSTM	**50**	**sigmoid**	**0.70**
LSTM	100	sigmoid	0.66
LSTM	**200**	**sigmoid**	**0.76**
LSTM	400	sigmoid	0.67
GRU	50	softmax	0.60
GRU	**100**	**softmax**	**0.73**
GRU	**200**	**softmax**	**0.76**
GRU	**400**	**softmax**	**0.74**
GRU	50	sigmoid	0.60
GRU	**100**	**sigmoid**	**0.73**
GRU	**200**	**sigmoid**	**0.76**
GRU	**400**	**sigmoid**	**0.74**
SimpleRNN	50	softmax	0.67
SimpleRNN	200	softmax	0.52
SimpleRNN	400	softmax	0.53
SimpleRNN	50	sigmoid	0.64
SimpleRNN	200	sigmoid	0.57
SimpleRNN	400	sigmoid	0.60
Simple Rule			**0.79**

Initially, three stumble classes were defined: obstacle, uneven, and bungee. However, preliminary work showed that specifying between these different types of stumbles decreased predictive power, increased error, and learning these classes ultimately led to overfitting. Therefore, all types of stumbling were combined into the one "stumble" class. Combining these classes led to an accuracy increase of 2% (78%) with a LSTM of size 200.

Transfer Learning

It was assumed that transfer learning would be beneficial for this work. A large dataset, HuGaDB, exists with IMU sensor data for healthy subjects during many activities including running, walking, standing, and sitting (Chereshnev & Kertész-Farkas, 2017). While the data did not directly overlap in sensor selection (the database does not include magnetometer data) nor prosthetic devices, the trends and pattern recognition from a large dataset (~600 minutes) were expected to aid the performance of the network. However, even though several problems, including corrupt data from the dataset, were overcome, the transfer learning ultimately did not achieve any decent results.

Ensembling

While it was planned to use different recurrent machine learning algorithms to ensemble and categorize the data, this ultimately met with some logistical alterations. Though several LSTM, GRU, or possibly even N-Beats networks running concurrently and ensembling their predictions would likely have a positive impact upon results, the practical reality of real-time predictions in a limited environment had to be considered. The Google Coral performed admirably well at making real-time predictions at a rate of 60-80 Hz. However, testing showed that adding even one alternative network slowed the predictions down enough that 60 Hz could not be maintained without compounding error.

4.4.3 Data Classification

As mentioned in the methods, the classification results were split into two different models, "Initial" and "Deployed." Both sets of results relate to the same dataset of three subjects. The improvements to these models has resulted in an increase in accuracy over the 78.8% of the simple rule baseline.

Initial

Table 4.5 shows the confusion matrix of a network that was trained by Subjects 1 and 2, and evaluated by Subject 4 with no data overlap. The right column shows the metrics measured: recall, precision, FPR, F-Score, and accuracy (83.0%).

Table 4.7: Confusion matrix for trained LSTM network without test-subject walking data. Right column is recall, precision, false positive rate (FPR), F-Score, and accuracy. Evaluated on Subject 4.

		Predicted			
		Stumble	**Walking**		
Actual	**Stumble**	1072	3247	**Recall**	24.8%
	Walking	558	17526	**Precision**	65.8%
				FPR	3.1%
				F-Score	36.0%
				Accuracy	83.0%

In later trainings, the training data also included a one-minute sample of the variable speed walking with no stumbles for the test subject (Subject 4). The confusion matrix for this preliminary network is shown in Table 4.8. All metrics of the network increased, and there was an overall accuracy of 85.5%.

Table 4.8: Confusion matrix for trained LSTM network including test-subject walking data. Right column is recall, precision, false positive rate (FPR), F-Score, and accuracy. Evaluated on Subject 4.

		Predicted			
		Stumble	**Walking**		
Actual	**Stumble**	1875	2444	**Recall**	43.4%
	Walking	809	17275	**Precision**	69.9%
				FPR	4.5%
				F-Score	53.5%
				Accuracy	85.5%

Deployed

Table 4.9 shows the breakdown of the LOO training for each subject. It shows the network metrics, such as accuracy and recall, calculated in the typical sample-by-sample method.

Additionally, it shows an analysis of from the perspective of individual steps with the metrics listed accordingly. Each subject's data is the average of three trainings.

Table 4.9: LOO comparison and average for deployed ML model. "Network" data is sample evaluation. Data includes evaluation of step events. Each subject's data is an average of three runs of the same model.

Metric	**Subject 1**	**Subject 2**	**Subject 4**	**Average**
Network Accuracy	84.7%	87.8%	93.6%	**88.7%**
Network Precision	14.0%	32.2%	67.0%	**37.7%**
Network Recall	51.9%	50.5%	46.5%	**49.6%**
Network FPR	14.0%	9.1%	2.1%	**8.4%**
Network F1_Score	21.2%	37.9%	57.4%	**38.8%**
Total Steps	225	271	196	**231**
Total Stumbles	24	56	50	**44**
Caught Stumbles	23 (97%)	33 (57%)	41 (79%)	**33 (73%)**
False Stumbles	136 (60%)	54 (20%)	2 (1%)	**64 (28%)**
Step Accuracy	39.1%	70.7%	93.3%	**66.9%**
Step Precision	14.7%	38.0%	94.7%	**33.8%**
Step Recall	97.2%	57.5%	79.5%	**73.1%**
Step FPR	67.8%	25.6%	1.6%	**31.7%**
Step F1_Score	25.5%	45.8%	86.4%	**46.2%**

4.5 DISCUSSION

The objectives of this aim were to perform according to specific metrics for an ML model. However, first, the data had to be collected and the process of building an ML network iteratively considered.

Gait Trials

It is important to note that though the beginning of the stumble was an estimate, so was the tracked stumble event. The tracked stumble-events were recorded by a researcher when the

stumbles were induced, and the subject stumbled. However, the bungee method had significant delay between induction and actual stumble, whereas the obstacle was a near instantaneous stumble. Therefore, tracked events should be seen more as a highlighting of which gait cycle had stumbles as opposed to the exact instance of the stumble.

The various stumble modes were all selected to model scenarios encountered by amputees during normal gait. The bungee stumble was chosen to simulate a lack of adequate forward momentum of the prosthetic to lock the passive knee in preparation for heel strike. As can be seen in Figure 4.10, the data shows distinct differences. Arrow 1 highlights the magnetic sensor, which gives relative position and angle of the segment with respect to earth's magnetic field. In contrast with the walking data, X and Y for this sensor do not completely trough, as it previously did after MS, before peaking again. Arrows 2 and 3 point to large spikes in angular velocity and linear acceleration, respectively. This is especially apparent in the velocity graph, where X and Y show large spikes that are atypical for walking gait. The abnormal accelerations indicate rotation and imbalance.

The reactionary sensor data of stumbling with an obstacle does not seem to drastically vary from stumbling with a bungee. As seen in Figure 4.11: Arrow 1, the thigh and shank also do not fully trough in their movement. Similar to bungee stumble, the non-sagittal (X and Y) angular velocities and the Y and Z accelerations once again show spikes (Arrow 2). The key difference is the presence of the obstacle impact (Arrow 3). A large difference to the bungee Orthogonally to the frontal plane (X), there is a sudden spike of acceleration. This is the instance that the object struck the shank and began the stumble. This gait imbalance is resolved very similarly to bungee stumble, and gait appears to resume. In both instances, it appears as though the subject did not fully extend the knee (Arrow 1), and quickly proceeded into a shaky flexion again.

In contrast to both of these, Figure 4.12 shows stumbling with an uneven surface. This stumble once again has a diminished trough (Arrow 1), yet it can be highlighted by Arrow 2 that the subject has had a successful heel strike. This is supported by the spike in the X (orthogonal to transverse) acceleration and Z (sagittal plane) gyroscopes that can be seen in normal walking to accompany heel strike. Although, as in the previous stumble modes, there were increases in acceleration and angular velocity indicating imbalance, the subject clearly recovered and completed a step. Though the subject had to accelerate their step frequency, the sensor data shows that the subject resumed gait after stumbling.

Machine Learning Classification

As seen in the results, the initial networks showed very little difference between GRU and LSTM. Even though the scores above 70% favored the GRU on average, the LSTM performed an equal maximum (76 %) and was selected for its familiarity and success on predicting the HuGaDB dataset. The “Simple Rule” prediction performed better than all the initial ML models and was therefore used as a comparative standard. However, it was developed with very specific rules, and likely would have been limited in its accuracy bounds. Once the ML models were diversified beyond input and output layers, the models quickly exceeded the 78.8% accuracy of the simple rule baseline model with 83.0 and 85.5%. Likely, a rule-based model could be developed that approached such accuracies for stumbles, but the reliance on very specific conditions would plausibly limit its generalization. With a combination of the different stumble methods into one class, the ML models showed a generalization that would likely be applicable to chaotic stumbles that did not match defined expected stumble patterns.

Through the ML model development process, several oversights were discovered and addressed iteratively. This is to be expected when developing such a model. For instance, in typical

machine learning, data is randomized and divided into training, testing, and validation sets. However, this dataset contains time series data with historic datapoints. Therefore, a randomized dataset could very well yield a train and test sample separated by milliseconds. This was addressed by LOO, but initially many iterations of the network did not account for this. Previous experience with the similar HuGaDB dataset also encouraged the application of a similar successful ML network to this stumble dataset. Additionally, the original dataset was not cleaned to account for human error during the stumble marking. Though the methods of testing different networks were still employed, these assumptions and oversights potentially could indicate that the current model is not the most suited model for this data. The model methods were chosen almost exclusively with Subject 4 (S4) being the test set. This is reflected in the results where predictions for S4 are more accurate and precise. Though hyperparameter turning was performed with consideration for all three subjects, either S1 and S2 predict very well for S4 or the networks are still more fundamentally inclined to benefit that subject.

The objectives for this aim were to achieve a classification system with accuracy greater than 90%, precision greater than 75%, and recall greater than 60%. These values were optimistically chosen when considering the confusion matrix of Subject 4 during the Initial model trained with walking data (Table 4.8: accuracy 85.5%, precision 69.9%, and recall 43.4%). With more data and finer tuning, such goals may have been achieved. However, at that stage it was realized that the traditional approach of analyzing model performance by the sample was not practical. A stumble classification could be 0.01 seconds early and be considered a FP, and the stumble markings were never intended to perfectly extend to the end of each stumble. In fact, it became clear that the end of a stumble may be more difficult to ascertain and classify than the beginning. It was therefore realized that more realistic metrics would address not samples, but

stumble inductions and complete gait cycles for performance. The switch of metrics on the same output can be seen in Table 4.9. According to these new "instance" definitions, the model's performance on each subject averaged 66.9% accuracy, 33.8% precision, and 49.6% recall. This is a decrease for the traditional model metrics, and below the objectives of this aim. By these standards, the hypothesis of Aim 2 was unsupported by the results. However, for practical application, this result may still be improved with the implantation of a robust control system.

Limitations

A fundamental constraint of the ML model is the dataset of the three subjects. Though more subjects were scheduled, initial Covid-19 lockdowns prevented further data collection. Artificial data and data cleaning methods were used, but the fact remains that ML models require high quality, labeled data, and more diverse datasets yield more beneficial models. More data was collected during the Aim 4 trials, but this data was not available to train this model.

Because the control systems of commercial knees are proprietary, there are no direct comparisons that can be made for the evaluation of the ML model. Therefore, the limitation is that a "Simple Rule" model had to be developed to be used as a baseline comparison. However, the score achieved by this system is not meant to be seen as representative of what commercial systems would be able to achieve, rather as standard by which the new ML models could be evaluated against.

Further, many different approaches were attempted, and data was processed and reprocessed several times. It is not clear whether the current model is the absolute best model for the data as it is applied currently. Additionally, the low number of subjects potentially restricted the model selection with overfitting of the general parameter selection process. Therefore, while

the model has performed admirably, the model may be constrained to data similar to the training data.

Future Work

Future work would include revisiting the model selection process and hyperparameter tuning to validate the model and parameter choices with all data transformations, such as, feature selection, data cleaning, artificial data, and stumble mode combination.

Exploration of the data gathered in Aim 4 would likely improve the model's performance. With the early limitation of subjects during the initial trials, the expansion of the dataset provides opportunity to greatly improve the classification model and parameter choices. The additional methods, such as ensembling and transfer learning warrant further investigation. Particularly for ensembling, a feature selection method should be applied. The current system uses twelve channels of data, but it is unclear if all of them are useful or if some of the calculated data, such as angle or knee velocity, could not replace and improve on the current features. With proper feature selection, there is the possibly of a more accurate and more compact model. If the networks could be made smaller, the possibility of multiple parallel ensembled models in real-time could result in greater network performance. Overall, this research establishes the base protocol, but there are many avenues for exploration for continued work with this dataset.

Broader Implications

Time series data classification is no new task, and several studies have suggested networks that can be implemented in real-time either because of their usefulness or because their architecture was designed to be low impact. The deployed network demonstrated in this aim not only was designed to be deployed in real-time but was successfully deployed in the later aims. Additionally, as the HuGaDB initial classification suggested, the activities of standing, sitting, and running are

accurately classifiable by the LSTM architecture. Future work should include an expansion of the classes for more versatile knee applications.

A fundamental shift in the research occurred during the model development process and displaying their data. Though traditional sample-based analysis is a comfortable and known quantity, it does not apply to all datasets necessarily. For this study, it became apparent that sample-based metrics were giving much higher accuracy than was realistic. The imbalance of classes contributed to this especially. A new perspective on the data was developed and applied in Aim 3, but continued, future analysis of this data should be conducted. Different tuning methods, such as utilizing a cost-matrix with custom weights with penalties and rewards for step metrics, should be implemented, and the data should be retrained according to this different set of metrics. This would allow a more customized tuning based on perceived costs of certain types of errors.

Chapter 5: Aim 3 - Develop a control algorithm for the knee mechanism of Aim 1 using the machine learning system of Aim 2 as an environmental differentiator to switch between control models for the purpose of emulating human activity.

5.1 OBJECTIVE

The control algorithm will switch between true changes in state within 0.15 seconds and will switch out of a false change of state within 0.30 seconds. Using embedded sensor information from the device, the control algorithm will adapt to current gait speed and transition to alternative activity models based on the activity classifier. This will use the additional sensor information along with the predicted activity from Aim 2 to control the physical knee device from Aim 1 – achieving predictive, reactive, and adaptive control for each wearer.

5.2 INTRODUCTION

The control of a prosthetic knee depends on if the knee is passive or active. Passive systems are nonreactive to external input and are fully dependent on the current physical configuration of the knee. Such systems have a set, albeit somewhat adjustable, stiffness to the flexion and extension of the knee. Active knees depend on a form of computational system and are most commonly controlled by microprocessors. These systems are deemed active because they monitor the environmental factors through sensors and respond accordingly. The types of sensors used range from Hall Effect distance sensors to multi-axis accelerometers.

Passive Control

Up until 26 years ago, passive knees were the only type of prosthetic knees available. They are still heavily used as a reliable cost-effective alternative to microprocessor-controlled systems. The two distinctions between active and passive systems are that in passive systems the knee swing dampening is not dynamically adjusted, and the support phase of the knee occurs in a narrow

window of knee flexion angles. Knee flexion and extension are controlled through one of several methods of applying torque to the system, discussed in the Control Methods of Aim 1. Passive systems rely on preset torque control methods that may often be adjusted manually. However, the changes require gait interruption, which means the system can only be adjusted to one ideal speed. At that speed, the system functions most naturally and would be comparable to the swing control of an active system (Johansson et al., 2005). The diversity among activities, patients, and knee systems means that this ideal speed is not usually employed for all activities the wearer will experience.

With no sensing component, passive systems offer support in a limited range. The support system depends on the geometry of the knee. For example, the passive, four-bar LIMBS M3 knee has functional support from full extension to six degrees of flexion (Galey, 2016). At that point, the mechanical stability of the joint will no longer support the patient. This drawback often leads patients to adapt a gait with increased hip rotation and strong prosthetic heel impact to ensure that the knee system is fully extended. Such movement can lead to secondary injury throughout the hip and healthy limb (Gailey et al., 2008).

Active Control: Microprocessor Control

On the other end of the spectrum, active control represents a knee system that is capable of dynamically adjusting flexion and extension dampening parameters to adjust to the gait environment of the patient. This translates into stumble control, fall prevention, and dynamic gait capabilities (Goldfarb, 2013). Knee sensors detecting a rapid collapse or simple application of weight can allow the system to apply the appropriate preventative dampening torque. Current programming of active knees uses discrete states of gait that are altered by predetermined combinations of sensory inputs (Torrealba & Fonseca-Rojas, 2019; Wen et al., 2017). In addition

to the stability of patient gait, a reactive system allows for the recognition of gait speed. Dynamically adjusting to the preferred gait speed of the patient allows the system to behave more naturally and reduces secondary stresses on the patient. It should be noted that the only commercial knee that currently adds mechanical energy to gait is the Power Knee© by Össur. Though powered knees are currently a topic of research interest (Kalanovic et al., 2000; Kapti & Yucenur, 2006; Martinez-Villalpando & Herr, 2009; Sup, Bohara, et al., 2008), other active knees simply adjust the dampening applied to the system during flexion and extension. Overall, the use of MPKs has ensured reduced metabolic rate, up to 5% (Johansson et al., 2005), greater stability (Kahle et al., 2008), and dynamic gait adaption (K. R. Kaufman et al., 2007).

Control Overview

Control systems are used to manage behavior in electro-mechanical devices. A common example of a feedback control system is shown in Figure 5.1, below. Such systems compare the feedback elements to the input to adjust the output. These systems can function quite effectively if the variances and bounds of the variances are understood. However, effectively switching between control models is more challenging and is the subject of much research and development (Narendra & Balakrishnan, 1997).

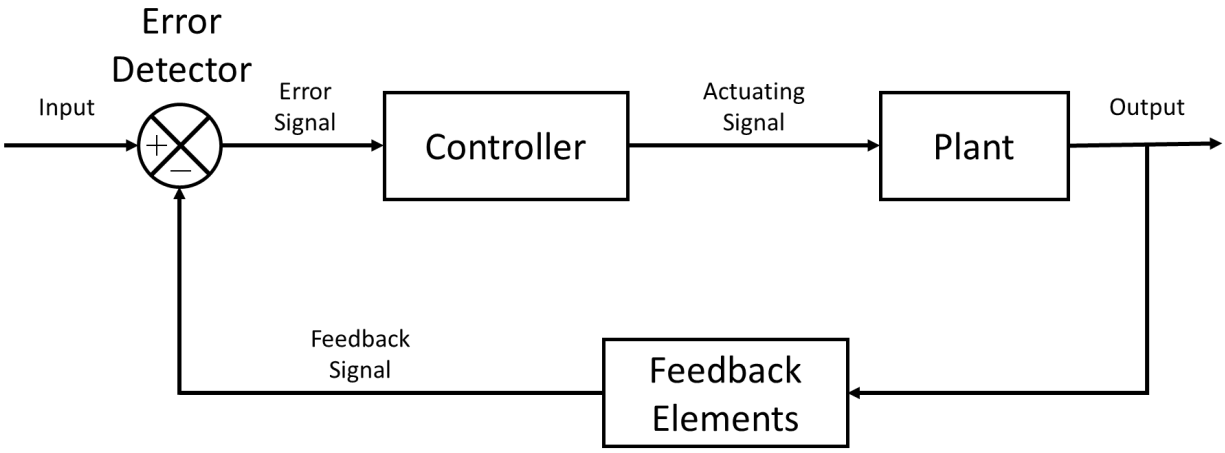

Figure 5.1: Example of a negative feedback closed loop control system. The controller is the model that governs the system. The plant is the component that is being controlled. Feedback elements in applied systems are often from sensors that monitor specific variables.

When only one gait mode is considered, the literature shows that adapting to gait patterns using simple adaptive control systems in powered prosthetic knees has been very successful (Kapti & Yucenur, 2006; Quintero et al., 2018). In fact, the literature shows that such systems often resort to more complicated measurements and controls due to the type of feedback received from sensors and the control variables available. Potential exists in applying the methods of machine learning models to the principles laid out by Narandra et al..

Components

As introduced thoroughly in Aim 1, the electrical system of the GKnee consists of the Google Coral, two gyroscope and accelerometer IMUs, a servo, and a few other assorted components. In terms of the control system, the Google Coral is the controller with the servo functioning as the plant. As the system goes through gait, the sensors function as the feedback elements. In a sense, the Google Coral adjusts its output and actions based on the interpretation of the feedback elements.

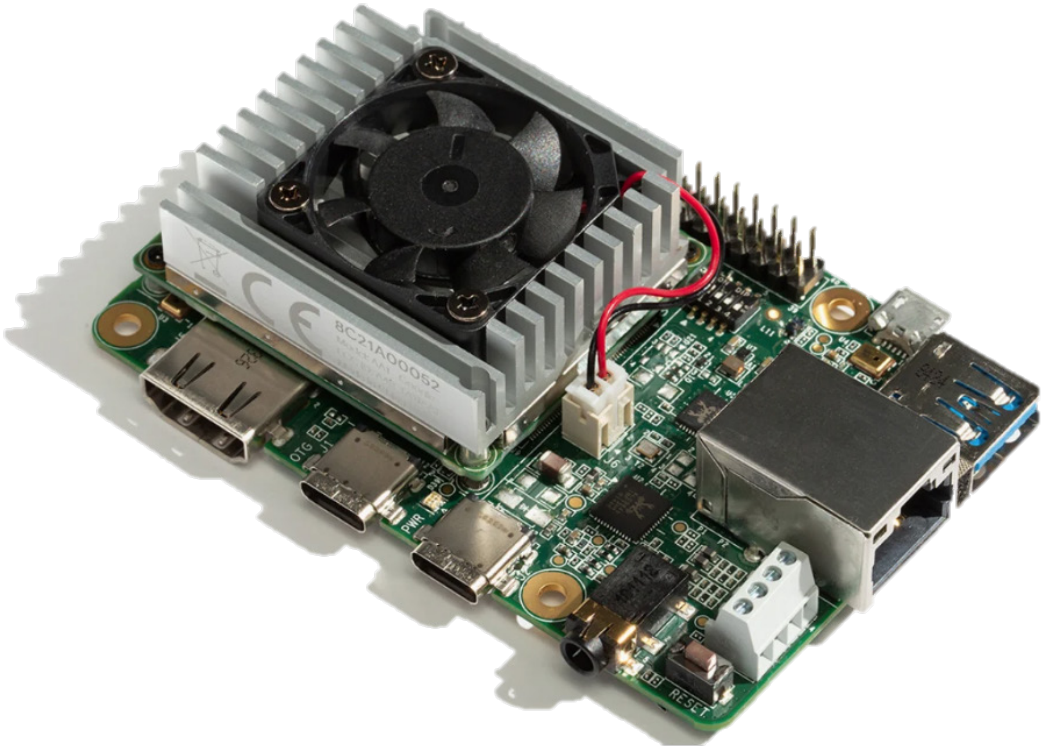

Figure 5.2: Google Coral used for Machine Learning model deployment. Image from coral.ai/products/dev-board.

5.3 METHODS

As the studies by Narandra et al. suggest, the difficulty in controlling one system within different environments lies in the control models not having the ability to adapt to multiple situations. Therefore, as Narandra proposes, this research uses a switching mechanism in the GKnee. Aim 2 developed an ML classifier that detects current gait activity. With this classifier as an environmental differentiator, the control models were simple constrained systems.

It was intended that the controller use a PID model for each of the activities. However, with the restraint of the Aim 2 to focus on walking and stumbling, there was no sophisticated ongoing tuning required. The actual controlled value of the knee system is the degree of valve opening of the damper. If the control system allowed maximum knee angular velocity, then the

damper valve was fully open. If the controller limited knee angular velocity like in the case of a stumble, then the damper valve was closed.

For each of the two activity categories, walk and stumble, a reactionary system was established. During stumble, the profile focused on support and therefore a minimization of knee angular velocity. On the other hand, the profile for walking focused on allowing knee freedom of motion—unrestrained angular velocity. The Aim 2 environmental differentiator was adapted to switch between the control models.

5.3.1 Control Algorithm

From a high-level view, the control system of the GKnee receives sensor data, uses the ML algorithm to determine gait mode, and then reacts to the gait mode by adjusting the valve of the hydraulic system. The main loop of the program operates at 60 Hz. However, a secondary, background thread is continually running and dedicated to collecting data and interpreting with the ML algorithm. Individually, each task could delay the program too long for one cycle to complete quickly enough before the next cycle begins. Therefore, the external loop cycles quickly and waits for timestamps that indicate another cycle is occurring. Each cycle is broken into four components: data collection, preparation, interpretation, and reaction. The following sections will elaborate on the methods used to accomplish those functions more specifically. Detailed code can be found in Appendix 9.1.

Data Collection

The only function of the first section of the control cycle is to initiate the data collection from the sensors. It was found that different sensors reacted at different rates, which seemed correlated with the ability of their software libraries to access the native peripherals of the Google Coral device. While some sensors, namely the LSM9DS1, were able to function in a sequential

loop at close to 120 Hz, others, such as the LSM6DS3, were unable to consistently perform beyond 80 Hz in a sequential loop. Additionally, even at 60 Hz, errors and cycle skips due to processing delays were common. However, implementing multithreading allowed the sensors to collect data while other processing steps occurred, which mitigated the errors and cycle skips at 60 Hz.

Data Preparation

The data from the sensors were processed in a variety of ways to be used throughout the control methods. First, raw data from the sensors were transformed by the same scaler that had been fit to the training data previously. This transformed data was saved in a rolling history (0.33 seconds long) for predicting. Additional rolling histories were kept for angular velocity (20 seconds), flatness of angular velocity (15 seconds), and angle (15 seconds).

Angular velocity was calculated using the sagittal plane gyroscope sensors. Since the velocities of these sensors are angular, no additional transformation was needed other than to subtract the velocity of the thigh segment from the shank segment. The resulting angular velocity categorizes the angular velocity of the knee joint.

To determine if the knee was relatively stable and unflexing, the change in velocity of the knee joint was categorized. The "flatness of angular velocity" was calculated using a 0.33 second window. A standard deviation function was used and a 15 second memory was kept.

Early attempts of the angle calculation (θ) depended on the integration of the knee angular velocity, shown below. Though such an integration would be relative, showing change in angle, it was believed that enough ground truth existed in the physical system to account for this because a reference angle could be inferred from full knee extension during gait.

Equation 5.1 $$\theta_{gyr}(t) = \frac{\omega(t) + \omega(t-1)}{2}\Delta t + \theta_{gyr}(t-1)$$

Though the angle would drift over time (seen in Figure 5.3), depending on the biases of the gyroscope sensors, the fully extended knee could be used to correct the angle to zero. It was found, however, that this depended on the magnitude of bias in each gyroscope sensor. In an attempt to detrend the signal drift, the approximate bias of the sensors was found and continuously subtracted from the input. Testing of this method on the preliminary data was successful, but experimentally, ineffective during the final trials, most likely due to sensor differences. An additional correcting bias was introduced to return the angle to zero periodically, which occurred naturally during gait, but ultimately this method was also inconsistent.

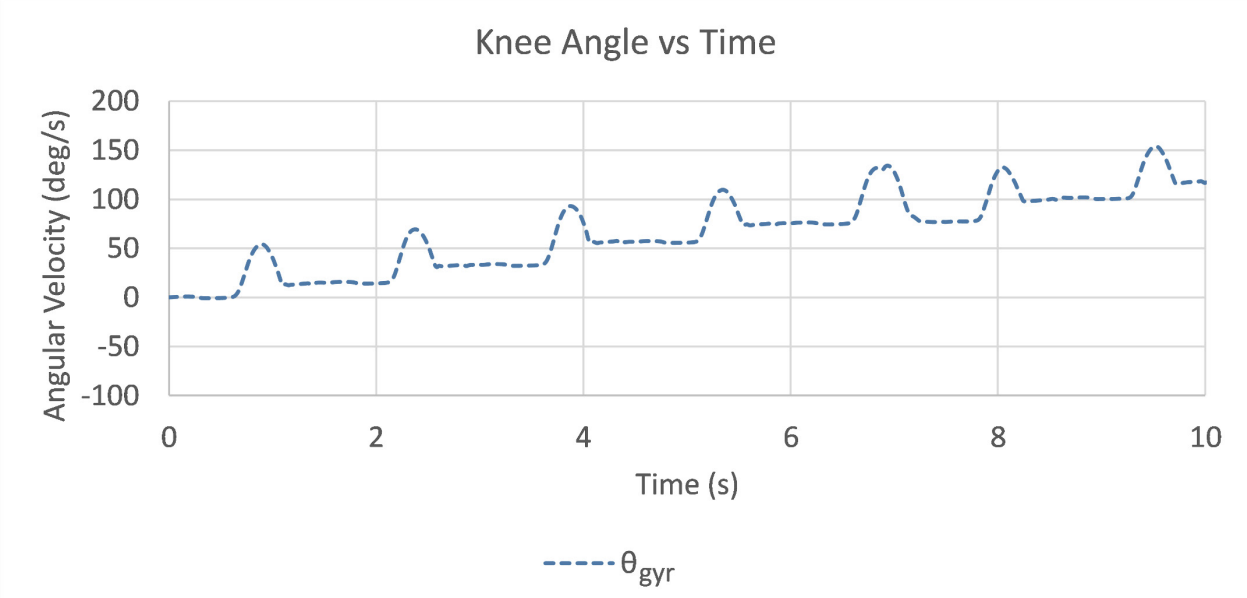

Figure 5.3: Example of calculated knee angle drift during trials. Angle (θ_{gyr}) calculated using gyroscope integration.

Angle can be calculated from accelerometers because of the constant pull of gravity. However, these calculations work best in a static environment because accelerometers are highly sensitive to vibrations. The angle formulas for each segment are given below where "α" represents acceleration in the x, y, and z planes, Equation 5.2 and Equation 5.3.

Equation 5.2 $$\theta_x = tan^{-1}\left(\frac{a_x}{\sqrt{{a_y}^2 + {a_z}^2}}\right)$$

Equation 5.3 $$\theta_y = tan^{-1}\left(\frac{a_y}{\sqrt{{a_x}^2 + {a_z}^2}}\right)$$

As shown in the formulas, the angles of each segment are calculated by accelerometers on different planes; therefore, the knee angle was first calculated separately on the x and y planes by taking the difference of each segment's angle. Lastly, the two planar knee angles were averaged. Theoretically, these angles should have been identical, however, in practice, the sensor limitations (according to position relative to gravity) warranted an averaging to reduce error. As can be seen in Figure 5.4, these calculated angles (θ_{acc}) were prone to error during movement (sporadic spikes and noise in the signal).

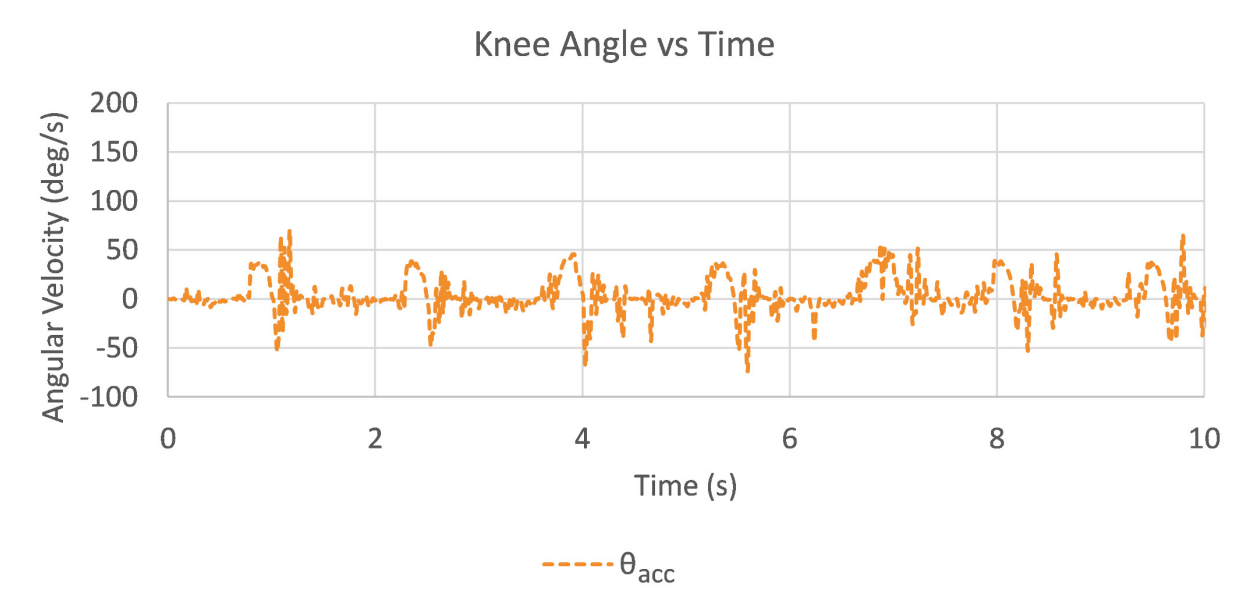

Figure 5.4: Example of acceleration calculated knee angle (θ_{acc}).

Fortunately, as demonstrated by Seel et al., knee angle can also be calculated by combining angle calculations from the gyroscopes and accelerometers with a Kalman filter (Seel et al., 2014).

An example of the Kalman filter is given below in Equation 5.4 where lambda (λ) represents the weight distribution of the angle source.

Equation 5.4 $$\theta_{acc+gyr}(t) = \lambda\theta_{acc}(t) + (1-\lambda)(\theta_{acc+gyr}(t-1) + \theta_{gyr}(t) - \theta_{gyr}(t-1)$$

In our use, λ was set to 0.025, which is somewhat higher than the example value by Seel et al. but proved to be more accurate with the gyroscope biases in this device. The result of the combination of angles from angular velocity and acceleration are shown below in Figure 5.5.

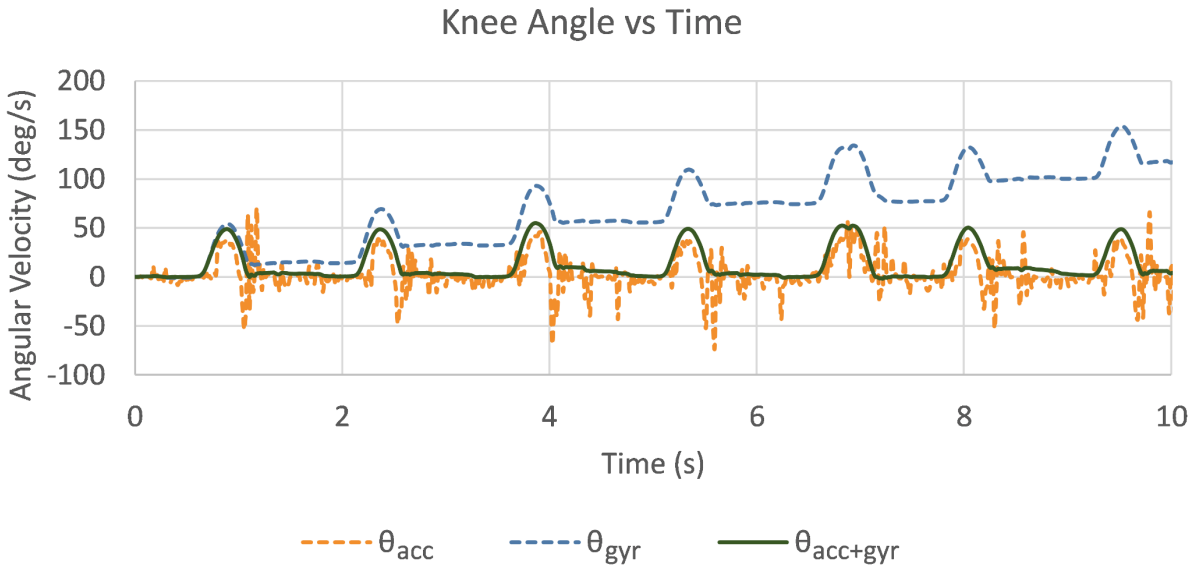

Figure 5.5: Example of combination of calculated knee angles. θ_{acc} is calculated from acceleration, θ_{gyr} is calculated from angular velocity, and $\theta_{acc+gyr}$ is calculated by combining both using the Kalman filter.

Data Interpretation

During the data interpretation step of the device-control process, the background thread is triggered to use the previously scalar-transformed sensor data in a ML prediction by the interpreter. The interpreter is given the current transformed sensor data and 10 historic data steps, skipping one step (0.0167 seconds) between each. The prediction is stored in a variable for later use.

Data Reaction

To address problems with false positives and improper stumble flagging, the data reaction step is broken into several components. First, it is determined whether the ML algorithm is predicting a stumble. Then, environmental factors, such as knee angle, are considered to determine if knee locking is necessary. Finally, the knee velocity and open position of the hydraulic dampener are evaluated for the base level of knee dampening.

The first portion of the reactionary period is observing the predictions from the previous data interpretation step. A stumble prediction (positive class) is made if the stumble probability score is greater than or equal to 0.8 and if more than three of the last six predictions were positive (stumbles) as well. A stumble prediction stays active until six predictions have been negative (non-stumble class). Once the stumble prediction has been obtained, the system continually evaluates the physical state of the prosthesis. If during that evaluation the knee angle exceeds 10 degrees and the knee is no longer extending rapidly (>-200 deg/s), then knee-locking is triggered. Locking sets the damper valve to its closed position which prevents further flexion but not extension.

The knee unlocks if the following three conditions are met: there is no active stumble prediction, the knee velocity standard deviation (STD) is less than 10 degrees per second for the last 0.167 seconds, and the knee angle is less than 15 degrees.

Similar to Herr et al., the system is designed with a maximum acceptable knee flexion during gait (Herr & Wilkenfeld, 2003b). While their system had an upper limit of 70 degrees, a review of the initial data and literature showed that a 60-degree limit for prosthetic knees appeared more realistic (Chauhan & Bhaduri, 2011; Ochoa-Diaz et al., 2014; Torrealba & Fonseca-Rojas, 2019). Therefore, if the historic max flexion angle was less than 50 degrees, the valve open position would be decreased by four degrees, which allowed more flow and decreased dampening. Conversely, if the max flexion angle exceeded 60 degrees, the servo would increase by four

degrees, which restricted flow and increased dampening. A maximum flexion angle between 50 and 60 degrees was the desired outcome during gait.

Supplemental Program Features

Though data collection and reaction occurred at 60 Hz, the program was designed to operate up to 120 Hz. The program was designed to maintain accurate time counts by establishing future timestamps and only initiating the appropriate functions during the corresponding periods. This also allows the program to account for delays in processing and record when delay errors occur. Additionally, the program received inputs from the testing button Input/Output pin to record stumble markers manually as described in the Recorded Data section of Aim 2 and the Electronic System section of 1.

The output of the program is a file containing the following features: timestamp, accelerometer and gyroscope data, manual stumble marker, any error messages, knee angle, stumble prediction, and program reactions, such as when the system locked or confirmed a stumble prediction. The control algorithm was designed to operate in real-time with all predictions, calculations, reactions, and recordings happening on the device's Coral microprocessor.

5.3.2 Aim Evaluation

To prevent rapid model switching (between knee locking and unlocking) for possible false positives (FP) or false negatives (FN), a small buffer of six samples, over 0.10 seconds, was applied. The buffer requires that classifications be positive for multiple, consecutive iterations before the control models switch, but no longer than 0.15 seconds. This was done to mitigate the potentially disastrous effects of arresting knee flexion during FP would cause to natural gait. There was an expectation that the control methods would affect the activity classification metrics (recall, precision, f-score, and accuracy). Because the control system of this aim was developed with, built

on, and transformed the classification of Aim 2, the assessment of this control system was done on activity classification during the deployed gait trials of Aim 4. As such, detailed procedures can be found in the methods section of Aim 4. The evaluation of the current aim was performed by categorizing the reaction times of the control system, and by evaluating the final activity classification using recall, precision, f-score, and accuracy in confusion matrices.

Reaction Times

To assess the achievement of this aim, different sets of reaction times needed to be calculated – switching between true states and switching out of false states. The transition from walking to stumble was clearly marked in the trials, albeit with some human errors. As described in the control system above, however, the transition from stumbling to walking depends on the system recognizing gait stability first. Therefore, the true state-switching-reaction-time will be the time taken to switch from walking to stumbling states as predicted by the device. Likewise, if the prediction switched from stumbling to walking incorrectly, such a prediction would be subsumed by the control system awaiting gait stability. Thus, the reaction time in switching from a false state must be calculated from how quickly incorrect stumbles are reverted to the correct walking state. It should be noted that in all calculations, the stumbling and walking classifications should be understood to be when the control system activated or deactivated the locking mechanism, respectively.

Switches between true walking and stumbling were calculated using the timestamps of the manually marked stumbles along with the timestamps of the system beginning locking. Marking code seen in Appendix 9.3. As an additional point of clarification, the timestamps of the prediction system classifying a stumble were calculated as well. Every marked stumble was manually

examined to confirm whether a stumble or locking prediction truly occurred near it. The reaction times were calculated as the differences between the predicted and true timestamps.

The reaction time of the control system when switching from an incorrect locking was calculated by averaging the durations of every locking instance that was classified as a FP. This meant that the locking did not align with a marked stumble and was manually marked as such.

5.4 RESULTS

Shown in the sections below are the results of the implemented control algorithm: visualized processing steps of the algorithm, reaction times when switching between states, and the effect on the prediction outcomes.

5.4.1 Processing Steps

While the data collection and ML predictions are difficult to visualize, the processing steps of the control algorithm outlined in the methods were plotted sequentially in Figure 5.6, Figure 5.7, and Figure 5.8.

Figure 5.6, below, highlights the data collection and preparation steps presented in the methods. The incoming sensor information was used to calculate the knee angular velocity and knee angle. This data was tracked and used in future reactions. Periods of flatness, such as between gait cycles, were tracked in the STD history for locking release.

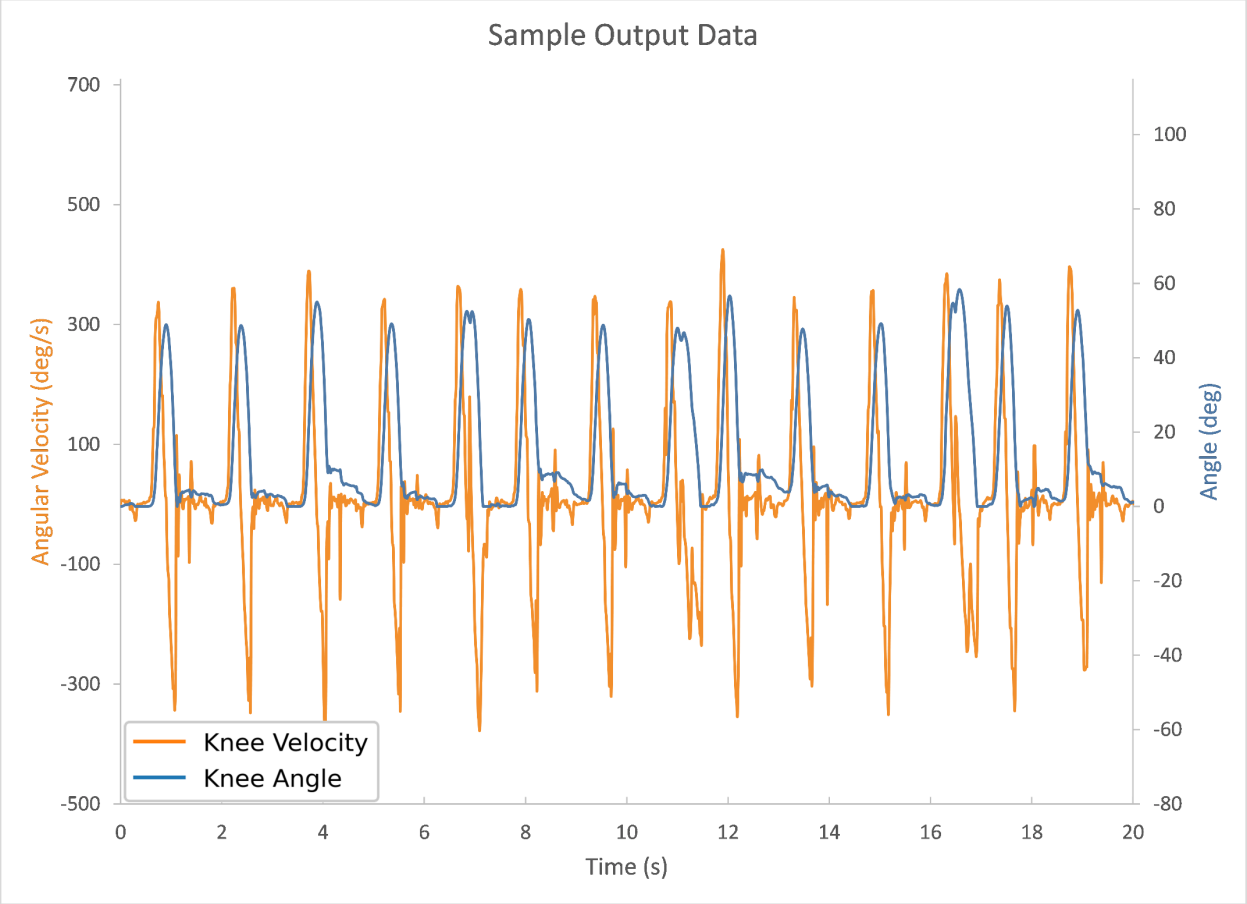

Figure 5.6: Knee Velocity and Knee Angle as calculated by the control algorithm and output during data collection. Sample from P1 during Obstacle stumbles.

As a part of the patient trials, stumbles were induced, and each induction was marked by the researchers pressing a button tethered to the GKnee. An example of this can be seen in Figure 5.7, below. It should be noted that the rest of the control system was unaware of this trigger information. The stumble indications were saved to the files but were not used in any part of the control process. Below, stumble inductions can be seen as the dotted lines labeled “Actual.”

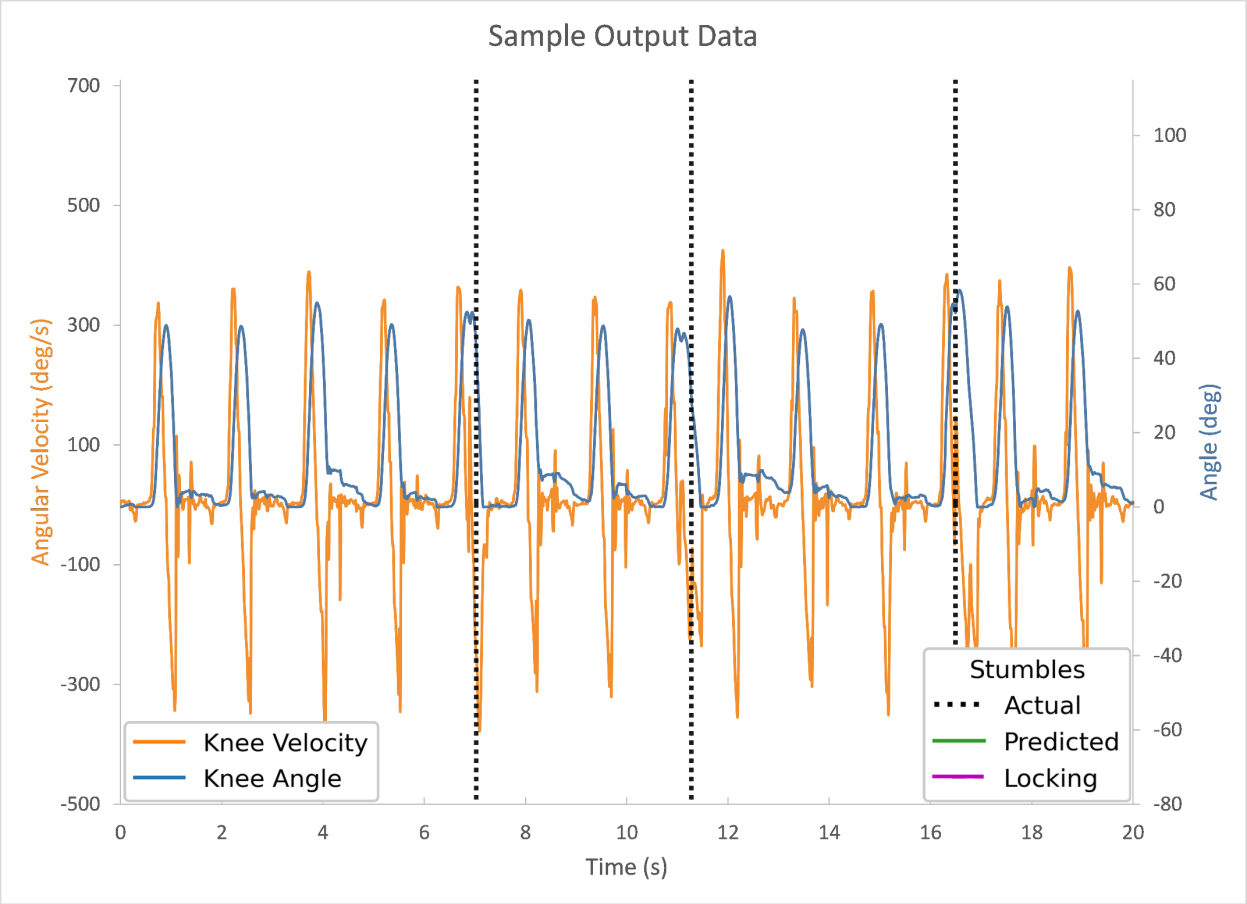

Figure 5.7: Knee Velocity and Knee Angle as calculated by the control algorithm, and Actual Stumbles as marked during data collection by researchers. Sample from P1 during Obstacle stumbles.

Lastly, Figure 5.8, highlights both the interpretation and prediction steps of the control algorithm. The green line, "Predicted," demonstrates the output of the ML algorithm interpreter. The purple "Locking" line shows the instances where the control algorithm reacted to the prediction conditions. As can be seen below, predictions were not always responded to immediately, and in some situations the marked stumbles were not predicted or responded to at all (as in the first stumble induction for example).

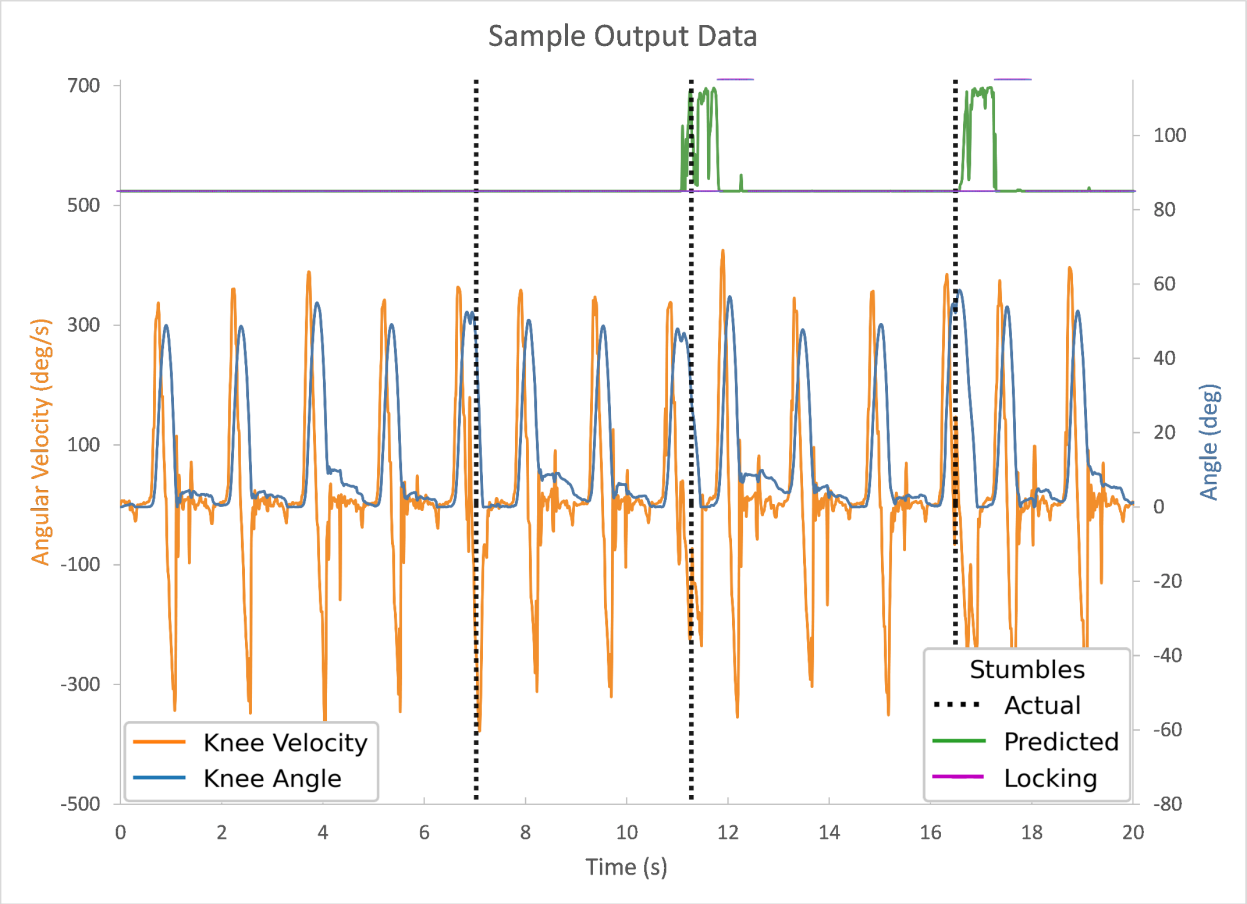

Figure 5.8: Knee Velocity and Knee Angle as calculated by the control algorithm, Actual Stumbles as marked during data collection by researchers, Predicted Stumbles as predicted by the ML algorithm, and Locking Stumbles as when the control algorithm determined reactions to stumbles. Sample from P1 during Obstacle stumbles.

5.4.2 Reaction Times

The following two tables show the reaction time results for the system. Table 5.1 shows the difference in time between the marked stumbles and the control system recognizing stumble or locking across all patient data. Table 5.2 shows the system reaction time in correcting after a false positive during which the system locked and then turned back to the true class of walking.

Table 5.1: Control system reaction times for recognizing stumble and locking events. The right column shows the difference between these two numbers. The bottom row shows the respective standard deviation (STD). Data shown is the average between all Patients 1, 2, 3, 4, and 6 for all reactions of that type.

	Stumble Reaction	Locking Reaction	Difference
Average	0.147 s	0.158 s	0.013 s
STD	0.241 s	0.255 s	0.059 s

Table 5.2: Control system reaction time in correcting been a false predicted stumble, during which the control system locked, back to the true class of walking. The average is the time in seconds of how long the system locked. The bottom cell shows the standard deviation (STD). Data shown is the average between all Patients 1, 2, 3, 4, and 6.

	Recovery Reaction
Average	0.034 s
STD	0.385 s

5.4.3 Stumble and Walking Positive and Negative Rates

Similar to the confusion matrices presented for the ML network in Aim 2, the summary of the effectiveness of the control algorithm is given below in a set of confusion matrices. The right column shows the metrics measured: recall, precision, FPR, F-Score, and accuracy. As discussed in greater detail in the methods of Aim 2, the counted data are the individual categorized steps. The steps were obtained during the Aim 4 stumble induction trials while patients were wearing the GKnee, (see Aim 4 for more details). The following tables display the confusion matrix by individual patient: Table 5.3, Table 5.4, Table 5.5, Table 5.6, and Table 5.7. On the other hand, Table 5.8 displays the effectiveness of control system as a whole in a summative table.

Individual

Table 5.3 : Confusion matrix of the counted step data gathered during the Aim 4 trials. Data shown is for P1 during the GKnee stumble induction data runs. Right column is recall, precision, false positive rate (FPR), F-Score, and accuracy.

		Predicted			
		Stumble	**Walking**	**Recall**	92.3%
				Precision	100.0%
Actual	**Stumble**	36	3	**FPR**	0.0%
	Walking	0	161	**F-Score**	96.0%
				Accuracy	98.5%

Table 5.4: Confusion matrix of the counted step data gathered during the Aim 4 trials. Data shown is for P2 during the GKnee stumble induction data runs. Right column is recall, precision, false positive rate (FPR), F-Score, and accuracy.

		Predicted			
		Stumble	**Walking**	**Recall**	69.6%
				Precision	97.0%
Actual	**Stumble**	32	14	**FPR**	0.8%
	Walking	1	130	**F-Score**	81.0%
				Accuracy	91.5%

Table 5.5: Confusion matrix of the counted step data gathered during the Aim 4 trials. Data shown is for P3 during the GKnee stumble induction data runs. Right column is recall, precision, false positive rate (FPR), F-Score, and accuracy.

		Predicted			
		Stumble	**Walking**	**Recall**	81.3%
				Precision	89.7%
Actual	**Stumble**	26	6	**FPR**	2.4%
	Walking	3	121	**F-Score**	85.2%
				Accuracy	94.2%

Table 5.6: Confusion matrix of the counted step data gathered during the Aim 4 trials. Data shown is for P4 during the GKnee stumble induction data runs. Right column is recall, precision, false positive rate (FPR), F-Score, and accuracy.

		Predicted			
		Stumble	**Walking**	**Recall**	76.5%
				Precision	56.5%
Actual	**Stumble**	26	8	**FPR**	19.2%
	Walking	20	84	**F-Score**	65.0%
				Accuracy	79.7%

Table 5.7: Confusion matrix of the counted step data gathered during the Aim 4 trials. Data shown is for P6 during the GKnee stumble induction data runs. Right column is recall, precision, false positive rate (FPR), F-Score, and accuracy.

		Predicted			
		Stumble	**Walking**	**Recall**	69.4%
Actual	**Stumble**	25	11	**Precision**	71.4%
	Walking	10	170	**FPR**	5.6%
				F-Score	70.4%
				Accuracy	90.3%

Summative

Table 5.8: Confusion matrix of the counted step data gathered during the Aim 4 trials. Data shown is summative for P1, P2, P3, P4, and P6 during the GKnee stumble induction data runs. The right column is recall, precision, false positive rate (FPR), F-Score, and accuracy.

		Predicted			
		Stumble	**Walking**	**Recall**	77.5%
Actual	**Stumble**	145	42	**Precision**	81.0%
	Walking	34	666	**FPR**	4.9%
				F-Score	79.2%
				Accuracy	91.4%

5.5 DISCUSSION

The objectives for this aim were to switch between true states in 0.15 seconds and switch out of a false state within 0.30 seconds. As can be seen in the results Table 5.1, the achievement of this goal was moderately successful. A complete switch from walking to stumble locking took 0.158 seconds on average. However, it is important to consider that there are several layers of interpretation between the ML model prediction of stumble before the system physically reacts to the stumble by locking.

The final level of interpretation is an analysis of physical gait characteristics, such as knee extension velocity and current knee angle, before locking the system for a stumble prediction. The stumble reaction time given in Table 5.1 is the time at which the system has clearly recognized a stumble and awaits the physical gait characteristics to not be positive movements towards extension. By this metric, the system achieved the objective with a switch time of 0.15 seconds.

The average switch out of a false state took 0.034 seconds, thus achieving the objective of 0.30 seconds.

While it is clear then, that the system achieved the objectives of this aim, it should also be noted that these reaction times have deviations. In part, this can be attributed to the inaccuracy of stumble marking during the trials. Though good efforts were made, errors in timing must be considered. This is especially true for stumble induction methods such as bungee or uneven because both add uncertainty. The beginning of a bungee stumble was often marked when the researchers began pulling on it; however, at that stage a physical effect would have been impossible to detect visually or in sensor data, and the time to a measurable physical effect would vary with the speed of bungee pulling and current leg swing. In fact, stumble and locking reaction times for many stumble inductions take place before the recorded stumble induction marking; though these were not considered as negative values for the reaction average, but rather as zero-time reactions. A future, more thorough analysis of these reactions would require a manual re-marking of stumble inductions, as performed for Aim 2. Though these results appear to signify success, the more important factor is how the control algorithm affected the final classification results.

Despite using the same model that performed with an average step accuracy of 66.9%, a precision of 33.8%, and a recall of 73.1%, the final classification of the control system showed a step accuracy of 91.4%, a recall of 77.5%, and a precision of 81.0%. One difference between these models is that the deployed model of Aim 2 was trained with three subjects in a LOO, and the final implemented model of Aim 4 used a LOO with the three subjects as training and each subject from the Aim 4 trial as the test. The only other difference between the models is that the control system interpreted the predictions according to the physical knee environment and thus switched between

walking and stumbling with much greater reliability. It should be noted that with the addition of the control system, the objectives of Aim 2 are achieved.

Limitations

Though the control system accomplished the objectives established for it, there remain limitations in its application and evaluation. For example, the control system was designed to switch between just two different gait states, stumbling and walking. While this was constrained for the scope of this research, inclusion of further states, such as sitting or standing, would be beneficial to a multipurpose knee system. A primary overarching objective of the research was to diverge from rule-based control systems. While the ML classifier is a large step away from rules, the use of the knee environmental data still constrained the system to a small set of inputs, albeit inputs with a much-decreased room for error.

While statistical significance was achieved in Aim 4 with the number of subjects, the evaluation of the control system had no statistical analysis. Additionally, the three training subjects included one female, but the evaluation occurred on five males. There are no reasons why the results would not be applicable to a larger and more diverse patient population, but the current evaluation is limited in its evaluations on this population.

Lastly, one of the underlying assumptions of this work has been that a knee reaction or locking time in less than 200 milliseconds would coincide with established human reaction times, and thus be fast enough to enable a subject to use the knee for stumble recovery. However, as the deviation of the stumble and locking reaction times suggests, there are still large variances despite having an average stumble reaction time of less than 150 milliseconds. Coupled with the previously discussed potential error in stumble marking, this may indicate that though the objective

for this aim was achieved, the 200 milliseconds of standard human reaction may be exceeded regularly.

Future Work

Continued development of this research would ideally attempt to address many of the limitations. As mentioned, watching videos and manual marking stumbles would shed much greater light on the exact timings of both stumble and reaction. Further, with the expansion of the training pool for the Aim 2 classifier, the control system should both be re-evaluated in its accuracies and metrics with a larger pool, but also perhaps reduce the number of environmental tunings required to produce accurate labels. Additionally, the knee environment factors could be included into the system as an additional ML network, or input to the system as calculated features. Ideally, such a change would move the control system to a purely ML classifier-based system for all gait states.

Lastly, though implemented in code, the flexion angle dampening mechanism was never formally deployed nor tested on subjects walking at various gait speeds. Further research would include a set of gait trails that greatly varied walking speed and compared the various gait parameters with the active angle dampening system enabled.

Broader Implications

Human stumbles are a common occurrence, and while healthy gait recovery strategies and situational comparisons have been conducted, the literature is bereft of a comprehensive study that not only has many stumbles conducted, but also has conducted these stumbles across a wider range of approaches. While a couple of studies compare two different modes of stumble induction, this study has included three separate modes. Half of the studies in the literature have compared more than one knee system, but this study has compared three: a mechanical passive knee, an MPK, and

a prototype MPK that was tested both mechanically (passive) and electronically (active). Apart from Shawen et al. and Shirota et al., no studies conducted any significant number of stumbles per subject (Shawen et al., 2017; Shirota et al., 2015). This study inducted 100 stumbles per subject. The data presented alone in this study has a combination of knee systems, number of stumbles, and variance of stumble modes that has never been published before. This data will be useful for follow-up studies and allow the scientific community to better understand stumbles, especially in amputee patients, so that future injury and discomfort can be minimized.

Chapter 6: Aim 4 - Compare assembled system of Aims 1, 2, and 3 to industry standards and base prototype M3.

6.1 OBJECTIVE

It is hypothesized that the system will have a smaller rate of falls per stumble event than the M3 and will have no statistical difference in rate of falls per stumble compared to the subject's own MPK. It is also hypothesized that the system will exhibit increased gait symmetry compared to the M3 and decreased gait symmetry compared with the subject's MPK. This aim seeks to evaluate the GKnee when compared to current passive and active knee systems in the categories of falling prevalence and walking mechanics. Ultimately, the device's performance in this aim will be a summative performance of the overall research project.

6.2 INTRODUCTION

As described in the introduction of Chapter 4, many different sensors and measurements have been used to compare and evaluate gait characteristics in the literature. Stumble and fall kinematics in healthy gait are documented, but within amputee gait few studies have been conducted. Studies that observe the effects of rapid acceleration or deceleration of a treadmill during gait have successfully induced stumbles (Sessoms et al., 2013; Zhang et al., 2011). Additionally, stumble modes from specific gait activities have been evaluated in direct comparisons between different prosthetic knees (Blumentritt et al., 2009b). However, studies have not thoroughly evaluated stumble-events in prosthetic knee systems with the rigor that healthy stumbles have been addressed.

6.3 METHODS

This aim evaluated the final assembly of the previous aims and the GKnee's effectiveness as a whole system when compared to industry-available prosthetic knee systems. Further, the

primary objectives of this final aim were to measure rate of falls and biomechanical parameters between the GKnee, LIMBS M3, and the subjects' current MPKs. Therefore, the basic methodology of the walking trials described in Chapter 4 were repeated with the previous notable differences focusing on overall evaluation.

This trial took place at The University of Texas Southwestern Medical Center (UTSouthwestern) in the Physical Therapy Gym located on the 1st Floor of the School of Health Professions Building. The prosthetic fitting and adjustments were performed by UTSouthwestern's Tiffany Graham, CPO, and Jan Karel Petric, CPO. Subject qualifying criteria were as follows: age between 21 and 60, unilateral transfemoral amputation, no secondary health conditions or neurological disorders (specifically things that affect gait normalcy), activity level of at least K3, and general good health. Additionally, the subject's everyday prosthetic knee was to be an MPK.

The biomechanical parameters were compared through a hybrid system of knee sensor data and video recordings. The subjects completed intervals of walking and of stumble induction in a safety harness as described in Aim 2. Also, as in Aim 2, stumbles were induced via three modes: obstacle, bungee, and uneven surface. In contrast to Aim 2, gait parameters were compared, and the methods were repeated for the three leg systems.

Because the order in which the knees could be tested was limited (M3 before GKnee, discussed below in GKnee Methods), and to reduce the amount of time by limiting knee fittings, subjects used the knees in two different orders, which were first randomized, and then alternated. The subjects either used their MPK, then the M3, and then the GKnee; or they used the M3, then the GKnee, and then the MPK. Subjects used their MPK with their regular foot and used the GKnee and M3 with the Niagara Foot® v1. Though some gait parameters can be affected by different feet,

it was deemed not significant enough to not change their foot alignment or comfort for standardization of the feet across the trials.

6.3.1 Data Collection

The data collected was: three-axis linear acceleration and three-axis rotational velocity for both shank and thigh of the subject's prosthetic leg; videos of stumble inductions for classification of stumble type; temporal symmetry measurements of gait; and stumble activities. The M3 and the GKnee already had similar sensor configurations, and the MPK of each subject was also retrofitted with the same two sensors to extract knee angle information.

As established by Orendurff et al. and Segal et al., biomechanical parameters are only validly comparable during similar gait speeds (Orendurff et al., 2006; Segal et al., 2006). Therefore, subjects had a time for self-selected walking speed, within the range of 0.5 – 1.0 m/s but were set at 0.8 m/s for the stumble trials. The rest of the experimental procedure and protocol remained the same between all three devices with induced stumbling conditions, but now repeated while wearing different prosthetic devices. Thus, the flow of the trials for each knee was as follows: fitting by a prosthetist, acclimation period, static sample, walking at self-selected speeds, induced stumbles by bungee, induced stumbles by obstacle, and induced stumbles by uneven all at 0.8 m/s. This is outlined in Figure 6.1, below.

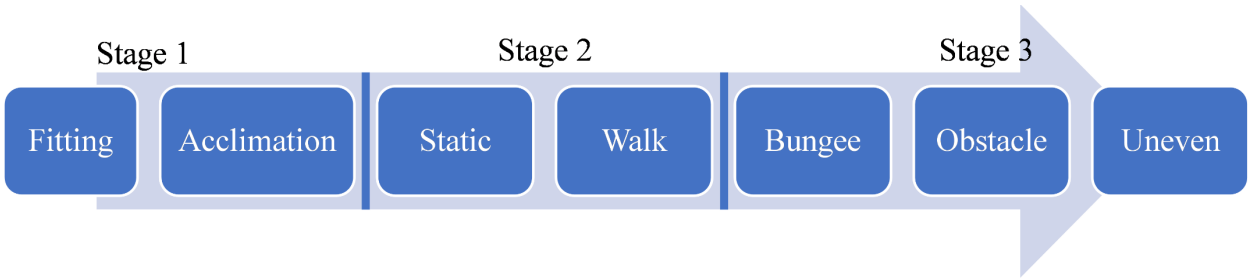

Figure 6.1: The order of each knee trial. Fitting and acclimation involved the prosthetist and did not include data recording. Static and walk were performed on the treadmill with the sensors recording for 1 and 3 minutes, respectively. Bungee, obstacle, and uneven were the different methods by which stumbles were induced. These were conducted until 10 of each type had been collected.

In total, each subject walked on three different prosthetic devices, was subjected to three different types of stumble inductions per knee and was induced approximately 10 times per type of stumble. This accounts for approximately 90 stumble inductions per subject.

GKnee Methods

As established in Aim 2, the addition of one minute of subject-specific gait data greatly improved the classification capabilities of the ML system. Therefore, subjects completed the knee trials on the M3 before the GKnee, and walking data from the M3 was used to train a subject-specific ML model for the GKnee trial. Details of this code can be found in Appendix 9.5.

While each subject was being fit for the GKnee, the trial data files were pulled from the M3 data collection system. The one-minute sample of walking data was pulled from the walking file and transformed into a usable input dataset for the ML model. The subject-specific data was then combined with all three subjects' data from the initial data collection and a scaler was fit to/transformed the data, as described in Models in the methods of Aim 2. The developed ML model discussed in Aim 2 was then retrained on this expanded data set, and the patient-specific model was converted and saved by the TensorFlow TFLiteConverter. The outputs of this process were a

subject-specific ML model and fit scaler. These were then loaded onto the GKnee with the control system file and used during that patient's GKnee trial.

GKnee Control Evaluation Recollection

During the trials, it became clear that angle calculation was not performing correctly. As discussed in Aim 3, this was addressed by the incorporation of accelerometer data to account for the drift of the gyroscope data. However, a solution was needed mid-trial when the performance issue was identified. Therefore, during the trials, the locking mechanism was disabled which allowed the trials to proceed with software-level data collection and classification, but without physical interaction with the knee system.

Therefore, a partial recollection of data was conducted within the same population. P2, P3, and P4 consented to the abbreviated retrial. The general methods of the trial remained the same, but the subjects walked and stumbled on the GKnee under two conditions: control system on, and control system off. Essentially, it tested whether the interaction of the servo would affect recovery or fall incidences. During this recollection, the M3 and MPK were not evaluated. The same statistical methods were employed for the patients during the on/off conditions.

6.3.2 Data Processing

Stumbles and falls were an incidence metric compared between the knee systems. Gait step parameters and knee angles were collected by analyzing video footage and sensor calculations (as described in Chapter 5). The resulting data was analyzed by the statistical repeated measures ANOVA method to determine significant variances between knee systems and subjects. Safety of the systems was comparatively assessed by the stumble and fall incidences. Gait symmetry was assessed by the gait parameters. Final comparisons between the three systems were made using these measurements and outcomes.

Stumble Classification

During the initial data collection for Aim 2, the gait classifications were either walking or stumbling, with manually marked differentiations between the three modes of stumbling (bungee, obstacle, and uneven surface). However, to evaluate the differences between knee systems it became apparent that additional stumble classifications were necessary. Originally, it was thought that a subclassification of stumbling into "falling" or "recovery" would allow the falling incidences to be evaluated between the different knee systems. While this was true, it assumed that each stumble induction was enough to trigger a stumble. In the bungee and uneven inductions, it quickly became clear that some inductions may be manually marked by researchers without any or minimal effect on the gait of the subjects.

Therefore, stumble inductions were classified into four categories: unaffected gait (U), affected gait (A), recovery (R), and fall (F). Unaffected gait (U) occurred when the stumble induction had no visible effect on the subject's gait. This occurred when the induction was too close to the heel strike or was done too lightly. The induction was classified as affected gait (A) when there were visible signs of the inductions' interference with the patient's gait, but no gait adjustment was necessary to overcome them. Examples of this included subjects stepping on the object during the uneven surface induction, or the bungee being pulled too lightly or too late in the gait cycle. These actions may have shown a hitch in gait, but the subjects were able to continue walking unperturbed. Therefore, U and A were considered ineffective inductions. Recovery (R) and fall (F) were considered effective inductions because in both instances the gait of the amputee was perturbed and recovery was attempted, successfully in R and unsuccessfully in F.

Though the specific stumble subclassification was originally recorded during the trial by the researcher who was also pressing the button to mark the stumble induction, this proved to split the researcher's attention too much to provide accurate information in both tasks (marking the

subtype and pressing the button). Therefore, when the U, A, R, and F subclassification system was applied, all previously classified stumble inductions were revised manually by watching back the video footage and recording the subclass using timestamps. This also created a manual data-verification process to ensure the classifications were marked uniformly across patients and knees. Additionally, every step and locking events were also manually reviewed and classified. Details of this code can be found in Appendices 9.3 and 9.4.

Statistical Analysis

Two statistical methods were used to quantitatively compare the knee systems: repeated measures ANOVA and Cohen's D. Repeated measures ANOVA was chosen because of the trial's design; the pool of participants remained constant, and each participant repeated the same trials as the others. Further, the same measurements were taken during every trial. Therefore, this statistical approach was chosen to establish that any differences found were the result of differences between the knees (M3, GKnee, or MPK) as opposed to differences between the subjects themselves. Cohen's D was chosen to compare the effect size of the differences between devices.

For the purposes of this analysis, the different stumble induction types were summed according to classification per knee per subject. This was done because each stumble induction type per knee per subject generally contained 10 samples. There was not enough difference between these samples to establish significant differences between the stumble induction method. The metric considered paramount was the rate of recoveries versus the amount of effective stumble inductions (R/(R+F)). The data was processed for this metric in Python through the "statsmodels" library. Statistical significance was considered at p less than or equal to 0.05.

Because statistical significance can be established for even small effect with a large enough population, Cohen's D was calculated between each the knees (M3-MPK, MPK-GKnee, and

GKnee-M3) on the same metric (recovery percentage per effective induction). Cohen's D was used to measure the effect size of the differences between the means of the knees. The benchmarks established by Cohen for effect sizes were small ($d = 0.2$), medium ($d = 0.5$), and large ($d = 0.8$) (Cohen, 1988). Though these benchmarks were somewhat arbitrary when established, the effects of the trials were measured accordingly.

6.4 RESULTS

Six subjects were recruited and completed the informed consent according to UTEP IRB Study 1842954-2 and according to UTSouthwestern IRB Study STU-2022-0145. Of these, one subject (P5), could not participate in the trials due to mobility and prosthetic leg abduction during gait, which prevented unsupported gait on a treadmill.

Table 6.1: Deployed knee trials subject demographics and amputation information.

Demographics	**P1**	**P2**	**P3**	**P4**	**P6**
Sex	M	M	M	M	M
Age	30	31	42	50	43
Height (m)	1.75	1.68	1.75	1.88	1.68
Mass (kg)	73	59	82	127	103
Amputation					
Age	12	26	13	3	15
Side	Left	Right	Left	Left	Right
Cause	Trauma	Bone Cancer	Trauma	Vascular Complication	Trauma
Socket	Double Wall (Pin)	Suction	Suction	Socketless Socket	Suction
Knee	C-Leg 4	Rheo XC	C-Leg 4	C-Leg 4	X3
Foot	Proprio	Flexfoot	Low Profile Triton	Triton	Low Profile Triton

The five subjects (all male) who completed the study ranged in age from 30-50 years (average 39 ±8 years), weighed on average 89 ±24 kg, and had an average height of 1.75 ±0.07 m. Table 6.1, above, highlights both the demographic and amputation-specific details of the subject group. The microprocessor knees used in this study are the C-Leg® 4 and X3 by Otto Bock (Duderstadt, Germany), and Rheo Knee® XC by Össur (Reykjavik, Iceland). The feet used were two version of the Triton Foot® by Otto Bock, and the Proprio Foot® and Flexfoot® by Össur.

6.4.2 Stumble Classification

The following tables show the gait trial results for each patient (P) wearing the three knee devices (M3, GKnee, and MPK) categorized by stumble induction mode (bungee, obstacle, and uneven surface) and outcome subclassification (unaffected gait, affected gait, recovery, or fall). The last two rows of each table show the counts per knee and stumble mode that were effective or ineffective at inducing a potential stumble. "Ineffective Induced" is the sum of "Unaffected Gait" and "Affected Gait," while "Effective Induced" is the sum of "Recovery" and "Fall" instances. Recovery rate is the sum of recoveries per knee over the sum of effective inductions per knee.

Individual Tables

Table 6.2: For P1, all counts of stumble inductions separated by various types of stumble induction modes and categorized by knee type. Each knee includes a sum of induction modes, a total sum for the subject is in the right column, and total summative values are shown for ineffective, effective, and total induced stumbles.

Knee	M3				GKnee				MPK				
Mode	bun	obs	une	sum	bun	obs	une	sum	bun	obs	une	sum	**SUM**
Unaffected Gait	0	1	0	1	5	0	0	5	4	4	0	8	14
Affected Gait	3	3	0	6	9	5	2	16	3	4	3	10	32
Recovery	5	4	1	10	2	5	7	14	4	3	8	15	39
Fall	4	4	10	18	0	0	1	1	2	1	1	4	23
Ineffective Induced	3	4	0	7	14	5	2	21	7	8	3	18	46
Effective Induced	9	8	11	28	2	5	8	15	6	4	9	19	62
Total Induced	**12**	**12**	**11**	**35**	**16**	**10**	**10**	**36**	**13**	**12**	**12**	**37**	**108**

For P1, the most effective inductions were seen on the M3 (n=28, 80%), followed by the MPK (n=19, 51%), with the lowest effective inductions occurring on the GKnee (n=15, 42%). P1 fell one time while wearing the GKnee, four times on the MPK, and eighteen times with the M3. Recovery rate was highest on the GKnee at 93%, followed by the MPK (79%) and the M3 (36%).

Table 6.3: For P2, all counts of stumble inductions separated by various types of stumble induction modes and categorized by knee type. Each knee includes a sum of induction modes, and total summative values are shown for ineffective, effective, and total induced stumbles.

Knee	M3				GKnee				MPK				
Mode	bun	obs	une	sum	bun	obs	une	sum	bun	obs	une	sum	**SUM**
Unaffected Gait	2	0	0	2	3	0	2	5	3	0	0	3	10
Affected Gait	7	2	4	13	3	5	5	13	4	4	7	15	41
Recovery	2	5	4	11	7	6	2	15	3	4	3	10	36
Fall	0	3	2	5	1	0	4	5	3	3	1	7	17
Ineffective Induced	9	2	4	15	6	5	7	18	7	4	7	18	51
Effective Induced	2	8	6	16	8	6	6	20	6	7	4	17	53
Total Induced	**11**	**10**	**10**	**31**	**14**	**11**	**13**	**38**	**13**	**11**	**11**	**35**	**104**

For P2, the most effective inductions were seen on the GKnee (n=20, 53%), followed by the M3 (n=16, 52%), with the fewest effective inductions occurring on the MPK (n=17, 48%). P2 fell five times while wearing the M3 and GKnee, and seven times on the MPK. Recovery rate was best for the GKnee (75%) followed by the M3 at 69% and lastly for the MPK (59%).

Table 6.4: For P3, all counts of stumble inductions separated by various types of stumble induction modes and categorized by knee type. Each knee includes a sum of induction modes, and total summative values are shown for ineffective, effective, and total induced stumbles.

Knee	M3				GKnee				MPK				
Mode	bun	obs	une	sum	bun	obs	une	sum	bun	obs	une	sum	**SUM**
Unaffected Gait	4	0	0	4	7	1	0	8	5	2	0	7	19
Affected Gait	4	6	6	16	4	6	5	15	2	4	5	11	42
Recovery	1	5	4	10	2	5	1	8	1	3	3	7	25
Fall	2	0	2	4	0	0	0	0	4	1	0	5	9
Ineffective Induced	8	6	6	20	11	7	5	23	7	6	5	18	61
Effective Induced	3	5	6	14	2	5	1	8	5	4	3	12	34
Total Induced	**11**	**11**	**12**	**34**	**13**	**12**	**6**	**31**	**12**	**10**	**8**	**30**	**95**

For P3, the most effective inductions were seen on the M3 (n=14, 41%), followed by the MPK (n=12, 40%), with the fewest effective inductions occurring on the GKnee (n=8, 26%). P3 fell five times while wearing the MPK, four times on the M3, and zero times on the GKnee. Recovery rate was 100% for the GKnee, followed by the M3 at 71% and MPK at 58%.

Table 6.5: For P4, all counts of stumble inductions separated by various types of stumble induction modes and categorized by knee type. Each knee includes a sum of induction modes, and total summative values are shown for ineffective, effective, and total induced stumbles.

Knee	M3				GKnee				MPK				
Mode	bun	obs	une	sum	bun	obs	une	sum	bun	obs	une	sum	**SUM**
Unaffected Gait	3	0	0	3	1	0	0	1	4	0	0	4	8
Affected Gait	4	13	0	17	9	10	2	21	6	7	6	19	57
Recovery	0	7	1	8	1	1	5	7	3	3	1	7	22
Fall	2	0	4	6	0	0	4	4	0	1	2	3	13
Ineffective Induced	7	13	0	20	10	10	2	22	10	7	6	23	65
Effective Induced	2	7	5	14	1	1	9	11	3	4	3	10	35
Total Induced	**9**	**20**	**5**	**34**	**11**	**11**	**11**	**33**	**13**	**11**	**9**	**33**	**100**

For P4, the most effective inductions were seen on the M3 (n=14, 41%), followed by the GKnee (n=11, 33%), with the fewest effective inductions occurring on the MPK (n=10, 30%). P4 fell six times while wearing the M3, four times on the GKnee, and three times on the MPK. Recovery rate was 63% for the GKnee, followed by the MPK at 70% and M3 at 57%.

Table 6.6: For P6, all counts of stumble inductions separated by various types of stumble induction modes and categorized by knee type. Each knee includes a sum of induction modes, and total summative values are shown for ineffective, effective, and total induced stumbles.

Knee	M3				GKnee				MPK				
Mode	bun	obs	une	sum	bun	obs	une	sum	bun	obs	une	sum	**SUM**
Unaffected Gait	1	2	1	4	3	0	1	4	4	0	0	4	12
Affected Gait	3	1	5	9	3	4	6	13	3	6	4	13	35
Recovery	1	2	2	5	5	7	3	15	2	5	1	8	28
Fall	5	5	0	10	1	0	0	1	4	1	2	7	18
Ineffective Induced	4	3	6	13	6	4	7	17	7	6	4	17	47
Effective Induced	6	7	2	15	6	7	3	16	6	6	3	15	46
Total Induced	**10**	**10**	**8**	**28**	**12**	**11**	**10**	**33**	**13**	**12**	**7**	**32**	**93**

For P6, the most effective inductions were seen on the M3 (n=15, 54%), followed by GKnee (n=16, 48%) and MPK (n=15, 47%). P6 fell ten times while wearing the M3, seven times on the MPK, and one time on the GKnee. Recovery rate was 94% for the GKnee, followed by the MPK at 53% and M3 at 33%.

Summed Tables

Below, in Table 6.7 and Table 6.8, are shown the total summed classification of stumble inductions by patient and by knee, respectively. More detailed summed tables can be found in Appendix 9.6.

Table 6.7: Summed counts of stumble inductions shown by patient. The categories of stumble induction mode and knee were summed for these counts. The rightmost column shows a summary of each row.

Patient	**P1**	**P2**	**P3**	**P4**	**P6**	**SUM**
Unaffected Gait	14	10	19	8	12	63
Affected Gait	32	41	42	57	35	207
Recovery	39	36	25	22	28	150
Fall	23	17	9	13	18	80
Ineffective Induced	46	51	61	65	47	270
Effective Induced	62	53	34	35	46	230
Total Induced	**108**	**104**	**95**	**100**	**93**	**500**

The percentage of ineffective and effective induced stumbles for all subjects is 54% and 46%, respectively. For P1, the ratio of effective versus ineffective is higher than for other subjects. P2 and P6 are nearly balanced. P3 and P4 have a lower effective versus ineffective ratio.

Table 6.8: Summed counts of stumble inductions shown by knee. The categories of stumble induction mode and patient were summed for these counts. The rightmost column shows a summary of each row.

Knee	**M3**	**GKnee**	**MPK**	**SUM**
Unaffected Gait	14	23	26	63
Affected Gait	61	78	68	207
Recovery	44	59	47	150
Fall	43	11	26	80
Ineffective Induced	75	101	94	270
Effective Induced	87	70	73	230
Total Induced	**162**	**171**	**167**	**500**

GKnee Recollection

Data in Table 6.9 are the total stumble incidences during the recollection trials for each individual patient and knee mode (on or off). “On” indicates that the control system was actively locking the piston during the trial. “Off” indicates the control system was only collecting data internally.

Table 6.9: Summed counts of stumble inductions shown by individual. The category of stumble induction mode was summed for these counts. The rightmost columns show a summary.

Patient	**P2**		**P3**		**P4**		**SUM**		
Mode	Off	On	Off	On	Off	On	Off	On	Total
Unaffected Gait	3	4	3	4	13	4	19	12	31
Affected Gait	15	12	16	19	15	16	46	47	93
Recovery	11	13	11	6	5	7	27	26	53
Fall	1	1	2	2	3	3	6	6	12
Ineffective Induced	18	16	19	23	28	20	65	59	124
Effective Induced	12	14	13	8	8	10	33	32	65
Total Induced	**30**	**30**	**32**	**31**	**36**	**30**	**98**	**91**	**189**

In total, the recovery rate was 81% for both the “on” and “off” systems.

6.4.3 Gait Parameters

Table 6.10: Average ± standard deviation of temporal parameters: step length (m), step duration (s), and knee angle (deg) across patients 1, 2, 3, 4, 6. Values separated by knee type and by leg side.

Parameter	Intact Limb			Prosthetic Limb		
	M3	**GKnee**	**MPK**	**M3**	**GKnee**	**MPK**
Step Length (m)	0.55 ± 0.01	0.57 ± 0.04	0.54 ± 0.05	0.55 ± 0.02	0.57 ± 0.04	0.54 ± 0.05
Step Duration (s)	0.91 ± 0.04	0.92 ± 0.07	0.89 ± 0.08	0.78 ± 0.03	0.82 ± 0.08	0.79 ± 0.06
Knee Angle (deg)				75.7 ± 9.5	57.2 ± 8.5	59.8 ± 4.9

6.4.4 Statistical Analysis

Below are the statistical analyses for the primary data collection (Figures 6.2 through 6.4) and the GKnee recollection (Figures 6.5 through 6.7).

Primary Collection

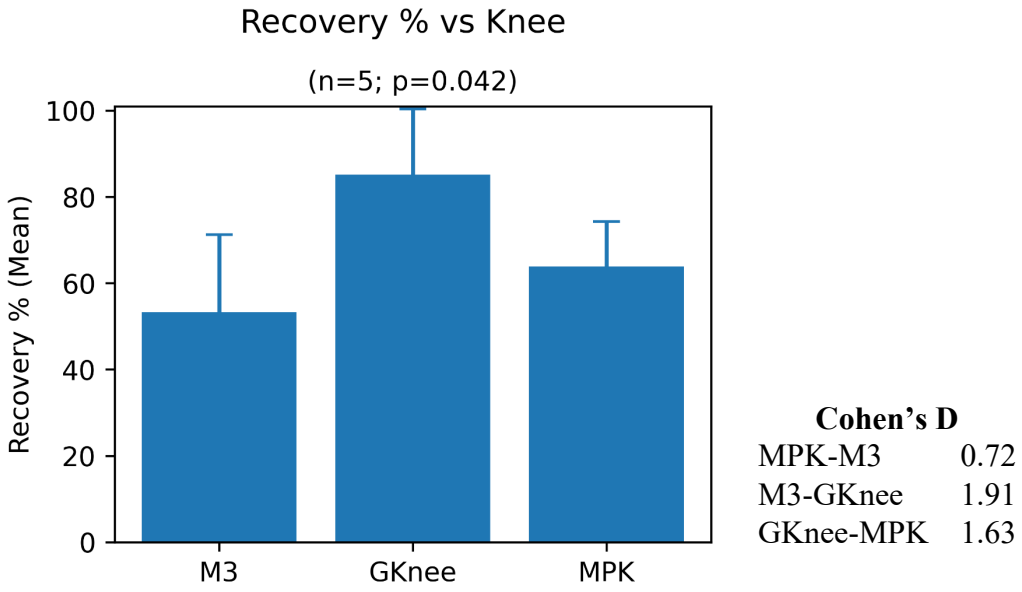

Figure 6.2: Recovery rate of total effectively induced stumbles (recovery and fall). Data is shown as mean with standard deviation bars. With five subjects, statistical significance was achieved (p=0.042). Right column shows Cohen’s D effect size between the means of paired knee systems.

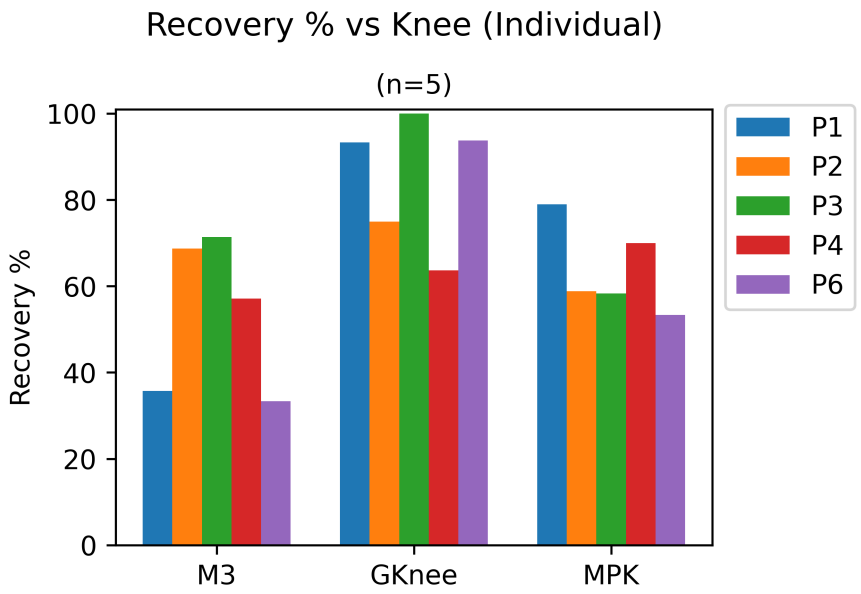

Figure 6.3: Recovery rate of total effectively induced stumbles (recovery and fall). Data is shown as individual data per knee.

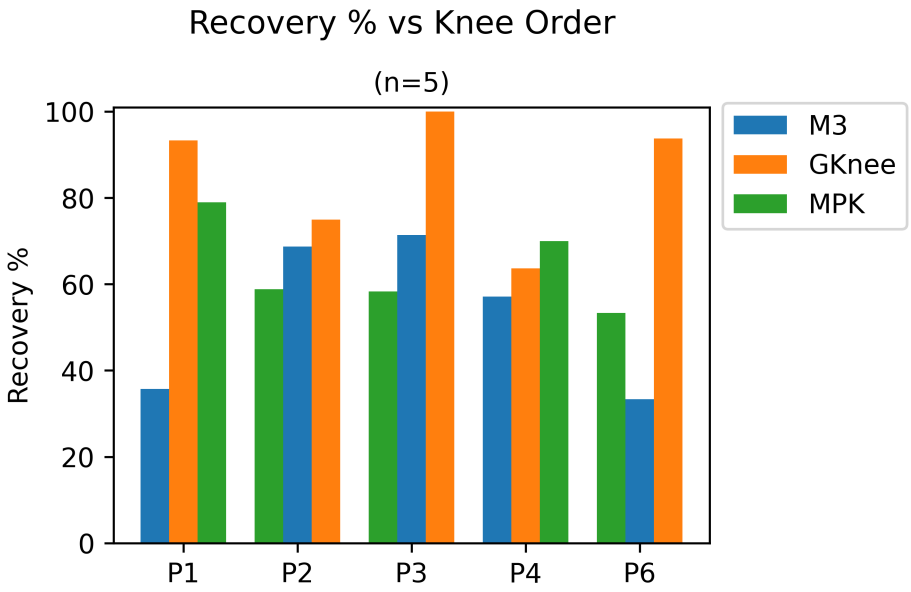

Figure 6.4: Recovery rate of total effectively induced stumbles (recovery and fall). Data is shown as knee order per individual.

GKnee Recollection

The recollection of data for the GKnee was performed on three subjects (P2, P3, P4), and compared the GKnee with the control system enabled (on) or disabled (off).

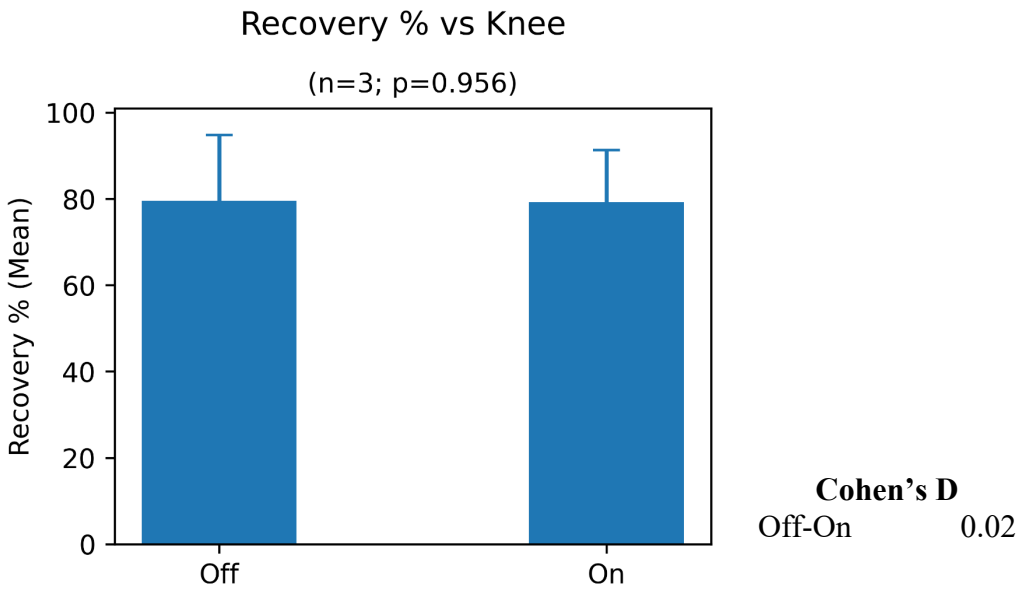

Figure 6.5: Recovery rate of total effectively induced stumbles (recovery and fall). Data is shown as mean with standard deviation bars. With three subjects, statistical significance was achieved in affirming the null hypothesis (p=0.956). Right column shows Cohen's D effect size between the means of two conditions.

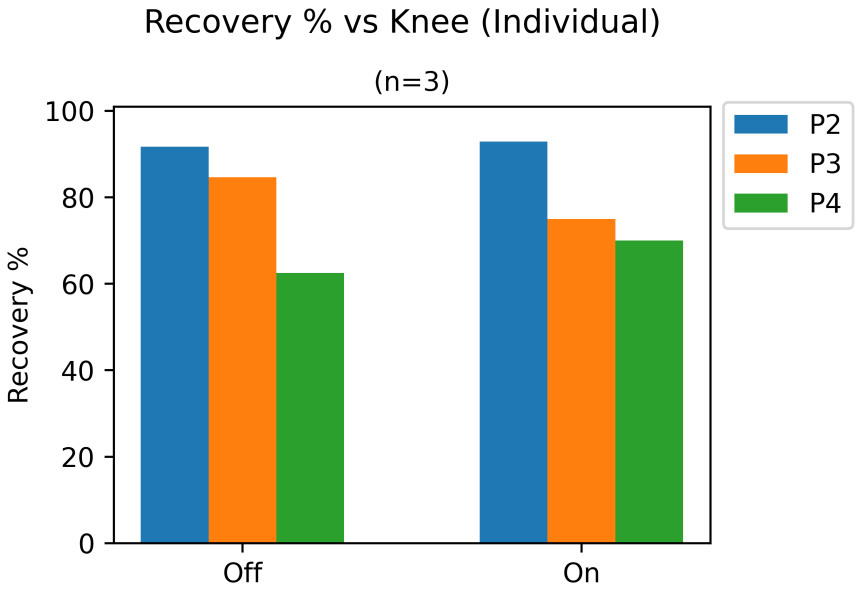

Figure 6.6: Recovery rate of total effectively induced stumbles (recovery and fall). Data is shown as individual data per condition.

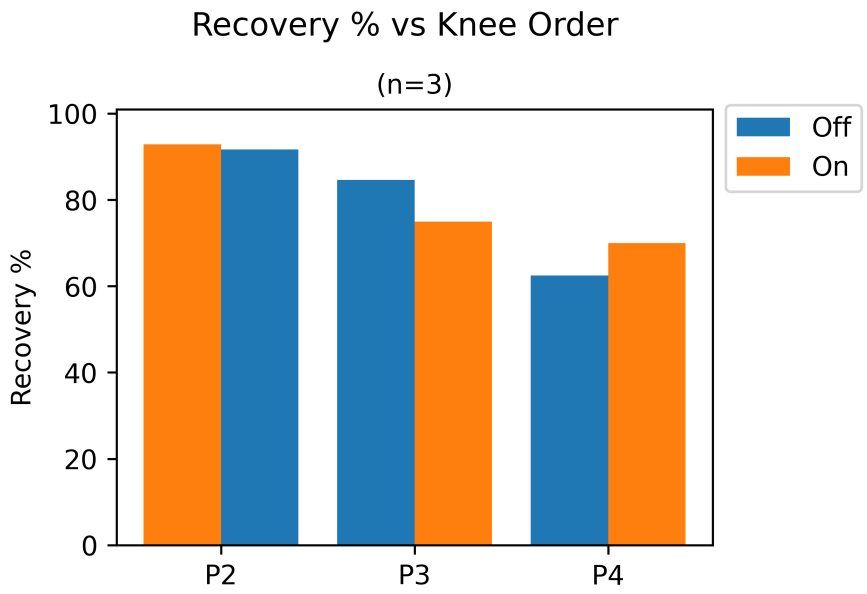

Figure 6.7: Recovery rate of total effectively induced stumbles (recovery and fall). Data is shown as knee condition order per individual.

6.5 Discussion

This aim was evaluated by classifying the stumbles and calculating gait parameters. Because the primary evaluation of this study was the fall rates between knee systems, the classified stumbles were subjected to statistical analysis.

Stumble Classifications

The subclassification of stumbles can raise questions of data classification subjectivity of the data classification. The original distinction between walking, stumbling, and falling was to be quite straightforward, but that stemmed from the false assumption that every stumble induction would lead to patients stumbling or falling. While this study attempts to maintain rigor without bias, the lines between the subclassifications of unaffected gait, affected gait, and recovery are partially subjective. To mitigate this, all subclassifications were done per patient in a continuum to preserve the classification "state of mind." However, fundamentally, the potential subjectivity has no effect on the statistical significance between the knees. Because the recovery percentage is

dependent enough on the fall rate, that even with recoveries reduced to zero, there is still a significant difference between the knees based on falling incidences alone. Also noting that the subclassification of falls was the most quantifiable measure because the conditions of falling were not open to subjective interpretation.

Gait Parameters

One of the hypotheses of this aim was that the gait symmetry across subjects would be different between the knee systems. However, for both step length and step duration, limited variance was seen when comparing knee system in terms of symmetry. While there are minor differences in step length between the knees themselves, they are within normal deviations. The step durations have evident differences between the intact and prosthetic limbs, but comparatively, these differences appear to be consistent across knees.

The symmetrical effect may be due to several factors. Though gait symmetry comparisons between knees is not uncommon in the literature, it appears that several studies compare two knee systems primarily, the Ottobock C-Leg and the Mauch SNS. From the studies, the C-Leg performs better than the Mauch SNS, but the results were generalized to apply to all non-microprocessor knees (K. R. Kaufman et al., 2007, 2012). Other studies have concluded that minimal biomechanical differences exist between the two (Segal et al., 2006).

The results presented in this aim are perhaps the least differentiated to date. A possible explanation could be the acclimation period allowed. In some studies, patients are given weeks to become acclimated. It could be that everyday MPK users do not vary their gait parameters much when switching between knees for a brief period. A more long-term study would bring clarity to this subject. The only gait parameter that shows certain differences between the knees is prosthetic limb angle during gait. The GKnee and MPK performed very similarly, while the M3 had a very

large angle. This could indicate that the stiffness of the knee should be adjusted. However, no patients made comments on this during the trials.

Statistical Analysis

The primary result of this study is the comparison of the recovery and fall rates between knee systems. As can be seen in Figure 6.2, the means of the recovery rate are statistically significant between the knee systems, and the effect of the difference is large or nearly so (MPK-M3). It should be noted that the order switching methodology appears to have been effective, and highlights that the order of the trials was not a predicting factor of the outcome, seen in Figure 6.4. Additionally, the recovery ratio is dependent on the fall rate, and the significant differences between the knees remain if the recoveries approach zero. Since the fall rate is the least subjective subclassification of the data, this gives assurance that the differences between the systems are legitimate.

However, it must be clarified that in these results, the GKnee was a purely mechanical system. Due to the angle errors discussed in Aim 3, the locking mechanism was disabled for the duration of the primary data collection of Aim 4. The recollection comparison between the GKnee with the control system "on" and "off" yields no statistically significant differences between the recovery rates ($p=0.956$). Therefore, it can be taken as affirming the null hypothesis and indicates that any difference is likely due to chance. This is affirmed by the comparison of subjects by knee order (Figure 6.7), where the visible trend is not the order or the GKnee mode, but rather the subjects themselves. Thus, the underlying objective of Aim 4 was not achieved despite a ML-based control system meeting the objectives of Aim 2 and 3. Why the reactions of the control system did not impact the ratio of recovery and falls is unknown but discussed below.

A possible reason for this is that of measurement error, namely the low rate of falls that are associated with the GKnee, which are shown in Table 6.8 and Table 6.9. The low rate of falls is what causes the large difference between the GKnee and the MPK and M3, but in this case also may be obscuring the legitimate differences that may be found between the purely physical GKnee and the GKnee with an ML-based control system locking in response to the stumble inductions. Another reason could be the reaction times of the control system. As discussed previously, the limitation of the marked stumbles could imply that the true reaction time is not fast enough to make a difference for the subject in recovery. If, as the data imply, there is no difference between the control system enabled or disabled, and the control system meets the objectives of Aim 2 and 3, then the assumptions of those objectives must be flawed.

Lastly, a potential reason for the GKnee's improved performance over the MPKs could be the weight of the system. Commercial MPKs advertise lightness and though an objective, this has not been realized in the GKnee. An estimate of the C-Leg's moment of inertia is 9.18e-3 $kg*m^2$. By contrast, the GKnee has a moment of inertia of 8.59e-2 $kg*m^2$. Though this difference should be mitigated by the foot, the difference in momentum may help patients overcome stumble inductions. The effectiveness of the physical GKnee would not have been realized if not for the errors in the control system during the primary data collection.

Limitations

This aim incorporates the objectives of the other three Aims; therefore, the constraints of each must also be considered for this final comparison. While the performance of the Aim 2 classifier appeared to be a large obstacle, the results of Aim 3 improved and overcame it. However, Aim 3 did highlight the issues of stumble marking and reaction times. The limitations of this aim are the subject pool, duration, stumble repeatability, and ultimately the performance of the GKnee.

The subjects of this Aim came from a relatively uniform pool. Though age and weight varied a considerate amount, height had little variation, and all subjects were male. Since one of the subjects in the initial data collection was female, there is no reason to assume that the stumble classifications would turn out differently if patients were female in the later trials, but this cannot be assumed. Additionally, though statistical significance was achieved with large effects, the number of subjects was relatively low.

The trails of this aim were relatively long, in some cases the data collection took three hours. However, though there is no definitive established standard, some literature studies compared knee systems after months of acclimation. This was not possible in the GKnee because it is both tethered and not configured for gait activities outside of walking and stumbling. While this is more common for short-term studies, the lack of change in the gait parameters may indicate that walking patterns had not adapted to the different systems sufficiently. Long-term studies may find greater differences between the gait parameters and the recovery rates.

Another limitation is the inexact timing of the gait perturbation. While this was intentional in both training and testing, to simulate the chaotic nature of stumbles, it restricts exact analysis of stumble inductions and simple comparisons to tightly controlled studies.

Future Work

The GKnee should be evaluated in further studies that are both long-term and with an expanded population. However, before this can occur, the continued work of the other Aims must first be addressed. Namely, the physical characteristics of the GKnee must be redesigned, improved, and streamlined; the classification system of Aim 2 should be improved with a larger training population and an overhaul of the methods; and the control system of Aim 3 must be revisited to ensure that the system reaction to stumbles occurs quickly enough to be effective.

After those improvements have been made the GKnee should be subjected to further studies and evaluation. Additionally, the data from these trials should be used to simulate further studies. To date, stumble data is very scarce, and the addition of these datasets should be used to develop systems that are even more effective and predicting stumbles.

Lastly, the recovery ratio results must be investigated more fully to find both the reasons for the physical GKnee outperforming the MPKs, but also to determine why the ML model and control system were unable to effect any changes. There are several possibilities for the GKnee's stability, such as, the four-bar mechanism, the moment of inertia, or the dampening characteristics of the hydraulic system; but each of these will require further tests and exploration to make a definitive statement. As stated, the reasons for the control system's lack of efficacy may be issues of timing in both marking and reacting to stumbles, or a function of the relatively low sample size for the recovery ratio of the GKnee. Each of these will be explored further in the future to establish whether the control system was simply ineffective in application or whether ML models are not suited to improving this process entirely.

Broader Implications

As stated previously, the literature contains many examples of MPKs outperforming passive mechanical systems. Specific studies reference energy expenditure (Johansson et al., 2005) or stability (Blumentritt et al., 2009a) as the factors to consider for greater MPK performance. However, as Segal et al. pointed out, biomechanical differences between the mechanical systems and MPKs can be minimal. Since the control system had no effect on the results, this study has outlined how a mechanical knee outperformed an MPK in the specific case of a multi-mode stumble induction trial.

Chapter 7: Conclusion

The field of biomechatronics is evolving quickly with advances in computer science, biology, and electrical and mechanical engineering. Coupled with increased interests in machine learning across all industry sectors, there are opportunities to leverage advanced analytics in uniquely complex problems. This study deployed a machine learning model in real-time to predict stumble inductions in a microprocessor-controlled prosthetic knee.

Aim 1 demonstrated the characteristics of a hydraulic prosthetic knee that could withstand the worst-case loading conditions of a 100 kg patient with a built-in safety-factor. It showed the feasibility of a servo actuated hydraulic with flow characteristics that did not impede gait and allowed the prosthetic knee to swing freely at 500 degrees/second.

A LSTM machine learning model was developed in Aim 2 to predict "stumbling" and "walking" in real-time. Though the accuracy and other metrics of this aim did not achieve the hypothesized values, the control system of Aim 3 demonstrated the hypothesized prediction switching times and compensated for Aim 2 by achieving step accuracy of 91.4%, recall of 77.5%, and precision of 81.0%.

Aim 4 established that the proposed system accomplished its two primary tasks: accurate stumble prediction with controlled reaction and a statistically significant improvement in the recovery rate among patients while wearing the GKnee. However, while the ML algorithm accurately predicted stumbles, its implementation in the control system did not result in any improvement to the recovery ratio of the GKnee trials. Therefore, this study demonstrated a control system with real-time prediction of stumbles; and surprisingly also demonstrated a passive cost-constrained prosthetic knee with a measurable reduction in fall incidences.

Chapter 8: References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/

Alaqtash, M., Sarkodie-Gyan, T., Yu, H., Fuentes, O., Brower, R., & Abdelgawad, A. (2011). Automatic classification of pathological gait patterns using ground reaction forces and machine learning algorithms. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, *2011*, 453–457. https://doi.org/10.1109/IEMBS.2011.6090063

Amputee Coalition. (2012). *Amputee statistics you ought to know*. http://www.advancedamputees.com/amputee-statistics-you-ought-know

Badawi, A. A., Al-Kabbany, A., & Shaban, H. (2019). Multimodal Human Activity Recognition From Wearable Inertial Sensors Using Machine Learning. *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, 402–407. https://doi.org/10.1109/iecbes.2018.8626737

Bellmann, M., Köhler, T. M., & Schmalz, T. (2019). Comparative biomechanical evaluation of two technologically different microprocessor-controlled prosthetic knee joints in safety-relevant daily-life situations. *Biomedizinische Technik*, *64*(4), 407–420. https://doi.org/10.1515/bmt-2018-0026

Bellmann, M., Schmalz, T., & Blumentritt, S. (2010). Comparative Biomechanical Analysis of Current Microprocessor-Controlled Prosthetic Knee Joints. *Archives of Physical Medicine and Rehabilitation*, *91*(4), 644–652. https://doi.org/10.1016/j.apmr.2009.12.014

Benedetti, M. G., Piperno, R., Simoncini, L., Bonato, P., Tonini, A., & Giannini', S. (1999). Gait abnormalities in minimally impaired multiple sclerosis patients. *Multiple Sclerosis*, *5*(5), 363–368. https://doi.org/10.1177/135245859900500510

Blumentritt, S., Schmalz, T., & Jarasch, R. (2009a). The safety of C-leg: Biomechanical tests. *Journal of Prosthetics and Orthotics*, *21*(1), 2–15. https://doi.org/10.1097/JPO.0b013e318192e96a

Blumentritt, S., Schmalz, T., & Jarasch, R. (2009b). The safety of C-leg: Biomechanical tests. *Journal of Prosthetics and Orthotics*, *21*(1), 2–15. https://doi.org/10.1097/JPO.0b013e318192e96a

Boonstra, a M., Fidler, V., & Eisma, W. H. (1993). Walking speed of normal subjects and amputees: aspects of validity of gait analysis. *Prosthetics and Orthotics International*, *17*(2), 78–82. https://doi.org/10.3109/03093649309164360

Center for Orthotic & Prosthetic Care. (2008). *Amputation Statistics Fact Sheet*. *200*(1999), 2002.

Chauhan, S. S., & Bhaduri, S. C. (2011). Evaluation of the Polycentric above Knee Prosthesis. *15th National Conference on Machines and Mechanisms*, *6*, 7. http://www.nacomm2011.ammindia.org/files/papers/nacomm2011_attachment_21_1.pdf

Chawla, N. V, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Chelius, G., Braillon, C., Pasquier, M., Horvais, N., Gibollet, R. P., Espiau, B., & Azevedo Coste, C. (2011). A wearable sensor network for gait analysis: A six-day experiment of

running through the desert. *IEEE/ASME Transactions on Mechatronics*, *16*(5), 878–883. https://doi.org/10.1109/TMECH.2011.2161324
Chen, S., & Ravallion, M. (2010). China is Poorer than We Thought, but No Less Successful in the Fight against Poverty. *Debates on the Measurement of Global Poverty*, *November*, 1577–1625. https://doi.org/10.1093/acprof:oso/9780199558032.003.0015
Chereshnev, R., & Kertész-Farkas, A. (2017). HuGaDB: Human Gait Database for Activity Recognition from Wearable Inertial Sensor Networks. In *Springer*. https://github.com/romanchereshnev/HuGaDB.
Chollet, F., & others. (2015). *Keras*.
Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences. In *Statistical Power Analysis for the Behavioral Sciences* (2nd Editio). Routledge. https://doi.org/10.4324/9780203771587
Cordero, A. F., Koopman, H. J. F. M., & Van Der Helm, F. C. T. (2004a). Mechanical model of the recovery from stumbling. *Biological Cybernetics*, *91*(4), 212–220. https://doi.org/10.1007/s00422-004-0508-0
Cordero, A. F., Koopman, H. J. F. M., & Van Der Helm, F. C. T. (2004b). Mechanical model of the recovery from stumbling. *Biological Cybernetics*, *91*(4), 212–220. https://doi.org/10.1007/s00422-004-0508-0
Crenshaw, J. R., Kaufman, K. R., & Grabiner, M. D. (2013). Trip recoveries of people with unilateral, transfemoral or knee disarticulation amputations: Initial findings. *Gait and Posture*, *38*(3), 534–536. https://doi.org/10.1016/j.gaitpost.2012.12.013
Eveld, M. E., King, S. T., Vailati, L. G., Zelik, K. E., & Goldfarb, M. (2021). On the basis for stumble recovery strategy selection in healthy adults. *Journal of Biomechanical Engineering*, *143*(7). https://doi.org/10.1115/1.4050171
Forner-Cordero, A., Ackermann, M., & De Lima Freitas, M. (2011). A method to simulate motor control strategies to recover from perturbations: Application to a stumble recovery during gait. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 7829–7832. https://doi.org/10.1109/IEMBS.2011.6091929
Freedom Innovations. (2015). *Plié® 3*. http://www.freedom-innovations.com/plie-3/
Gailey, R., Allen, K., Castles, J., Kucharik, J., & Roeder, M. (2008). Review of secondary physical conditions associated with lower-limb amputation and long-term prosthesis use. *Journal of Rehabilitation Research and Development*, *45*(1), 15–29. https://doi.org/10.1682/JRRD.2006.11.0147
Galey, L. J. (2016). Development and initial testing of a low-cost, electronic, microprocessor-controlled prosthetic knee. In *ETD Collection for University of Texas, El Paso*. http://digitalcommons.utep.edu/dissertations/AAI10251518
Galey, L. J., & Gonzalez, R. V. (2022). Design and Initial Evaluation of a Low-Cost Microprocessor-Controlled Above-Knee Prosthesis: A Case Report of 2 Patients. *Prosthesis*, *4*(1), 60–72. https://doi.org/10.3390/prosthesis4010007
Gard, S. A. (2006). Use of Quantitative Gait Analysis for the Evaluation of Prosthetic Walking Performance. *JPO Journal of Prosthetics and Orthotics*, *18*, P93–P104. https://doi.org/10.1097/00008526-200601001-00011
Gard, S. A., Childress, D. S., & Uellendahl, J. E. (1996). The Influence of Four-Bar Linkage Knees on Prosthetic Swing-Phase Floor Clearance. *Journal of Prosthetic and Orthotics*, *8*(2), 34–40. https://doi.org/10.1097/00008526-199603000-00006

Godest, A. C., Beaugonin, M., Haug, E., Taylor, M., & Gregson, P. J. (2002). Simulation of a knee joint replacement during a gait cycle using explicit finite element analysis. *Journal of Biomechanics*, *35*(2), 267–275. https://doi.org/10.1016/S0021-9290(01)00179-8

Goldfarb, M. (2013). Consideration of Powered Prosthetic Components as They Relate to Microprocessor Knee Systems. *JPO Journal of Prosthetics and Orthotics*, *25*(4S), 65–75. https://doi.org/10.1097/jpo.0b013e3182a8953e

Gosthipaty, A. R., Chakraborty, D., & Raha, R. (2022). *Long Short-Term Memory Networks*. PyImageSearch. https://pyimagesearch.com/2022/08/01/long-short-term-memory-networks/

Grabiner, M. D., Koh, T. J., Lundin, T. M., & Jahnigen, D. W. (1993). Kinematics of recovery from a stumble. *Journals of Gerontology*, *48*(3), 97–102. https://doi.org/10.1093/geronj/48.3.M97

Hafner, B. J., Willingham, L. L., Buell, N. C., Allyn, K. J., & Smith, D. G. (2007). Evaluation of Function, Performance, and Preference as Transfemoral Amputees Transition From Mechanical to Microprocessor Control of the Prosthetic Knee. *Archives of Physical Medicine and Rehabilitation*, *88*(2), 207–217. https://doi.org/10.1016/j.apmr.2006.10.030

Hajj Chehade, N., Ozisik, P., Gomez, J., Ramos, F., & Pottie, G. (2012). Detecting stumbles with a single accelerometer. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 6681–6686. https://doi.org/10.1109/EMBC.2012.6347527

Hak, L., Van Dieën, J. H., Van Der Wurff, P., Prins, M. R., Mert, A., Beek, P. J., & Houdijk, H. (2013). Walking in an unstable environment: Strategies used by transtibial amputees to prevent falling during gait. *Archives of Physical Medicine and Rehabilitation*, *94*(11), 2186–2193. https://doi.org/10.1016/j.apmr.2013.07.020

Hanlon, M., & Anderson, R. (2009). Real-time gait event detection using wearable sensors. *Gait & Posture*, *30*(4), 523–527. https://doi.org/10.1016/J.GAITPOST.2009.07.128

Herr, H., & Wilkenfeld, A. (2003a). User-adaptive control of a magnetorheological prosthetic knee. *Industrial Robot*, *30*(1), 42–55. https://doi.org/10.1108/01439910310457706

Herr, H., & Wilkenfeld, A. (2003b). User-adaptive control of a magnetorheological prosthetic knee. *Industrial Robot*, *30*(1), 42–55. https://doi.org/10.1108/01439910310457706

Highsmith, M. J., Kahle, J. T., Bongiorni, D. R., Sutton, B. S., Groer, S., & Kaufman, K. R. (2010). Safety, energy efficiency, and cost efficacy of the C-Leg for transfemoral amputees: A review of the literature. *Prosthetics and Orthotics International*, *34*(4), 362–377. https://doi.org/10.3109/03093646.2010.520054

Highsmith, M. J., Kahle, J. T., Shepard, N. T., & Kaufman, K. R. (2014). The Effect of the C-Leg Knee Prosthesis on Sensory Dependency and Falls During Sensory Organization Testing. *Technology & Innovation*, *15*(4), 343–347. https://doi.org/10.3727/194982413x13844488879212

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. In *MEMORY Neural Computation* (Vol. 9, Issue 8). http://www7.informatik.tu-muenchen.de/~hochreithttp://www.idsia.ch/~juergen

Jin, D., Zhang, R., Dimo, H. O., Wang, R., & Zhang, J. (2003). Kinematic and dynamic performance of prosthetic knee joint using six-bar mechanism. *Journal of Rehabilitation Research and Development*, *40*(1), 39–48. https://doi.org/10.1682/JRRD.2003.01.0039

Johansson, J. L., Sherrill, D. M., Riley, P. O., Bonato, P., & Herr, H. (2005). A clinical comparison of variable-damping and mechanically passive prosthetic knee devices.

American Journal of Physical Medicine & Rehabilitation / Association of Academic Physiatrists, *84*(8), 563–575. https://doi.org/10.1097/01.phm.0000174665.74933.0b

Kahle, J. T., Highsmith, M. J., & Hubbard, S. L. (2008). Comparison of nonmicroprocessor knee mechanism versus C-Leg on Prosthesis Evaluation Questionnaire, stumbles, falls, walking tests, stair descent, and knee preference. *Journal of Rehabilitation Research and Development*, *45*(1), 1–14. https://doi.org/10.1682/JRRD.2007.04.0054

Kalanovic, V. D., Popovic, D., & Skaug, N. T. (2000). Feedback error learning neural network for trans-femoral prosthesis. *IEEE Transactions on Rehabilitation Engineering*, *8*(1), 71–80. https://doi.org/10.1109/86.830951

Kapti, A. O., & Yucenur, M. S. (2006). Design and control of an active artificial knee joint. *Mechanism and Machine Theory*, *41*(12), 1477–1485. https://doi.org/10.1016/j.mechmachtheory.2006.01.017

Kaufman, K., Anderson, T., Schneider, G., Walsh, K., & Bme, M. S. (2008). Mechanisms of Stumble Recovery : Non-Microprocessor Controlled Compared To Microprocessor-Controlled Prosthetic Knees. *Knee, The*, 1–4.

Kaufman, K. R., Frittoli, S., & Frigo, C. A. (2012). Gait asymmetry of transfemoral amputees using mechanical and microprocessor-controlled prosthetic knees. *Clinical Biomechanics*, *27*(5), 460–465. https://doi.org/10.1016/j.clinbiomech.2011.11.011

Kaufman, K. R., Levine, J. A., Brey, R. H., Iverson, B. K., McCrady, S. K., Padgett, D. J., & Joyner, M. J. (2007). Gait and balance of transfemoral amputees using passive mechanical and microprocessor-controlled prosthetic knees. *Gait and Posture*, *26*(4), 489–493. https://doi.org/10.1016/j.gaitpost.2007.07.011

Kaufman, K. R., Levine, J. A., Brey, R. H., McCrady, S. K., Padgett, D. J., & Joyner, M. J. (2008). Energy Expenditure and Activity of Transfemoral Amputees Using Mechanical and Microprocessor-Controlled Prosthetic Knees. *Archives of Physical Medicine and Rehabilitation*, *89*(7), 1380–1385. https://doi.org/10.1016/j.apmr.2007.11.053

Keçeci, A., Yildirak, A., & Özyazici, K. (2018). Gait Recognition via Machine Learning. In *International Conference on Cyber Security and Computer Science (ICONCS'18)*. http://www.iconcs.org/papers/Paper_70.pdf

Kelleher, K. J., Spence, W., Solomonidis, S., & Apatsidis, D. (2010). *Disability and Rehabilitation The characterisation of gait patterns of people with multiple sclerosis The characterisation of gait patterns of people with multiple sclerosis*. https://doi.org/10.3109/09638280903464497

Keller, T. S., Weisberger, A. M., Ray, J. L., Hasan, S. S., Shiavi, R. G., & Spengler, D. M. (1996). Relationship between vertical ground reaction force and speed during walking, slow jogging, and running. *Clinical Biomechanics*, *11*(5), 253–259. https://doi.org/10.1016/0268-0033(95)00068-2

King, S. T., Eveld, M. E., Martínez, A., Zelik, K. E., & Goldfarb, M. (2019). A novel system for introducing precisely-controlled, unanticipated gait perturbations for the study of stumble recovery. *Journal of NeuroEngineering and Rehabilitation*, *16*(1), 69. https://doi.org/10.1186/s12984-019-0527-7

Kumar, N., Soni, S., Kumar, A., & Sohi, B. S. (2010). Low cost prototype development of electronic knee. *Industrial Research*, *69*(June), 444–448.

Lambrecht, S., Harutyunyan, A., Tanghe, K., Afschrift, M., De Schutter, J., & Jonkers, I. (2017). Real-Time Gait Event Detection Based on Kinematic Data Coupled to a Biomechanical Model †. *Sensors 2017, Vol. 17, Page 671*, *17*(4), 671. https://doi.org/10.3390/S17040671

Lawson, B. E., Varol, H. A., Sup, F., & Goldfarb, M. (2010a). Stumble detection and classification for an intelligent transfemoral prosthesis. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10*, 511–514. https://doi.org/10.1109/IEMBS.2010.5626021

Lawson, B. E., Varol, H. A., Sup, F., & Goldfarb, M. (2010b). Stumble detection and classification for an intelligent transfemoral prosthesis. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10*, 511–514. https://doi.org/10.1109/IEMBS.2010.5626021

Mannini, A., Trojaniello, D., Cereatti, A., & Sabatini, A. M. (2016). A machine learning framework for gait classification using inertial sensors: Application to elderly, post-stroke and huntington's disease patients. *Sensors (Switzerland)*, *16*(1). https://doi.org/10.3390/s16010134

Maqbool, H. F., Husman, M. A. B., Awad, M. I., Abouhossein, A., Iqbal, N., & Dehghani-Sanij, A. A. (2017). A Real-Time Gait Event Detection for Lower Limb Prosthesis Control and Evaluation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *25*(9), 1500–1509. https://doi.org/10.1109/TNSRE.2016.2636367

Marayong, P., Khoo, I.-H., Krishnan, V., Sciortino, A., & Daniel Crews BS, B. J. (2017). *Real-Time Estimation of Knee Angle, Heel-Strike, and Toe-Off Events for Gait Rehabilitation Devices*. https://search.proquest.com/docview/1949398483?pq-origsite=gscholar

Martinez-Villalpando, E. C., & Herr, H. (2009). Agonist-antagonist active knee prosthesis: A preliminary study in level-ground walking. *The Journal of Rehabilitation Research and Development*, *46*(3), 361. https://doi.org/10.1682/JRRD.2008.09.0131

Mentiplay, B. F., Banky, M., Clark, R. A., Kahn, M. B., & Williams, G. (2018). Lower limb angular velocity during walking at various speeds. *Gait and Posture*, *65*, 190–196. https://doi.org/10.1016/j.gaitpost.2018.06.162

Michael, J. (2001). *Article on Amputee Demographics*. http://www.oandp.com/news/jmcorner/2001-02/2.asp

Miller, W. C., Deathe, A. B., Speechley, M., & Koval, J. (2001). The influence of falling, fear of falling, and balance confidence on prosthetic mobility and social activity among individuals with a lower extremity amputation. *Archives of Physical Medicine and Rehabilitation*, *82*(9), 1238–1244. https://doi.org/10.1053/apmr.2001.25079

Mittal, A. (2019). *Understanding RNN and LSTM. What is Neural Network?* Medium. https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

Miyazaki, S. (1997). Long-term unrestrained measurement of stride length and walking velocity utilizing a piezoelectric gyroscope. *IEEE Transactions on Biomedical Engineering*, *44*(8), 753–759. https://doi.org/10.1109/10.605434

Mundell, B., Maradit Kremers, H., Visscher, S., Hoppe, K., & Kaufman, K. (2017). Direct medical costs of accidental falls for adults with transfemoral amputations. *Prosthetics and Orthotics International*, 030936461770480. https://doi.org/10.1177/0309364617704804

Narendra, K. S., & Balakrishnan, J. (1997). Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, *42*(2), 171–187. https://doi.org/10.1109/9.554398

Ochoa-Diaz, C., Rocha, T. S., de Levy Oliveira, L., Paredes, M. G., Lima, R., Bo, A. Padilha. L., & Borges, G. A. (2014). An above-knee prosthesis with magnetorheological variable-damping. *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 108–113. https://doi.org/10.1109/BIOROB.2014.6913761

Olson, D. L., & Delen, D. (2008). Advanced data mining techniques. In *Advanced Data Mining Techniques*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-76917-0

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., & others. (2019). *KerasTuner*.

Orendurff, M. S., Segal, A. D., Klute, G. K., McDowell, M. L., Pecoraro, J. a, & Czerniecki, J. M. (2006). Gait efficiency using the C-Leg. *Journal of Rehabilitation Research and Development*, *43*(2), 239–246. https://doi.org/10.1682/JRRD.2005.06.0095

Össur. (2016). *Rheo Knee 3*. Össur Catalog. http://www.ossur.com/prosthetic-solutions/products/dynamic-solutions/rheo-knee-3

Ottobock. (2016). *C-Leg® 4*. http://www.ottobockus.com/prosthetics/lower-limb-prosthetics/solution-overview/c-leg-above-knee-system/

Pirker, W., & Katzenschlager, R. (2017). Gait disorders in adults and the elderly: A clinical guide. In *Wiener Klinische Wochenschrift* (Vol. 129, Issues 3–4, pp. 81–95). Springer-Verlag Wien. https://doi.org/10.1007/s00508-016-1096-4

Quintero, D., Villarreal, D. J., Lambert, D. J., Kapp, S., & Gregg, R. D. (2018, June). Continuous-Phase Control of a Powered Knee–Ankle Prosthesis: Amputee Experiments Across Speeds and Inclines. *IEEE Transactions on Robotics*, *34*(3), 686–701. https://doi.org/10.1109/TRO.2018.2794536

Radcliffe, C. W., & Lamoreux, L. (1968). Uc-Bl Pneumatic Swing-Control Unit for Above-Knee Prostheses. *Bulletin of Prosthetics Research-Fall*, 73–89. http://www.rehab.research.va.gov/jour/68/5/2/73.pdf

Ren, B., Liu, J., & Chen, J. (2019). Simulating human–machine coupled model for gait trajectory optimization of the lower limb exoskeleton system based on genetic algorithm. *International Journal of Advanced Robotic Systems*, *16*(6). https://doi.org/10.1177/1729881419893493

Riley, P. O., Paolini, G., Della Croce, U., Paylo, K. W., & Kerrigan, D. C. (2007). A kinematic and kinetic comparison of overground and treadmill walking in healthy subjects. *Gait and Posture*, *26*(1), 17–24. https://doi.org/10.1016/j.gaitpost.2006.07.003

Rueterbories, J., Spaich, E. G., Larsen, B., & Andersen, O. K. (2010). Methods for gait event detection and analysis in ambulatory systems. In *Medical Engineering and Physics* (Vol. 32, Issue 6, pp. 545–552). Med Eng Phys. https://doi.org/10.1016/j.medengphy.2010.03.007

Sagawa, Y., Turcot, K., Armand, S., Thevenon, A., Vuillerme, N., & Watelain, E. (2011). Biomechanics and physiological parameters during gait in lower-limb amputees: A systematic review. *Gait and Posture*, *33*(4), 511–526. https://doi.org/10.1016/j.gaitpost.2011.02.003

Schillings, A. M., Van Wezel, B. M. H., & Duysens, J. (1996a). Mechanically induced stumbling during human treadmill walking. *Journal of Neuroscience Methods*, *67*(1), 11–17. https://doi.org/10.1016/0165-0270(95)00149-2

Schillings, A. M., Van Wezel, B. M. H., & Duysens, J. (1996b). Mechanically induced stumbling during human treadmill walking. *Journal of Neuroscience Methods*, *67*(1), 11–17. https://doi.org/10.1016/0165-0270(95)00149-2

Seel, T., Raisch, J., & Schauer, T. (2014). IMU-based joint angle measurement for gait analysis. *Sensors (Basel, Switzerland)*, *14*(4), 6891–6909. https://doi.org/10.3390/s140406891

Segal, A. D., Orendurff, M. S., Klute, G. K., McDowell, M. L., Pecoraro, J. A., Shofer, J., & Czerniecki, J. M. (2006). Kinematic and kinetic comparisons of transfemoral amputee gait

using C-Leg® and Mauch SNS® prosthetic knees. *Journal of Rehabilitation Research and Development*, *43*(7), 857–870. https://doi.org/10.1682/JRRD.2005.09.0147

Sessoms, P. H., Wyatt, M., Grabiner, M., Collins, J. D., Kingsbury, T., Thesing, N., & Kaufman, K. (2014). Method for evoking a trip-like response using a treadmill-based perturbation during locomotion. *Journal of Biomechanics*, *47*(1), 277–280. https://doi.org/10.1016/j.jbiomech.2013.10.035

Sessoms, P. H., Wyatt, M., Grabiner, M., Collins, J.-D., Kingsbury, T., Thesing, N., & Kaufman, K. (2013). Method for evoking a trip-like response using a treadmill-based perturbation during locomotion. *Journal of Biomechanics*, *47*, 277–280. https://doi.org/10.1016/j.jbiomech.2013.10.035

Shawen, N., Lonini, L., Mummidisetty, C. K., Shparii, I., Albert, M. V, Kording, K., & Jayaraman, A. (2017). Fall Detection in Individuals With Lower Limb Amputations Using Mobile Phones: Machine Learning Enhances Robustness for Real-World Applications. *JMIR MHealth and UHealth*, *5*(10), e151. https://doi.org/10.2196/mhealth.8201

Shepherd, M. K., & Rouse, E. J. (2017). Design of a quasi-passive ankle-foot prosthesis with biomimetic, variable stiffness. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 6672–6678. https://doi.org/10.1109/ICRA.2017.7989788

Shirota, C., Simon, A. M., & Kuiken, T. A. (2015). Transfemoral amputee recovery strategies following trips to their sound and prosthesis sides throughout swing phase. *Journal of NeuroEngineering and Rehabilitation*, *12*(1), 1–11. https://doi.org/10.1186/s12984-015-0067-8

Shrider, E. A., Kollar, M., Chen, F., & Semega, J. (2021). Income and Poverty in the United States: 2020. *US Census Bureau, Current Population Reports*, 60–273. http://www.census.gov/content/dam/Census/library/publications/2021/demo/p60-273.pdf

Sup, F., Bohara, A., & Goldfarb, M. (2008). Design and Control of a Powered Transfemoral Prosthesis. *The International Journal of Robotics Research*, *27*(2), 263–273. https://doi.org/10.1177/0278364907084588

Sup, F., Varol, H. A., Mitchell, J., Withrow, T., & Goldfarb, M. (2008). Design and control of an active electrical knee and ankle prosthesis. *Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2008*, *2008*, 523–528. https://doi.org/10.1109/BIOROB.2008.4762811

Tahir, N. M., & Manap, H. H. (2012). Parkinson Disease gait classification based on machine learning approach. *Journal of Applied Sciences*, *12*(2), 180–185. https://doi.org/10.3923/jas.2012.180.185

Tang, P. C. Y., Ravji, K., Key, J. J., Mahler, D. B., Blume, P. A., & Sumpio, B. (2008). Let Them Walk! Current Prosthesis Options for Leg and Foot Amputees. *Journal of the American College of Surgeons*, *206*(3), 548–560. https://doi.org/10.1016/j.jamcollsurg.2007.10.007

Torrealba, R. R., & Fonseca-Rojas, E. D. (2019). Toward the Development of Knee Prostheses: Review of Current Active Devices. In *Applied Mechanics Reviews* (Vol. 71, Issue 3, p. 030801). American Society of Mechanical Engineers. https://doi.org/10.1115/1.4043323

Torrealba, R. R., Pérez-D'Arpino, C., Cappelletto, J., Fermín-León, L., Fernández-López, G., & Grieco, J. C. (2010). Through the development of a biomechatronic knee prosthesis for transfemoral amputees: Mechanical design and manufacture, human gait characterization, intelligent control strategies and tests. *Proceedings - IEEE International Conference on Robotics and Automation*, 2934–2939. https://doi.org/10.1109/ROBOT.2010.5509674

Weiss, K., Khoshgoftaar, T. M., & Wang, D. D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1). https://doi.org/10.1186/s40537-016-0043-6

Wen, Y., Brandt, A., Liu, M., Huang, H., & Si, J. (2017). Comparing Parallel and Sequential Control Parameter Tuning for a Powered Knee Prosthesis Joint Department of Biomedical Engineering. *IEEE Int., Conf. Sys., Man and Cybern.*, 1716–1721. http://www.smc2017.org/SMC2017_Papers/media/files/0831.pdf

White, R., Agouris, I., Selbie, R. D., & Kirkpatrick, M. (1999). The variability of force platform data in normal and cerebral palsy gait. *Clinical Biomechanics*, *14*(3), 185–192. https://doi.org/10.1016/S0268-0033(99)80003-5

WHO. (2011). *World Report on Disability. Geneva*.

Winter, D. A. (1990). *Biomechanics and Motor Control of Human Movement* (2nd ed.). John Wiley & Sons.

Wise, K. D., Anderson, D. J., Hetke, J. F., Kipke, D. R., & Najafi, K. (2004). Wireless implantable microsystems: High-density electronic interfaces to the nervous system. *Proceedings of the IEEE*, *92*(1), 76–97. https://doi.org/10.1109/JPROC.2003.820544

Yoo, D., Seo, K. H., & Lee, B. C. (2019). The effect of the most common gait perturbations on the compensatory limb's ankle, knee, and hip moments during the first stepping response. *Gait and Posture*, *71*, 98–104. https://doi.org/10.1016/j.gaitpost.2019.04.013

Zhang, F., D'andrea, S. E., Nunnery, M. J., Kay, S. M., & Huang, H. (2011). Towards design of a stumble detection system for artificial legs. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *19*(5), 567–577. https://doi.org/10.1109/TNSRE.2011.2161888

Chapter 9: Appendices

9.1 Appendix A: Simple Rule-Based Control

```
import numpy as np
import pandas as pd

def angles():
    global cax1, cay1, caz1, cax2, cay2, caz2
    ang_x1 = np.degrees(np.arctan(cax1/(np.sqrt(cay1**2+caz1**2)+0.0001)))
    ang_y1 = np.degrees(np.arctan(cay1/(np.sqrt(cax1**2+caz1**2)+0.0001)))
    ang_x2 = np.degrees(np.arctan(cax2/(np.sqrt(cay2**2+caz2**2)+0.0001)))
    ang_y2 = np.degrees(np.arctan(cay2/(np.sqrt(cax2**2+caz2**2)+0.0001)))
    ang_x = ang_x1-ang_x2
    ang_y = ang_y1-ang_y2

    ang = np.mean((ang_x, ang_y))
    return ang

run_name = 'Simple_Control'

subjects = ["S1", "S2", "S4"]
in_dir = "..//Compiled//"
headers =
["time","var1(t)","var2(t)","var3(t)","var4(t)","var5(t)","var6(t)","var7(t)"
,"var8(t)","var9(t)","var10(t)","var11(t)","var12(t)","var13(t)","var14(t)","
var15(t)","var16(t)","var17(t)","var18(t)","sbu","sbo","sw","walking"]

TP = 0
FP = 0
TN = 0
FN = 0

for sub in subjects:
# sub = "S1"

    print(sub)
    data = pd.read_excel(in_dir+sub+".xlsx", usecols=headers).to_numpy()

    times = data[:,0]
    data_x = data[:,1:19]
    data_y = np.stack((np.sum(data[:,19:22], axis=1), data[:,22]), axis=1)

    pred_y = np.zeros(data_y.shape)

    history = np.zeros((21, 12))
    history_long = np.zeros((1200)) #angular velocity
    angle = np.zeros((900))

    stumb_count = 0
    flex = False
    ext = False
    cross = False
    step = 0
    bun = False
    obs = False
    une = False
```

```
    knee_vel_off = 0

    std_win = 20
    old_stamp = 0
    angle_gyr = 0
    angle_acc = 0
    epsilon = 0.025

    for i in range(1,len(times)-1):
        cgx1, cgy1, cgz1, cax1, cay1, caz1, cmx1, cmy1, cmz1, cgx2, cgy2,
cgz2, cax2, cay2, caz2, cmx2, cmy2, cmz2 = data_x[i, :]
        stamp = times[i]
        old_stamp = times[i-1]

        history = np.roll(history, -1, axis=0)
        history_long = np.roll(history_long, -1, axis=0)
        angle = np.roll(angle, 1, axis=0)
        mag_hist = np.roll(history, 1, axis=0)

        history[-1] = cgx1, cgy1, cgz1, cax1, cay1, caz1, cgx2, cgy2, cgz2,
cax2, cay2, caz2
        history_long[-1] = cgz2-cgz1 + knee_vel_off
        area_gyr = sum(history_long[-2:]) / 2 * (stamp-old_stamp)
        old_stamp = stamp

        angle_gyr += area_gyr
        angle_acc = angles()

        angle[0] = epsilon*angle_acc+(1-epsilon)*(angle[1]+area_gyr)
        if angle[0] < 0: angle[0] = 0

        if history_long[-1] > 100:
            flex = True
            cross = False
        if history_long[-1] < -200:
            flex = False
            ext = True
            cross = False
        if history_long[-1] > -20:
            ext = False
            if history_long[-1] < 20: cross = True


        if history_long[-1] > 100 and angle[0] > 25:
            step = 30
        else: step -= 1

        if step < 0: step = 0


        if step > 0:

            #bungee detection
            if not flex and not ext and history_long[-1] < -100:
                bun = True

            #obstacle detection
```

```
            if flex and cross and history_long[-1] > 30:
                obs = True

            #uneven detection
            if angle[0] > 25 and abs(caz2) > 1.5:
                une = True

            if bun or une or obs:
                stumb = True
                stumb_count = 30
                bun = False
                une = False
                obs = False

        stumb_count -= 1

        if stumb_count > 0:
            pred_y[i,0] = 1

        if stumb_count < 0:
            stumb = False
            stumb_count = 0
        if stumb_count == 0:
            pred_y[i,1] = 1

        if pred_y[i,0] == data_y[i,0]:
            if data_y[i,0] == 1: TP += 1
            else: TN += 1
        else:
            if data_y[i,0] == 1: FN += 1
            else: FP += 1

acc = (TP+TN)/(TP+FP+TN+FN)
prec = (TP)/(TP+FP)
rec = (TP)/(TP+FN)
fpr = (FP)/(FP+TN)
fscore = 2*(prec*rec)/(prec+rec)

print("Accuracy: {:.1f}%".format(acc*100))
print("Precision: {:.1f}%".format(prec*100))
print("Recall: {:.1f}%".format(rec*100))
print("FPR: {:.1f}%".format(fpr*100))
print("F-Score: {:.1f}%".format(fscore*100))
```

9.2 APPENDIX B: CONTROL SYSTEM

```
import numpy as np
import math
import tflite_runtime.interpreter as tflite
import board
import busio
import digitalio
import pickle
import time
import threading
from scipy.signal import find_peaks
from periphery import PWM
import argparse
parser =
argparse.ArgumentParser(formatter_class=argparse.ArgumentDefaultsHelpFormatte
r)

class myThread(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
    def run(self):
        tester()

def tester():
    global pred, interpreter, input_details, output_details, history,
interpret, collect, sensor1, sensor2, cgx1, cgy1, cgz1, cax1, cay1, caz1,
cgx2, cgy2, cgz2, cax2, cay2, caz2

    while not end:
        if collect:
            cax1, cay1, caz1 = sensor1.acceleration
            cgx1, cgy1, cgz1 = sensor1.gyro
            cax2, cay2, caz2 = sensor2.acceleration
            cgx2, cgy2, cgz2 = sensor2.gyro
            collect = False
        if interpret:
            input_data = np.expand_dims(history[::2],
axis=0).astype("float32")
            interpreter.set_tensor(input_details[0]['index'],input_data)
            interpreter.invoke()
            pred = interpreter.get_tensor(output_details[0]['index'])[0]
            interpret = False
        time.sleep(.002)

def angles():
    global cax1, cay1, caz1, cax2, cay2, caz2
    ang_x1 = np.degrees(np.arctan(cax1/(np.sqrt(cay1**2+caz1**2)+0.0001)))
    ang_y1 = np.degrees(np.arctan(cay1/(np.sqrt(cax1**2+caz1**2)+0.0001)))
    ang_x2 = np.degrees(np.arctan(cax2/(np.sqrt(cay2**2+caz2**2)+0.0001)))
    ang_y2 = np.degrees(np.arctan(cay2/(np.sqrt(cax2**2+caz2**2)+0.0001)))
    ang_x = ang_x1-ang_x2
    ang_y = ang_y1-ang_y2

    ang = np.mean((ang_x, ang_y))
    return ang
```

```
parser.add_argument("-f","--freq", type=int, help="set frequency in hertz
(default", default=480)
parser.add_argument("-t","--time", type=float, help="set time in minutes",
default=10)
parser.add_argument("-n","--name", help="set file name", default='')
parser.add_argument("-r","--ratio", type=int, help="set how many cycles pass
to collect one IMU sample", default=8)
parser.add_argument("-m","--model_f", help="patient for model file",
default='')
parser.add_argument("-s","--sensor", type=int, help="use LSM6 sensor (default
is LSM9)", default=0)
parser.add_argument("-e","--enable", type=int, help="master enable")
args = parser.parse_args()

if args.sensor: from new_lsm6ds import LSM6DS as lsm
else: import new_lsm9ds1 as lsm

if type(args.enable) != int: print("Need to specify master enable")

master_enable = args.enable
model_file = "Patient_Walk_" + args.model_f
name = args.name
freq = args.freq
tot_time = args.time
dur = int(float(tot_time)*60)
ratio = args.ratio
cycles = dur*freq
dec = 1/freq*1000

#Load LSTM Model
interpreter = tflite.Interpreter(model_path="model_"+model_file+".tflite")
interpreter.allocate_tensors()
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

scaler = pickle.load(open(model_file+'_Scaler.pkl', 'rb'))


print('EMG Fequency = {}Hz; IMU Fequency = {}Hz; Time =
{:2}:{:2}:{:2}'.format(freq, int(freq/ratio), int(tot_time//60),
int(tot_time%60//1), int(tot_time%60%1*60)))

file_name = time.strftime("%y.%m.%d_%I.%M.%S_", time.localtime())+ name +
".csv"

file = open("data/"+file_name, 'w')
file.write("frequency,duration,date,time\n")
file.write("{},{:2}:{:2}:{:2},{}\n".format(freq, int(tot_time//60),
int(tot_time%60//1),
           int(tot_time%60%1*60),time.strftime("%y.%m.%d,%I.%M",
time.localtime())))
file.write("time,g1_x,g1_y,g1_z,a1_x,a1_y,a1_z,g2_x,g2_y,g2_z,a2_x,a2_y,a2_z,
ch1,ch2,ch3,stumble,error,knee angle,prediction,stumb,locking\n")
print("Started. Saving to file "+file_name)

#Sensors, Button, and Servo Declaration
```

```
i2c = busio.I2C(board.SCL, board.SDA, frequency=400000)

if args.sensor:
    sensor1 = lsm(i2c, address=0x6B) #Thigh high
    sensor2 = lsm(i2c, address=0x6A) #Shank ground
else:
    sensor1 = lsm.LSM9DS1_I2C(i2c, mag_address=0x1E, xg_address=0x6B) #Thigh
    sensor2 = lsm.LSM9DS1_I2C(i2c, mag_address=0x1C, xg_address=0x6A) #Shank

button = digitalio.DigitalInOut(board.D24)
button.direction = digitalio.Direction.INPUT

range_low = .9 #ms from spec
range_high = 2.1 #ms from spec
duty_range = range_high - range_low

pwm = PWM(2, 0)
pwm.frequency = 50
pwm.enable()
def servo(pwm, angle, duty_range, range_low):
    pwm.duty_cycle_ns = int(1000000 * ((angle/180 * duty_range) + range_low))
    return


#Tracked Info
stamps = np.trunc(np.arange(dec, dur*1000+81*dec, dec)).astype(int)
history = np.zeros((21, 12))
history_long = np.zeros((1200))
angle = np.zeros((900))
std_hist = np.zeros ((900))
std_peaks = np.empty((1))
out=[]
end = False
interpret = False
collect = False

#Calculation Variables
gait_len = np.zeros((int(cycles)))
error=0
stumb = False
counter = 0
stumb_count = 0
act = True
act_count = 0
locking = False
knee_vel_off = 0
cal1 = True
cal2 = True
area_out = np.zeros((0))
j = 0
i = 0
c = 0
e = 0
k = 0
cgx1, cgy1, cgz1, cax1, cay1, caz1, cgx2, cgy2, cgz2, cax2, cay2, caz2 =
0,0,0,0,0,0,0,0,0,0,0,0
pred = [0, 1]
```

```
ratio_prep = 2
ratio_interp = 2
ratio_react = 0
ratio1 = 4

#Customization Parameters
threshold = .8
pred_offset = 6
pred_sum =  3
cal1_period = int(5*freq/ratio)
cal2_period = int(15*freq/ratio)
std_win = 20
prev = 0
servo_open = 60
servo_closed = 155
servo_move = np.copy(servo_open)
servo(pwm, servo_move, duty_range, range_low)
adjustment = False
old_stamp = 0
angle_gyr = 0
angle_acc = 0
epsilon = 0.025

try:
    thread1 = myThread()
    thread1.start()
    start = time.time()
    while c+1 < cycles:
        stamp = np.round(time.time() - start, 3)
        stamp_trunk = math.trunc(stamp*1000)
        if stamp_trunk > stamps[c]:
            if stamp_trunk > stamps[c+80]:
                print("Got Real Issues Now", stamp_trunk, stamps[c])
                break
            for j in range(1,80):
                if stamp_trunk == stamps[c+j]:
                    e = j
                    c+=j
                    break


        if stamp_trunk == stamps[c]:
            k += 1
            ch0, ch1, ch2 = 0, 0, 0

            if k == ratio: k = 0

            if k == ratio1: collect = True


            if k == ratio_prep:
                # print(stamp)
                i = c // ratio
                history = np.roll(history, -1, axis=0)
                history_long = np.roll(history_long, -1, axis=0)
                angle = np.roll(angle, 1, axis=0)
```

```
                history[-1, :] = scaler.transform([[cgx1, cgy1, cgz1, cax1,
cay1, caz1, cgx2, cgy2, cgz2, cax2, cay2, caz2],[cgx1, cgy1, cgz1, cax1,
cay1, caz1, cgx2, cgy2, cgz2, cax2, cay2, caz2]])[0]
                history_long[-1] = cgz2-cgz1 + knee_vel_off
                area_gyr = sum(history_long[-2:]) / 2 * (stamp-old_stamp)
                old_stamp = stamp
                # if area_long[0] < 0: area_long[0] -= 0.5
                # print(area_long[0])
                angle_gyr += area_gyr
                angle_acc = angles()

                angle[0] = epsilon*angle_acc+(1-epsilon)*(angle[1]+area_gyr)
                if angle[0] < 0: angle[0] = 0


                if i > std_win:
                    std_hist = np.roll(std_hist, -1, axis=0)
                    std_hist[-1] = np.std(history_long[-std_win:])

                peaks = find_peaks(history_long[:], height=100,
distance=30)[0][1:-1]
                # print(history_long[peaks])
                if len(peaks) > 1: gait_len[i] = np.average(np.diff(peaks))


            if k == ratio_interp:
                # pass
                interpret = True

            if k == ratio_react:
                try: std_peak = find_peaks(-std_hist[-300:], height=-20,
distance=30)[0][-1]+i-300
                except : std_peak = np.nan
                if np.diff((std_peaks[-1], std_peak)) > 20:
                    # adjustment = True
                    j += 1
                    std_peaks = np.append(std_peaks, std_peak)


                if pred[0] > threshold and i >= cal1_period:
                    counter+=1
                    prev = 1
                    if counter > pred_offset: counter = pred_offset
                else:
                    if prev == 0: counter-=1
                    prev = 0
                    if counter < 0: counter = 0

                if counter > pred_sum:
                    stumb = True
                    #print(stumb)
                    stumb_count = 20
                elif counter == 0:
                    stumb = False
                    #print(stumb)
                    if stumb_count > 0: stumb_count -= 1
```

```
                if stumb and angle[0] > 10 and history_long[-1] > -200:
                    if master_enable: servo(pwm, servo_closed, duty_range,
range_low)
                    locking = True
                    if act:
                        act_count += 1
                        print(stamp, 'Locking')
                        act = False
                if not stumb and (np.std(history_long[-10:]) < 10) and
angle[0] < 15:
                    if master_enable: servo(pwm, servo_move, duty_range,
range_low)
                    act = True
                    locking = False

                if adjustment:
                    if servo_move > servo_open and max(angle[-180:])<50:
servo_move -= 4
                    if max(angle[-60:])>=60: servo_move += 4
                    adjustment = False


out.append("%.4f,%.4f,%.4f,%.4f,%.4f,%.4f,%.4f,%.4f,%.4f,%.4f,%.4f,%.4f,%.4f,
%i,%i,%i,%i,%.2f,%.2f,%.4f,%i,%i" % (stamp, cgx1, cgy1, cgz1, cax1, cay1,
caz1, cgx2, cgy2, cgz2, cax2, cay2, caz2, ch0, ch1, ch2, button.value, e,
angle[0], pred[0], stumb, locking))

            if e >= 1: error+= int(e // 1)
            elif e % 1 > 0: error+=1
            if stamp_trunk % 60000 == 0: print("Collecting...", stamp_trunk
// 60000, "min")
            e=0
            c+=1

except KeyboardInterrupt:
    print("Cancelation")
except:
    print("Error")

stop = time.time() - start
end = True

time.sleep(.1)

for lines in out:
    file.write(lines + "\n")
print("%i Errors (%i%%), %.1f Errors/s" % (error, int(error/len(out)*100),
int(error/stop)))
print("Done. Saved to file "+file_name)

file.close()
print("Finished")
```

9.3 APPENDIX C: LOCKING EVENT MARKER

```
import matplotlib.pyplot as plt
import matplotlib as mpl
```

```
mpl.rcParams['figure.dpi'] = 300
import numpy as np
import pandas as pd
import os

def find_files(input_directory, end="xlsx"):
    file = []
    name = []
    paths = []
    for path, subdirs, files in os.walk(input_directory):
        for filename in files:
            if str(filename).endswith(end):
                f = os.path.join(path, filename)
                file.append(str(f))
                name.append(filename)
                paths.append(path)
    return file, name, paths


root_dir = "C:\\Users\\lucas\\OneDrive - University of Texas at El
Paso\\Research\\Programming\\Python\\Dissertation\\Data - UTSoutwestern\\"
if not os.path.isdir(root_dir):
    root_dir = "C:\\Users\\ljgaley\\OneDrive - University of Texas at El
Paso\\Research\\Programming\\Python\\Dissertation\\Data - UTSoutwestern\\"

data_dir = root_dir + "ReDone_Processed"
notes_dir = root_dir + "Patient Notes"
# folder = "\\P3\\M3"

pats = ["P1", "P2", "P3", "P4", "P6"]
# pats = ["P6"]
# knees = ["M3", "MPK", "GKnee"]
knees = ["GKnee"] #only here are we actually having reactions to events
stumbles = ["bun", "obs", "une"]
# stumbles = ["bun"] #only got bun the first time

saving = True

catch = []
beg = 60

for pat in pats:
    for knee in knees:
        folder = "\\"+pat+"\\"+knee

        files, names, paths = find_files(data_dir+folder)

        for j, f in enumerate(files):
            # file = 5
            # print(names[j])
            data0 = pd.read_excel(f, index_col='time')
            # plt.plot(data['stumble'])

            for stumble in stumbles:
                if stumble in names[j]:
                    file_name = ("_").join(names[j].split("_")[2:])[:-5]
                    stumb = 1
```

```
                    prev = 0
                    for i, num in enumerate(data0['locking']):
                        if num == 1 and prev == 0:
                            if i < 60: beg = i
                            for k in range(300):
                                if data0['locking'].iloc[i+k] == 0: break
                                else: pass
                            end_time = data0.index[i+k]
                            data = data0.iloc[i-beg:i+120]



                            fig, ax = plt.subplots()
                            ax.set_xlim(data.index[0], data.index[-1])
                            ax2 = ax.twinx()
                            ax3 = ax.twinx()
                            ax4 = ax.twinx()

                            ax.set_title(file_name+" locking "+str(stumb))
                            ax.set_xlabel("Time (s)")
                            ax.set_ylabel("Angular Velocity (deg/s)", color =
"tab:orange")
                            ax2.set_ylabel("Angle (deg)", color = "tab:blue")
                            ax3.axis('off')
                            ax4.axis('off')


                            # ax3.axvline(data.index[60], color = 'black',
label='Actual', linewidth=2, linestyle=':')

                            ax3.plot(data.index, data['prediction']*.8, color
= "tab:green", label='Predicted', zorder=-1)
                            ax3.plot(data.index, data['locking']*.90, "_",
color = "m", label='Locking', zorder=0)
                            ax3.plot(data.index, data['stumble']*.3, color =
'black', label='Actual', linewidth=2, linestyle=':', zorder=0)


                            lns1 = ax.plot(data.index, data['g2_z']-
data['g1_z'], color = "tab:orange", label='Knee Velocity')
                            ax.plot(np.zeros((len(data.index))), color =
"black", linewidth = .5, zorder=1)
                            lns2 = ax2.plot(data.index, data['knee angle'],
color = "tab:blue", label='Knee Angle')

                            ax.set_ylim(-550,800)
                            ax2.set_ylim(-75, 110)
                            # ax2.set_ylim(-0, 50)
                            ax3.set_ylim(-5, 1)


                            lns = lns1+lns2
                            labs = [l.get_label() for l in lns]
                            ax3.legend(bbox_to_anchor=(1,0), loc="lower
right", framealpha=0.9, title="Stumbles")
                            ax4.legend(lns, labs, bbox_to_anchor=(0,0),
loc="lower left", framealpha=0.9)
```

```
                            plt.show()

catch.append([pat,file_name,stumble,stumb,input("Classify this locking:
"),data.index[beg], end_time])
                            #
catch.append([pat,file_name,stumble,stumb,1,data.index[beg], end_time])
                            # 1 true lock
                            # 2 false lock
                            # 3 repeat lock (within window of true)
                            # 4 not real lock (start or stop of file)


                            stumb += 1
                            beg = 60

                        prev = num
```

9.4 APPENDIX D: STEP EVENT MARKER

```
import matplotlib.pyplot as plt
import matplotlib as mpl
mpl.rcParams['figure.dpi'] = 300
import numpy as np
import pandas as pd
import os
from scipy.signal import find_peaks

def find_files(input_directory, end="xlsx"):
    file = []
    name = []
    paths = []
    for path, subdirs, files in os.walk(input_directory):
        for filename in files:
            if str(filename).endswith(end):
                f = os.path.join(path, filename)
                file.append(str(f))
                name.append(filename)
                paths.append(path)
    return file, name, paths

def plotter(data, file_name):
    fig, ax = plt.subplots()
    ax.set_xlim(data.index[0], data.index[-1])
    ax2 = ax.twinx()
    ax3 = ax.twinx()
    ax4 = ax.twinx()

    ax.set_title(file_name)
    ax.set_xlabel("Time (s)")
    ax.set_ylabel("Angular Velocity (deg/s)", color = "tab:orange")
    ax2.set_ylabel("Angle (deg)", color = "tab:blue")
    ax3.axis('off')
    ax4.axis('off')

    if (data['stumble'] == .5).any(): color = 'orange'
    else: color = 'black'

    ax.axvline(data.index[60], color = 'red', label='Current', linewidth=2,
linestyle=':')

    ax3.plot(data.index, data['prediction']*.8, color = "tab:green",
label='Predicted', zorder=-1)
    ax3.plot(data.index, data['locking']*.90, "_", color = "m",
label='Locking', zorder=0)
    ax3.plot(data.index, data['stumble']*.5, color = color, label='Actual',
linewidth=2, linestyle=':', zorder=0)

    lns1 = ax.plot(data.index, data['g2_z']-data['g1_z'], color =
"tab:orange", label='Knee Velocity')
    ax.plot(np.zeros((len(data.index))), color = "black", linewidth = .5,
zorder=1)
    lns2 = ax2.plot(data.index, data['knee angle'], color = "tab:blue",
label='Knee Angle')
```

```
    ax.set_ylim(-550,800)
    ax2.set_ylim(-75, 110)
    # ax2.set_ylim(-0, 50)
    ax3.set_ylim(-5, 1)


    lns = lns1+lns2
    labs = [l.get_label() for l in lns]
    ax3.legend(bbox_to_anchor=(1,0), loc="lower right", framealpha=0.9,
title="Stumbles")
    ax4.legend(lns, labs, bbox_to_anchor=(0,0), loc="lower left",
framealpha=0.9)

    plt.show()
    return

root_dir = "C:\\Users\\lucas\\OneDrive - University of Texas at El
Paso\\Research\\Programming\\Python\\Dissertation\\Data - UTSoutwestern\\"
if not os.path.isdir(root_dir):
    root_dir = "C:\\Users\\ljgaley\\OneDrive - University of Texas at El
Paso\\Research\\Programming\\Python\\Dissertation\\Data - UTSoutwestern\\"

data_dir = root_dir + "ReDone_Processed"
notes_dir = root_dir + "Patient Notes"
# folder = "\\P3\\M3"

pats = ["P1", "P2", "P3", "P4", "P6"]
pats = ["P1"]
# knees = ["M3", "MPK", "GKnee"]
knees = ["GKnee"] #only here are we actually having reactions to events
stumbles = ["bun", "obs", "une"]
# stumbles = ["bun"] #only got bun the first time


data_dir = root_dir + "Round3_Processed"
pats = ["P2", "P3", "P4"]
knees = ["Off", "On"]
stumbles = ["bun", "obs", "une"]


saving = True

catch = []
beg = 60

for pat in pats:
    for knee in knees:
        folder = "\\"+pat+"\\"+knee

        files, names, paths = find_files(data_dir+folder)

        for j, f in enumerate(files):
            # file = 5
            # print(names[j])
            data0 = pd.read_excel(f, index_col='time')
            # plt.plot(data['stumble'])
```

```
            for stumble in stumbles:
                if stumble in names[j]:
                    file_name = ("_").join(names[j].split("_")[2:])[:-5]
                    stumb = 1
                    prev = 0

                    angles = data0['knee angle']
                    peaks = find_peaks(angles, height=20, distance=25)


                    for num in peaks[0]:
                        beg = 60
                        if num < 60: beg = num

                        file_name = "{} {} {}
{:.1f}s".format(pat,knee,stumble,data0.index[num])


                        if (data0["locking"].iloc[num-beg:num+60] == 1).any()
or (data0["stumble"].iloc[num-beg:num+60] == 1).any():
                            plotter(data0.iloc[num-beg:num+120], file_name)

catch.append([pat,knee,stumble,data0.index[num],input("Classify this step:
")])
                            # catch.append([pat,stumble,data0.index[num],1])
                        else:
                            catch.append([pat,stumble,data0.index[num],1])

                        # 1 step empty
                        # 2 step stumble marked and predicted (R or F)
                        # 3 step stumble marked and predicted (U or A)
                        # 4 step stumble marked
                        # 5 step stumble predicted
                        # 6 discard (beginning, end)
```

9.5 APPENDIX E: DEPLOYED LEARNING

```
# %%
# Initialize
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

import os
import pickle
os.environ["TF_CPP_MIN_LOG_LEVEL"] = "3"
import tensorflow as tf

from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.models import load_model
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import LSTM
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import TensorBoard
from datetime import datetime

gpus = tf.config.experimental.list_physical_devices('GPU')
if gpus:
  try:
    # Currently, memory growth needs to be the same across GPUs
    for gpu in gpus:
      tf.config.experimental.set_memory_growth(gpu, True)
    logical_gpus = tf.config.experimental.list_logical_devices('GPU')
    print(len(gpus), "Physical GPUs,", len(logical_gpus), "Logical GPUs")
  except RuntimeError as e:
    # Memory growth must be set before GPUs have been initialized
    print(e)

def plotter(i, data, act_col, pred_col):
    start = i-200
    if start < 0: start = 0
    stop = i+200


    x_axis = np.arange(start, stop)
    data_stumb = np.ma.masked_where(data[start:stop, act_col] < 1,
data[start:stop, act_col])*480
    data_stumb2 = np.ma.masked_where(data[start:stop, pred_col] < 1,
data[start:stop, pred_col])*475

    fig, ax = plt.subplots()
    ax.set_ylim(-500, 500)
    ax.plot(x_axis, data[start:stop, 0], 'b', label="Knee Angular Velocity")
    ax.plot(x_axis, data_stumb, 'r', label="Stumble Actual")
    ax.plot(x_axis, data_stumb2, 'orange', label="Stumble Predicted")
    ax.axvline(i, color='orange', linewidth=.5)
    ax.legend(loc='lower right')
    plt.show()
```

```
# %%
# load dataset
global class_total
#Both load_data and load_npy will load the walking data
load_data = False
load_npy = True
transfer = False
if load_data and load_npy:
    input("Disable either load_data or load_npy")
load_prev_model = False
prev_model = "5.0.0"
run_name = "5.0.1"
note = "Fine Tuning"
com_stumble = True
sep_subjects = True
modified = True
deleting = True
epochs = 30
learning_rate = 0.001
dropout = .1
# train_array = np.arange(1, 19)
scaler = MinMaxScaler(feature_range=(0, 1))
scaler_save = MinMaxScaler(feature_range=(0, 1))
n_hist = 11
n_features = 18
# Deleting mag cols
delete_cols = np.hstack([np.arange(6, n_hist*n_features, 9), np.arange(7,
n_hist*n_features, 9), np.arange(8, n_hist*n_features, 9)])
class_total = 4
events = np.array(["Bungie", "Obstacle", "Uneven", "Walking"])
data_mod = "_art"
preq = "Mod_"

train_files = ["S1_train_art", "S1_val_art", "S2_train_art", "S2_val_art",
"S4_train_art", "S4_val_art"]
val_files = ["S1_test", "S2_test", "S4_test"]

if modified:
    train_files = [preq + s for s in train_files]
    val_files = [preq + s for s in val_files]

# if modified:
#     train_files = ["Mod_" + loo + s for s in train_files]
#     val_files = ["Mod_" + loo + s for s in val_files]
#     test_files = ["Mod_" + loo + s for s in test_files]


if deleting: n_features -= int(len(delete_cols)/n_hist)

# train_files = [s+"_train"+data_mod for s in train_files]
# val_files = [s+data_mod for s in val_files]
# test_files = [s+"_test"+data_mod for s in test_files]

logs_weights_dir = "C:\\Users\\lucas\\Documents\\Learning\Machine
Learning\\Dissertation\\Logs and Weights"
```

```
logs_weights_dir = "C:\\Users\\ljgaley\\Documents\\Learning\Machine
Learning\\Dissertation\\Logs and Weights"
in_dir = "Compiled\\"
out_dir = "Results\\"
send_dir = "Deploy\\On Device\\Sending 2\\"
walking_data = "Patient_Walk"

# %%
# Load Data
print('Loading Patient Walk')
walk = pd.read_excel(in_dir+walking_data+".xlsx",
index_col='time').to_numpy()
n_obs = n_hist * n_features
x_walk_u, y_walk = walk[:, :n_obs], walk[:, -class_total:]

if com_stumble: y_walk = np.stack([np.abs(y_walk[:,-1]-1), y_walk[:, -1]],
axis=1)

if load_data:
    print('Loading New Data')
    train = []
    val = []

    for i in range(len(train_files)):
        print("Loading Train File: "+train_files[i])
        train.append(pd.read_excel(in_dir+train_files[i]+".xlsx",
index_col='time'))
    train = np.vstack(train)


    for i in range(len(val_files)):
        print("Loading Validation File: "+val_files[i])
        val.append(pd.read_excel(in_dir+val_files[i]+".xlsx",
index_col='time'))
    val = np.vstack(val)


    if deleting:
        train = np.delete(train, delete_cols, 1)
        val = np.delete(val, delete_cols, 1)

    # split into input and outputs
    x_train_u, y_train = train[:, :n_obs], train[:, -class_total:]

    x_val_u, y_val = val[:, :n_obs], val[:, -class_total:]


    if com_stumble:
        y_train = np.stack([np.abs(y_train[:,-1]-1), y_train[:, -1]], axis=1)
        y_val = np.stack([np.abs(y_val[:,-1]-1), y_val[:, -1]], axis=1)

    descr = np.sum(y_train, axis=0)/np.sum(y_train)

    x_train_u2 = np.vstack([x_train_u, x_walk_u])

    # Scale data according to training set
    _ = scaler_save.fit(x_train_u2[:,:n_features])
```

```
    x_train = scaler.fit_transform(x_train_u2)
    x_val = scaler.transform(x_val_u)

    # reshape input to be 3D [samples, timesteps, features]
    x_train = x_train.reshape((x_train.shape[0], n_hist, n_features))
    x_val = x_val.reshape((x_val.shape[0], n_hist, n_features))

    np.save(in_dir+"Deploy - x_train_u.npy", x_train_u)
    np.save(in_dir+"Deploy - x_val_u.npy", x_val_u)

    np.save(in_dir+"Deploy - y_train.npy", y_train)
    np.save(in_dir+"Deploy - y_val.npy", y_val)
    pickle.dump(scaler_save, open(out_dir+walking_data+'_Scaler.pkl', 'wb'))
    y_train = np.vstack([y_train, y_walk])

    print('Data Loaded')

if load_npy:
    x_train_u = np.load(in_dir+"Deploy - x_train_u.npy")
    x_val_u = np.load(in_dir+"Deploy - x_val_u.npy")

    x_train_u2 = np.vstack([x_train_u, x_walk_u])

    # Scale data according to training set
    _ = scaler_save.fit(x_train_u2[:,:n_features])
    x_train = scaler.fit_transform(x_train_u2)
    x_val = scaler.transform(x_val_u)

    # reshape input to be 3D [samples, timesteps, features]
    x_train = x_train.reshape((x_train.shape[0], n_hist, n_features))
    x_val = x_val.reshape((x_val.shape[0], n_hist, n_features))

    y_train = np.load(in_dir+"Deploy - y_train.npy")
    y_train = np.vstack([y_train, y_walk])
    y_val = np.load(in_dir+"Deploy - y_val.npy")
    pickle.dump(scaler_save, open(send_dir+walking_data+'_Scaler.pkl', 'wb'))

    print('Npy Data Loaded')

if com_stumble:
    class_total = 2
    events = np.array(["Stumble", "Walking"])

metric_list = ['accuracy']
# for i in range(class_total):
#     metric_list.append(Precision(class_id=i, name='Precision_'+ events[i]))

# %%
# Learning
# Model
model = Sequential()
model.add(LSTM(200, return_sequences=True, input_shape=(x_train.shape[1],
x_train.shape[2])))
model.add(LSTM(50))
model.add(Dense(300))
model.add(Dropout(dropout))
model.add(Dense(class_total, activation='softmax'))
```

```
filepath=logs_weights_dir+"\\Weights\\model_improve_{epoch:02d}.h5"
checkpoint = ModelCheckpoint(filepath, monitor='val_loss', verbose=0,
save_best_only=True, save_freq='epoch')
logdir = logs_weights_dir+"\\Logs\\" + datetime.now().strftime("%Y%m%d-
%H%M%S")
tensorboard_callback = TensorBoard(log_dir=logdir, profile_batch=0,
update_freq='epoch')
stopping = EarlyStopping(monitor='val_loss', patience=25,
restore_best_weights=True)
callback_list = [checkpoint, tensorboard_callback]

if not load_data: print('Previous Data Reused')
print('Current Run: '+run_name)
if load_prev_model or transfer:
    model = load_model('Weights\\model_'+prev_model+'.h5')
    print("Loaded Previous Model: "+prev_model)

    model.layers[0].trainable = True
    model.layers[1].trainable = True


    if transfer:
        # model.layers[0].trainable = False
        # model.layers[1].trainable = False
        model2 = Sequential()

        for layer in model.layers[:-1]:
            model2.add(layer)
        # model2.add(Dense(class_total, activation='softmax'))
        model2.add(Dense(class_total, activation='softmax'))
        model = model2
        print(model.summary())

model.compile(loss='categorical_crossentropy',
optimizer=RMSprop(centered=True, learning_rate=learning_rate),
metrics=[metric_list])

print(model.summary())
history = model.fit(x_train, y_train, epochs=epochs, batch_size=400,
validation_data=(x_val, y_val), callbacks=callback_list, verbose=1,
shuffle=True)
model.save('Weights\\model_'+run_name+'.h5')
print('\n', "Finished Training Run: "+run_name)

# %%
# Saving

converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_model = converter.convert()
open(send_dir+'model_'+walking_data+'.tflite', "wb").write(tflite_model)

# %%
# Plotting
plot_list = list(history.history)
plot_num = int(len(plot_list)/2)
```

```
# Set up plots
fig, ax = plt.subplots(plot_num,1,figsize=((10,2*plot_num+5)))

# Plot each set of metrics
for i in range(plot_num):
    ax[i].plot(history.history[plot_list[i]])
    ax[i].plot(history.history[plot_list[i+plot_num]], ':')
    ax[i].set_ylim([0,1])
    ax[i].set_title('Model '+plot_list[i])
    ax[i].set_ylabel(plot_list[i])
    ax[i].set_xlabel('Epoch')
    ax[i].legend(['Train', 'Validation'], loc='upper left')
plt.tight_layout()
plt.savefig(out_dir+run_name+'_Graph.png', bbox_inches='tight', dpi=150)
plt.show()
```

9.6 APPENDIX F: STUMBLE COUNTS SUMMARIZED

Table 9.1: Counts of stumble inductions shown by patient, by knee, and by mode.

Patient	P1									P2									P3									P4									P6								
Knee	M3			Gknee			MPK			M3			Gknee			MPK			M3			Gknee			MPK			M3			Gknee			MPK			M3			Gknee			MPK		
Mode	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une	bun	obs	une
Unaffected Gait	0	1	0	5	0	0	4	4	0	2	0	0	3	0	2	3	0	0	4	0	0	7	1	0	5	2	0	3	0	0	1	0	0	4	0	0	1	2	1	3	0	1	4	0	0
Affected Gait	3	3	0	9	5	2	3	4	3	7	2	4	3	5	5	4	4	7	4	6	6	4	6	5	2	4	5	4	13	0	9	10	2	6	7	6	3	1	5	3	4	6	3	6	4
Recovery	5	4	1	2	5	7	4	3	8	2	5	4	7	6	2	3	4	3	1	5	4	2	5	1	1	3	3	0	7	1	1	1	5	3	3	1	1	2	2	5	7	3	2	5	1
Fall	4	4	10	0	0	1	2	1	1	0	3	2	1	0	4	3	3	1	2	0	2	0	0	0	4	1	0	2	0	4	0	0	4	0	1	2	5	5	0	1	0	0	4	1	2
Ineffective Induced	3	4	0	14	5	2	7	8	3	9	2	4	6	5	7	7	4	7	8	6	6	11	7	5	7	6	5	7	13	0	10	10	2	10	7	6	4	3	6	6	4	7	7	6	4
Effective Induced	9	8	11	2	5	8	6	4	9	2	8	6	8	6	6	6	7	4	3	5	6	2	5	1	5	4	3	2	7	5	1	1	9	3	4	3	6	7	2	6	7	3	6	6	3
Total Induced	**12**	**12**	**11**	**16**	**10**	**10**	**13**	**12**	**12**	**11**	**10**	**10**	**14**	**11**	**13**	**13**	**11**	**11**	**11**	**11**	**12**	**13**	**12**	**6**	**12**	**10**	**8**	**9**	**20**	**5**	**11**	**11**	**11**	**13**	**11**	**9**	**10**	**10**	**8**	**12**	**11**	**10**	**13**	**12**	**7**

Table 9.2: Counts of stumble inductions shown by patient and by knee. Mode counts have been summed.

Patient	P1			P2			P3			P4			P6		
Knee	M3	Gknee	MPK	M3	Gknee	MPK	M3	Gknee	MPK	M3	Gknee	MPK	M3	Gknee	MPK
Unaffected Gait	1	5	8	2	5	3	4	8	7	3	1	4	4	4	4
Affected Gait	6	16	10	13	13	15	16	15	11	17	21	19	9	13	13
Recovery	10	14	15	11	15	10	10	8	7	8	7	7	5	15	8
Fall	18	1	4	5	5	7	4	0	5	6	4	3	10	1	7
Ineffective Induced	7	21	18	15	18	18	20	23	18	20	22	23	13	17	17
Effective Induced	28	15	19	16	20	17	14	8	12	14	11	10	15	16	15
Total Induced	**35**	**36**	**37**	**31**	**38**	**35**	**34**	**31**	**30**	**34**	**33**	**33**	**28**	**33**	**32**

Chapter 10: Vita

Lucas Jonathan Galey was the firstborn son of Tom and Gabi Galey. His mother homeschooled him through the 10th grade, and he graduated from Cassville High School in Missouri in 2010. Lucas earned his Bachelor of Science, Magna Cum Laude, with a Biomedical Concentration from LeTourneau University in May 2014. In Fall of 2014, Lucas followed his professor, Dr. Roger Gonzalez, from LeTourneau to The University of Texas at El Paso; and began pursuing his M.S. in Biomedical Engineering. He completed the M.S. in the fall of 2016 and stayed to complete his Ph.D. in Biomedical Engineering.

Lucas has one publication, "Design and Initial Evaluation of a Low-Cost Microprocessor-Controlled Above-Knee Prosthesis: A Case Report of 2 Patients", in the journal *Prosthesis* (Volume 4, 2022). He worked as a TA for the Engineering Education and Leadership capstone design course from Fall 2017 to Spring 2022, which included teaching a Friday lecture. In Fall 2022 and Spring 2023, he helped coteach freshmen engineering courses in Engineering Education and Leadership; and taught "Modeling and Simulation" and "Engineering Measurements".

Since the start of his graduate career, Lucas has helped out with LIMBS International, a prosthetic supply nonprofit. In 2017, he joined as a volunteer research associate; and became the Interim Director during the fall of 2021. Since the beginning of 2022, he has been the Executive Director of LIMBS International.

Contact Information: lucasgaley@hotmail.com