Title
Development of a Human Machine Interface for a Wearable Exoskeleton for Users with Spinal Cord Injury

Author
Strausser, Katherine Ann

Publication Date
2011

Peer reviewed|Thesis/dissertation

Development of a Human Machine Interface for a Wearable Exoskeleton for Users with Spinal Cord Injury

By

Katherine Ann Strausser

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Homayoon Kazerooni, Chair
Professor J. Karl Hedrick
Professor Stephen Derenzo

Fall 2011

ABSTRACT

Development of a Human Machine Interface for a Wearable Exoskeleton for Users with Spinal Cord Injury

By

Katherine Ann Strausser

Doctor of Philosophy in Engineering – Mechanical Engineering

University of California, Berkeley

Professor Homayoon Kazerooni, Chair

For millions of individuals, a spinal cord injury has taken away their ability to walk. While wheelchairs and leg braces offer mobility options, none offer a means to stand up and walk. For these individuals, secondary injuries can be prevalent, and special care must be taken to avoid the pain and cost of pressure sores, urinary tract infections, and other such ailments. Furthermore, there is an emotional benefit to being able to stand and walk. Events such as choosing your own seat at the theater or sports game, walking your daughter down the aisle at her wedding, reaching the pasta on the top shelf at the grocery store, or checking out of a hotel at the main counter, are taken for granted by those who can walk, but for those who use a wheelchair for mobility, these are stark reminders of the limitations of the chair.

Exoskeletons provide a means by which these individuals can get up again and walk. They offer power joints and a support for the body so that a user with a spinal cord injury can rely on the robot's power to replace what their body no longer provides. While the architecture and design of such an exoskeleton is complex, the control of the exoskeleton offers numerous challenges.

This thesis presents the development and testing of a method to allow the user to communicate his desired motion to the robot. For an exoskeleton to truly provide freedom for the user, the user must be able to operate the exoskeleton independently. To do this, the exoskeleton must know what the user wants to do and when and then decide if that maneuver is safe. The user communicates his desired action to the exoskeleton using the Human Machine Interface (HMI).

This thesis describes development of the hardware and software for the HMI beginning with the conception of the structure of the HMI based on end-user surveys and observations of users. The hardware was then developed to determine the state transitions and the software was written to determine desired state changes. The Human Machine Interface was then verified using a mockup to test and then was tested on the eLEGS exoskeleton. The software was verified through experiments and theoretically using classifiers. The Human Machine Interface was tested by subjects with a wide range of injuries and abilities to ensure that it performed safely for all users. Based on experience with the Human Machine Interface, improvements in robustness and usability were made.

This thesis also presents the development of some of the continuous controllers used to achieve the sitting and standing motions. While traditional control strategies rely on models, control of exoskeletons includes a human in the loop, which can be a sizeable disturbance. Therefore, the controller development must be robust to this disturbance and also take into account the comfort and safety of the user.

The results presented here show numerous spinal cord injury patients of varying levels and completeness able to ambulate independently using the HMI developed for eLEGS. They are able to walk, sit, and stand naturally, thus providing wheelchair users a viable means of walking again.

TABLE OF CONTENTS

Abstract 1

Acknowledgements iv

1 Introduction 1

1.1 Motivation 1

1.2 Major Contributions and Outline 1

2 Background 3

2.1 Spinal Cord Injury 3

2.2 Spinal Cord Injury Treatment Options 5

2.3 Current Gait Training Devices 5

2.3.1 Manual Gait Training & Braces 6

2.3.2 Treadmill-Based Robots 8

2.3.3 Mobile Robots 9

3 Biomechanics 13

4 System Description 16

4.1 MedX 16

4.2 eLEGS 17

4.3 Mechanical Components 18

4.4 Interface to User 19

5 State Identification 21

5.1 PT Mode 21

5.2 HMI Mode 21

5.3 Existing Human Machine Interfaces 22

6 End User Surveys 24

6.1 HMI System Requirements 26

6.2 HMI Proposed Concepts 27

6.2.1 Voice Recognition 27

6.2.2 External sensors- crutch sensor 27

6.2.3 On-board Sensors 28

6.2.4 External Sensors- Arm Sensor 28

7 HMI Hardware 29
7.1 Crutch Force Sensor 29
7.1.1 Specifications 29
7.1.2 Design 29
7.1.3 Analysis and Testing 31
7.2 Arm Angle Sensor 32
7.2.1 Arm Sensor Calibration 34
7.2.2 Arm Sensor Testing 36
7.3 Foot Sensors 39
7.4 Angle Sensors 41
8 HMI Software 42
8.1 Finite State Machine 42
8.1.1 Walking 43
8.1.2 Start Walk & End Walk 44
8.1.3 Sit & Stand 45
8.2 Initial Testing 46
8.3 Medical Exoskeleton Testing 49
8.4 Preliminary SCI Subject Testing 50
8.5 Improvements to the HMI Sensors 55
9 Classification for Verification of Sensor Choice 56
9.1 Support Vector Machine (SVM) 56
9.2 Development of the SVM for the HMI 56
10 Clinical Testing 62
11 Adaptive Threshold 65
12 Extensions: Step Length Adjustment 68
12.1 Training Procedure 69
12.2 ε-Greedy Algorithm 70
12.2.1 Greedy Search 70
12.2.2 ε-Greedy Search 70
12.2.3 Results 71

12.3 Non-Linear Regression ... 72
12.3.1 Non-Linear Regression Algorithm ... 72
12.3.2 Results ... 72
12.3.3 Linear Approximation ... 74
12.4 Conclusion ... 75
13 Continuous Controls ... 77
13.1 Control architecture ... 77
13.2 Dynamic model ... 78
13.3 Standing up ... 80
13.4 Sitting Down ... 83
14 Conclusion ... 88
14.1 Future Work ... 88
Appendix A: Linear Potentiometer Data Sheet ... 90
Appendix B: Standing & Sitting Software ... 92

ACKNOWLEDGEMENTS

Throughout my graduate career, there have been many who have offered their support, advice, knowledge, and help. Without these people, this dissertation would not be possible. Professor Kazerooni has been a constant supporter of my graduate work. I am privileged to have been able to work with such a brilliant professor who is constantly seeking new and innovative projects that will change the world. I would like to thank Professor Kazerooni for his advice, guidance, and the opportunity to be a part of this project and his lab. It was truly an amazing experience.

The development of eLEGS was truly a team endeavor. I am grateful to the entire eLEGS team, especially to Tim Swift and Adam Zoss. Thank you for every discussion we had around a whiteboard, for all of the hours of testing, and for helping me fix everything that I broke. I could not have asked for better teammates.

I would also like to thank the other members of the Human Engineering & Robotics Lab for their years of support and camaraderie. I am honored to have worked with all of you and look forward to the amazing work that I know will continue coming from the lab.

Berkeley Bionics has also been a constant source of support throughout my graduate work. I am grateful for the opportunity to work with such talented engineers. I also would like to thank all of our test pilots without whom this work would not be possible. Thank you for your willingness to help us with our testing and for all of the feedback you have given.

To all of those who supported me throughout my graduate career, including my housemates and my friends: thank you. Your intellect and dedication to your work continually inspire me. I am forever grateful to you for making graduate school such an enjoyable experience.

And finally, I thank my family. I am grateful to have such a wonderful brother with whom I am proud to share the title of Berkeley Alumni. Thank you, Matthew, for allowing me to share your campus for a bit. Go Bears! I have had the privilege of the unwavering support of my parents throughout my life. I could not have made it to this point without their advice, guidance, and example. Thank you for challenging me to strive for higher goals, for being there to support me in my successes and failures, and for editing my papers.

1 INTRODUCTION

1.1 MOTIVATION

Spinal cord injuries limit the mobility of thousands of people. 250,000 Americans have spinal cord injuries with approximately 11,000 more each year[1]. Most spinal cord injuries result from trauma, but others causes include spinal cyst, infection, and tumors. For these individuals, daily tasks that most take for granted can become difficult if not impossible. Reaching for the top shelf in a grocery store or having a conversation with a friend eye-to-eye become challenges. Furthermore, there are also health risks involved with spinal cord injury including blood clots, osteoporosis, and pressure sores. These secondary injuries increase healthcare costs in hospital stays and treatments. These can be minimized by correct padding, stretching, and using a standing frame. Walking can also be therapeutic and help prevent many of these injuries since walking improves circulation, loads the bones, and moves the muscles through their range of motion.

1.2 MAJOR CONTRIBUTIONS AND OUTLINE

While wheelchairs provide mobility, there are very few devices that allow a paralyzed individual to walk. The device presented here, eLEGS, allows users to stand and walk using a natural gait. However, just walking is insufficient to provide true mobility. This thesis presents a method by which the user can independently initiate steps while using eLEGS, known as the Human Machine Interface. This novel approach, which uses natural gestures, is comprised of both hardware and software developed for the application. The selection of sensors and software is verified both analytically and in practice. The method was tested in the laboratory and in the clinic with users with spinal cord injuries to show its safety and efficacy. Additional improvements to this method to increase robustness and functionality were also developed.

The exoskeleton must also be able to accomplish additional maneuvers such as sitting and standing in order to be useful in daily living and provide true mobility. Therefore, a new trajectory and controller were developed to allow a user to sit and stand in the exoskeleton. This was also tested on numerous users.

For a user with a spinal cord injury, the independence gained from a mobile exoskeleton that can be independently used to walk, sit, or stand is life changing. eLEGS, with the Human Machine Interface and the ability to naturally walk, sit, and stand, provides mobility and independence.

Chapter 2 and 3 give an overview of biomechanics and existing exoskeletons and human machine interfaces.   These foundational pieces form the basis for the development of the HMI presented in this thesis and in the building of eLEGS.

In Chapter 4, I describe the exoskeleton, eLEGS. I helped develop this exoskeleton with my colleagues at University of California, Berkeley and Berkeley Bionics. This chapter goes into detail describing the initial prototype and eLEGS as well as some of the key features we implemented during our development with regards to interfacing to the user.

The Human Machine Interface is described in Chapter 5-9. This is the main focus of my research and is my own work. I utilized end-user feedback and analyzed numerous initial concepts, and I developed the hardware and software for the Human Machine Interface for eLEGS. I validated these design decisions with control theory based in finite state machines and classification, and I implemented this design into the exoskeleton for testing (which is presented in Chapter 10).

Chapters 11 and 12 present some of the advanced control concepts which I have developed for the Human Machine Interface. These include implementing an additional filter to the sensor data to improve the robustness of the system. I also apply traditional learning methods to this specific application to increase the robustness and performance.

Finally, in chapters 13, I present my work on the continuous controls for eLEGS. While other states were developed by my colleagues, I developed the methods for sitting and standing. These controllers, while based in traditional control theory, must be robust to unknown inputs by including a human in the loop. This controller has been extensively tested with a variety of users, showing the efficacy and safety of the method.

2 BACKGROUND

2.1 SPINAL CORD INJURY

Spinal cord injuries are classified in a number of means. The first is the level of the injury. For those classified as paraplegics, the injury affects only the lower body. Quadriplegia, on the other hand, affects the arms and legs[2].

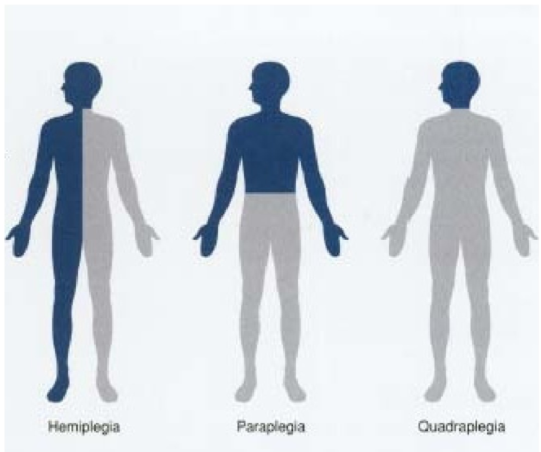


Figure 2-1. Types of Paralysis - grey area indicates area of paralysis (www.humanillnesses.com)

The level of injury can be more specifically described by indicating the vertebra injured. The "C" indicates a cervical vertebra, "T" is thoracic, "L" is lumbar, and "S" is sacral. The number indicates which number vertebrae in that grouping with 1 being the highest on the spinal cord.

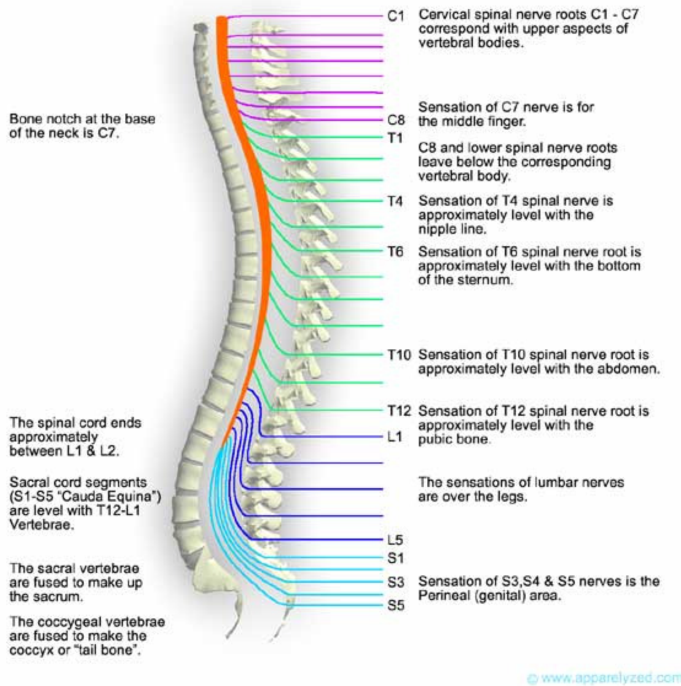


Figure 2-2. The vertebrae of the spinal column are numbered with C1 at the top and S5 at the bottom. This figure shows the muscles that are affected by the nerves at each level.

The completeness is indicated on the American Spinal Injury Association (ASIA) scale. A indicates complete paralysis with no motor or sensory function preserved at S4-S5, B is preserved sensory but not motor function at S4-S5, C is reduced motor function below the level of the injury, D is slightly reduced motor function below the level of the injury, and E is normal sensory and motor function.

For spinal cord injury patients, secondary injuries, such as bed sores, spasticity, respiratory dysfunction, osteoporosis, and fractures, threaten their well-being. These injuries occur due to their inability to move around or stand up to increase blood flow and relieve pressure points. By maintaining the ability to move, through exercise and rehabilitation, and decreasing the amount of time spent in a wheelchair, many of these can be avoided. An exoskeleton will allow the patient to move around as well as remain upright, increasing blood flow to the lower body.

Though many of the discussions and testing done discussed here focus on spinal cord injury patients, users with other gait disorders are also possible end users. Spinal cysts, Guillain-Barre syndrome, cerebral palsy, and stroke are among the target group for the device.

2.2 Spinal Cord Injury Treatment Options

Therapy options for those with a spinal cord injury have not progressed much in the recent years, but evidence that gait training is important is increasing. For many, the tedious manual gait training with a body weight support system may seem like it is not worth the effort. Patients with complete paralysis (ASIA A) will not be able to walk again on their own (with current treatment options). In fact, according to one study, "The majority of subjects (94.4%) who had a neurologically complete injury at 1 year remained complete at 5 years post-injury, with 3.5% improving to AIS grade B, and up to 1.05% each improving to AIS grades C and D [3]." While this may seem like an argument against using rehabilitation for complete injuries, there is evidence that this exercise is still beneficial to their long-term health and even more benefits are suspected but have not yet been sufficiently proven. Body-weight support treadmill training has been shown to improve muscle mass in spinal cord injury patients as opposed to no training [4]. There are also studies examining the benefits of the muscles moving across the bones as a means to decrease the effects of osteoporosis. Previously it was thought that simply standing would help improve bone density by putting weight through the bones. However, studies have shown that standing alone is not sufficient, and it is now thought that gait therapy may help improve bone integrity. Other benefits of therapy include relief from pressure points, improved blood circulation, and stretching of joints to maintain full range of motion.

These other benefits also hold for patients with a chance of recovering muscle function; however, for them, this therapy is crucial to regaining this strength and control. Studies have shown that "the locomotion of spinal animals can be improved by training that provides complex temporal patterns of sensory information related to stepping that is interpreted by the spinal cord [5]." The repetitive motion of gait therapy can help strengthen the limbs and retrain the brain to walk again. Unfortunately, the limited amount of time that many people can be in therapy may limit their ability to reach their full potential. Time is limited due to space, therapist time, insurance coverage, and cost. The following describes the current devices and methods used for gait training in rehabilitation centers. While some rehabilitation centers are beginning to use robotic gait trainers, most are limited to manual gait training and braces.

2.3 Current Gait Training Devices

Current gait training methods can be divided into two categories. One category is traditional clinical treatments. The second category includes the newer treatments, mostly based on robotic systems.

2.3.1 *MANUAL GAIT TRAINING & BRACES*

Traditional clinical gait training usually begins with manual gait training. This is done using a body weight support system over a treadmill. In this system, one physical therapist helps support the user's body and balance while two others move the legs in a walking motion. This can be seen in Figure 2-3. During manual gait training, two therapists move the patient's legs while one supports the patient's hips and aids with weight transfer. This therapy is done over a treadmill with a body weight support harness. http://www.physicaltherapyjournal.com/content/80/7/688.fullFigure 2-3. While this is effective for initial training, it is tiring for therapists and requires two or three therapists to support a single gait training session.

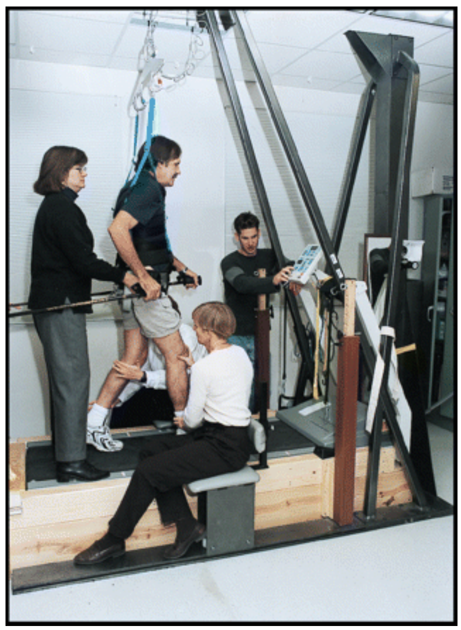


Figure 2-3. During manual gait training, two therapists move the patient's legs while one supports the patient's hips and aids with weight transfer. This therapy is done over a treadmill with a body weight support harness. http://www.physicaltherapyjournal.com/content/80/7/688.full

Once a patient has sufficiently progressed, he may move on to over-ground gait training. This is usually done using braces to support the weak joints. Braces vary in length based on what joints need support and range from HKAFOS (hip knee ankle foot orthosis) to KAFOS (knee ankle foot orthosis) to AFOs (ankle foot orthosis). Over-ground training is usually done in parallel bars for safety and support. This method allows the patient practice walking, but requires a lot of energy which limits the number of steps he can take in one session.

Figure 2-4. Examples of various braces used in gait training. a) HKAFO b) KAFO c) AFO http://orthoticconsultants.com/services.html

For many spinal cord injury patients, braces do allow them to stand and move about, but the energy expenditure is very high, especially for those patients requiring an HKAFO or a KAFO. Therefore, even with these options, they usually return to their wheelchair as a form of mobility and rarely, if ever, use these braces.

Another brace available for SCI patients is a Reciprocating Gait Orthosis (RGO). While the energy requirement for the RGO is still high and thus rarely used, it does give the patient a reciprocal gait by coupling the hips together.

Figure 2-5. A reciprocating gait orthosis supports the trunk, hips, knees, and ankles of a patient. The hips are coupled using a bar across the back, which allows for a reciprocating gait. http://www.centerfororthoticsdesign.com/isocentric_rgo/index.html

In the second category of gait training devices, there are more robotic options for spinal cord injury subjects. These devices, whether over-ground or treadmill based, provide increased repetition of gait as opposed to manual treadmill training. They also provide mobility with lower energy expenditure than traditional braces.

2.3.2 Treadmill-Based Robots

In the second category of gait training devices, there are more robotic options for spinal cord injury subjects. These devices, whether over-ground or treadmill based, provide increased repetition of gait as opposed to manual treadmill training. They also provide mobility with lower energy expenditure than traditional braces.

One of the most widely used treadmill-based devices is Lokomat, which is built by Hocoma [6]. Lokomat features a natural and repeatable gait pattern, intense training, motivation through biofeedback, and sensor information from the user's interaction with the machine **Invalid source specified.**. Lokomat has been used by numerous hospitals with positive results. Studies have shown that functional abilities increased more in sub-acute spinal cord injury patients who used Lokomat as opposed to those who did not [2]. Other studies likewise indicate that this training may be beneficial to those who use it. However, the Lokomat does require a large amount of floor space in a rehabilitation center to be used.

A second treadmill-based rehabilitation robot is LOPES. LOPES is developed by the University of Twente [7]. The robot has multiple degrees of freedom including knee and hip flexion and extension for walking. It also has hip abduction and adduction for balance training, which supported by horizontal translation of the treadmill support rig for safety. The pelvis also has vertical translation freedom for natural hip motion [8]. This robot has undergone clinical tests to determine efficacy and safety for chronic stroke patients. The study was too limited to make general conclusions but researchers feel that it offers a method by which stroke patients can recover some function [9].

While treadmill-based robots have been able to provide consistent therapy for many patients, their use is limited. Because they are coupled to a treadmill and the joints are moved for the user, the user may become passive in the rehabilitation. For this reason, many people have focused on making interactive displays to help maintain focus and engagement while undergoing rehabilitation [10]. Furthermore, the user is not responsible for balance, which can be an important skill to develop during gait rehabilitation. For rehabilitation clinics, space availability can be a concern, so treadmill based robots, which take up valuable floor space, may be too large.

2.3.3 Mobile Robots

There are many benefits to mobile robots for gait training. Their smaller size and increased involvement from the user make them of interest to clinics. However, to the users, the vision of using these at home or outside of the clinic is the main draw. Mobile robots offer the ability to continue gait training or provide gait support outside of the clinic. The increased time with the robot may yield better rehabilitation results. For those who rely on wheelchairs for mobility, the mobile robots provide a means to move through their daily lives while standing.

There are a few companies that are developing mobile exoskeletons for rehabilitation and mobility, much like the one discussed in this thesis. They are Rex by Rex Bionics, HAL by Cyberdine, ReWalk by Argo Medical Technologies. While each of these has varying features, their overall goal is to provide a mobile platform to assist walking.

Rex is a mobile exoskeleton for use by spinal cord injury patients and “manual wheelchair users who can self-transfer and operate hand controls” [11]. Rex’s outstanding feature is its stability. The user does not need to use an external walking aid because of the amount of support Rex provides. However, the hands are still engaged holding onto the joystick controller and another handle for added stability. Rex also assists with the lateral weight shift while walking, which is not provided for in the other mobile robots. However, Rex is bulky and slow. The added degrees of freedom for stability come with the cost of weight and size. Rex also appears to have very little torso support and current videos show patients who have control over their abdominal muscles and torso [12]. Rex does provide the user the ability to walk (both forward and backward) as well as climb stairs, sit and stand.

Figure 2-6. Rex, a mobile exoskeleton by Rex-Bionics. Photo from www.rexbionics.com

Argo Medical Technologies is an Israeli company that has been working on developing a mobile exoskeleton for spinal cord injury users. They have focused their efforts on making a small, lightweight machine that can be used for daily activities. Argo's website requires that their users are able to use their hands and shoulders for walking with crutches and have a healthy cardiovascular system and bone density. They have distinguished two different products: ReWalk-I, for institutional use, and ReWalk –P for personal use. ReWalk-I is currently available in rehabilitation centers, and ReWalk-P is expected to be released in 2011[13]. ReWalk has undergone trials at the Moss Rehab center, but study results are not yet published. ReWalk provides the user the ability to walk, climb stairs, sit, and stand. The gait appears to have less knee flexion than many other devices.

Figure 2-7. ReWalk, a mobile exoskeleton developed by Argo Medical Technologies, is a slim robot which allows a spinal cord injured user to walk. Photo: www.argomedtec.com

The third mobile exoskeleton is HAL, which was originally developed as a strength augmentation exoskeleton, but has applications in the medical realm. HAL utilizes bio-feedback through EMG signals from the surface of the skin, which makes it applicable for some injury types where muscle signals can still be read. These are then translated into the motion, allowing the user to move. HAL comes in multiple versions, and some versions include both upper- and lower-body components. However, the upper-body exoskeleton is mainly focused on strength augmentation for lifting.

Figure 2-8. HAL is an upper- and lower-body exoskeleton which aids the user's strength. It uses bio-feedback for control.

There are additional research projects developing exoskeletons for medical applications. The University of California, Santa Cruz has previously focused their research on upper-extremity exoskeletons and robots, but has now developed a prototype for a 12-DOF leg exoskeleton[14][14][14][14]. Additionally, the Human Engineering Laboratory at University of California, Berkeley is developing a second lower-extremity exoskeleton aside from the one presented in this thesis. Austin is a mobile exoskeleton that utilizes only two actuators to allow a spinal cord injury patient to walk, sit, and stand. This exoskeleton seeks to be a low-cost option for spinal cord injury patients[15].

3 BIOMECHANICS

In order to understand how an exoskeleton works with a person, we first must understand the biomechanics of walking. There are three anatomical planes used for describing biomechanics: the sagittal, frontal (or coronal), and axial plane (Figure 3-1). The hip and knee flex and extend in the sagittal plane during walking, so the exoskeleton actuators are designed to allow actuation in this plane.

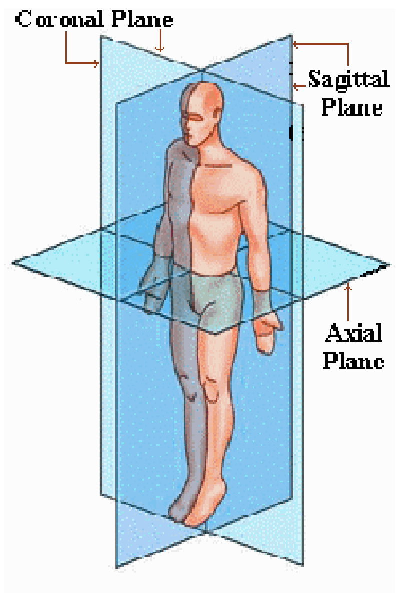


Figure 3-1. The anatomical planes of the body. http://www.spineuniverse.com/anatomy/anatomical-planes-body

The walking cycle is divided into two main phases per leg. The stance phase describes when the foot is on the ground, whereas the swing phase is when the foot is not on the ground. The swing phase begins with heel off, which is when the trailing foot begins to come off of the ground as the knee buckles in preparation for swing. Swing ends when the heel hits the ground during heel strike. These phases of walking are shown in Figure 3-2.

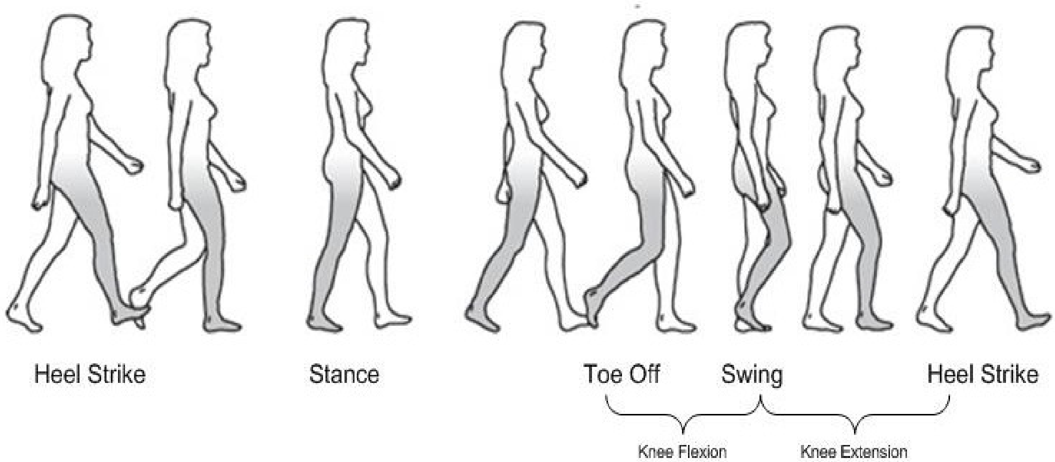


Figure 3-2. The phases of walking.

The two legs move through each of these phases in a coordinated manner. While the left leg is in swing, the right leg is in stance as there is no flight phase during walking (as opposed to running where both legs can be in swing at the same time). There is, however, a double stance phase when both legs are on the ground. This begins when the leading foot reaches heel strike and before the trailing foot has come off the ground.

A normal gait will have symmetric knee flexion on both legs and the knee flexion will be sufficient to allow the toe to clear the ground during swing. Researchers have analyzed gaits of healthy individuals to come up with "clinical gait analysis" data, which indicates a normal trajectory for the hip and knee angles during walking. Three sets of such data are shown here. This data shows the angle ranges and angular velocities of the joints, which we have used to determine the necessary range of motion and speed for the exoskeleton.

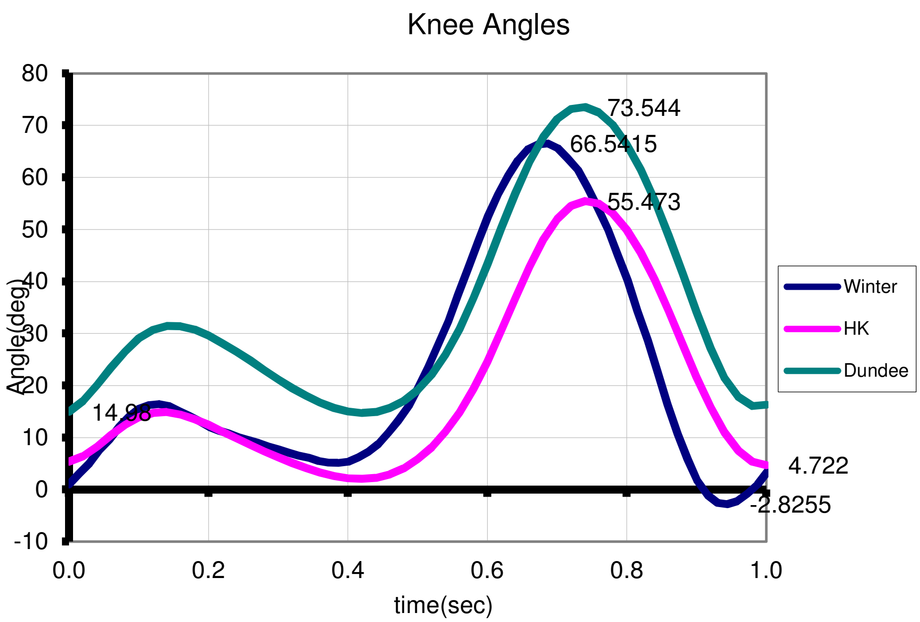


Figure 3-3. CGA Data for Knee Angle during Walking

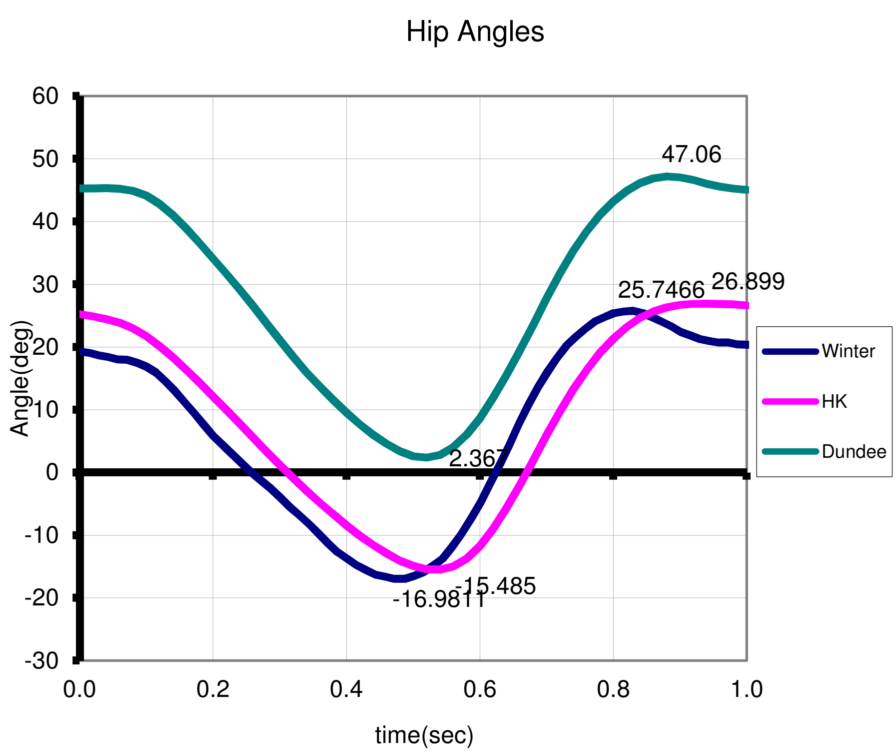


Figure 3-4. CGA Data for Hip Angle during Walking

4 System Description

A robotic exoskeleton is a device which a person wears externally, much like animals with exoskeletons. The robotic exoskeleton, however, augments the human's capabilities in some way, such as by adding extra strength, transferring weight loads, adding power, or increasing speed. Many exoskeletons developed are used by able-bodied individuals to add to their strength or endurance. Berkeley Bionics' HULC (Human Universal Load Carrier) can carry up to 200 lbs. without impeding the user and decrease metabolic costs (http://berkeleybionics.com/Unrestricted/HULC.html). The Raytheon Sarcos exoskeleton allows people to lift extra weight with powered legs as well as powered arms. The benefits of these human augmentation exoskeletons are in strength and metabolic costs providing for diverse applications in the military, industrial, and medical fields. The exoskeleton considered here, however, provides control and strength that those with spinal cord injuries no longer have.

The exoskeleton works by fitting close to the human body, with joints that are aligned with the human joints. By connecting the robot limbs to the human's limbs and powering the joints, the robot can help lift the person's limbs.

The main physical components of the exoskeleton are shown here. They are: the torso and backpack, the hip actuation units, the upper leg, the knee, the lower leg, and the feet.

4.1 MedX

During the course of the development of the exoskeleton and the HMI, there have been two generations of the exoskeleton design. The first is MedX. MedX was built as a proof-of-concept prototype using a HULC exoskeleton as the basis. It was hydraulically actuated at both the hips and the knees. Because of its prototype nature, the exoskeleton weighed over 80 pounds and had to be partially supported by an overhead gantry. This augmented the performance of the exoskeleton, and thus it was only used in one set of trials, which took place at the University of Virginia. Though the mechanical setup of the machine did limit our performance, we did learn valuable information about the user interaction with the exoskeleton. Initial data about arm motion, foot pressures, and coupling the body with the machine were all used in designing our second exoskeleton, eLEGS.

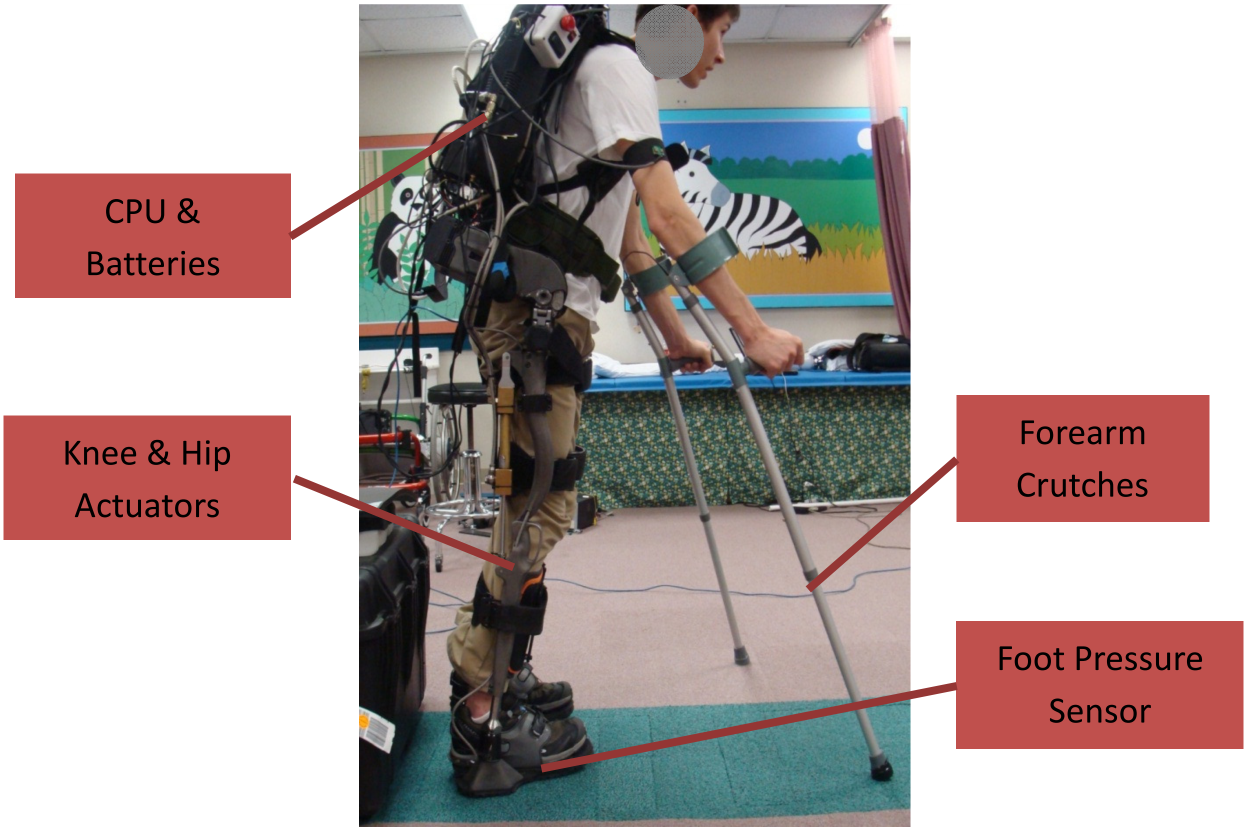


Figure 4-1. MedX Components include two actuators on each leg, foot pressure sensors, a backpack which supports the computer and batteries, and forearm crutches for support.

4.2 eLEGS

The second, which was used in clinical trials and for the majority of the development, is eLEGS. eLEGS (exoskeleton Lower Extremity Gait System) is an electromechanical system. This device is much lighter, weighing about 45 pounds, which allows the user to support the machine independently. The weight of the exoskeleton is supported by the exoskeleton itself; however, the inertia is still felt by the user. The mechanical components of the eLEGS exoskeleton are discussed here as this is the exoskeleton used for testing the HMI.

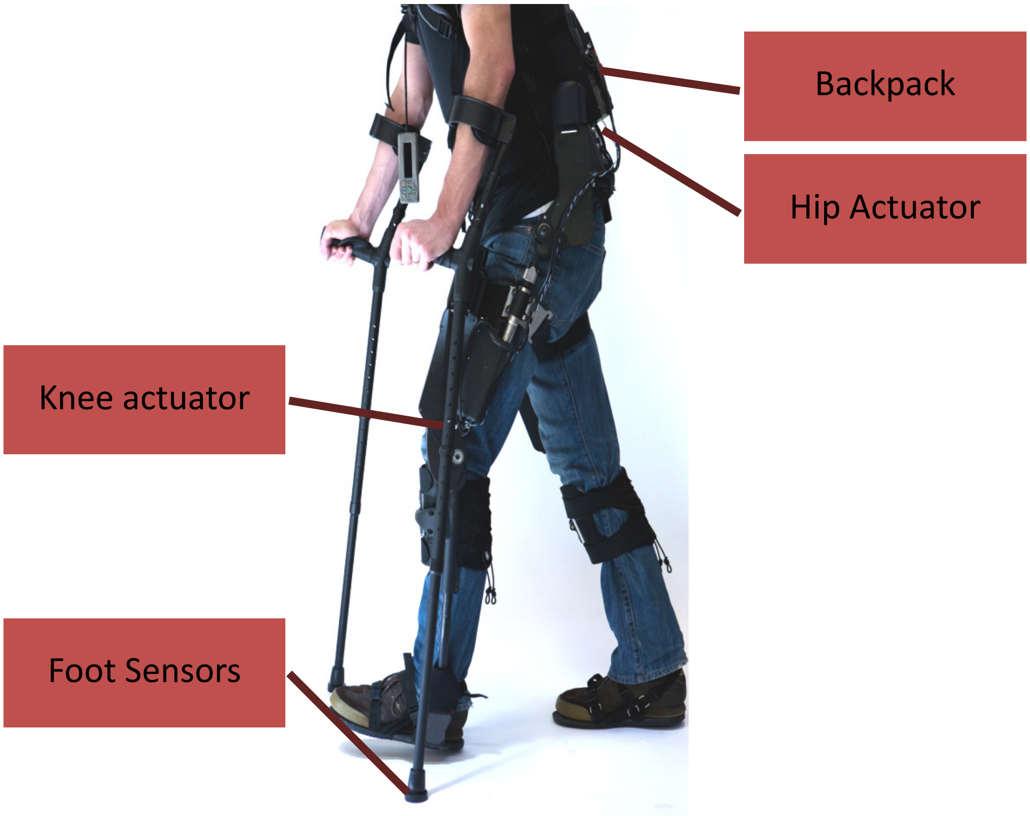


Figure 4-2. The eLEGS robot has four actuators, one at each hip and one at each knee. The foot sensors are located below the sole of the shoe. The backpack supports the computer and batteries.

4.3 Mechanical Components

The torso structure supports the computer, batteries, and serves as a connection point for the hips actuation units. The torso is rigid in order to protect the electronics components from bending or breaking. The hips actuation units are connected to the torso structure by a hip adjustment assembly, which allows the width of the machine to be adjusted for different sized users. The hip adjustment assembly also includes provisions for abducting and adducting the legs at the hip.

The hip actuators provide actuation in the sagittal plane. They have a range of motion of -20 to 135 degrees. The hips are connected to the upper leg through a hip abduction lock. This allows the legs to be abducted to allow the user to transfer into the machine easily. However, to minimize uncontrolled degrees of freedom, the joint is then locked in the abduction/adduction direction. The upper leg, however, is able to rotate slightly against a spring. The spring provides a force to center the leg while it is in spring, but also allows the person to pivot slightly on one foot to make a turn.

(a) (b) (c)

Figure 4-3. The degrees of freedom are shown individually here.

(a) The hip actuator is positioned along the hip of the user and actuates the joint in the sagittal plane. (b) The knee actuator is along the thigh of the user and has one degree of freedom in the sagittal plane. (c) The hip joint is passively actuated with springs.

The upper leg is a stiff structure which runs the from the hip actuation unit to the knee joint along the user's leg. The knee actuation unit is located within the upper leg assembly. The upper leg length is adjustable for different lengths of legs.

The knee joint is actuated in the sagittal plane only and is locked from motion in other planes. The knee has a range of 0 to 120 degrees. This range of motion allows for standing, walking, and sitting, and it prevents the knee from being able to move beyond the user's range of motion, thus preventing injury.

The lower leg is a stiff structure that goes from the knee to the ankle joint. This, like the upper leg, is adjustable to allow for different heights of users. The lower leg can also be rotated to allow for adjustment of the foot rotation in the coronal plane.

The ankle joint is passive and sprung in the sagittal plane. This provides a force to prevent foot drop, a common issue among spinal cord injury patients. However, the spring allows the entire foot to stay in contact with the ground as the user transfers their weight over the ankle, pivoting at the ankle. The other degrees of freedom of the ankle are locked. The ankle is connected to the foot plate, which is a semi-rigid structure that supports the user's foot.

4.4 Interface to User

The exoskeleton is attached to the user at multiple points to help support their body and couple them to the exoskeleton. The hip and knee joints must stay aligned to the exoskeleton hip and knee joints, so the padding and straps must provide this coupling. Furthermore, spinal cord injury patients are very susceptible to skin wounds and pressure sores. Because they cannot feel an irritation or pressure point, the pads must be sufficient to prevent any skin issues without the benefit of feedback from the patient during use of the exoskeleton.

The exoskeleton has a backpack harness to support the back of the robot and hold the computer and batteries against the person's back. The backpack is padded where the computer sits against the user's back, which helps protect the back from pressure as well as any heat from the computer. The backpack straps are like those from a traditional backpacking pack and have padded shoulder straps as well as a chest strap.

A torso harness helps to support the weight of the user and couple the torso to the hips. The torso harness is attached to the backpack and has a front section that is attached at the user's chest and straps to hold it tight. The front section can be small or tall depending on the level of torso support needed by the user.

5 State Identification

The intended motion of the exoskeleton must be determined. Therefore, the exoskeleton operates in two modes, which allow the machine to transition between actions. The first is PT mode, which is a manual input mode. The second is HMI mode, which is an automatic transition between states. These transitions should not require input which disturbs the user's motion.

The states identified may include the states of walking, such as heel strike, toe off, and swing. They may also include sitting, standing, and advanced maneuvers such as stair climbing.

5.1 PT Mode

In PT Mode, the state is selected using a control pad, which is an LCD screen with input buttons. The allowable states are determined by the same state machine used by the automated HMI. However, the physical therapist must select the state and push a button to cause the exoskeleton to move.

During walking, the PT pushes a button to take a step. The exoskeleton alternates between right and left steps, therefore, only one button is needed to walk. In order to select the sit down or stand up mode, the PT must select these on the control pad menu.

Figure 5-1. The control pad allows the physical therapist to select the state.

5.2 HMI Mode

A human machine interface (HMI) is the interface through which people interact with a machine. However, the HMI should be based on natural motions that allow the user to interact with the machine and use it independently.

The HMI is a combination of hardware and software that allows the user to indicate their motions. The development and testing of this mode is presented in the remainder of this thesis.

5.3 Existing Human Machine Interfaces

Treadmill based exoskeletons do not require an elaborate HMI because they are limited in function. Often keypads or buttons are used to adjust speed, amount of assistance, and step length, but these can be done by a physical therapist supervisor. A human machine interface for a mobile exoskeleton that can be used independently, however, requires that the user can easily switch between modes independently.

Rex Bionics' exoskeleton Rex utilizes a joystick and button control to initiate steps and other motions. Because Rex does not require the use of crutches, the hands are free to control the joystick. However, because the medical exoskeleton presented here requires the use of crutches for stability and safety, this is not a viable option.

ReWalk, an exoskeleton developed by Argo Medical Technologies Ltd., is controlled through "subtle changes in the center of gravity" and a button interface worn on the user's arm [3]. The user leans forward to start walking and leans back to stop walking. At the time of the writing of this paper, the method by which the user switches from sitting to standing or from standing into the walking mode requires the user to push a button to switch modes. This is cumbersome for the user, especially since he is relying on the crutches for stability. Furthermore, leaning forward and backwards can be difficult for those with higher spinal cord injuries resulting in a lack of control of the torso.

Additional motions, such as using a keypad or moving the tongue or arm in a specific way is not desirable because it limits the ability of the user to learn and use while performing other tasks. One group has developed a method by which the user's tongue motions can be read by studying the airflow [16]. While this showed promising results for being able to distinguish various commands, it did require a training period and did not allow the person to freely speak while using the tongue control method. For this exoskeleton, we would like the training period for use of the HMI to be minimal so that people can start using it immediately since rehabilitation time is limited due to insurance coverage. Also, it is unreasonable to ask people to not talk while walking if they are using the exoskeleton out in the community.

Finally, Brain Machine Interfacing is a developing field which allows brain activity to be read and translated into intended motion. For spinal cord injured patients, this could allow the signal to be read above the injury level and translated into a desired motion. However, this field is still new and does not yet have a reliable means by which to accomplish this for walking [17][18][19]. While Brain Machine Interface (BMI) is a viable method for future controls for the exoskeleton, currently there are a variety of issues which must first be addressed. BMIs rely on either reading signals from the surface of the brain or implanting electrodes to read the signals from inside the brain. Both of these methods require invasive surgery to achieve the best signal strength. For invasive electrodes, which offer the best possibility for reading clear signals, the brain treats them as foreign objects and forms scars around them, decreasing the signal strength. As these scar layers form, the electrodes have to be replaced. Therefore, this is not currently a long term solution for exoskeleton use. Furthermore, with all of these methods, a training period is needed to determine what signals map to which motions. Most research in this mapping has been done for upper extremity grasping and manipulation. As this field develops, the integration of BMI with the HMI presented here could provide even more accurate means of controlling the exoskeleton.

For the medical exoskeleton HMI, multiple methods of input were considered. First, however, basic requirements of the system were established based on end user input.

6 END USER SURVEYS

To determine the system requirements, we surveyed potential end users using an online survey which was advertised at the University of Virginia, the Berkeley Center for Independent Living, and other wheelchair users from internet forums. The exoskeleton should accommodate daily activities of the users. Of the 52 survey respondents, 42 responded that they would use the exoskeleton in their home for normal daily activities. Those who responded that they would not use it cited that it was too bulky, that the exoskeleton would be slower than their wheelchair that they were already adept at, and that they could walk with braces and crutches already. Since these features will not be affected by the HMI, this did not lead to any specific requirements, but did indicate that there is a user market for the exoskeleton. 42 respondents also indicated that they would use the exoskeleton in public for errands and other activities. Many cited that though their wheelchair is faster, this would allow them to go places which were formerly difficult for them, such as up stairs. Stability, however, was a concern in that in the community, they would be in close proximity to others going about their daily activities as well and would be subject to being bumped into. Since the exoskeleton is only actuated in the sagittal plane and thus can only affect balance in that plane, it is important that the user has access to another support device for additional balance.

Potential users were asked what activities they most desired the exoskeleton to allow them to do. These answers help to determine what states the HMI needs to be able to determine. The most common answer was walking, closely followed by stairs and reaching for things while standing. This means that the exoskeleton must be able to determine when the user wants to walk and identify when the user wants to climb stairs. Furthermore, it must determine when the user wants to stand and be able to remain standing while the user moves their arms to reach for something, cook, or give a person a hug. Users also wanted to be able to avoid obstacles which are currently difficult to avoid in a wheelchair. This means that the user must be able to tell the exoskeleton when to turn, vary the step length, and vary the step height. Users also wanted to be able to get up from a chair unassisted, so the HMI will need to detect when the user wants to stand up or sit down.

Additional desired activities included carrying things, entering cars, walking on uneven terrain such as trails, beach, and grass, dancing, and side stepping. These are considered advanced maneuvers and the continuous controls are not yet being developed for these maneuvers. However, these are possible expansions and so the HMI will be designed with consideration for these maneuvers.

Users were also asked what walking aid they would be willing to use. Multiple answers were permitted for this question and there were 51 total responses. Out of the 51, 36

(70.6%) were willing to use forearm crutches with the exoskeleton. The second most acceptable walk aid was a cane, which 66.7% of respondents were willing to use. However, since a cane does not provide sufficient stability and balance, this is not a viable option. The next most choices were the walkers (standard and front wheeled). Based on these responses, the HMI will be designed primarily to work with forearm crutches, but will also have the goal of being compatible with a walker.

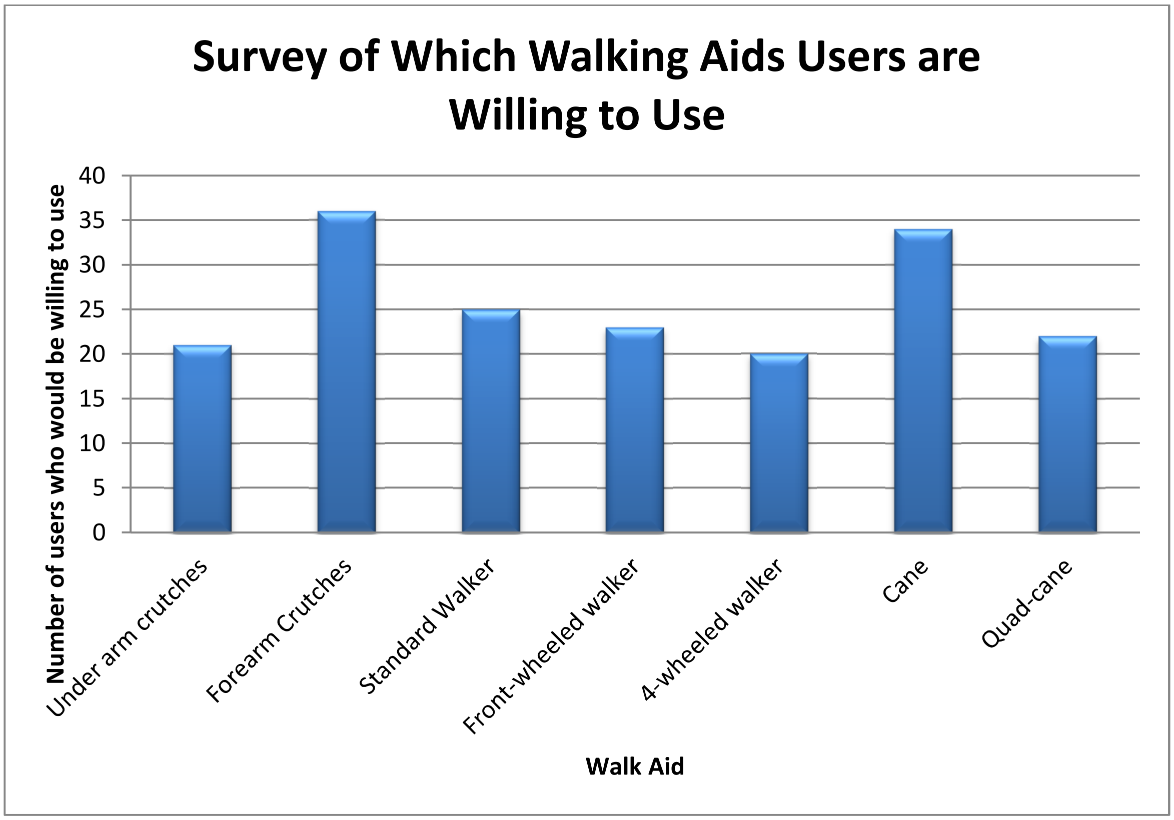


Figure 6-1. Walking aid preferences as identified by potential end users.

When asked for an ordered preference of walking aid among quad cane, 4-wheeled walker, front wheeled walker, standard walker, forearm crutches, and underarm crutches, users chose forearm crutches most frequently as their top choice. Out of 50 responses to the question, 35 users listed forearm crutches as one of their top three support device preferences. While many would prefer the use of a cane, this was deemed not practical for providing sufficient balance for new users and is not traditionally used in gait rehabilitation for those with spinal cord injuries.

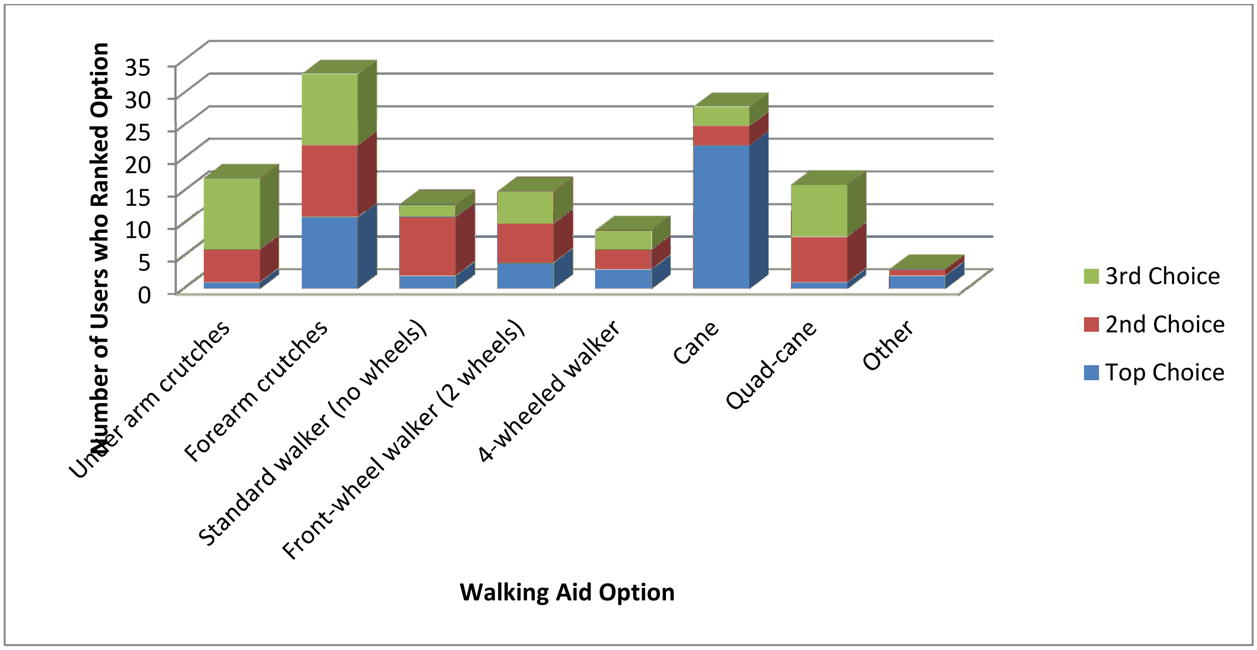


Figure 6-2. The preferences of the support devices are ranked in terms of preferences.

Further investigation into user's desired methods of walking has been done during research and development testing. Throughout trials in rehabilitation hospitals such as Shepherd Center in Atlanta, GA, Good Shepherd in Allentown, PA, Santa Clara Valley Medical Center in Santa Clara, CA, and Rehab Institute of the Pacific in Honolulu, Hawaii, the HMI has been demonstrated and tested with various patients and physical therapists. During each of these tests, the physical therapists have indicated a desire to start working with the patient using a walker then progressing to forearm crutches as their balance and weight shift ability improves. The crutches then provide a natural option for the Human Machine Interface as this is the preferred method used by physical therapists during gait rehabilitation. Furthermore, when the patient has progressed sufficiently, the therapist will ask the patient to indicate when he is ready to step. This starts to give the control of the step to the user rather than the therapist. This makes a natural progression to the HMI where the user has full control over his steps. When the users make this transition, they feel more in control of the robot and independent. This is a feature which must be retained throughout the development of the HMI.

6.1 HMI System Requirements

The system must determine when the user wants to take a step. This is a fundamental requirement of the system as the primary use of the device will be walking. Furthermore, because of the frequency with which a step is taken, the method to trigger a step should be as simple as possible in order to make the system easy to use.

The system must be able to tell when to stop walking as well as when to sit and stand. Though these functions can be triggered by external buttons, incorporating them into a simple HMI would ensure ease of use. The HMI should also have the ability to adjust characteristics of the step, such as how fast or how far. These should be adjustable during walking, which requires a hands-free method of adjustment since the user will be using crutches.

The HMI should be a minimal disturbance to the user's walking. This means that the HMI should not require extra motions that the user would not otherwise make. The HMI must also maintain the safety of the user. Safety requires that the exoskeleton only take a step, sit, or stand when the user is ready. It also means that the HMI cannot interfere with the use of crutches for balance or be sensitive to false triggers.

6.2 HMI Proposed Concepts

The following are the initial concepts considered for the Human Machine Interface.

6.2.1 Voice Recognition

A voice-based HMI system would include a microphone and software which allows a user to speak commands into a headset. These commands would direct the robot's motion. This system would be able to recognize any action in the robot's set of actions. It could also be used with any balance aid, such as crutches or a walker. However, it is susceptible to background noise or accidental triggers from normal conversation. In order to prevent these accidental triggers, a trigger word or button would need to be added, which would increase the complexity. Furthermore, when interacting in society on a day to day basis, verbally commanding the unit to start walking or stop walking may be embarrassing.

6.2.2 External sensors- crutch sensor

The crutch sensor would be a sensor board attached to the crutch. The sensor can measure the angle and velocity to determine the step length and speed. The crutch could also be outfitted with a pressure sensor to determine when it is safe to step. This system could be integrated to any crutch and provides the safety assurance that the crutch must be on the ground in order for the robot to take a step. The position measurements allow for step length adjustments in real time. There are possible inconsistencies in measurement due to sensor noise and variable methods of placing the crutches. This system also does not provide a method for identifying sitting, standing, or going up stairs.

6.2.3 On-board Sensors

The on-board sensors are those included the exoskeleton design for control. These include joint angle measurements, back angle, and foot pressure sensors. Utilizing the on-board sensors, the HMI could simply be a method of combining known information about the state to determine when to take a step. It could determine when to take a step by determining where the pressure is on the feet and calculating a desired action based on the person's pose and the pressure location. Extra motions, such as reaching for an object or holding a bag could disrupt the reliability of the system and cause unwanted steps.

6.2.4 External Sensors- Arm Sensor

The arm sensor method consists of a sensor band worn on the upper arm. This would allow the HMI to read the upper arm angle and use this to determine when to step, how far, and which leg. Like the crutch sensor method, this can also utilize a sensor on the crutch to determine when the crutch is providing stability by being on the ground. The arm band sensor is small and light weight and is worn on the upper arm so as to minimize interference with other actions. They can also be made to be adjustable to fit any user. The arm band can be used with crutches or a walker. However, because the arm is cylindrical, the arm band will be susceptible to twisting, and thus the arm motion may not be reliably read. Furthermore, additional safeties are needed to prevent the machine from taking a step while the person is reaching for an object.

The arm sensor, in conjunction with the crutch sensor, appeared to be the most viable option in terms of ease of use and expandability. This is the concept that is further developed for the medical exoskeleton HMI.

7 HMI Hardware

The sensors which make up the HMI hardware are used to determine the intended state.

7.1 Crutch Force Sensor

7.1.1 Specifications

The sensor was originally designed as a push button to indicate on or off the ground. The button was embedded in the foot of a standard crutch. This measurement was sufficient for identifying on or off the ground, but could not be accurately adjusted to ensure that the person was bearing weight on the crutch. Furthermore, many physical therapists expressed interest in knowing exactly how much weight is being transferred through the shoulder as this information can be used to ensure safety of the user as well as to coach the user in gait training.

The following crutch sensor specifications were developed based on testing of other crutches and specifications of the weight expected to go through the shoulder. I examined existing crutches with "shock absorption" to determine what industry standards consider comfortable deflection for the user. The sensor was also required to fit inside of a standard crutch without modifying the crutch. This allows the user to pick a crutch which is comfortable for their use.

Outer diameter: <0.72"

Maximum sensor length (beyond length of crutch): 5"

Maximum deflection: 0.5"

Minimum weight detected: 50 lbs.

Maximum weight supported: 200 lbs.

Sensor resolution: 1 lb.

7.1.2 Design

The crutch sensor developed for the medical exoskeleton HMI utilizes a linear potentiometer. The potentiometer has infinite resolution, and thus is limited only by the ADC conversion resolution. The potentiometer is from State Electronics, LCP8, and has a travel of .433-.5". The datasheet for LCP8 is shown in Appendix A. JIS Springs from Anchor Daily were chosen because they had sufficient spring rate to measure 50 lbs. with allowable deflection for infinite cycles while fitting in the space constraint of the system. Two options were specified. The first, 93-1440, has a spring rate of 3 kg/mm (167.99 lb./in) and can thus measure 68 lbs. in a 0.40" deflection. 94-1455 was chosen to measure up to 103.6 lbs. in 0.42" deflection. These two springs have the same diameters but different lengths, so the crutch sensor was designed to be able to accommodate either of these, depending on what final load limits were determined after testing.

Figure 7-1 shows the crutch sensor assembly. The grooves in the upper section allow for o-rings, which hold the assembly in the crutch tube. The potentiometer fits into this top section, which is machined out for easy assembly. The lower section, shown in green, remains outside of the crutch. There is a locating surface which allows the crutch to be firmly seated at a specific location, maintaining the calibration. The outer surface of the lower section is designed to eliminate any pinch surfaces that result from the sprung bottom. The bottom also fits a standard crutch tip which provides traction for the crutch.

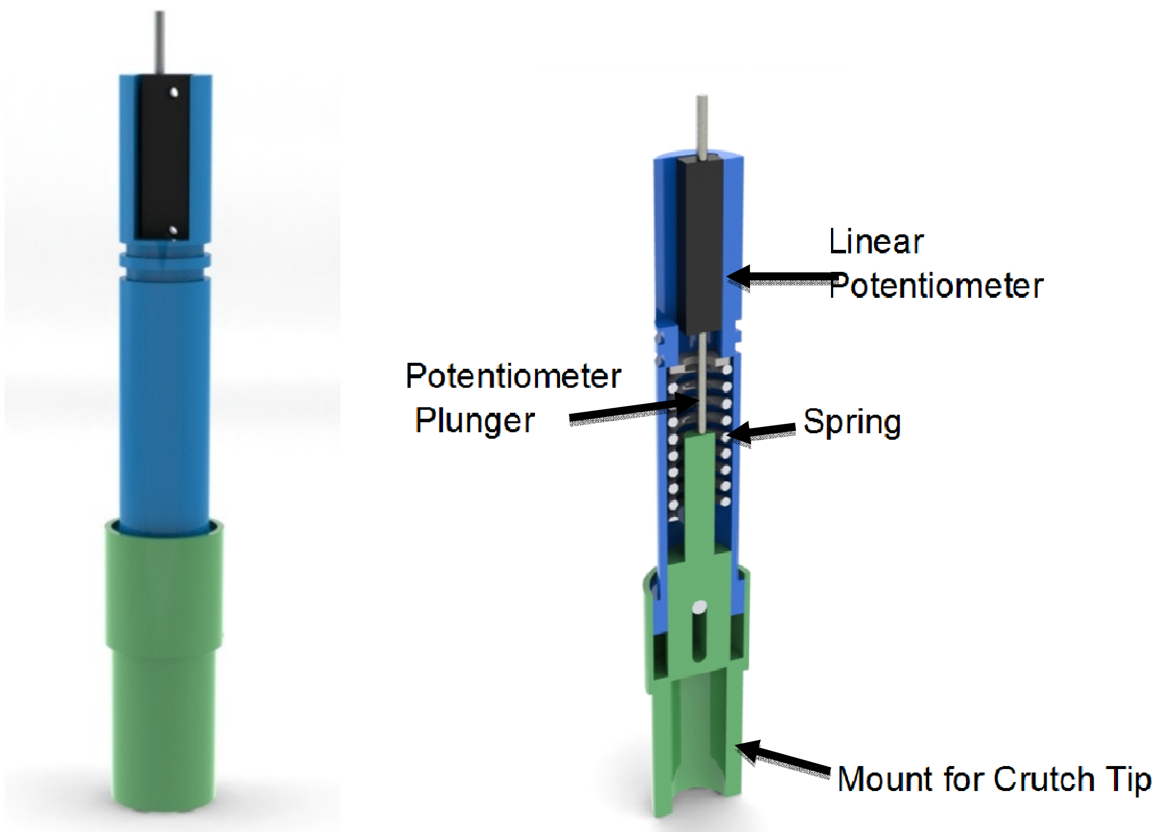


Figure 7-1. a) The crutch sensor assembly, which fits into a standard crutch. b) A cross-section of the crutch. The spring can be interchanged to allow for different force ranges to be measured.

The cross-section view of the crutch sensor assembly shows the bottom plunger, which compresses the spring and thus moves the potentiometer shaft. Because the force is proportional to the potentiometer shaft displacement, a simple calibration can be used to determine how much force is going through the crutch assembly.

7.1.3 *Analysis and Testing*

The sensor was calibrated using Larson Systems Inc. Force Digital Hand Tester. This allowed for the application of a precise force and then the potentiometer output was read. The sensor assembly displayed some hysteresis, so the reading was always made before releasing the force. Four sensor assemblies were calibrated. Because of machining tolerance and tolerance in the length of the spring, the values were not expected to be exact. Figure 7-2 shows the calibration of the four sensors. Two were calibrated to approximately y = 0.2x-.8 while the other two were approximately y = 0.18x. These sets were paired for use so that the calibration between left and right was similar. Because these sensors are linear, the output voltage from the circuit is also linear. These no-load reading is 0 lbs. and the highest reading is 80 lbs. The force applied can then be found as a function of the voltage, which varies from 0 to 3.3 V.

$$\text{Force} = 80*(\text{Voltage}/3.3\text{V})$$

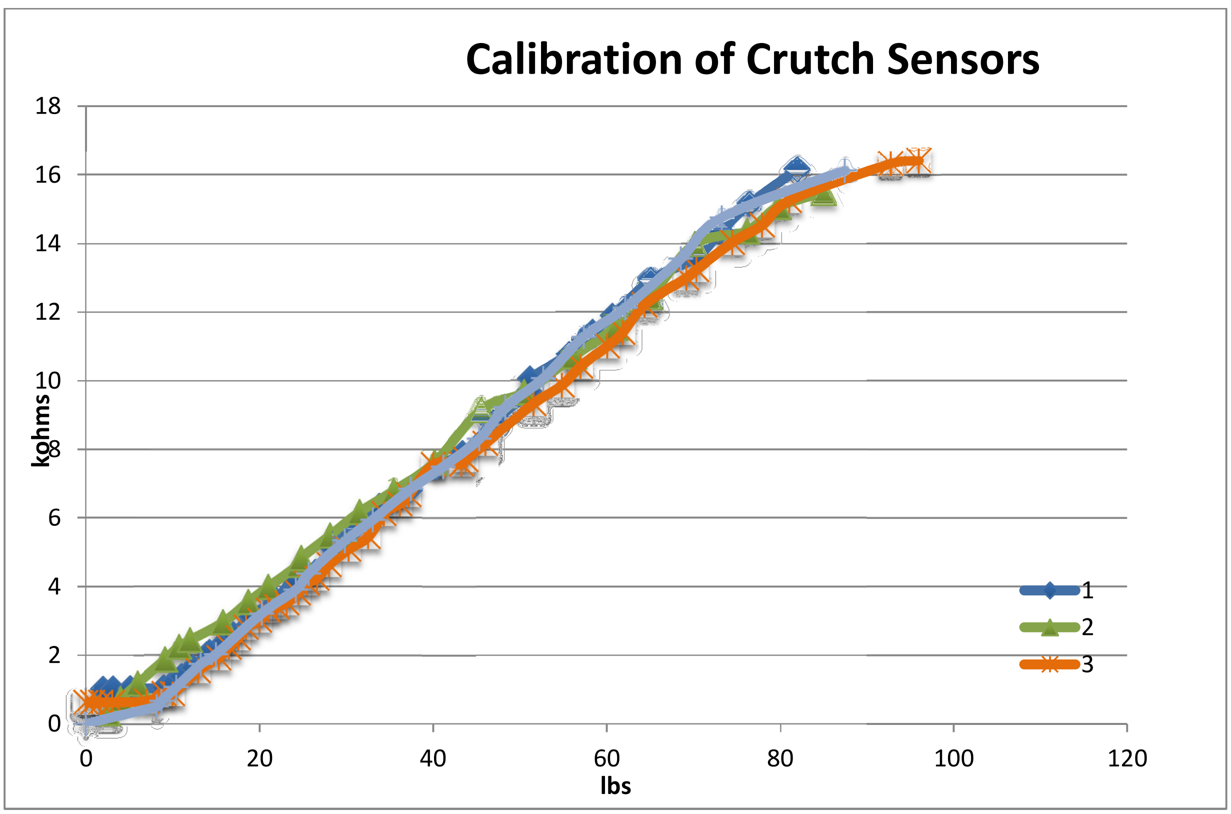


Figure 7-2. Calibration of Crutch Sensors

7.2 Arm Angle Sensor

The arm angle sensor measures the projection of the arm angle in the sagittal plane. This can be used to estimate where the end of the crutch is placed. However, because the state machine only needs a relative position of the crutch, this estimation is sufficient.

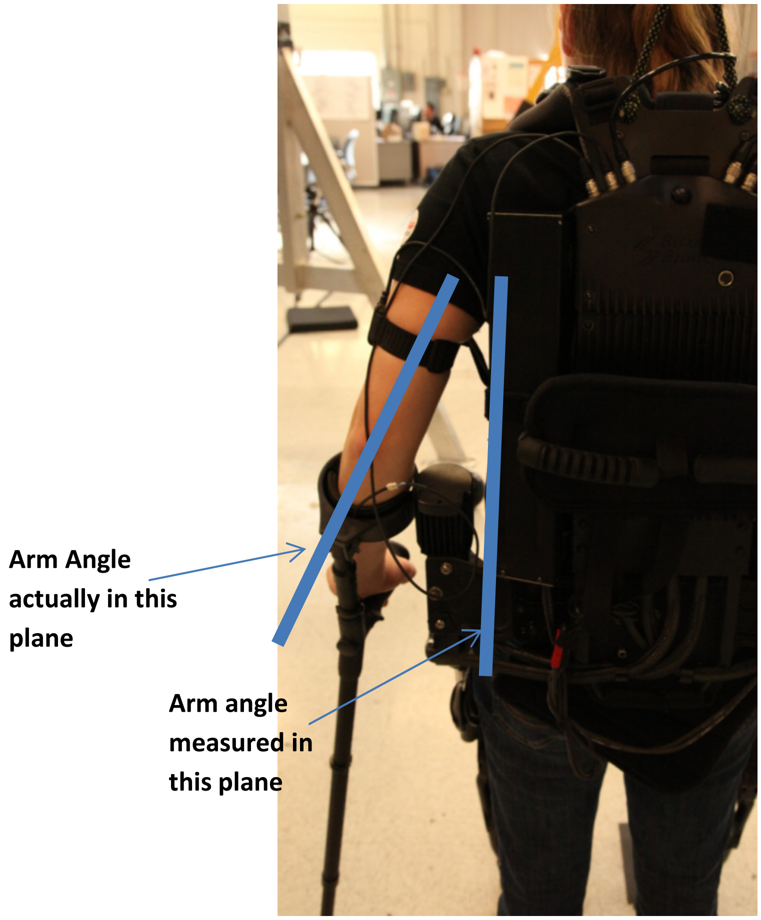


Figure 7-3. The measured arm angle is a projection of the actual arm angle on the sagittal plane.

The arm angle sensor uses an inertial measurement unit from Sparkfun, SEN-0918. The gyroscope (Analog Devices ADXRS610) has a rate of ±300º/sec. The accelerometer (Analog Devices ADXL203) has a range of ±1.7 g. These two components are mounted on a single board to maintain their relative orientation.

The case is fabricated using fused deposition modeling (FDM), so curves and other features are not difficult to create. The FDM process creates a case that can be manufactured quickly and cheaply despite having a smooth and curved profile. FDM utilizes a plastic which is layered, so it is not as durable as a molded case, but since the arm sensor is not expected to withstand any impact forces or loading, this is sufficient.

The full case is made of two parts. The bottom provides locating holes for the sensor, which ensures that the sensor is mounted in the same place on each arm sensor band. The support beam in the center allows the sensor to be mounted with the components facing down with sufficient support to keep it from deflecting during use. This minimizes the height of the case. There is a strain relief feature in the case which allows the wires from the IMU to be zip tied onto the case, which will prevent them from breaking in the event that the wire outside of the case is pulled. The opening fits a standard rubber strain relief component, which prevents the wires from being pulled out as well as prevents them from breaking from being turned around too tight of a corner.

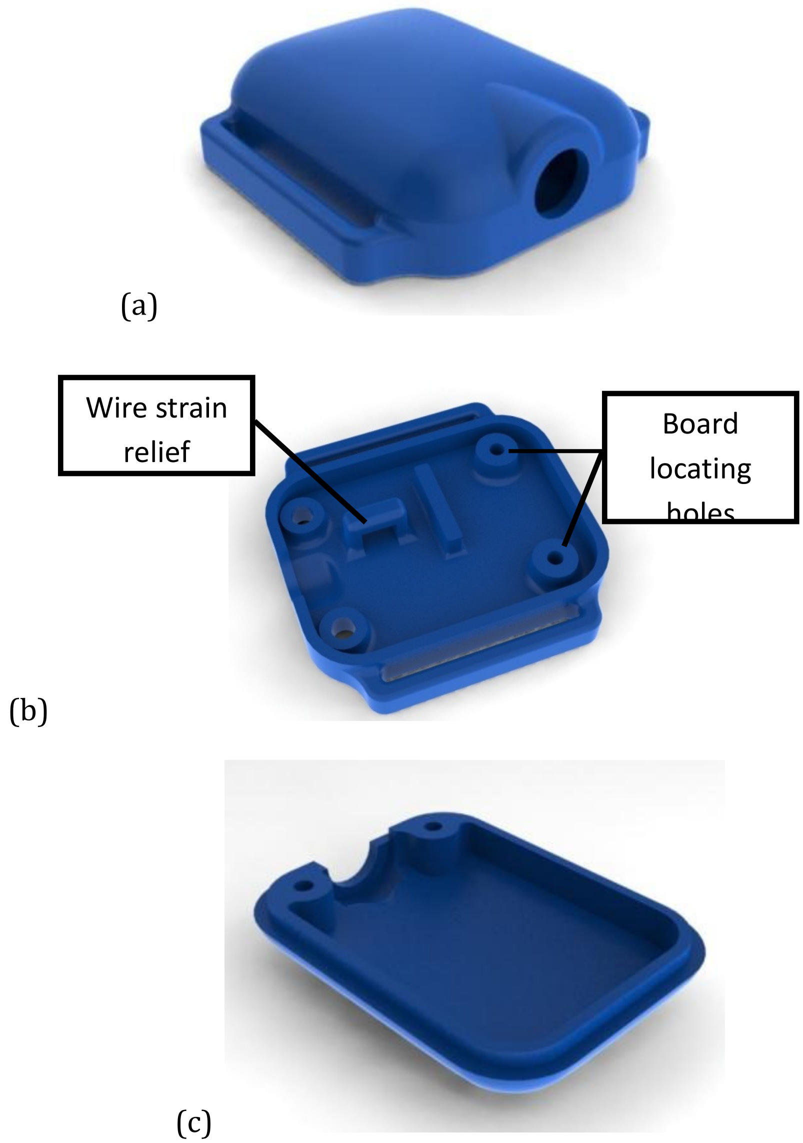


Figure 7-4. The arm sensor case is designed to protect the IMU and hold it in place on the upper arm. (a) The assembled case. (b) The bottom section of the case where the IMU is attached. (c) The top cover of the case.

The case also provides an attachment point for the elastic band, which is used to secure the sensor to the user's arm. The elastic provides a comfortable fit while ensuring that the sensor is securely fixed in one location. It is adjustable for various users. Figure 7-5 shows the sensor held in place on a user's arm. The cover has been removed for clarity. The arm band wiring plugs into the top of the back panel; therefore the wire does not interfere with the user's motion.

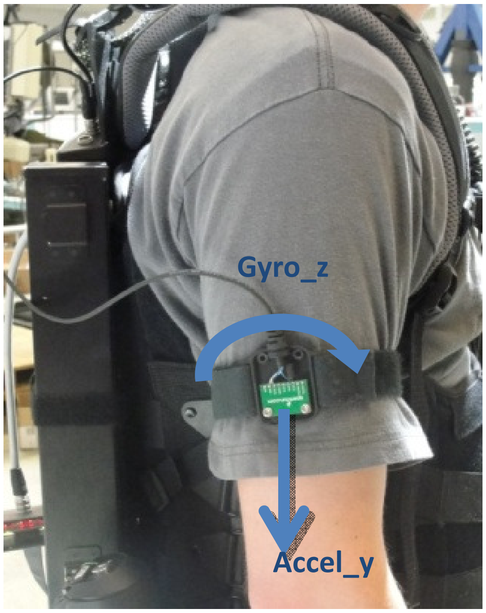


Figure 7-5. The arm sensor case is shown here on a user's arm (without the cover). The sensor is held in place by an elastic band. The accelerometer is oriented down the user's arm. The gyroscope reads the acceleration in the plane of the IMU.

7.2.1 Arm Sensor Calibration

The accelerometer measures the acceleration of a mass with respect to the reference frame of the sensor. This gives an accurate angle measurement in the stationary case, as $a_y \propto \sin(\theta)$. The sensor offset, which is found during calibration, is then subtracted from the output and the measurement is scaled to the full range of the sensor to give the angle estimate. The offset is found by holding the arm at the 0 degree position (straight down along the person's side), which should give a 0 degree output. The range of the sensor is found by holding the arm at the 90 degree position (straight out in front of the person) and subtracting the 0 degree position reading. This calibration is done through the user interface.

The gyroscope measures the angular velocity about the axis perpendicular to the IMU board on the arm. The gyroscope is calibrated by holding the arm still to obtain the 0°/s reading to remove any offset error. The range is assumed to accurate from the datasheet because the calibration of these devices done by the manufacturer is more accurate than can be done on the exoskeleton. Furthermore, this calibration is not likely to drift. The gyroscope reading is only valid at high frequencies and is subject to drift. Therefore, it is combined with the accelerometer, which gives an accurate absolute angle at low frequencies to get accurate arm angle estimation.

In order to combine these two, a complementary filter was used. This filter is selected because it uses the high frequency readings of the gyroscope and the low frequency readings of the accelerometer. It is also computationally simpler than a Kalman filter and provides sufficient accuracy. A low pass filter is applied to the accelerometer output and a high pass filter is applied to the gyroscope output. These filters are done at the same cutoff frequency (0.15 Hz) such that when added together, they sum to 1. The cutoff frequency is chosen based on the response of the hardware.

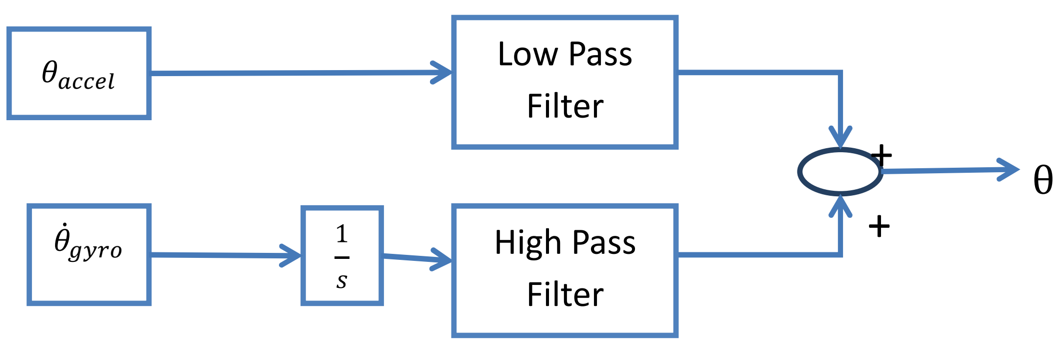


Figure 7-6. This schematic shows the complementary filter used to determine the arm angle from the IMU readings.

Initially, a first order filter is chosen for simplicity. However, the response here is not sufficient. The accelerometer should only be read at low frequencies since it is influenced by motion, which will cause errors in our estimation. In order to get a sharper roll-off and thus a better attenuation of high frequency noise on the accelerometer, a second order filter can be used. Therefore, the continuous time filter equations become as follows:

$$Low\ Pass\ Filter = \frac{1}{\tau^2 s^2 + 2\tau s + 1}$$

$$High\ Pass\ Filter = \frac{\tau^2 s^2 + 2\tau s}{\tau^2 s^2 + 2\tau s + 1}$$

This is a complementary filter since the filters add to 1. The comparison of these responses can be seen in Figure 7-7.

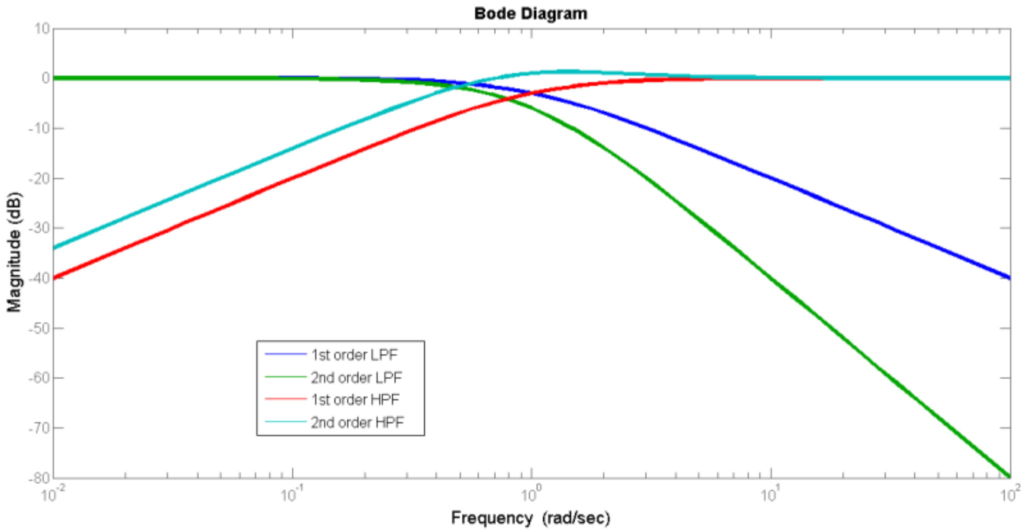


Figure 7-7. A comparison of the 1st order and 2nd order filters for the accelerometer and gyroscope. The 2nd order filter has a faster roll-off.

Since the calculations are done in discrete time in this system, the filter is also transformed into discrete time using a bilinear transformation. The bilinear transformation maintains the frequency response characteristics and approximates the mapping of the s-plane to the z-plane.

$$s = \frac{2}{T}\frac{z-1}{z+1}$$

T is the sampling time and is 0.008 seconds for our system. This mapping is applied to the filters to transform the equation into discrete time.

7.2.2 Arm Sensor Testing

The arm sensor could also be on the crutch to measure the crutch angle since the ultimate goal is to determine that approximate crutch tip position. Once the sensors were built and configured for the electronics board, testing in both positions was done. The intent of the tests was to determine what location was most robust for the sensor placement. The sensor should give consistent performance regardless of user. The sensor should also indicate the intended step length by measuring a difference in crutch angle.

The sensors were tested and plotted for various conditions. The plots shown here give uncalibrated sensor values as the data was taken before the calibration and are given in ADC values.

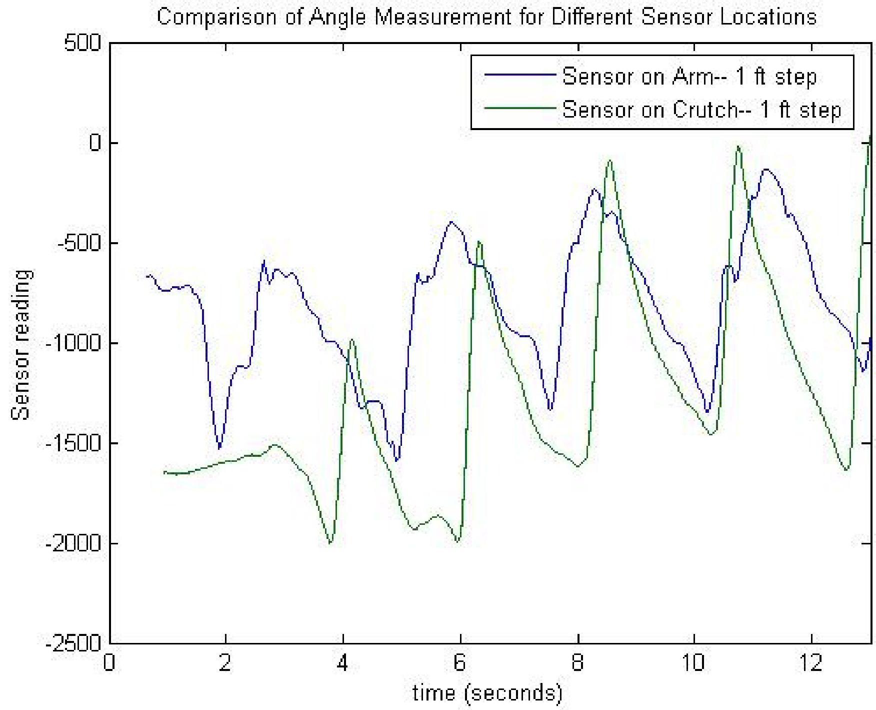


Figure 7-8. The arm angle sensor was tested on both the arm and the crutch. The signals on both are shown here. The sensor reading is given in ADC counts, which is proportional to angle output.

The comparison of arm to crutch does not indicate that one is decidedly easier to detect. Both motions give detectable motions towards the front and back. The crutch is more easily read because of the sharp peaks, of the angle, but the arm peaks are sufficiently easy to detect.

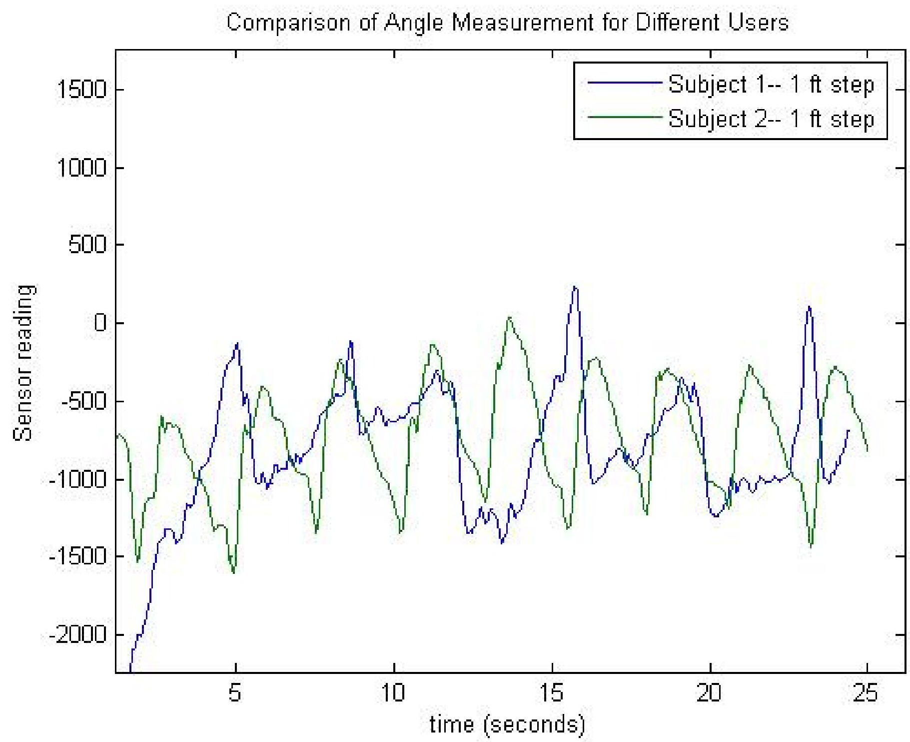


Figure 7-9. The variation between two different users with the sensor on the arm is shown here. The magnitudes between users are reasonably consistent.

The arm angle leads to an easier sensor placement, since the wiring necessary can be run along the arm and into the robot's computer. Therefore, the difference of the response of the arm angle between users was investigated. The above plot shows the arm angles of two users, each taking 1 foot steps and thus positioning the crutches for a 1 foot step. The two subjects have very similar magnitude changes between the forward and backward position of the arm. This indicates that this measurement will be robust enough for use by various users.

The final step is to determine if the sensor is sufficient to determine different length steps. In order to test this, the user placed his crutch for stepping 1 foot, and then for stepping 2 feet. I expected to find that the 1 foot step had a smaller change in angle, as expected. The magnitude of the 2 foot step arm angle, especially when the arm is behind the user, is much greater than the 1 foot steps.

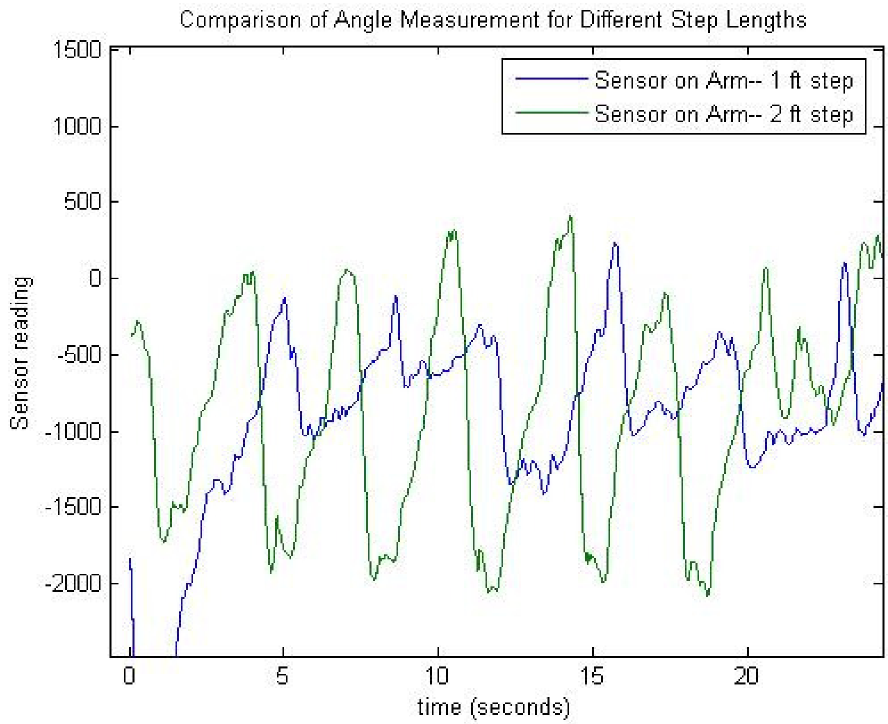


Figure 7-10. The arm angle signal should also differ based on step length. Here, the difference between a 1 foot and 2 foot step is shown. As expected, the change in arm angle is much greater for a 2 foot step.

These measurements are consistent from user to user and can be used to indicate when the user has placed their crutch out in front, indicating the desire to take a step. In future applications, the crutch placement can also be used to augment step length based on the distance forward that the user places the crutch. This length can either be determined by a geometric estimate based on the user's height or by a learning algorithm or other function which maps arm angle to step length.

7.3 Foot Sensors

The foot sensors measure the reaction force between the ground and the user's foot. The foot sensor gives two reaction force measurements: one for the forefoot, and one for the heel. Therefore, by reading the value of these sensors, we can determine if the user's foot is on the ground, which indicates whether it is in stance or swing. With this verification, the HMI will prevent the user from going into swing until the other leg is on the ground.

The foot sensors are designed as part of the exoskeleton foot. The design must be compact and flexible so as to not interfere with the gait. When the user shifts their weight forward and the weight is on the toe, the heel must be able to come off of the ground. The sensor used must also be robust to the bending forces as it will undergo numerous cycles over the lifetime of the product. In order to achieve this, we utilize a conductive fabric a flexible metal layer, separated by perforated foam. As the foam is compressed, the resistance between the two layers of conductive fabric decreases. By measuring the resistance between the two layers, we can determine the level of compression of the sensor, which is indicative of the force across the sensor.

The sensor is divided into two regions which give independent signals: the toe and the heel. In order to accommodate a wide range of foot sizes, these sensor sections fill the majority of the region of the foot, so that the force is detected no matter where the foot is on the footplate. This larger segment also accommodates pronation or supination. The separation of the heel and toe allows for the distinction of heel strike and toe off.

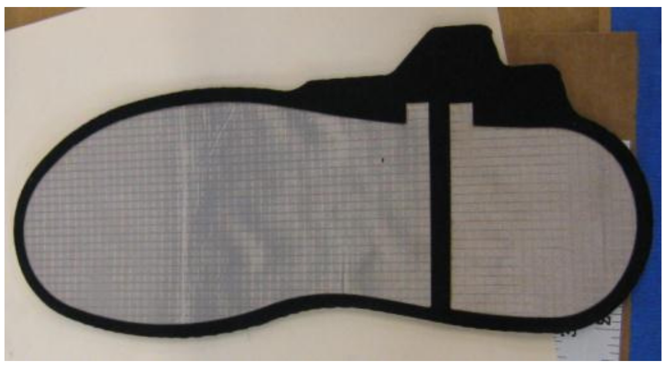


Figure 7-11. The foot sensor has two segments for accurately determining heel strike and toe off. The front section is larger to accommodate a range of foot sizes.

Figure 7-12 shows the foot sensors during normal walking. The heel hits first and thus that sensor output goes up. As the person shifts forward, his weight transfers onto the toe. This can be seen in the transition of the sensor output from the heel to the foot. When the foot goes into swing, there is no longer any weight on the foot and both sensors read 0.

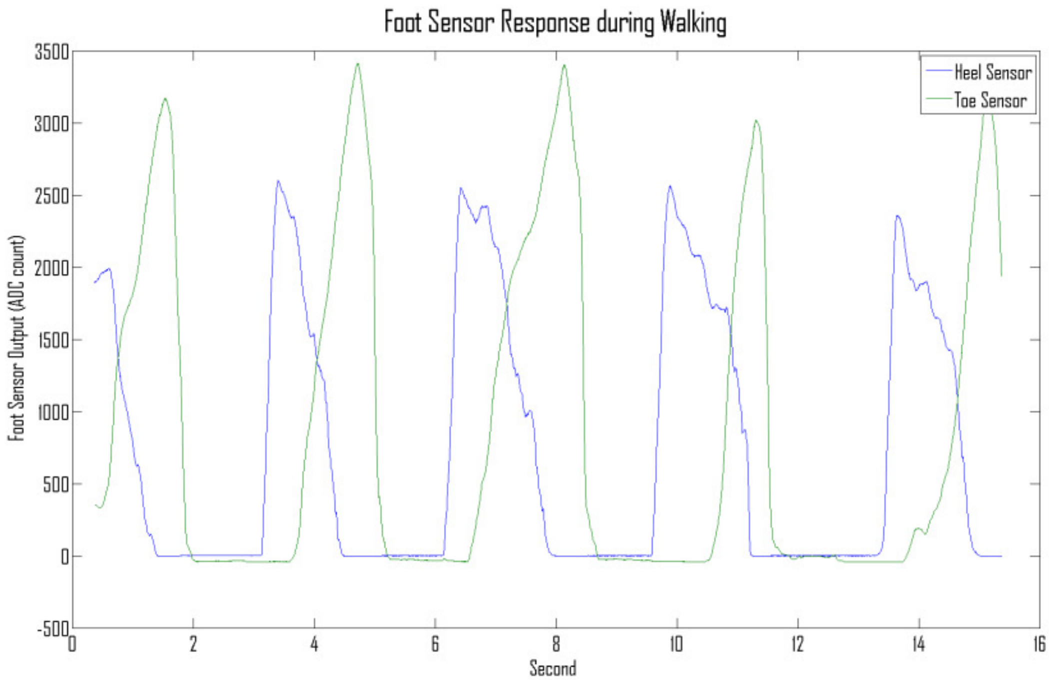


Figure 7-12. The response of the foot sensors during walking. Heel strike results in a large increase in the reading from the heel. This decreases as the user shifts his weight forward onto the toe, which causes the output of the toe sensor to increase. These readings both drop to zero as the foot comes off the ground through swing.

7.4 ANGLE SENSORS

The hip and knee angles are measured in the sagittal plane. They are measured by high precision relative encoders at the motor with potentiometers used to verify the absolute position. The geometry of the linkages is used to convert the motor rotation to the knee and hip joint angles.

8 HMI SOFTWARE

8.1 FINITE STATE MACHINE

The HMI Software is structured as a hierarchical finite state machine. A finite state machine has a finite number of states combined with state transitions between those states. For each transition, there is a gate, which is the set of conditions that must be met in order to transition to the next state.

The sensors added to read the motion of the user can then be added as conditions on the state transitions. Using these sensors as gates ensures that the person's crutches are on the ground before swing is initiated, ensuring stability. It can check that the stance foot is on the ground before the other foot is lifted off of the ground. Also, because these gates can easily be changed without disrupting the overall structure of the state machine, different sensors can be tested and thresholds can be adjusted quickly and easily for different users.

The state machine here is hierarchical, meaning that it has multiple levels that operate simultaneously. The top level governs the mode, which is the type of action that the user is doing. The modes include: walking, start walk, end walk, seated, stand up, standing, and sit down. The allowable transitions from one state to the next are shown in the diagram below. From the standing position, the user can either sit down or start walking. From start walk, the user enters walking, then proceeds to end walk, and back to standing. From sit down, the user transitions into the seated position. From seated, the user can stand up. The user can begin and end use of the exoskeleton either in the seated or standing position depending on whether the exoskeleton was donned while seated or standing. All spinal cord injury patients will don the exoskeleton while seated, so the exoskeleton should start in the seated position. However, developers and other able bodied users can also have the option to begin in the standing position.

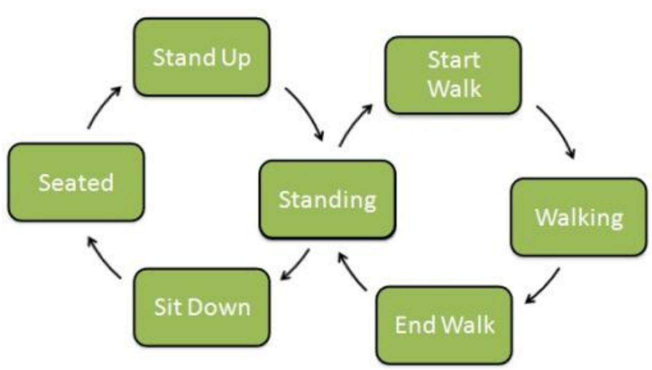


Figure 8-1. The state machine for the top level modes of the HMI. The user can begin and end in seated or standing. Only the transitions shown with arrows are permitted to improve safety of the user.

Each of the top modes shown above contains finite state machines. This hierarchical structure maintains the simplicity of the state machines while allowing for additional states and guards to be added. For Standing and Seated, there is only one state. The only allowable transitions are to remain in the current position or continue to the next top level mode. The other modes have more complex state machines and are described in the following sections.

8.1.1 WALKING

The finite state machine structure is well suited for the walking cycle. The walking cycle is made of four finite states: left leg swing, left double stance (stance with the left leg in front), right leg swing, and right double stance (stance with the right leg in front). In order to transition from one state to the next, certain conditions must be met. For example, to transition from swing to stance, the knee is mostly extended and the heel contacts the ground. To transition from stance to swing, the heel comes off the ground and the opposite foot must be on the ground. Furthermore, the finite state machine protects from unwanted transitions, such as a transition from right swing directly to left swing without a double stance phase.

Even though in swing, one leg will be going through swing and the other will be in stance, the state machine treats both actions as part of one state. This minimizes the complexity of the system and inherently prevents situations like double swing, where both feet are off the ground at once.

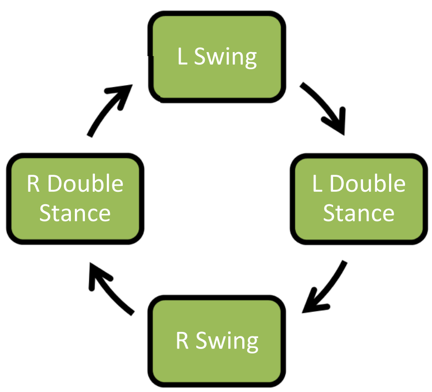


Figure 8-2. The walking cycle consists of four states: left swing, double stance (left foot in front), right swing, and double stance (right foot in front).

The user begins in one of the double stance phases. If the left leg is in front, the right leg will then enter the swing phase. Once the right leg hits the ground in heel strike, this begins right double stance. As the person shifts his weight forward and the left heel starts to come off of the ground, the left leg enters the swing phase while the right leg is still in stance, supporting the weight of the body.

For the HMI of the medical exoskeleton, the guards have additional conditions. This takes into account the user's safety and allows their gestures to control when the robot takes a step. Therefore, in order to transition from right double stance to left swing, the user must move the right crutch forward and put weight on it. In order to protect from both legs trying to swing at the same time, the right foot must be on the ground. To transition from swing to stance, the swing foot must contact the ground. Similar guards exist for all of the state transitions as shown in Table 1.

Table 8-1. The guards for the state transitions ensure that the user is ready to take a step and will be safely supported.

Conditions / State Transitions	Right Arm Forward	Right Crutch on Ground	Left Arm Forward	Left Crutch on Ground	Left Foot on Ground	Right Foot on Ground
L Double Stance → R Swing			✓	✓		✓
R Swing → R Double Stance						✓
R Double Stance → L Swing	✓	✓			✓	
L Swing → L Double Stance				✓		

8.1.2 Start Walk & End Walk

The start walk state machine looks very much like the walk state machine. However, the state machine begins with a transition from standing, so the feet are together. This means that the swing state is augmented as the foot is only moving half the distance as it would in a normal step. Furthermore, once a single swing state, left or right, is completed, the state machine automatically transitions into the walk mode. The conditions to enter swing from stance, however, are the same as during walking. The crutch must be on the ground and the opposite arm is forward. However, because both of the legs are together, either can lead the first step. In order to determine which leg should step first, the HMI determines which arm is in front and the opposite leg will swing.

Table 8-2. Guards for transitions between states in the start walk state machine.

Conditions / State Transitions	Right Arm Forward	Right Crutch on Ground	Left Arm Forward	Left Crutch on Ground	Left Foot on Ground	Right Foot on Ground	L Arm in front of R Arm	R Arm in front of L Arm
Stand → R half Swing			✓	✓		✓	✓	
Stand → L half Swing	✓	✓			✓			✓
half Swing → Double Stance					✓	✓		

The end walk state machine begins when the user signals the desire to end the walking cycle by bringing their arms to the same angle. The leg which was previously in back, then swings forward, but only a half step, thus bringing the swing foot in line with the stance foot to end in the standing position. This state machine transitions directly to standing.

8.1.3 Sit & Stand

The exoskeleton normally begins in the seated position. It then moves to the stand up state machine when the user signals a desire to stand. This is done by putting the arms back and the crutches on the ground. The user must start standing from this position because the exoskeleton does not have sufficient range of motion for the user to move their center of mass over their feet to remain statically stable while standing. As the user begins standing, their center of mass is behind his feet, so the crutches must be behind to help prevent the user from falling backwards. Once the user has transitioned into standing, the previously selected standing program helps stand the user. The details of these programs will be discussed in subsequent chapters. When the knees have fully extended, this indicates that standing is complete, and the HMI transitions into the stand mode.

From the standing position, the user can transition into the sit down mode by putting their arms behind them with the crutches on the ground. Like standing, this position provides support to keep the user from falling backwards as the sit down program lowers them to the chair.

Table 8-3. The guards for the state transitions from seated to standing, and standing to seated. The transitions from the sit down motion to seated and from stand up to standing are automatic based on the joint angles.

Conditions / State Transitions	Arms backwards	Crutches on the ground	Knees and hips at 0 degrees	Knees and hips at 90 degrees
Standing → Sit Down	✓	✓	✓	
Sit Down→Seated				✓
Seated → Stand Up	✓	✓		✓
Stand up→Standing			✓	

8.2 Initial Testing

Initial tests of the HMI were done using a model of the exoskeleton and LabView software. In order to test the state transitions without interference of the dynamics and control of the medical exoskeleton, I designed a simplified version to be used on able-bodied individuals which would allow the testing of all of the state transitions. The hardware consisted of an off-the-shelf brace system. Custom designed mounting brackets held potentiometers at the joints. These potentiometers were used to measure the joint angles. This platform allowed for testing of various foot sensors, arm bands, and crutch sensors.

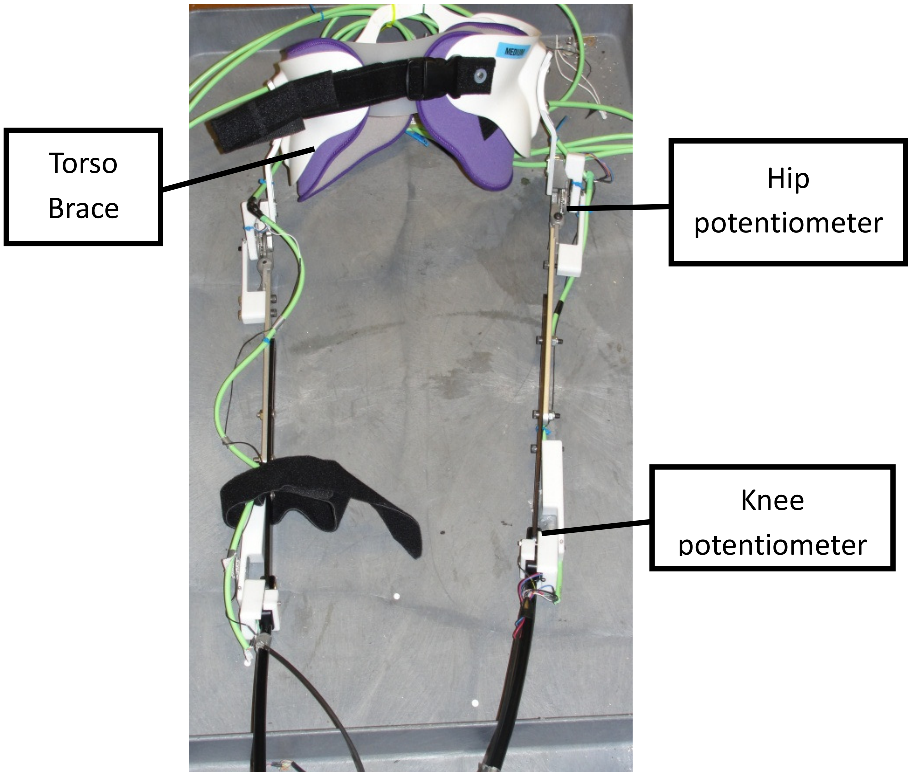


Figure 8-3. The HMI test platform.

This platform has an encoder on each hip and knee joint to measure the joint angle. The sensors are read into LabView along with the HMI sensors and foot sensors. The torso brace helps couple the sensors to the user.

LabView was used to test the finite state machine. A National Instruments PXI 1031 box was used to read in the sensor signals. Filtering of the sensor signals was done in software.

The HMI software tests showed that the angle measurements, foot sensors, crutch sensors, and upper arm sensors were sufficient to transition between the seated, standing, and walking states. In order to do this, I developed a front panel display which would show the pose of the user from the sensor readings as well as the assumed state from the state machine. The test subject then donned the test platform and proceeded to walk, calling out their state as they did so. The front panel display was used so that the test administrator could see if the HMI software identified the proper state.

The front panel display is shown below (Figure 8-4) displays the raw and filtered sensor values, including the arm angles, crutch output (on/off the ground), joint angles, and foot sensor values. The threshold of the arm angles could be adjusted from the front panel as well. The center of the front panel (#2 in Figure 8-4) gives a visual representation of the person’s body pose. This is used for debugging and sensor verification. The rightmost section (3) shows the output of the HMI state machine. It displays the mode (Sit, Stand, Walk, End Walk, Start Walk) as well as the state of each leg.

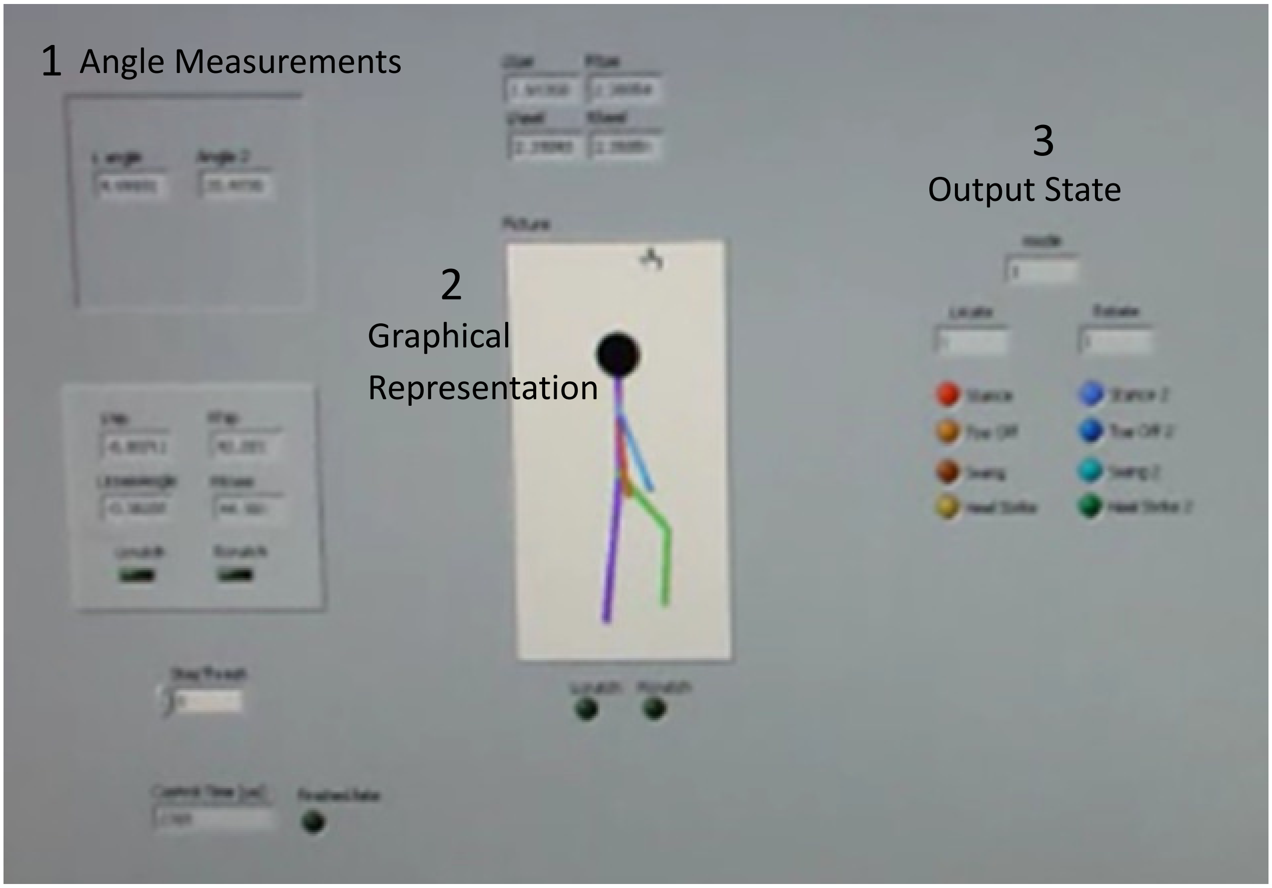


Figure 8-4. The front panel display of the HMI test software.

The display shows the raw sensor values (1), a visual representation of the person's body pose (2), and the output of the HMI state machine (3).

This set of tests showed that multiple users could reliably transition from seated to stand up, then to walking, and back to seated. Figure 8-5 shows the results of testing walking. Since the user was able to control his knee, the knee angle indicates what state the user was in. As expected, right before the knee bends, the state machine correctly identifies toe-off (state 3). During swing (state 2), the knee is bending. The HMI then identifies heel strike (state 4) when the knee has completely straightened. This was delayed on the first right step to ensure that the HMI did not falsely trigger. Furthermore, this plot shows that many times, heel strike and toe-off are not identified because the person lands with a flat foot or picks their foot up rather than rolling onto their toe. Missing these states frequently happens during rehabilitation because of the user’s overall body pose, and so the HMI must respond to this situation by identifying stance and swing despite the missed transition states. The second right step shows such as case. However, the HMI is still able to accurately determine when the user enters the swing phase and the stance phase.

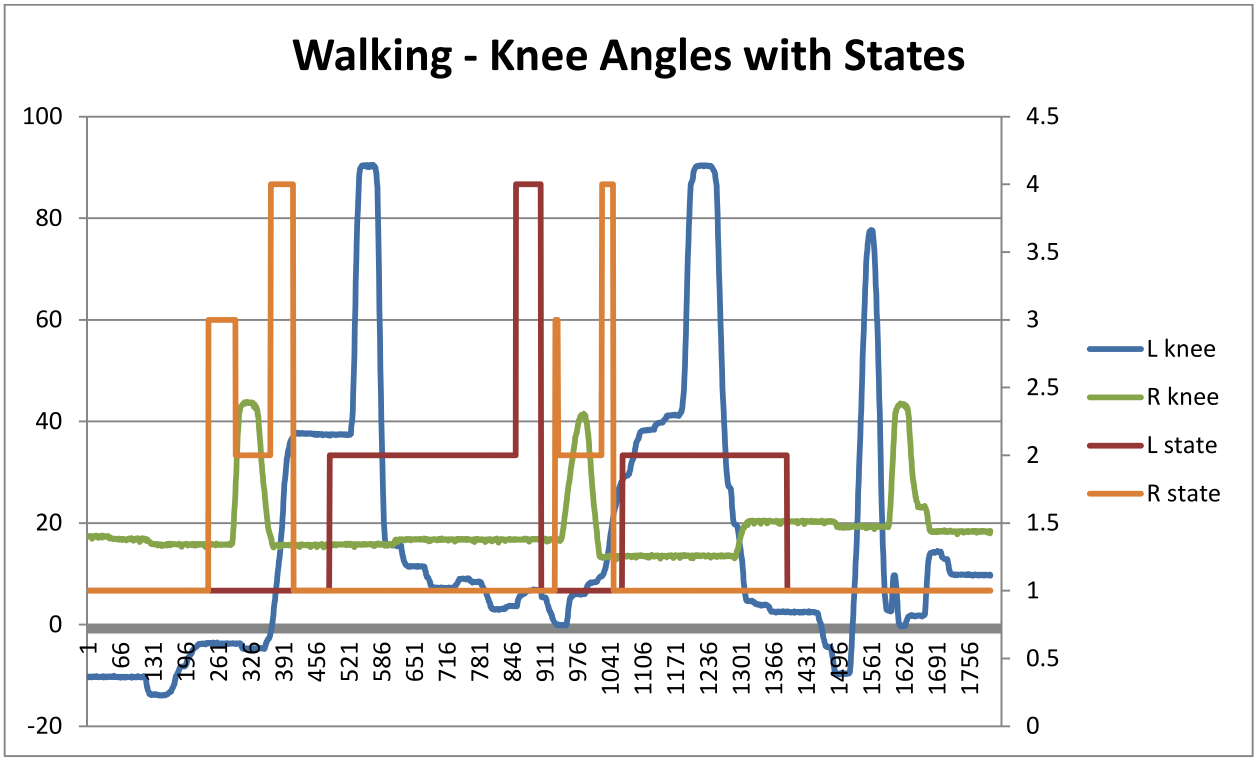


Figure 8-5. Results of HMI software testing using LabView and model system.

8.3 Medical Exoskeleton Testing

Once the medical exoskeleton was built and the controller for the step code verified, the HMI software and hardware were integrated into the system. The medical exoskeleton has integrated hip and knee measurements as well as foot pressure sensors. The arm angle sensors and crutch sensors are specific to the HMI and thus are not built into the exoskeleton. Both sensors are currently wired, though there are plans to make them wireless. In the wired configuration, however, the wires run up the user's arm, keeping them out of the way, and are plugged into the top of the electronics panel on the back of the exoskeleton.

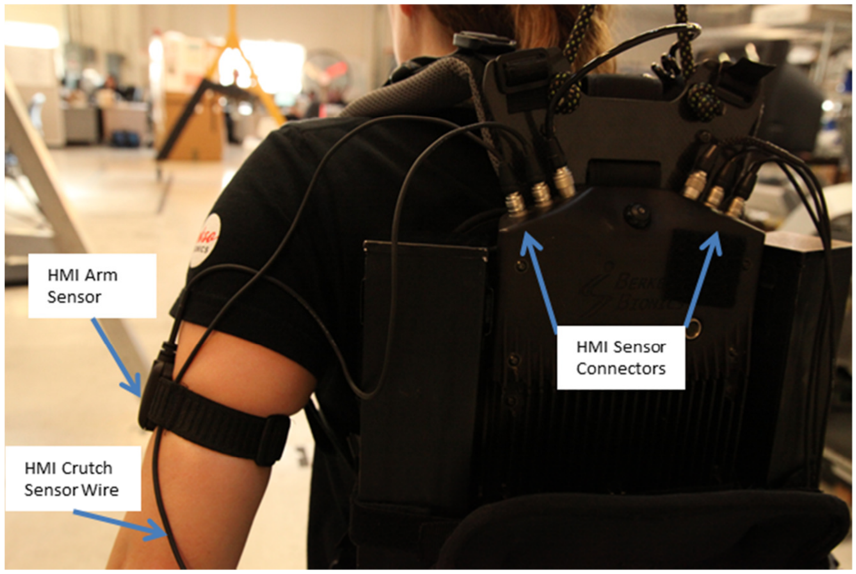


Figure 8-6. The configuration of the HMI Sensors on eLEGS.

The HMI Sensors are plugged into the back of the eLEGS device and the wires run down the arms to the arm band sensor and crutch.

Both the accelerometer/gyroscope board worn on the user's arm and the crutch sensors are read in directly to the main computer and filtered in the software. The accelerometer and gyroscope are filtered using a frequency-based filter. The accelerometer measures the steady state orientation while the gyroscope measures the transient orientation. Therefore, by combining these sensors, taking a higher weighting of the high-frequency signals from the gyroscope and, likewise, a higher percentage of the low-frequency signals from the accelerometer, we can get a more accurate representation of the orientation of the sensor.

The crutch sensor uses a linear potentiometer, which is read in as an analog value and converted to a digital signal. The potentiometer is used as an adjustable switch. The threshold of the potentiometer can be adjusted in software and is adjusted to trigger when a force slightly greater than the weight of the crutch is applied.

8.4 Preliminary SCI Subject Testing

The HMI was tested with able-bodied users before test subjects were able to use it. Subject 1 was the first paraplegic subject to use the HMI. . Subject 1 is a 24 year old male. He has a spinal cord injury at T9 resulting in complete paralysis (motor and sensory). The positioning of the arm sensors is critical for an accurate arm angle, and therefore the sensors must be calibrated for each use. However, once the sensors were calibrated, Subject 1 was able to reliably trigger steps. In future development, this lack of robustness will be addressed. Figure 7-7 shows Subject 1 taking steps using the HMI. He is able to reliably take a step, and the timing of the steps is very natural. In the 3rd frame, the right arm has just moved forward as the left heel begins to come off of the ground. As the heel comes off the ground, it progresses into the swing phase, which is triggered by the crutch placement. The crutches are then in the proper place to support the user while swinging through, which is critical for the patient's safety. The user's motions are natural and the motions needed to trigger the steps do not impede his ability to walk.

Subsequent tests of eLEGS and the HMI were done at the University of Virginia Gait Laboratory. During testing, a full body marker set was applied to the subject and the exoskeleton, and a Vicon 8 camera motion analysis system captured movement data. Sensor information was also collected from the exoskeleton processor. The Vicon data was collected to verify the angles of the sensors and the exoskeleton.

The first test wit this Vicuna system was done with an able-bodied subject who is familiar with walking in eLEGS with and without the HMI. The step speed was set the same for both runs. Therefore, the only variation in walk speed is the length of the double stance phase, which depends on how long it takes the person to shift their weight forward so that the trailing leg can flex. As the weight shifts forward, the trailing heel lifts off of the ground. This can be measured by the exoskeleton foot sensors. In normal walking, the heel coming off of the ground immediately precedes the knee flexion and the start of the swing phase. However, because the eLEGS system is waiting for a trigger to start the swing, there is a delay. Decreasing this delay is essential to making the walking natural and smooth. This delay gets smaller as the weight shift becomes more natural and the step is triggered quicker. This test was designed to determine the difference in fluidity of step based on this delay.

eLEGS can be operated in "PT Mode", which is when the physical therapist or operator pushes a button to initiate the step. Initial tests were done with able bodied users so that the effect of learning how to use the machine did not influence the timing. For an able bodied user walking in PT Mode, the average time between the heel coming off of the ground and the swing phase beginning was 0.938 seconds (Figure 8-8). This can be seen in the delay between the heel pressure going to zero and the leg state switching to swing

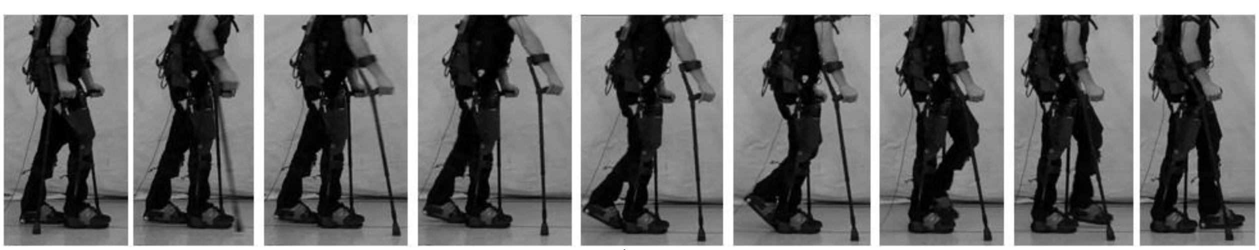


Figure 8-7. Subject 1 walking while using the HMI to trigger steps

(shown as 1).

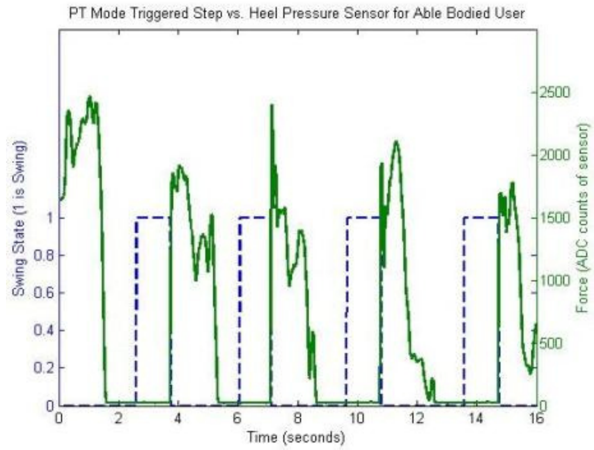


Figure 8-8. The swing phase is triggered by a push of a button (PT Mode) rather than by the HMI. The heel force value shows when the heel lifts off of the ground.

The next test was done with the user utilizing the HMI mode of the exoskeleton. The average time after the heel is off the ground was only .474 seconds. This improvement indicates that the walk is more natural in HMI mode and is also faster. In fact, in 16 seconds, using HMI mode, the user was able to take 6 steps as compared to the 4 steps that he could take using the PT mode.

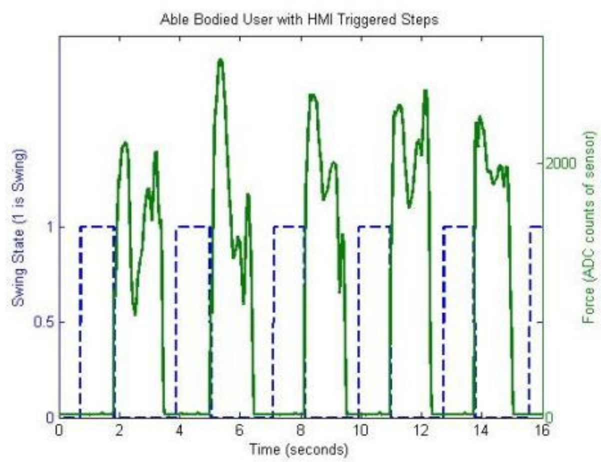


Figure 8-9. An able bodied user using eLEGS in HMI mode to trigger steps. The time in double stance is decreased in comparison to the PT triggered steps.

The exoskeleton was also tested on spinal cord injury subjects. Since the time between the steps is dependent on the user's ability to shift his weight forward in preparation for the next step, for a spinal cord injury subject, the effectiveness of the HMI in terms of timing steps is demonstrated by the difference between when the HMI starts the step and when the Physical Therapist or other operator would trigger the step using the control pad.

While the HMI allowed an able bodied subject to walk faster and trigger the steps quickly, it takes longer for a spinal cord injury subject to adjust to this new system and become proficient. Also, because the subject is still limited in speed by shifting his weight forward, we cannot expect him to be as fast as the able-bodied user. Figure 8-10 shows the transition from stance to swing when the step was triggered by the physical therapist when eLEGS is used by Subject 1. The point at which the HMI would have triggered the swing state is also shown. There was very short delay between the swing phase when the physical therapist triggered it and when the HMI would have initiated the step. The second step was delayed because the subject was not leaning on the crutch, which may indicate an unsafe step initiated by the PT. The average delay over all of the steps is only 0.25 seconds. It is expected that this delay will decrease as the user becomes more familiar with the machine.

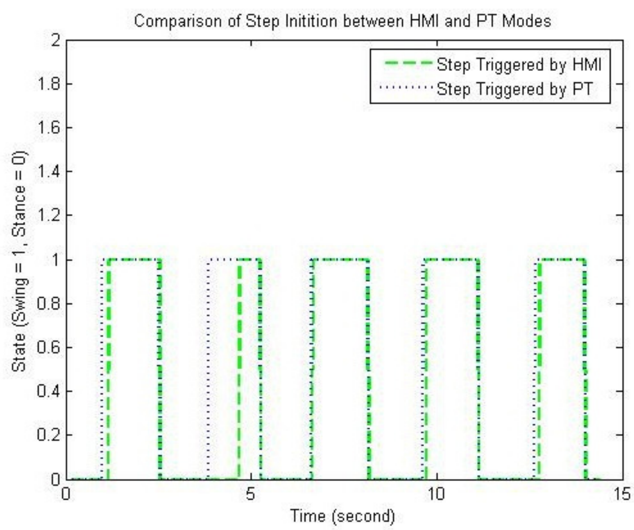


Figure 8-10. The steps shown here were triggered by a physical therapist. However, the dashed line shows when the HMI would have initiated the step. The lag between the steps averages only 0.25 seconds.

The HMI is intuitive and easy to learn to use. In order to quantify this, eLEGS was tested by eight other subjects at the University of Virginia, Kluge Research Center. Four of these users were able to test the HMI (Table 2). The remaining four did not reach proficiency in the two-hour session to move to using the HMI. The subjects were introduced to the exoskeleton and the HMI during a two-hour session. During this session, anatomical measurements were taken and the exoskeleton was adjusted for them. They were given a demonstration of the exoskeleton and then began walking with the device with a walker for support. Once they were comfortable with the walker, they began using forearm crutches.

Those subjects who were comfortable on the forearm crutches were taught to use the HMI. Out of eight subjects, four were able to progress to using the HMI. All four of these subjects were able to successfully use the HMI to walk in eLEGS. The subjects presented a range of injury levels and completeness and yet were able to use the HMI without requiring customization of the program.

Table 8-4. Five subjects with varying spinal cord injury levels and completeness have tested the HMI with eLEGS. Subject 1 tested the HMI in the laboratory setting while Subjects 2-5 used eLEGS at the University of Virginia.

Subject Number	Injury Type	Injury Level	ASIA Completeness
1	Paraplegic	T9/10	A
2	Friedreich's ataxia	N/A	N/A
3	Spinal Cyst	C7/T10	C
4	Paraplegic	T5/6	A
5	Quadriplegic	C5/6	C

Subject 2 is a motor incomplete spinal cord injury patient with Friedreich's Ataxia. Subject 2 was introduced to the exoskeleton at the beginning of a 2-hour long session. He walked in the exoskeleton for about 30 minutes. After this time, he was introduced to the HMI and walked about 50 feet before data was taken.

Both subjects were able to quickly learn the HMI. Both had an increased time between heel off and the step as compared to the able–bodied user; however, this time was decreased as they gained experience. While the new user had an average lag time of 0.859 seconds, the experienced user was able to reduce the lag time to 0.590 seconds.

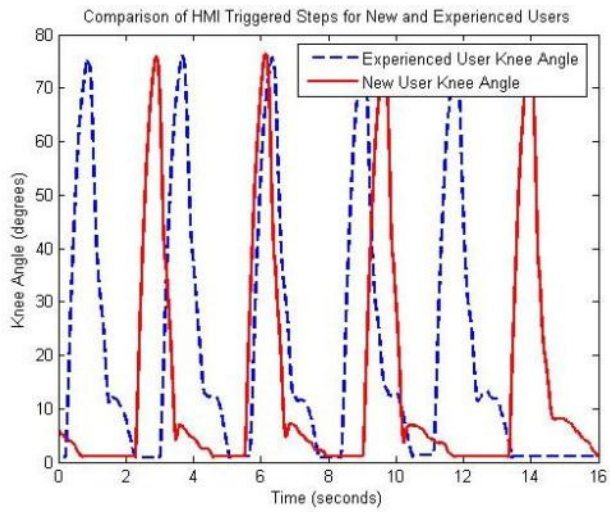


Figure 8-11. Comparison of the HMI Triggered Steps for a New User and an Experienced User.

Figure 8-11 shows a comparison of the knee angle of the users over time. Both users were able to maintain a consistent cadence to their steps, which indicates that the HMI, though slower than walking for an able-bodied subject, is steady. Once the user has learned how to use the HMI, it becomes a natural step in the walking cycle.

8.5 Improvements to the HMI Sensors

The original 2 Degree-of-Freedom sensor does not allow for measuring out of plane motion. However, this does not accurately capture the motion. Originally it was thought that measuring the motion in the XY plane of the sensor would be sufficient, and this did provide acceptable results. However, by measuring the motion in all of the sensor planes and then combining them to determine the motion in the user's sagittal plane, we can achieve much greater robustness. Furthermore, during the timeframe of this research, the cost of inertial measurement sensors dropped sufficiently such that a 6 degree-of-freedom sensor is now a viable option for a consumer product of this level.

The 6 degree-of-freedom sensor allows for the measurement of out of plane motion. This gives a more accurate picture of what the user is doing. We can continue to use the sagittal plane angle, but our estimation of the sagittal plane angle is much more accurate due to the increased information.

9 Classification for Verification of Sensor Choice

In order to verify the selection of the features used for the HMI, it is useful to compare data of when the step should be taken to when the step was taken based on the HMI algorithm. Since there are many possible features which could indicate that a step should be taken, a method must be used that can handle the high dimensionality of the state space. Classifiers provide this feature. In order to allow for an effective method of classifying while allowing for some error points, a support vector machine is used.

9.1 Support Vector Machine (SVM)

A support vector machine uses supervised learning to use a set of training data to determine a classification by determining which category a data point should fall into. The classification can then be used on uncategorized data to predict which category the points are in. This is done by finding a hyperplane that creates the maximum distance to the nearest training points. The vectors from these nearest points to the separating plane are the support vectors.

A support vector machine can be used to find a separating plane between data points which are given by features of the user, machine, or sensors. By creating a separating plane, the relevance of each feature in determining the desired state can be ascertained. Furthermore, this plane can be used with new data to determine what state the point belongs in.

In this application, a best-fit N-dimensional plane is used to fit the data, where N is the dimension of the data. Because there is a large number of inputs to the system and thus a high dimensionality, a linear function is used as a separating function in order to simplify computation.

9.2 Development of the SVM for the HMI

Data sets from walking were used where the button on the control pad was used to trigger the step. The data was then labeled to determine when the step was taken. To do this, the data at the time that the button was pushed and a few steps after (as taking the step a few control cycles late is acceptable) is labeled as “step” data. The rest is labeled as “not step”. I then use this data to train the support vector machine and determine the separating hyperplane. This returns the equation for a plane, and based on the magnitude of the coefficients for the matrix, we can determine the relative importance of the variables. In this way, we can determine which features to consider or sense in the HMI.

Initially, I considered the relationship between two individual features. This is easier to visualize and gives a good indicator of what to investigate as a feature. The classifier here compares the relationship of the knee angle to the opposite (normalized) arm angle. This shows that these two variables both have an influence on determining if the system should step or not. This is as expected as the knee angle increases when the step is instigated, thus indicating that a step is taken. However, it is evident that many of the training data for small knee angles were misclassified in the verification step. This is because the distinction was clearest after the knee had started to buckle, which indicates that the step had been taken. Instead, we want to be able to determine when a 0 or slight knee bend (due to the buckling of the knee as the weight shifts forward) indicates an intended step.

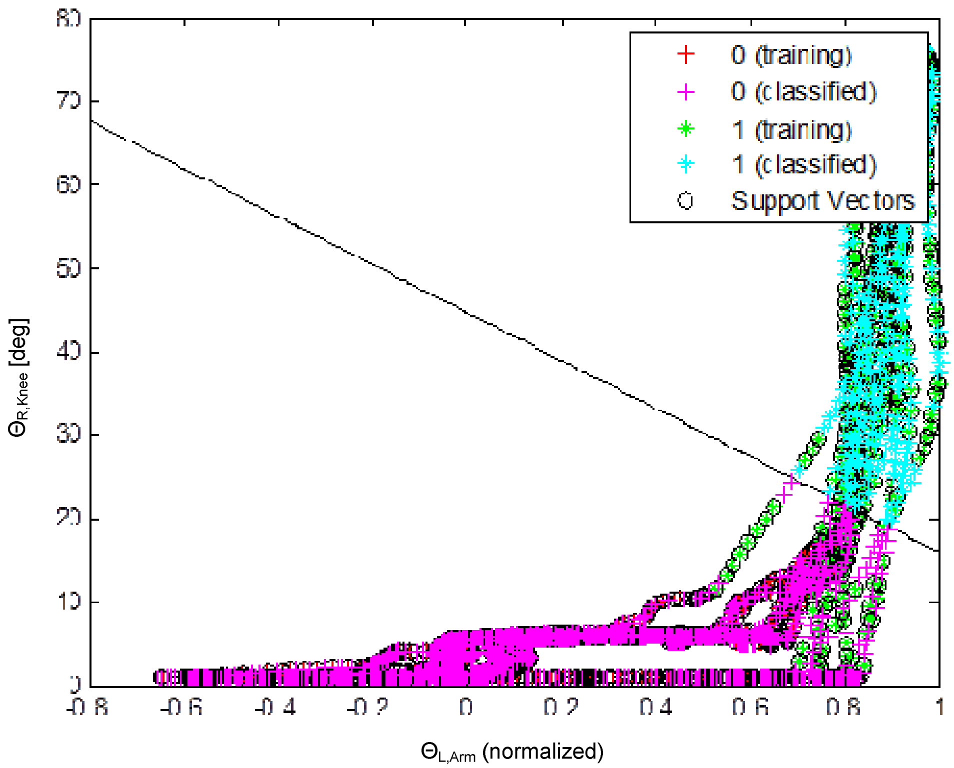


Figure 9-1. Classification of Intended Step using Right Knee Angle and normalized Left Arm Angle features.

The next two features considered were the arm angle with the opposite hip angle. The classifier was able to determine stance and swing easily from the hip angle and the arm angle with over 93% accuracy. However, since the data shown uses the hip angle during the entire swing cycle, it is difficult from this result to determine if the hip angle may play a more critical role if data is only considered when a right step is allowable (i.e. during left stance). By focusing only on these points, we find that the hip angle is more relevant.

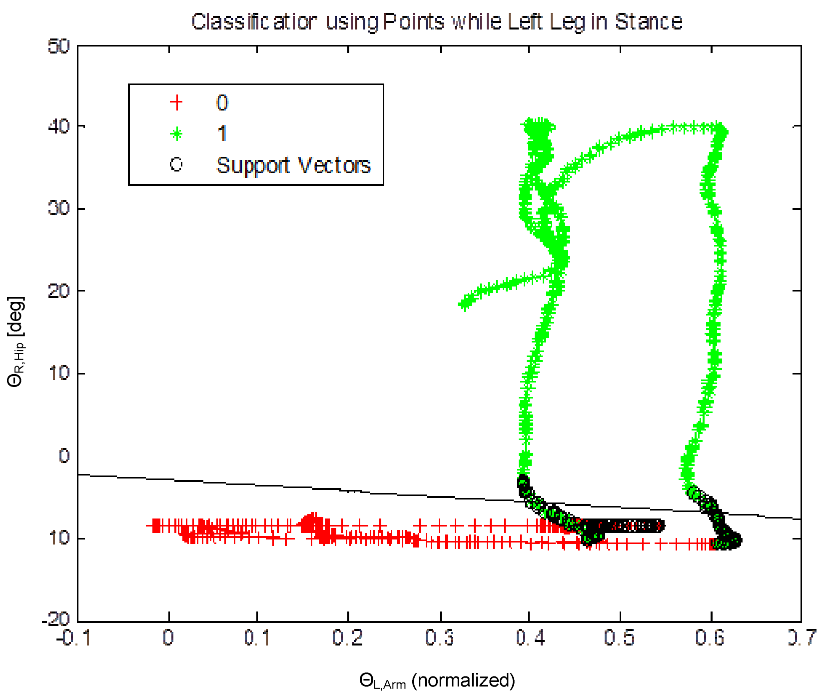


Figure 9-2. Training Data and Classification of Intended Step using Right Hip Angle and normalized Left Arm Angle features using only data points when the left leg is in stance.

Based on this initial finding, we should not consider taking a step if the hip angle is above the line shown. However, this does not change considerably with the arm angle. Furthermore, we do not expect the hip to be positive before the step is taken since that is the trailing leg. Therefore, this feature does not aid in determining when to take a step.

It is reasonable to expect that the foot sensor data would influence the decision to step or not. For example, when a person rolls forward onto their toe, this indicates that the weight shift is complete or ending and thus a step is about to be taken. Looking at the heel and toe sensors of the right foot during the time when the left foot is in stance, we expect to see some trend for classifying. However, the classifier is unable to converge to a classification for these sensors with a linear, quadratic, or cubic function.

The next feature explored is the Arm Angular Velocity. The arm is swinging forward and comes to a stop as the crutch hits the ground, indicating the position for a step. Therefore, it would be reasonable to assume that a low velocity indicates a step should be taken. However, based on using all of the data points, it is clear that the arm is also stationary before the step is taken and therefore the arm angular velocity is not a good indicator that a step should be taken. However, bias on the arm angle, the classifier was able to classify the intended step 93.9% of the time. Many of the data points misclassified, however, are the transition steps and thus those most important to classify correctly.

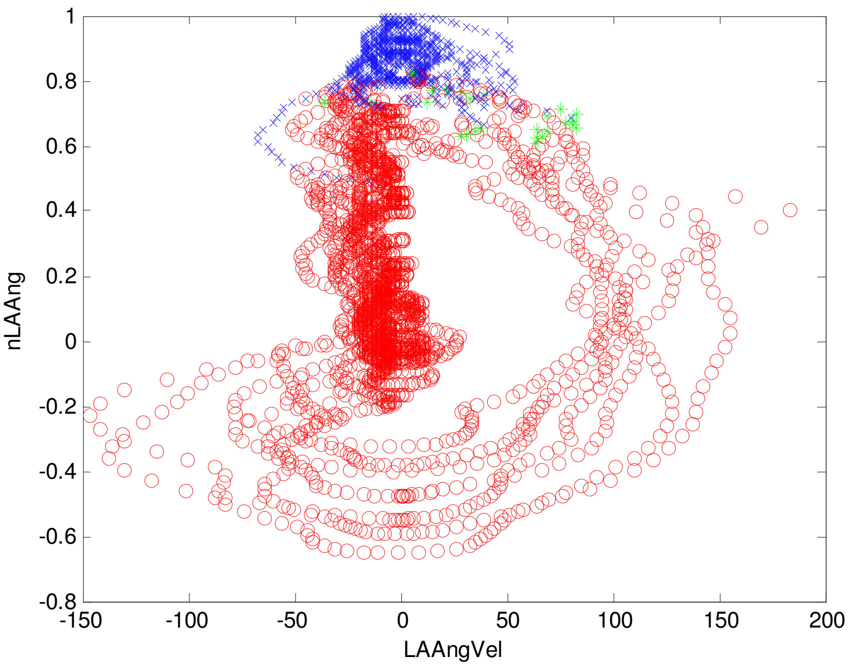


Figure 9-3. Training Data for Classifier using L Arm Angular Velocity and L Arm Angle as features.

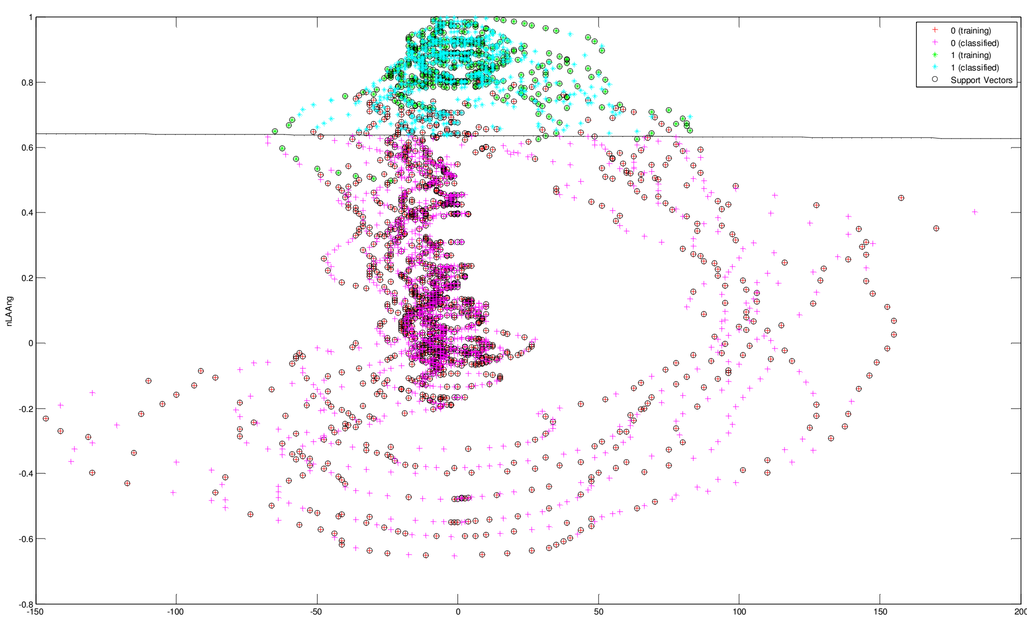


Figure 9-4. Classifier Results using L Arm Angular Velocity and L Arm Angle as features

While it may seem like the torso angle is a likely feature, tests across multiple users show that the torso angle is user dependent and thus not a good choice for a general classification tool. For many users, adjusting the torso angle to a proper walking gait is one of the hardest parts of learning to walk in the exoskeleton. Some users must use their torso more to help propel them forward while others maintain an upright posture. While an upright posture is desired, this comes with coaching and becoming more proficient at using the exoskeleton, and thus it is not desirable to have the HMI dependent on this.

The classifier can be run in N-dimensions to compare more features. However, computation power is limited. Visualizing three dimensions, we can create a classifier using the Right Arm Angle, the Left Arm Angle, and the Left Hip Angle as features. Figure 9-5 shows the classification, which was able to classify with 94.49% accuracy. The weighting vector for [Right Arm Angle, L Arm Angle, L Hip Angle] is [-.0084 -0.0710 -0.0130] for this dataset indicating that the strongest indicator of an intended step is the Left Arm Angle. This conclusion is upheld using the left arm angular velocity, the left arm angle, and the left hip angle. The weighting vector is [0.0013 -0.0641 -0.0153]. This validates the decision to use an arm angle measurement to determine the step data. In the future, this method could also be used with motion capture data to analyze more features that are not sensed directly by the exoskeleton.

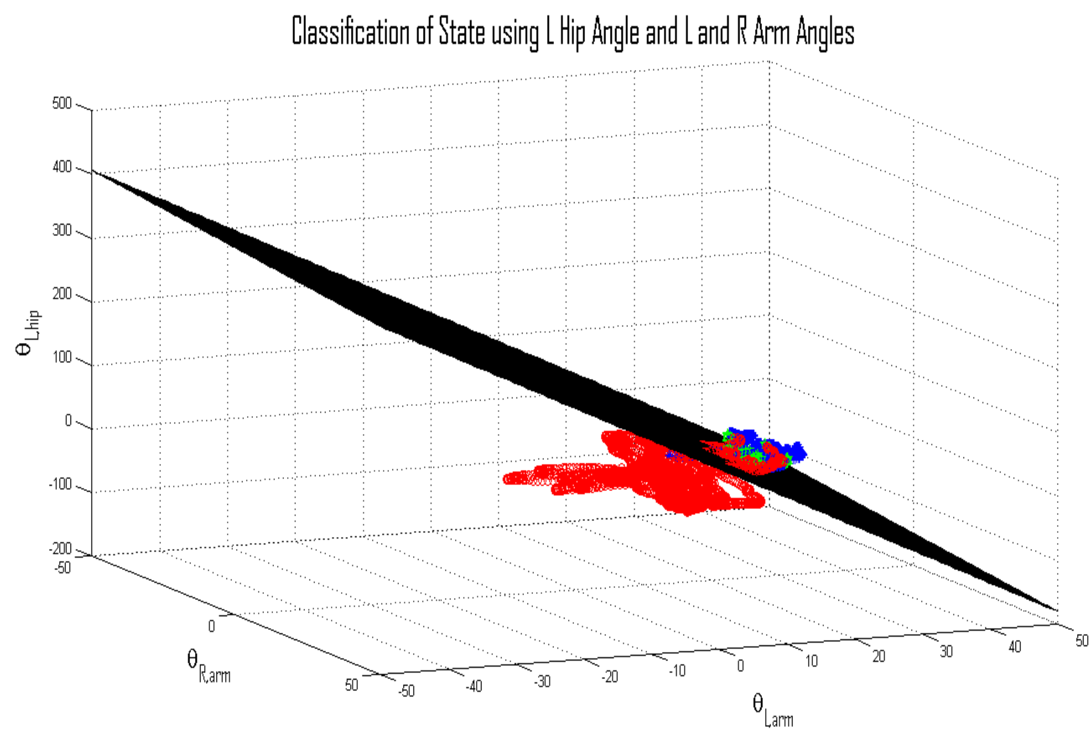


Figure 9-5. Classifier using the Right Arm Angle, Left Arm Angle, and Left Hip Angle as features.

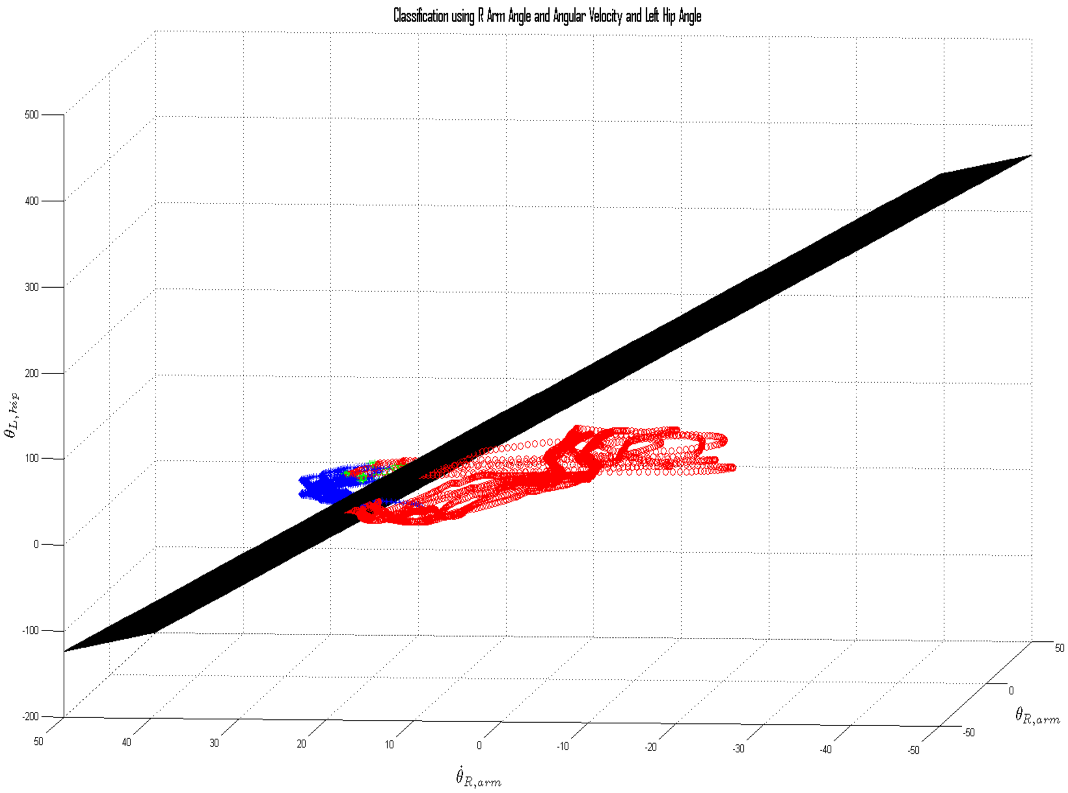


Figure 9-6. Classifier using Right Arm Angle, Right Arm Angular Velocity and Left Hip Angle

10 CLINICAL TESTING

The eLEGS exoskeleton has been tested at a variety of rehabilitation centers throughout the United States. During these clinical sessions, the main purpose of the visit is twofold: first, to train the physical therapists to use eLEGS and secondly, to determine what type of subjects can successfully use the exoskeleton and what to expect of their outcome. Most subjects are brought in for a single visit, but some are brought in for up to four visits throughout the week. The first session with a subject is two hours long to perform measurement and evaluation. Subsequent sessions are 1 hour, which reflects a typical rehabilitation session.

Each session begins with an explanation of the machine and the goals of the study. The subject then consents to the study and is measured and evaluated by the physical therapists. Once the evaluation is complete and the subject is determined to have met the inclusion criteria for the study, the subject is shown a brief video of the exoskeleton to introduce them to the features of the exoskeleton and what it will do. The exoskeleton is then sized for the subject and the subject transfers into the exoskeleton. Once they are strapped in and the exoskeleton is set up properly for them, they stand using a walker. From this point, the therapists continue with a rehabilitation session as they see fit for the subject. They will practice balancing and shifting weight from foot to foot and front to back. They will then start walking using the walker as a support device. If the subject demonstrates competency on the walker, they will move on to the forearm crutches. Many of the balance and shifting lessons may be repeated with the new walking aid. The subject will then learn to walk with the crutches. Once the subject has mastered crutches, the subject will be introduced to the HMI. Once the sensors are fit and calibrated, the subject will begin walking with the HMI sensors and the physical therapist only spotting.

During these two-hour sessions, it is rare that a subject progresses sufficiently to try the HMI, especially since the therapists are also learning how to teach and coach the subject during this time. Not all subjects reach the HMI mode of control during their rehabilitation sessions. For those who return, in follow up sessions, they may reach the HMI. At the end of each session, the subject does a 2 minute test, which provides a metric to determine how well the subject is doing.

One study was done at Shepherd Center in Atlanta, GA. During this week, nine users were brought in to use the exoskeleton. Of these users, three were able to progress sufficiently to try the HMI. This number may have been increased with experienced physical therapists, but since they were also learning how to coach the user, progress was inhibited.

The users who were able to use the HMI were able to do so after only about 30 minutes of walking time in the exoskeleton. Subjects 3 and 8 were able to use the HMI after only 1 visit, and Subject 1 started using it on his second visit. These subjects, however, had incomplete injuries. Therefore, the sensory feedback and limited muscle control they possess may aid them in the ability to learn to use the exoskeleton quicker. However, their level of injury did not affect their ability to use or learn the HMI once the use of the exoskeleton was mastered. When the users switched to using the HMI, their speed was not inhibited. In fact, for many of these users, they were able to walk faster using the HMI. The only exception to this was Subject 1, who was able to reliably transfer his weight and support himself and thus was able to switch to a 2-point walking gait rather than a 3-point gait, which means that his opposite arm and leg were moving synchronously. The HMI does not support this gait, which is faster.

Table 10-1. Subjects at Shepherd Center and their final support device. Three of the users progressed to using HMI.

Subject	Level	AISA	Final Walk Aid
1	C6	C	HMI
2	T10	A	Crutches
3	L1	C	HMI
4	C6	C	walker
5	T1	A	walker
6	T9	A	walker
7	T7	A	walker
8	C2	D	HMI
9	L1	C	crutches

The following table shows the subjects who reached the HMI during a clinical week session at Shepherd Center (Atlanta, GA) and a second week at Craig Hospital (Denver, CO). Those users who were able to use the HMI at Craig Hospital were both complete injuries, but required three sessions before progressing to the HMI.

Table 10-2. This shows the subjects during a clinical week session and their 2 minute speeds.

Subject	Visit #	Repeat?	Injury Level	AIS	2 min speed [mph]
S3	1	n	L1	C	0.31
S8	1	n	C2	D	0.36
S1	2	y	C6	C	0.49
C1	3	y	T4	A	0.38
C2	3	y	T8	A	0.29
S1	4	y	C6	C	0.72

While these results are somewhat limited due to the constrined duration of these sessions, they do indicate that the HMI can be learned quickly. Furthermore, the walking speed does not decrease when switching to this new method. Subject S1 used the HMI for the 2 minute walk on his 2nd and 4th visits. During the 1st and 3rd visit, he used the crutches. (He switched back to crutches on day 3 to improve his skills and speed). On the first day, his speed was .49 mph, while he increased his speed on the 3rd and 4th days to .71 and .72 mph respectively.

Table 10-3. Subject S1's 2-minute test shows that the speed does not decrease as a function of the use of the HMI versus the physical therapist initiating the step with the controller.

Subject S1 2 Minute Test		
Visit	**Speed (mph)**	**Walking Aid**
1	0.49	crutch
2	0.49	HMI
3	0.71	crutch
4	0.72	HMI

Subjects C1 and C2 had similar results. On their third visit, they both used the HMI, but were unable to use it on their fourth visit due to technical glitches. Subject C1's speeds for each visit were (in mph): 0.37 (Visit 1), 0.31 (Visit 2), 0.38 (Visit 3), and 0.43 (Visit 4). We see an improvement in his gait speed, which is not hindered by the switch to the forearm crutches. While he was .06 mph faster on the first day, this could be explained by extra assistance from the physical therapists as he was learning to weight shift and may not be significant. However, as the subject progresses, the physical therapist is simply spotting, and so the increases in speed are due to their proficiency with the device. Subject C2 showed very consistent walking speeds throughout his four visits. He walked at 0.26 mph on Visit 2 and only improved to 0.28 on Visits 2 and 4, and 0.29 on Visit 3. This shows that the HMI did not impede his speed progress.

11 ADAPTIVE THRESHOLD

A critical aspect to the robust implementation of the HMI is the sensitivity of the arm sensors. The arm angle is used as a threshold to signify when the step should be taken. This may vary from person to person depending on how they walk with the crutches and how the sensor is placed on their arm. However, the threshold should be sufficiently high that it does not accidentally trigger a step and sufficiently low that the user does not have to adjust his gait to get the sensor to trigger. As the arm sensors are worn on the arm, they are subject to twisting, being placed in different spots for various people, and rotating during use. Anti-slip elastic bands and pads on the sensor box help to mitigate this, but do not solve the problem. Furthermore, for each user, the sensor must be calibrated before it can be used. This means that the user must hold their arm at 90 degrees and then at 0, to allow the accelerometer to determine the limits. This is very difficult to do independently because the person does not have the stability of the crutches when holding their arms straight forward. Furthermore, the user must hold the control pad and push a button to confirm that their arms are in the proper position. This is difficult to do while using the crutches as support.

Unlike many applications, exact knowledge of the angle is not necessary. Instead, a relative magnitude can be used. Therefore, we can utilize techniques which cannot be used for inertial navigation or position alignment such as tracking or mapping where an exact location is needed.

Using adaptive threshold techniques allows us to develop a system which is robust to calibration or positioning errors. In addition, this technique allows us to use the relative measurement of the arm angle within each stride allowing the system to adapt to the users gait. The calibration routine for the adaptive threshold procedure is also simpler than the calibration procedure discuss above. For this implementation, the gyroscopes are calibrated by holding the sensor stationary; however, the accelerometers are not calibrated. Thus some misalignment can still be observed in the sensor data. This misalignment is adjusted for by modifying the current threshold with respect to a moving average of the minimum and maximum arm angles.

The threshold is a function of the minimum and maximum angle seen during the step. At each step, the threshold is updated by the following filter:

$$NewThreshold = 0.85 * PreviousThreshold + 0.15 * \left[\frac{1}{2}(ArmAngleMax - ArmAngleMin) + (ArmAngleMin)\right]$$

When the user is using the HMI independently, this method will still require the user to pass the original threshold to take a step. Therefore, the only means by which to reduce the threshold angle is to move the arm back significantly in order to reduce the minimum angle, thus making the average arm angle smaller. However, if the user is walking with a physical therapist or other operator, the step button can still be used as an override. In this method, the button can set the max arm angle as it triggers the step.

The adaptive threshold can be used throughout walking to continue to adapt the arm angle, which helps adjust to any movement of the arm band or changes in crutch motion. It can also be used in a training period during which the user "trains" the exoskeleton to determine the threshold and then holds that threshold throughout the use of the exoskeleton.

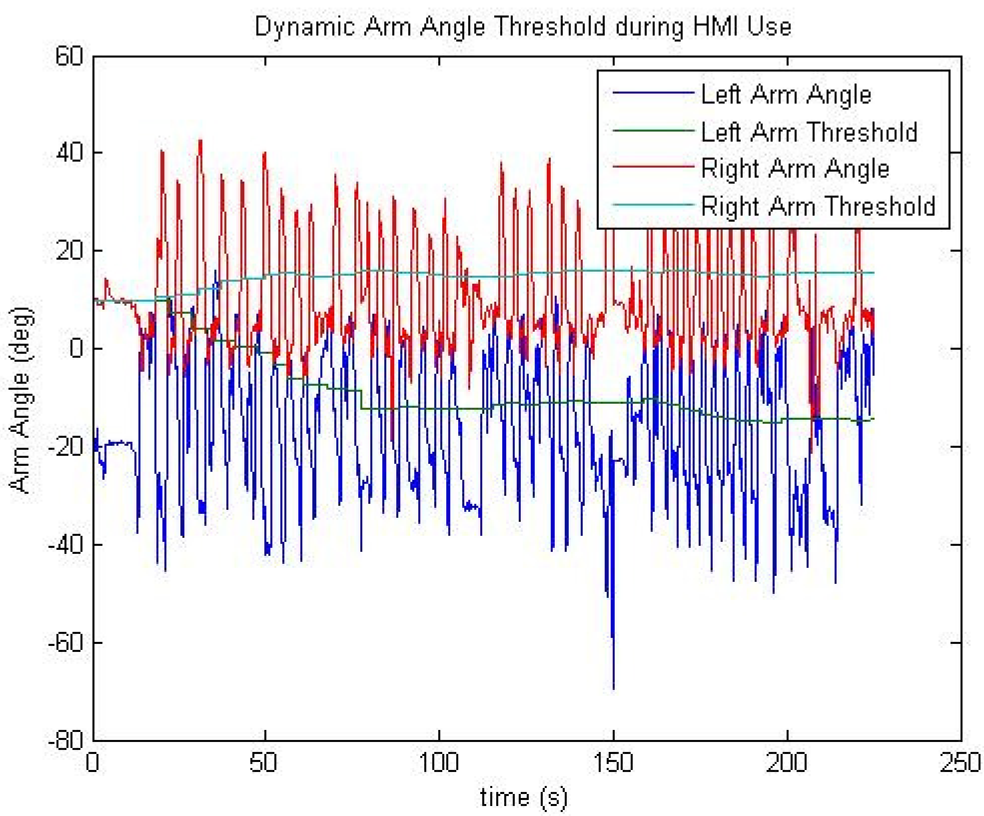


Figure 11-1. The HMI dynamic arm angle limit finds an acceptable arm angle threshold. The left arm sensor is skewed on the user's arm and therefore the arm angle threshold is adjusted from 10 degrees to approximately -15 degrees through the walking cycle.

Figure 11-1 shows a trial during which one of the arm sensors was purposefully skewed on the arm from the alignment of the opposite sensor. The gyroscope was calibrated, but the accelerometers were not. It is evident that if the original threshold parameter was unchanged, the right steps would not have triggered as the left arm rarely got above the original threshold. However, by utilizing the dynamic threshold program, the arm angle threshold adjusts for both arms to make them more reliable and easier for the pilot to trigger the steps reliably. Furthermore, after about 70 seconds of training, the threshold levels off as the pilot reaches a consistent gait and arm motion and the threshold converges to this behavior.

12 EXTENSIONS: STEP LENGTH ADJUSTMENT

The Human Machine Interface (HMI) must determine not only when the user wants to take a step, but how long that step should be. Because the arm angle can easily be measured and the crutch position, which is proportional to the arm angle, determines how far the person can step and maintain stability, this is a logical indicator of step length. Rather than creating a mapping of arm angle to step length that the user must conform to, we want to create a means to learn what the user's natural mapping is. Furthermore, because the user cannot walk on their own, the training is done while in the machine. This means that incorrect steps could be uncomfortable for the user, so training should be minimal. For clinical trials, we would like for 5-10 steps to be sufficient to train the HMI. This means that the learning algorithm must be able to find an acceptable (though not necessarily optimal) solution to the mapping with only 5-10 sets of training data. However, the training data can give information such as how much bigger or smaller the step should be.

The machine must learn what arm angles correspond to what step lengths. Since this is a continuous mapping, we can determine a function which maps the two. We cannot, however, determine the exact function as this would require testing an infinite number of functions with a possibly infinite number of parameters. This would likely result in over-fitting also since for N trials, the best fit line would be an N-degree polynomial. Therefore, finding a function form a priori is necessary.

Based on known characteristics of the function, we can choose a form that adequately describes the mapping and then determine the parameters. The function should range from 0 to an adjustable maximum step length. The function should be positive, since an increased arm angle should always result in an increased or equal step length. The limit of the function should be an adjustable maximum step length.

The arc tangent function fulfills all of these qualities and has the added benefit of being defined by 2 parameters, rather than a cube root function which may have more parameters. This means that the state space over which we must search for the desired function is only 2-dimensional.

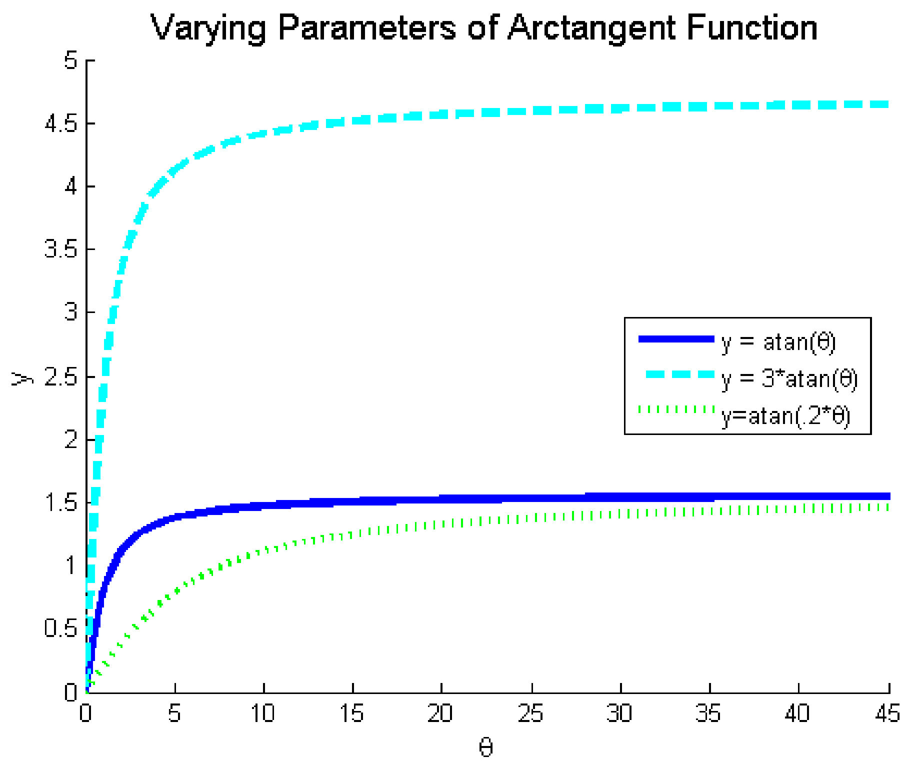


Figure 12-1. Varying parameters of the atan function provides different curve fits.

This function can easily be expanded to a third dimension to incorporate a deadband. The deadband accommodates a range where the user can move his crutch forward without triggering a step. This can be represented by a 3rd parameter which shifts the function so that the step length is zero below an arm angle threshold: max{0, A*atan(B*(θ*C))}. This can also be implemented such that step lengths below a certain threshold are treated as 0 length and not steps. This will prevent the user from shuffling forward using very small steps. This is represented by: if(steplength = A*atan(B*θ) < C), steplength = 0;. This third parameter is not addressed in this paper, but is the subject of future work.

12.1 Training Procedure

In order to train the system, the user will be wearing the exoskeleton with the HMI connected. The user will move their crutch and prepare to step. The HMI will use the current best estimate of the step length function to take a step. At the completion of the step, the uesr will give feedback on the step, either ranking it as way too small, too small, just right, too big, or way too big, or giving an estimate of how much bigger or smaller the step should be (in inches). The HMI will then update the function and repeat the process for the given number of trials.

By taking the step first, the user will get a better feel for exactly how long the step is and how comfortable they feel taking a step of that length. Since the feedback is based on the user's opinions, it will be somewhat stochastic. As the mapping function is adjusted throughout the training process, the difference between actual and desired step length will be reduced, enabling the user to give more accurate feedback as training progresses.

12.2 E-GREEDY ALGORITHM

12.2.1 GREEDY SEARCH

The state space can be discretized since the patient is unlikely to notice the difference between small variations in step length. In the worst case scenario, the maximum step length will be 30 inches. A variation of 0.05 of the B parameter will result in a maximum step length variation of 0.513", which is an acceptable difference. A difference of 1 of the A variable results in a change in step length of 1.5486", which is still acceptable. Therefore, the state space can be discretized from A = 0 to A=20 incremented by 1, and B from 0 to 1 in increments of 0.05.

In the ε-greedy algorithm, we start with an initial guess for the A and B parameters. The user then gives feedback on the step length (way too small, too small, just right, too big, way too big). If the step is too small, that step is given a value of -1 or -2 depending if it was slightly short or very short. This increases both the A and B parameters proportionally based on how short the step was. Likewise, if the step is too long, the A and B parameters are decreased based on how long they were. If the step is deemed "just right", the parameters are held constant and that combination is marked with a reward value of 1. For the remaining trials, we explore the states around that one that have not yet been given a negative reward. The mapping equation is the most recent mapping that received a value of 1. If no states have been given a value of 1, we guess the next state that we would have chosen to test and use that.

12.2.2 E-GREEDY SEARCH

In order to more thoroughly explore the state space, I added a bit of randomness to the search algorithm. Because only a small number of samples are taken, diverging from the current exploration path can be very costly if we were close to an optimal solution. However, without any randomness, much of the state space is left unexplored. In order to take into account both of these situations, the randomness was bounded to an area around the next step.

12.2.3 *Results*

Because at the time of this testing, there was not a working exoskeleton, the HMI algorithms must be tested in simulation. In order to obtain results, I established a series of "arm angles" and the "desired step sizes" that correspond to those arm angles. The output step was then calculated based on the current guess of the function and recorded.

Table 12-1. Arm Angle, Desired Step Length, and Determined Step Length from Greedy Algorithm.

Arm Angle	Desired Step Length (in)	Step Length (in)
45°	22	21.90
20°	10	21.78
40°	20	21.89
30°	18	21.85
25°	15	21.82
30°	17	21.85
45°	21	21.90
45°	21	21.90

This iteration results in two steps that are deemed correct. A lot of space, however, is left unexplored. This method seems to converge to an acceptable function. However, when we look at the final equation and compare that to the desired step sizes for the entire trial period, there are large errors. The final equation fit the final points very accurately, but did not fit the initial data well even after eight training steps.

Table 12-2. Arm Angle, Desired Step Length, and Determined Step Length from Greedy Algorithm.

Arm Angle	Desired Step Length (in)	Step Length (in)
45°	22	20.403
20°	10	18.477
40°	20	20.206
30°	18	19.621
25°	15	19.159
30°	17	19.621
45°	21	20.403
45°	21	20.403

Because the slope of the function found through this method was so steep, I altered the state space to search B from 0 to 2 in increments of 0.02. The increased resolution should provide a more accurate function as well as not allow the slope to reach as high, since the data shows that a slower curve is necessary. However, even with this alteration, the function was $y = 14\mathrm{atan}(1.32\theta)$. This slope is at the same point in the state space, but since the spacing is different results in a different function. The resulting function is very sensitive to the gridding of the state space.

The function is also dependent on the order of the training steps. This is because the search algorithm searches based on the current feedback, and thus favors the more recent steps. For this trail, the same angles and desired step lengths were used, but were given in a different sequence. We want an algorithm that is more robust to this as there is no "correct" order to take steps of various lengths and the patient should be able to vary their steps as they wish.

12.3 Non-Linear Regression

12.3.1 Non-Linear Regression Algorithm

In order to do non-linear regression, there must be an initial set of data to fit a non-linear function to. The algorithm requires an equation form, a preliminary guess at the parameters, and the trial data. The equation form is the arctangent function. The data can be obtained by doing the trials, but in order to improve the initial guesses, the function parameters are adjusted during the trials.

As in the greedy algorithms, we begin by choosing initial parameters in the middle of the range of typical parameters. These parameters may not be close to the actual for any given user, and in order to get a more accurate initial guess and to allow the patient to give more accurate feedback, the parameters must be adjusted throughout the training phrase. The errors are given in whole inches since the ability to estimate desired step length is limited in precision. The parameters are adjusted as follows: if the step is more than 7 inches too long, the parameters are both divided by 2. This ensures that the resulting steps will be much smaller. If the step is only slightly long (4 in or less) the A and B parameters are decreased by a 1/4. Likewise, if the step is more than 7 inches too big, the parameters are increased by 50%, and if it is slightly too long (4in or less), they are increased by 25%.

The data set used in the non-linear regression is the arm angle input and the desired step size as specified by the user. These are stored as a tuple. By using the desired step size, the algorithm fits a line to the data set which represents the step lengths that the patient desires.

12.3.2 Results

The non-linear regression finds a curve which accurately fits the dataset given. During the training, the parameters are adjusted giving different step lengths during training. Therefore, as in the ε-greedy algorithm, the step lengths during training are a function of the adjustments made to the parameters.

Table 12-3. The Desired Step vs. the Step Length using Nonlinear Regression Training

Arm Angle	Desired Step Length (in)	Step Length (in)
45°	22	30.53
20°	10	13.73
40°	20	14.71
30°	18	18.31
25°	15	18.04
30°	17	18.31
45°	21	18.75
45°	21	18.75

The estimated function during this trial was: $y = 12.5\text{atan}(0.3125\theta)$. However, these parameters are only an initial guess. The nonlinear regression function returns the function $y = 22.2677\text{atan}(0.0312\theta)$. This has a much smaller B parameter, which accounts for the large variation in step lengths within the range of arm angles. Using the non-linear regression function, the largest error between the step output and the desired step is 2.4", which is within the range of error in the feedback from the patient.

The function is adapted during training and gets closer to the data. However, non-linear regression utilizes all of the data points and a more thorough optimization to find the best fit function. Though the function found before non-linear regression could be optimized by using a more sophisticated method of adjusting the parameters, finding a function that performs as well as non-linear regression would be difficult and performance would probably vary based on the training data.

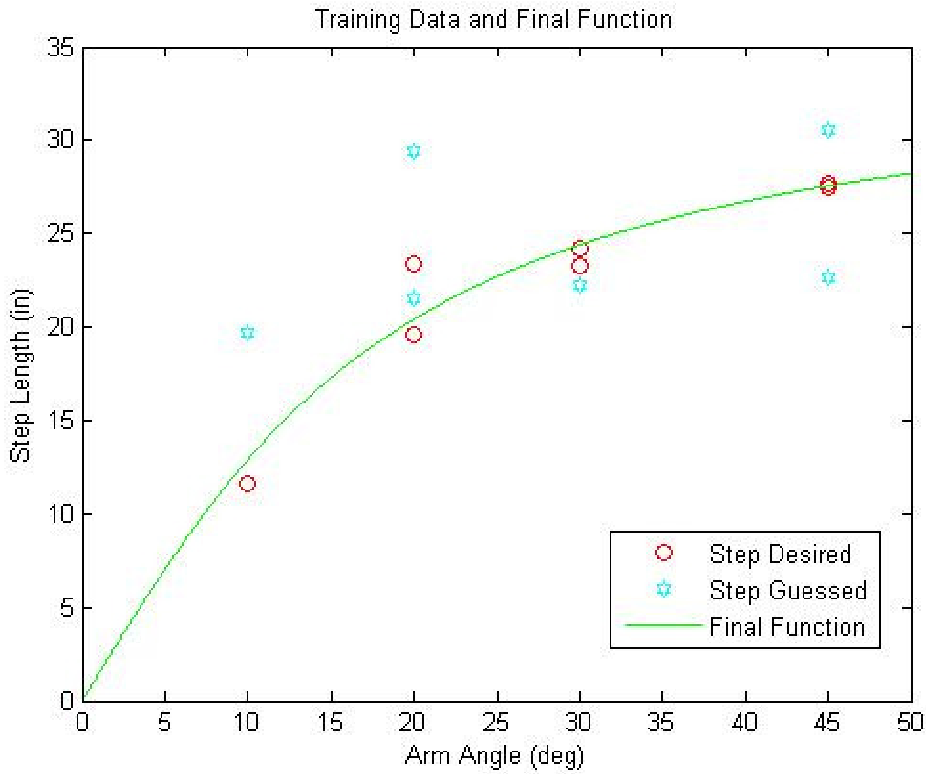


Figure 12-2. The non-linear regression algorithm gives a function which fits the desired data.

The non-linear regression method is also robust to changes in the training data order. Changing the sequence of trial steps, which altered the results with the ε-greedy algorithm, resulted in only a small change to the parameters found using non-linear regression. The original function was 22.2677atan(0.03120θ) whereas the second training data set returned 22.8366atan(0.030θ). These functions are within an acceptable error range and thus are functionally equivalent. The difference in performance is due to the adjustment of the parameters during training resulting in slightly different estimates. Since the error is based on a rounded number, these small variations may cause a slightly different function to be returned.

12.3.3 Linear Approximation

Since the curve in the arctangent function is fairly gradual, I wanted to examine approximating the function as a linear function for ease of implementation. The easiest way to implement this would be to skip the nonlinear regression and fit a linear function to the data points. However, this is not a very good fit based on the experimental data. Furthermore, by setting the x-intercept to zero as we expect, the fitting is worse. By linearizing the arctangent function given by the non-linear regression, we can obtain a better fit. The linearized arctangent had an R^2 value of .9711. Throughout the operating range, the error is within +/-2 inches. This is acceptable performance. However, we must still use non-linear regression, which requires more computational complexity, so the savings of linearizing it is minimal.

12.4 Conclusion

The two methods both utilize minimal sets of training data, thus allowing the patient to start using the exoskeleton quickly. The ε-greedy search algorithm, however, does not adequately explore the state space. It also favors fitting the later steps, which results in a function with a very high slope such that most of the angles in the operating range give the same output. Furthermore, the algorithm is dependent on the order of the trials.

The non-linear regression method, in contrast, does look at all of the training data and is less dependent on the order the training data is given. The resulting function from the regression has a slope which gives a lot of variation between the step lengths in the operating range of the arm angle. This is as we would expect.

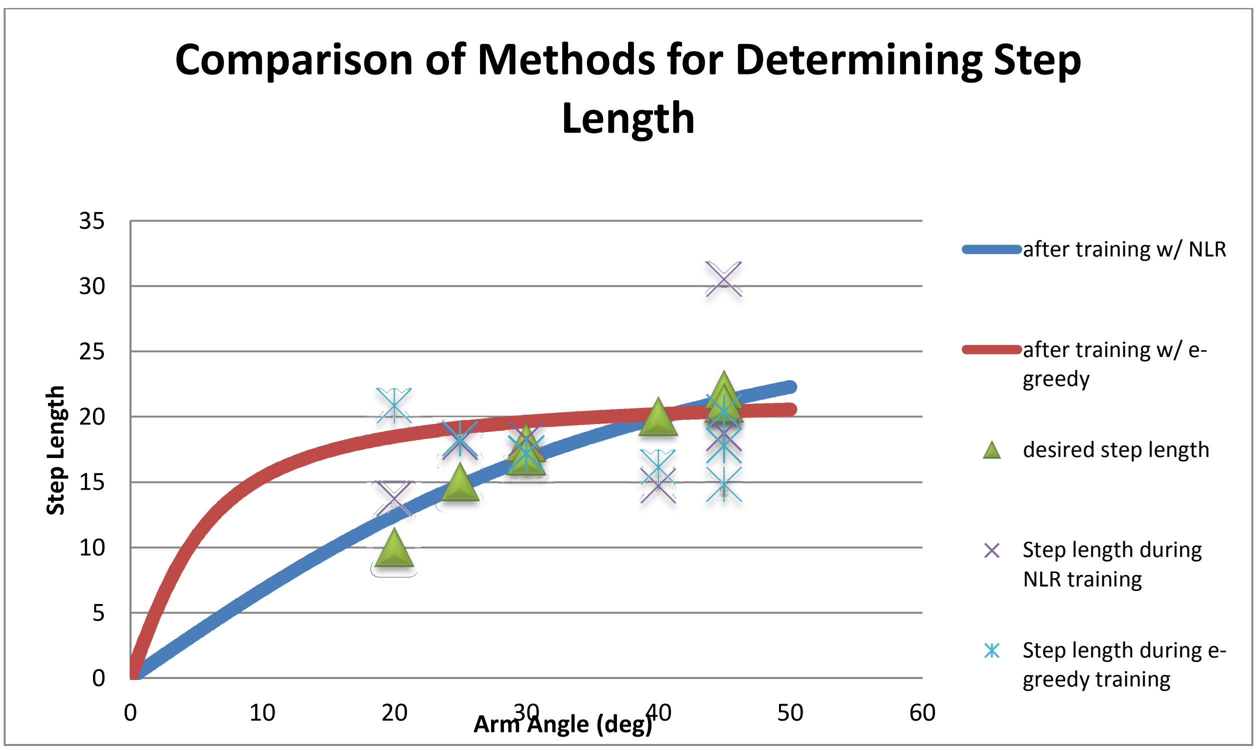


Figure 12-3. The comparison of Nonlinear Regression and Epsilon-Greedy methods to determine the step length.

Figure 12-3 shows a side-by-side comparison of the outcomes of the two functions. The step lengths during training vary due to the adjustment of the parameters during the training process. However, the purple triangles are the desired step lengths during training. For the same arm angle, there are multiple possible step lengths because we expect some stochastic behavior in the response from the person. The non-linear regression method is able to handle this and still determine a function which describes the mapping. The ε-greedy algorithm finds a function which works for 45 degrees, which was the final test point. However, for most other data points, there is a large error. Therefore, the non-linear regression method is a good means by which to determine the mapping from arm angle to step length.

13 CONTINUOUS CONTROLS

13.1 CONTROL ARCHITECTURE

The exoskeleton software is designed to use the sensor input from the HMI and the exoskeleton to achieve the desired motions such as sitting and standing. Figure 13-1 shows the design of the software for eLEGS. It is a modular design in that each group of functions achieving a single task can be considered a software module. The architecture allows for utilizing various trajectory generators, HMIs, or low level controllers without having to augment the other modules.

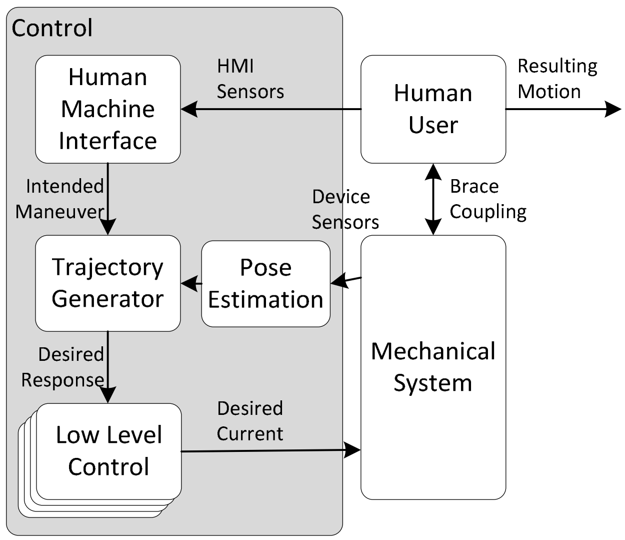


Figure 13-1. The software architecture of the software for the exoskeleton.

The HMI or control pad will determine the intended maneuver. The pose estimation module uses the sensor input to calculate joint angles, the leg segment and torso angles, the arm angles, the planar positions of the joints relative to a ground reference.

The trajectory generator then utilizes joint angles and positions from the pose estimator and the HMI to determine a trajectory for each joint. The trajectory generator calculates the desired joint angle in most cases, but could also provide a torque profile if specific torque profile is desired. This module is used to design stepping trajectories to achieve a natural gait based on the current pose and the desired step length. This is also used to generate joint profiles to sit and stand for the hip and the knee joints.

The desired angle given by the trajectory generator is used as a reference for the low level controller which determines a desired motor current based on the difference between the current angle and the desired angle. In most cases, a PID controller is used; however, because of the modularity of this architecture, other controllers can be easily used. This current is commanded to the motor drivers which provide current to the exoskeleton motors thereby moving the .exoskeleton. Because the subject is coupled to the exoskeleton, the subject moves with the desired motion.

13.2 DYNAMIC MODEL

The exoskeleton is also used to move the user from a seated position to standing and from a standing position to seated. In order to model the dynamics of a person sitting, I used Lagrangian dynamics. Because only the knees and hips are controlled, the arms and arm weight can be considered a disturbance and thus are not included in the dynamic model. During sitting and standing, the left and right legs move in unison. Therefore, the dynamic model becomes a three-link model where the lower legs and upper legs are each combined into a single link.

The notation used for the equations are shown in the graphics below. The three-link model shows the torso, thigh, and shank (or lower leg). The ankle joint is assumed to be at the ground for simplicity. The locations of the center of mass are taken from the Winter's Biomechanics and Motor Control of Human Movement [20]. The knee angle is measured relative to the thigh, and the thigh angle is measured relative to the hip. The torso is measured with respect to gravity, thus the thigh and shank angles can also be determined.

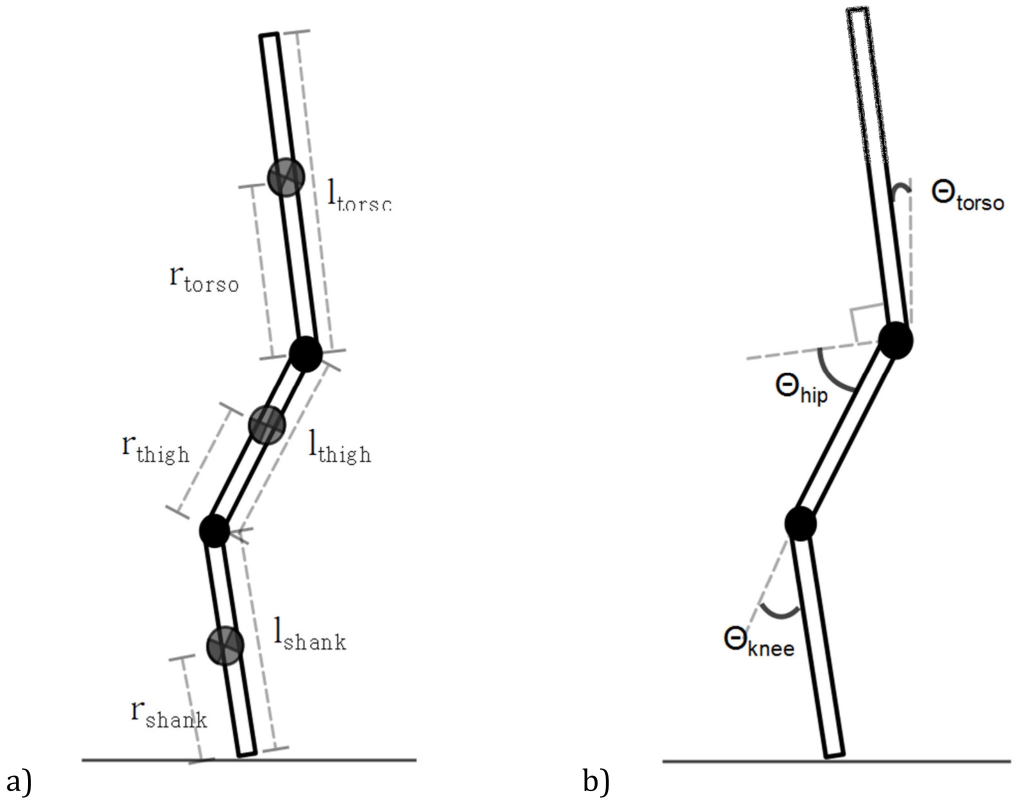


Figure 13-2. Standing Models showing the distance measurement convention and the angle measurement convention.

The dynamic model then allows for modeling of the system and controlling the exoskeleton based on these dynamics. These dynamics can then be used to control the person from a sitting to standing position. However, this model does not take into account that there is a person in the loop. The weight of the person can be approximated, but the user's reaction to the machine cannot. Therefore, if the muscle spasms or they change the weight distribution using their crutches, there is a large, unaccounted for disturbance. Each user will have a different reaction: some will aid the standing by supporting their weight; some will act against it due to a muscle spasm or loading the exoskeleton poorly. Also, the magnitude of these disturbances may be greater than the actual force exerted by the motors. For example, if a user puts a lot of weight into the exoskeleton at the hips, the hip motors may be unable to overcome that torque and may not be able to lift the user's torso until the gain becomes very large or the position reaches a more optimal configuration for the motor. In this case, the controller will overshoot and cause the exoskeleton to go unstable and oscillate at the hips. A non-ideal standing maneuver may be acceptable as the user is able to aid and learn to work with the machine in this case; however, an unstable stand maneuver is not as this could lead to injury of the user. Therefore, without more information about the user's input, maintaining an ideal trajectory and keeping the user's weight over his feet is not possible without risking this instability.

13.3 STANDING UP

The stand algorithm should be stable and smooth while standing. Therefore, a trajectory that does not try to adjust to the person's position as aggressively is desirable. This will minimize the oscillations in the trajectory and allow for more predictability, which will help the user to work with the exoskeleton. Also, by creating a smoother trajectory, we can turn down the P gain on the PID controller used to control the exoskeleton, which will help the exoskeleton to remain stable. This will hurt tracking, but because the exact timing of the stand is less critical than getting to the standing position, this is acceptable. This is a recurring problem in the control of the exoskeleton and thus tracking and optimal trajectories are sacrificed for stability, predictability, and safety.

The basic stand algorithm should look like a person trying to stand without much force on their arms. Figure 13-3 below shows this idealized stand (with crutches shown providing balance). The person tries to get their center of mass over their feet and then begins to rise up to a standing position.

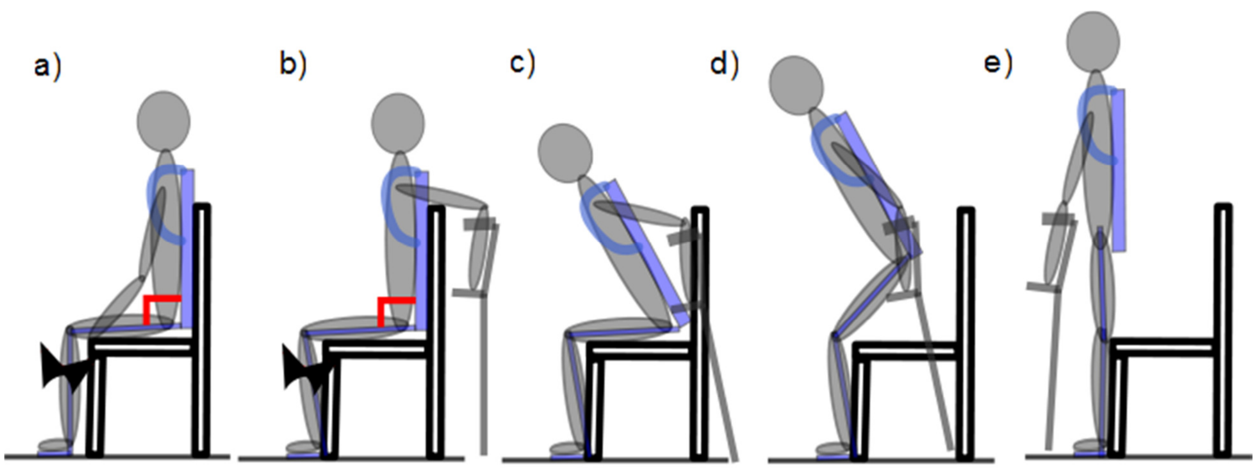


Figure 13-3. a) The exoskeleton supports the user in the seated position. The back is held upright. b) The user positions his feet and arms to prepare for standing. c) The user's torso is leaned forward to move the center of mass over his feet. d) The knees begin to extend while maintaining torso angle until the weight is over the feet. e) The hips extend as well, resulting in a standing posture. The user must move his arms from behind to in front to prepare for walking.

The stand algorithm attempts to keep the torso center of mass over the feet. The center of mass is a function of the hip and knee angles:

$$\theta_{thigh} = \theta_{hip} - \theta_{torso}$$

$$\theta_{shank} = \theta_{thigh} - \theta_{knee}$$

$$x_{comtorso} = [L_{shank}\sin(\theta_{shank}) + L_{thigh}\sin(\theta_{thigh}) + 1/2 L_{torso}\sin(\theta_{torso})\,]$$

The stand algorithm I designed for the exoskeleton is divided into three sections. The first is the "Sit Up" state. This state brings the user's torso upright and holds it into place. The user can then use his hands to move his feet into position for a stand. The stand is easiest if the legs are under the person, and therefore, bringing the feet back as far as possible is beneficial. Here, the user also positions his crutches behind him to provide a forward force during standing. Because the weight is behind the feet, the user must support his weight using his arms. However, the exoskeleton will provide the upwards force.

Once the user is in position, they can progress to the "Lean Forward" state by pushing a button on the control pad. The "Lean Forward" state brings the user's center of mass over the feet or as close as possible. The hips move from the upright position to this max lean forward in a linear motion over the amount of time designated by the "stand time", which is set by the user. A max lean angle is defined, which limits the amount that the torso can lean forward either based on the exoskeleton hardware or the comfort of the user. The hip will not lean past this angle. Therefore, the center of mass will not be directly over the feet, but will be as far forward as possible.

In states "Sit up" and "Lean Forward", the hips are controlled to a reference position while the knees are locked with a brake.

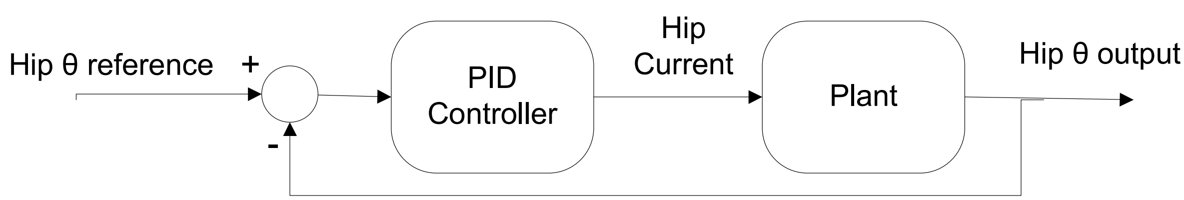


Figure 13-4. The hip angle is controlled by a PID controller based on the reference input. The reference guides the hip from the current position to the upright position in "Sit Up" and holds it there. In "Lean Forward", the hip is guided to the leaning position.

When the exoskeleton reaches the max lean angle, it automatically transitions to the "Stand Up" mode. This mode begins to extend the knees while maintaining the hip angle to keep the center of mass over the toes. The knee trajectory is simply a linear path between the original seated angle and 0 degrees over the designated stand time. The hips extend from the angle reached during "Lean Forward" to 0 degrees in the time remaining during the stand. During this state, the knees control the height of the hips while the hip angle controls the center of mass of the user.

The controller in this state moves the hip as a function of the knee position to maintain the center of mass over the toe as the knee angle raises the user.

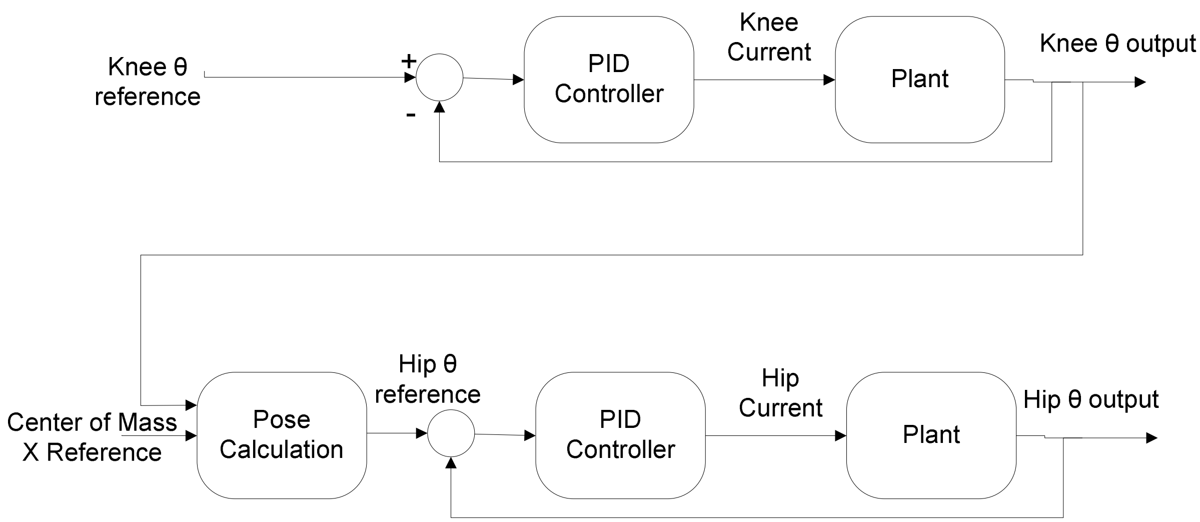


Figure 13-5. The hip reference is based on the knee angle and the center of mass position in the horizontal direction. The knee reference raises the person up to a standing position.

Assuming a constant shank angle of 0°, the trajectory for the stand up maneuver is shown in Figure 13-6. The knee and torso angles during standing if the shank angle is held at 0 degrees.Figure 13-6.

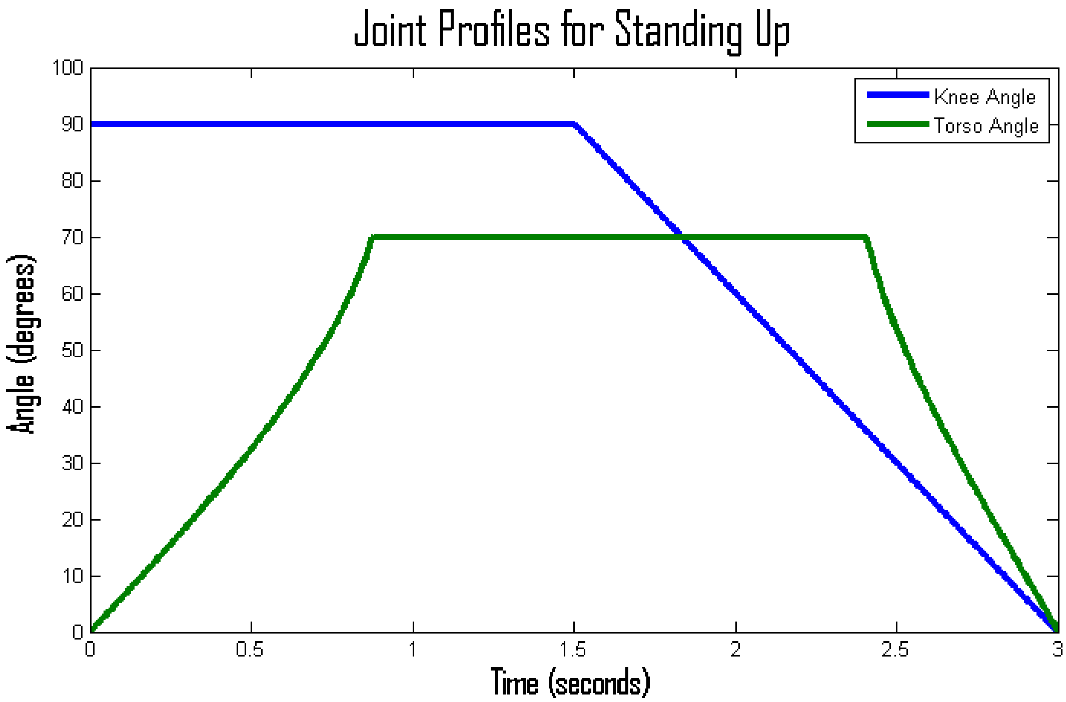


Figure 13-6. The knee and torso angles during standing if the shank angle is held at 0 degrees.

Experimental results show a similar behavior in reality to what we expected based on our model of standing.

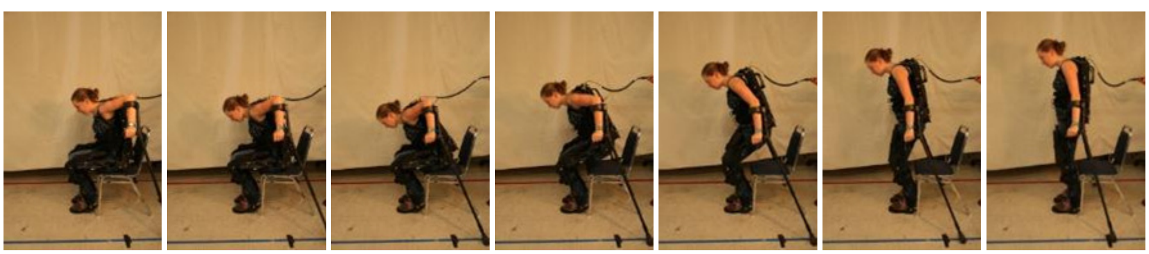


Figure 13-7. The standing maneuver includes a lean forward motion, followed by a stable rise with the center of mass maintained over the feet. The first three frames show "Lean Forward" and the last four show "Stand Up".

13.4 Sitting Down

The exoskeleton must also control the knee and hip joints during the sitting motion. When the subject is standing and wants to move to a sitting position, the exoskeleton must assist with this maneuver in such a way as to minimize the stress put on the user's shoulders, as shoulder injuries are common and very debilitating for wheelchair users.

The most basic approach to sitting is to allow the user to support their torso using their crutches for balancing. The knees are then bent in a controlled manner from 0 to 90 degrees with the hips free to move. By leaning back slightly, the user can then guide himself into the chair. However, because the user is always leaning back onto the crutches during this maneuver, this puts a lot of strain on the shoulders. Furthermore, because the arms cannot be moved in the middle of the sitting maneuver since the body weight is on them, the user ends with their shoulder rotated back very far, which can be painful for some users.

Figure 13-8. The seated position of the sitting maneuver may cause strain on the pilot's shoulders.

In order to help avoid these downfalls, a balanced approach to sitting is used. In this approach, the center of mass of the torso is kept over the user's feet. The center of mass is the same as used in standing. In the ideal control method, the torso is controlled to keep the center of mass over the feet while the knees are bent to a sitting position. However, in order to estimate this necessary angle, the full body pose must be known. Since we do not have direct measurement of the ankle joint angle (or whether the feet are flat on the ground), we cannot determine the exact desired position. However, by assuming that the feet are flat on the ground, we can use the torso angle and the joint angles to determine a shank angle. This can then be used to determine the overall body position and where the torso should ideally be in order to balance.

However, because the back moves to adjust the body position and this affects the shank position, this becomes a positive feedback loop and can easily become unstable.

Therefore, we must assume a constant shank angle and adjust the torso based on this. While this is likely not the case, it does provide a sufficient estimation. The torso can then move to an angle which maintains the weight over the estimated foot position.

The range of ending shank angles is relatively small and is mostly based on the position of the feet relative to the chair and the chair height. Figure 13-9 shows that the variation of the knee and hip angles based on a 10 degree difference of assumed joint angle is relatively minor. Therefore, assuming a final joint angle of zero should not affect the quality of the stand maneuver, especially because the user will support some of his weight with the crutches.

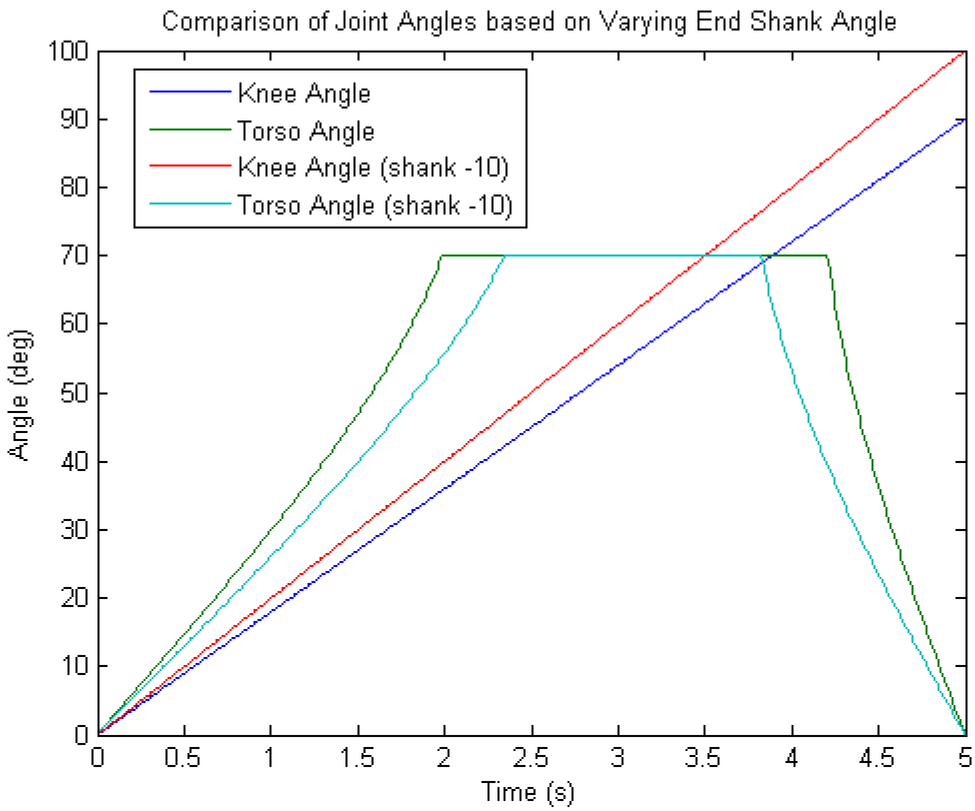


Figure 13-9. Comparison of Different Shank Angles. The difference between the angles is minimal enough that this is a viable option for a sitting algorithm.

A further simplification is linearizing the torso angle over a set percentage of the sitting cycle. While this is less accurate to the desired model, it does provide simple calculations for quick processing and an easy method to explain to users and observe. Figure 13-10 shows this approximation. The linear approximation, while the lean is slower, still achieves the desired angle before the knee is fully bent, as expected. Since the lean is slower, we expect more weight to be behind the person's feet, but since the crutches are providing force to support the weight and push the person forward, this is acceptable.

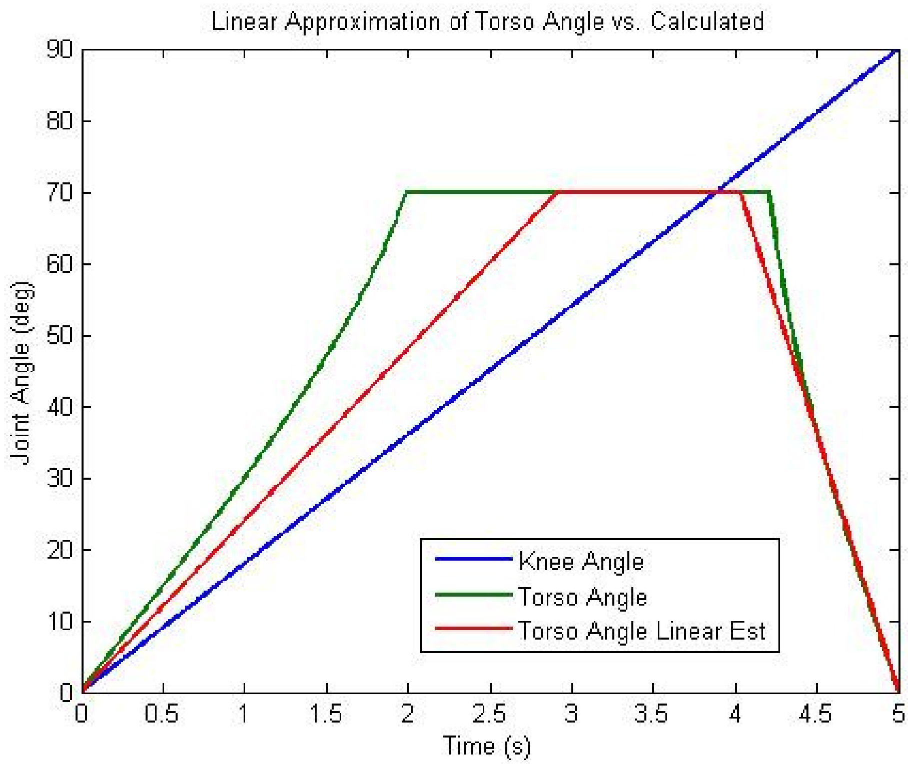


Figure 13-10. Linear Approximation of the Torso Angle During Sitting. The linear approximation is sufficient for the maneuver.

The sitting maneuver begins from the standing position. The user puts their crutches slightly behind their feet. This allows for a base of support while they sit down, since they will have more weight behind their feet. The hips and knees then begin to bend simultaneously to create an accordion-like motion. This brings the user's torso over their feet while lowering them towards the chair. Because some chairs are high and do not allow the knees to bend fully before the person is seated, the sit algorithm has two stopping conditions. The motion is stopped when the knees are bent to 90 degrees or the feet have come off the ground. The foot state can be determined by reading the pressure sensors under the feet and waiting for the pressure on the heel and toe to read sufficiently low that it indicates that they are no longer on the ground.

The sit algorithm allows the user to sit smoothly and easily. However, there are some downfalls to this method. Because the knee and hip trajectories are not dependent on the back angle, the user can be leaning forwards or backwards when the maneuver begins by pivoting about the ankle. This will lead to a variation on where they land when they sit.

This means that the person may land far back in the chair, far forward, or miss the chair completely. Furthermore, depending on the length of the crutches and the position, the elbows can end up very bent in the final position, causing undesirable stretching of the shoulder.

Figure 13-11. The sitting maneuver starts from standing and attempts to keep the center of mass over the feet while sitting.

14 CONCLUSION

This thesis describes the development and testing of the Human Machine Interface for a mobile exoskeleton for users with spinal cord injuries. The Human Machine Interface consists of both hardware and software which allows a user to control the exoskeleton independently. The hardware is small, lightweight, and unobtrusive, allowing this to be used on concert with the eLEGS exoskeleton device. The software utilizes a finite state machine which examines the current state as well as the sensor inputs to determine the user's intent. This method for walking has been developed, tested in the laboratory, and tested on eLEGS in rehabilitation centers. Methods indicating sitting and standing have been developed and tested in the laboratory. The HMI has been successful in allowing a user to control the exoskeleton and ambulate independently as has been demonstrated by a variety of test pilots with spinal cord injuries. Safety of the system has also been shown through use in the rehabilitation centers. This method is unique in that it can be learned quickly by users in a rehabilitation center because of its use of natural gestures.

Once the user's intended action is known, the exoskeleton must move to accomplish that maneuver. This thesis also presents the methods by which the user can sit and stand using the exoskeleton. These controllers are made complicated by the fact that the user is in the exoskeleton and greatly affects the motion. For example, a user may have some muscle and be able to aid the exoskeleton, a user may spasm as the exoskeleton moves and thus their muscles may fight against the motors, or the user may support their body weight with the crutches in different positions thus affecting their overall lean in the device. However, despite these uncertainties, this control method is robust and reliably allows the user to sit and stand. This has also been tested on a variety of users of different levels and completeness of spinal cord injury.

14.1 FUTURE WORK

While the HMI allows the user to walk in straight lines or turn using their crutches to help shift their body, the methods for adjusting the step length as a function of the arm have not yet been introduced. One of the main reasons is due to the inaccuracy of the upper arm measurement. When these sensors were selected, the 3-degree-of-freedom IMU was selected in order to keep the cost of the system down. However, by only measuring the angles in one plane, the understanding of what the arm is doing is limited. Therefore, rotations and out of plane motion are not seen but do degrade the ability to accurately measure the upper arm angle. As this work has progressed, however, the cost of multi-axis IMUs has decreased making 6 or even 9 DOF IMUs affordable. I have begun initial investigations into these devices and have seen that the arm angle measurement from these devices is far more accurate and can give a more clear indication of the user's intent.

Furthermore, by increasing the accuracy of the upper arm angle sensor, I hope to further reduce the need to calibrate the sensor for each use.

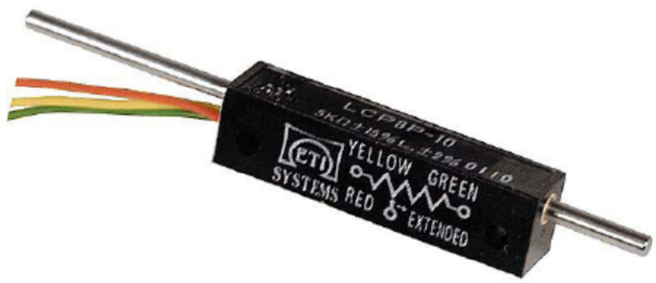
If eLEGS is to be a true wheelchair replacement, it will also need to be able to perform a wide range of motions beyond just walking, sitting and standing, and the HMI will need to accommodate all of these. Therefore, one of the biggest open issues is how to determine actions such as stair climbing, ramp ascent and descent, turning in place, and bending down. These maneuvers are necessary to truly make the user independent and able to go about their daily life using the exoskeleton; however, it is questionable whether this initial set of HMI sensors will be sufficient to achieve these maneuvers. We have begun considering options such as vision systems, sonar systems, and additional IMUs to give a better picture of the surrounding environment.

The HMI sensor placement is another area of ongoing research. Requiring the user to place the armbands on their arms adds an extra component to the machine and causes additional setup time. Furthermore, it is susceptible to the placement on the arm and may move during use. Users would also rather have the sensors less visually conspicuous. The crutch may also be a viable location for the HMI sensors and would allow for more compact packaging. With a more accurate sensor and increased degrees of freedom, the difference in accuracy and repeatability between the upper arm and the crutch may be negligible.

APPENDIX A: LINEAR POTENTIOMETER DATA SHEET

LCP8
Linear Conductive Plastic

STATE ELECTRONICS



Series LCP8 linear potentiometers feature Conductive Plastic Element. Provides essentially infinite resolution, long life, high speed tracking ability and good high frequency characteristics

---

Features
- Conductive Plastic resistive element
- Stroke Length: 10mm, 12mm
- Stainless steel shaft - threaded or plain
- Bronze bearings
- Gold plated terminals
- Platinum Alloy multi-finger wiper
- Internal spring return optional

Electrical Specifications

Resistance Range:
1K to 50K ohms standard

Resistance Tolerance: ±15%

Independent Linearity: ±2%

Power Rating: 0.2 watt

Electrical Stroke: 0.433 inches

Insulation Resistance: 1,000 Megohms @ 500VDC

Dielectric Strength: 500 VAC min.

Mechanical Specifications

Stroke Length: 10mm, 12mm

Shaft: Stainless steel - plain, chamfered, or threaded (#2-56 UNC-2A).

Friction: 30 grams

Weight: 0.2 oz

Life Expectance: 20,000,000 strokes

Environmental Specifications

Operating Temperature: -30°C to +105°C

Temperature Coefficient: ±400 PPM/°C

Vibration: 10HZ to 2,000HZ 15G

Shock: 50G 11ms

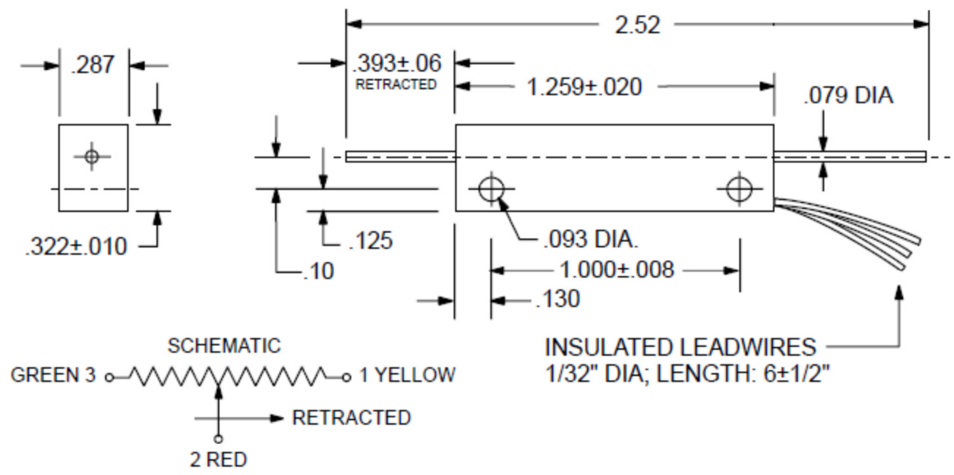
Moisture Resistance:
40°C 95%RH 120 Hours

---

36 Route 10 East Hanover, NJ 07936 (800) 631-8083 Local: (973) 887-2550
e-Mail: sales@state-elec.com website: http://www.potentiometers.com

LCP8

Linear Conductive Plastic



Ordering Information

Example Part Number: **LCP8-P-10-10K**

LCP8	P	10	10K	SB
Model	**Shaft Style**	**Stroke Length**	**Resistance**	**Sealed Body Option**
LCP8	**P** = Plain **PS** = Plain w/ Spring Return **C** = Chamfered **CS** = Chamfered w/ Spring Return **T** = Threaded #2-56 **TS** = Threaded w/ Spring Return	**10** = 10mm **12** = 12mm	**1K** = 1K ohms **2K** = 2K ohms **5K** = 5K ohms **10K** = 10K ohms **20K** = 20K ohms **50K** = 50K ohms	**SB** = Sealed Body

On-line Quotations are available at http://www.potentiometers.com

STATE ELECTRONICS

36 Route 10 East Hanover, NJ 07936
(800) 631-8083 Local: (973) 887-2550
e-Mail: sales@state-elec.com
http://www.state-elec.com http://www.potentiometers.com

Rev. 042209

APPENDIX B: STANDING & SITTING SOFTWARE

```
//set hip reference position to 0 degrees
//blend from current angle to the 0 degree point over 2 seconds
//set the knees free
case Sit_Up:
	pHip->ref = 0.0;
	pHip->fMode = CT_PosPD;
	linBlend( Count, 2.0, pPose->tAng, pHip);

	pKnee->ref = 0.0;  pKnee->fMode = CT_Free;
break;

//lean torso over ankles
//first calculate the desired lean angle then lean to that or max lean
//Knees are locked
case Lean_Forward:
	percent = (Count/pCtrl->gait.standT)*100.0;
	if( percent < 50 )
	{
		tmpIG[1] = 1;
		if( Count <= .002)
		{
			//geometry for desired torso COM at 0 x.
			inAsin = ( 0.05 + (pPose->P.shL)*sin(DEG2RAD*(pPose->L.shAng)) +
		pPose->P.thL)*sin(DEG2RAD*(pPose->L.thAng)))/(.5*pPose->P.toL);
			if( inAsin > .8)
				inAsin = .8;
			refHip1 = RAD2DEG*(asin(inAsin));
			//max lean angle for safety
			if(refHip1 > pCtrl->gait.leanAngle)
				refHip1 = pCtrl->gait.leanAngle;
			}
		// torso is between -90 and 90 so can use asin if inasin is limited.
		pHip->ref = refHip1;

		pHip->fMode = CT_PosPD;
```

```
                    linBlend( Count, pCtrl->gait.standT*.50, pPose->tAng, pHip);
          }
break;

//calculate the torso angle based on the rest of the position of the body
//knees raise gradually from current position to 0 degrees.
case Stable_Rise:
                    inAsin = ( 0.05 + (pPose->P.shL)*sin(DEG2RAD*(pPose->L.shAng)) + (pPose-
>P.thL)*sin(DEG2RAD*(pPose->L.thAng)))/(.5*pPose->P.toL);
                    inAsin = LIMIT(inAsin, 0, 0.8);
                    //if still need to be leaned forward
                    if(inAsin >= 0.8)
                    {
                              pHip->ref = refHip1;
                              stTime = Count;
                              pHip->fTarget = TA_Tor;
                    }
                    //leaned forward enough, just stand now
                    else
                    {
                              pHip->ref = stAng;
                              pHip->refD = 0.0;
                              linBlend(Count - stTime, pCtrl->gait.standT*0.5-stTime, pLeg->hAng,
pHip);
                              pHip->fTarget = TA_Jnt;
                    }
                    //stand the knees up over 50% of the stand time
                    pKnee->ref = 0.0;
                    linBlend( Count, pCtrl->gait.standT*0.5, pLeg->kAng, pKnee);
                              pHip->fMode  = CT_PosPID;
                    pKnee->fMode = CT_PosPID;
break;

//Sit by moving the knees down to 90 degrees and maintaining the torso CG approximately over
the feet
//then straighten torso
case RG_Balanced_Sit:
```

```
        /***** Calculations *****/
        //estimate a final torso lean and blend to that
        //assume shanks end straight up and down
        //active calculation gives better results, but more susceptible to not being smooth, error
is at max 30%
        inAsin = ( (pPose->P.shL)*sind((0)) + (pPose->P.thL)*sind((90)))/(.5*pPose->P.toL);
        if( inAsin > .8)
                inAsin = .8;  //cap for valid range of angles
        refHip1 = (asind(inAsin)); //convert to deg

        //cap at the lean angle
        refHip1 = MIN(refHip1, pCtrl->gait.leanAngle);

        percent = (Count/(pCtrl->gait.standT))*100.0;
        if( percent < 100 )
        {
                //only stop the bending after a sufficient amount of the cycle has passed
                if( percent > 75 )
                {
                        //check if there is sufficient weight on the foot
                        if(!pPose->R.bFootOnGnd && !pPose->L.bFootOnGnd)
                                toeFlag = TRUE;
                }
                else
                        toeFlag = FALSE;

                if( toeFlag == FALSE)
                {
                        //during the "Stand time", bend the knees to 90
                        //after the knees are at 90, the shank angles should be well known, so
calculate the needed knee angle
                        kneeGoal = 90 - (pPose->R.shAng);
                        kneeGoal = MAX(kneeGoal, pLeg->knee.ang);
                        /***** Hip Outputs *****/
                        traj = smoothTraj_Time(Count, pCtrl->gait.standT, pPose->tAngPitch, 0.0,
refHip1, pHip->fJoint);
                        refGenOutput(pHip, CT_PosPD, TA_Tor, traj.pos);
```

```
                    /***** Knee Outputs *****/
                    traj = smoothTraj_Time(Count, pCtrl->gait.standT, pLeg->knee.ang, 0.0,
90.0, pKnee->fJoint);
                    refGenOutput(pKnee, CT_PosPD, TA_Jnt, traj.pos);
                }
        }
        else
        {
                if( !pPose->R.bToeOnGnd && !pPose->L.bToeOnGnd)
                    toeFlag = TRUE;
                     //if the toes are off the ground, stop.  Allow heels to come off ground b/c
we may rock forward
                    if( toeFlag == FALSE)
                    {
                            //control knees to the rest of the way down (so thighs are
horizontal)
                            //Check out of range knee
                            if( kneeGoal > 120) kneeGoal = 120;

                    /***** Knee Outputs *****/
                    traj = smoothTraj_Time(Count-pCtrl->gait.standT, 1.0, pKnee->refP, 0.0,
kneeGoal, pKnee->fJoint);
                    refGenOutput(pKnee, CT_PosPD, TA_Jnt, traj.pos);
                }

                /***** Hip Outputs *****/
                //control the hips back up to 90 to sit up straight
                traj = smoothTraj_Time(Count-pCtrl->gait.standT, 1.0, pLeg->hip.ang, 0.0, 90.0,
pHip->fJoint);
                refGenOutput(pHip, CT_PosPD, TA_Jnt, traj.pos);
        }
        break;
}
```

Appendix C: HMI Software

```
/**
 * @brief      Determine if the HMI conditions have been met for the next step
 *
 * @param      *pStanceLeg          The leg structure within the main Pose structure
corresponding next stance leg
 * @param      *pSwingLeg           The leg structure within the main Pose structure
corresponding next swing leg
 *
 * @return     TRUE/FALSE whether HMI conditions have been met for next step
 */
static int HMIStep(PoseData *pPose, int side)
{
	PoseLeg *pStanceLeg;
	PoseLeg *pSwingLeg;
	float armThresh;

	//based on which leg, determine the stance and swing leg as well as the arm threshold
for that side
	switch(side)
	{
		case RIGHT:
			pSwingLeg = &pPose->R;
			pStanceLeg = &pPose->L;
			armThresh = pPose->P.LArmThresh;
			break;

		case LEFT:
			pSwingLeg = &pPose->L;
			pStanceLeg = &pPose->R;
			armThresh = pPose->P.RArmThresh;
			break;


		default:	// error... you shouldn't be here
			break;
	}

	//Conditions for step
	//If these conditions are met, a step can be taken.
	if((pStanceLeg->aAng > armThresh) &&
      pStanceLeg->cGnd &&
     (pStanceLeg->aAng > pSwingLeg->aAng) &&
```

```
    (pStanceLeg->hGnd || pStanceLeg->tGnd))
  {
        return TRUE;
  }
  else
  {
        return FALSE;
  }
}

/**
* @brief  The walking state machine as driven by the HMI
* If the HMI condition is met, the step will be taken
* Transitions from swing to stance are based on heel sensors
**/

static void WalkingStateMachine(PoseData *pPose, HMIData *pHMI, GaitParam *pGait)
{
        //Switch states based on the current state of walking
        switch(pHMI->subState)
        {
                //The arm threshold is updated in swing
                //The states are set during this state
                //Conditions for transition to stance are looked for
                case WLK_RSwing:                // Right Swing

                        //Adaptive Arm Threshold
                        if( pGait->tCountR < 0.0025)
                        {
                                //update arm threshold
                                pPose->P.LArmThresh = .85*pPose->P.LArmThresh +
.15*((curr_Larm_max-curr_Larm_min)*(.5)+curr_Larm_min);
                                curr_Larm_min = 0;
                                curr_Larm_max = 0;
                                // end update arm threshold
                        }
                        //update max & min R arm angle
                        if( pPose->R.aAng > curr_Rarm_max )
                                curr_Rarm_max = pPose->R.aAng;
                        if( pPose->R.aAng < curr_Rarm_min )
                                curr_Rarm_min = pPose->R.aAng;
                        //update max & min L arm angle
                        if( pPose->L.aAng > curr_Larm_max )
```

```
				curr_Larm_max = pPose->L.aAng;
			if( pPose->L.aAng < curr_Larm_min )
				curr_Larm_min = pPose->L.aAng;

			//Set current states
			pHMI->lState = RG_Stance;
			pHMI->rState = RG_ToePSwing;

			//Condition to transition into stance
			// When foot hits the ground switch to double stance (must wait enough
time to give it a chance to leave ground)
			ResetTCount(pHMI, pGait, 2);
			if ((pPose->R.hGnd==TRUE || pPose->R.tGnd==TRUE) && (pGait-
>tCountR>=pGait->stepT*0.40))
			{
				pHMI->subState = WLK_RDStance;
			}

		case WLK_RDStance:		//DS - Right in front of Left
			//The states are set for each leg
			//Conditions for transition to swing are looked for based on HMIStep()
function


			pHMI->lState = RG_DStance_TO;
			pHMI->rState = RG_DStance_HS;

			if (HMIStep(pPose, LEFT) == TRUE)
				pHMI->substate = WLK_LSwing;

			break;


		case WLK_LSwing:		// Left Swing
			//The arm threshold is updated in swing
			//The states are set during this state
			//Conditions for transition to stance are looked for


			if( pGait->tCountR < 0.0025)
			{
				//update arm threshold
```

```
				pPose->P.RArmThresh = .85*pPose->P.RArmThresh +
.15*((curr_Rarm_max-curr_Rarm_min)*(.50)+curr_Rarm_min);
				curr_Rarm_min = 0;
				curr_Rarm_max = 0;
				// end update arm threshold
			}
			//update max & min R arm angle
			if( pPose->R.aAng > curr_Rarm_max )
				curr_Rarm_max = pPose->R.aAng;
			if( pPose->R.aAng < curr_Rarm_min )
				curr_Rarm_min = pPose->R.aAng;
			//update max & min L arm angle
			if( pPose->L.aAng > curr_Larm_max )
				curr_Larm_max = pPose->L.aAng;
			if( pPose->L.aAng < curr_Larm_min )
				curr_Larm_min = pPose->L.aAng;

			//Set current states
			pHMI->lState = RG_ToePSwing;
			pHMI->rState = RG_Stance;

			// When foot hits the ground switch to double stance (must wait enough
time to give it a chance to leave ground)
			ResetTCount(pHMI, pGait, 2);
			if ((pPose->L.hGnd==TRUE || pPose->L.tGnd==TRUE) && (pGait-
>tCountL>=pGait->stepT*0.40))
			{
				pHMI->subState = WLK_LDStance;
			}


			break;

		case WLK_LDStance:		// DS - Left in front of Right
			//The states are set for each leg
			//Conditions for transition to swing are looked for based on HMIStep()
function


			pHMI->rState = RG_DStance_TO;
			pHMI->lState = RG_DStance_HS;

			//State transitions to Swing based on HMI conditions
```

```
				if (HMIStep(pPose, RIGHT) == TRUE)
					pHMI->substate = WLK_RSwing;

				break;

			default:		// error... You shouldn't be here
				break;
		}
}
```

[1] "Spinal Cord Injury Facts & Statistics." [Online]. Available: http://www.sci-info-pages.com/facts.html. [Accessed: 18-Oct-2011].

[2] "Spinal cord injury - MayoClinic.com." [Online]. Available: http://www.mayoclinic.com/health/spinal-cord-injury/DS00460. [Accessed: 18-Oct-2011].

[3] S. Kirshblum, S. Millis, W. McKinley, and D. Tulsky, "Late neurologic recovery after traumatic spinal cord injury.," *Archives of Physical Medicine and Rehabilitation*, vol. 85, no. 11, pp. 1811-1817, 2004.

[4] L. M. Giangregorio et al., *Body weight supported treadmill training in acute spinal cord injury: impact on muscle and bone.*, vol. 43, no. 11. 2005, pp. 649-657.

[5] S. J. Harkema, "Neural plasticity after human spinal cord injury: application of locomotor training to the rehabilitation of walking.," *The Neuroscientist a review journal bringing neurobiology neurology and psychiatry*, vol. 7, no. 5, pp. 455-468, 2001.

[6] "Hocoma - Lokomat®." [Online]. Available: http://www.hocoma.com/en/products/lokomat/. [Accessed: 18-Aug-2011].

[7] "Lopes Webpage." [Online]. Available: http://www.bw.ctw.utwente.nl/research/projects/lopes.doc/index.html. [Accessed: 18-Aug-2011].

[8] R. Ekkelenkamp, J. Veneman, and H. van der Kooij, "LOPES: a lower extremity powered exoskeleton," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, no. April, pp. 3132-3133, Apr. 2007.

[9] "LOPES Congress Posters." [Online]. Available: http://www.bw.ctw.utwente.nl/research/projects/lopes_posters/^index.html/photo_page_1.html. [Accessed: 18-Aug-2011].

[10] R. Riener, A. Koenig, S. Member, M. Bolliger, M. Wieser, and H. Vallery, "Bio-Cooperative Robotics: Controlling Mechanical, Physiological and Mental Patient States," *Heart*, pp. 407-412, 2009.

[11] "Rex: Who is Rex Suitable For." [Online]. Available: http://www.rexbionics.com/Technical/Who-is-Rex-suitable-for.aspx. [Accessed: 18-Aug-2011].

[12] "1st Rex commercial bionic legs go to Paralympian | Crave - CNET." [Online]. Available: http://news.cnet.com/1st-rex-commercial-bionic-legs-go-to-paralympian/8301-17938_105-20058955-1.html. [Accessed: 18-Aug-2011].

[13] "Argo Medical Technologies." [Online]. Available: http://www.argomedtec.com/. [Accessed: 18-Aug-2011].

[14] "UCSC | Bionics Lab; Wearable Robotics." [Online]. Available: http://bionics.soe.ucsc.edu/research/exoskeleton_index.html. [Accessed: 18-Aug-2011].

[15] "Austin | Berkeley Robotics & Human Engineering Laboratory." [Online]. Available: http://bleex.me.berkeley.edu/research/exoskeleton/medical-exoskeleton/. [Accessed: 18-Aug-2011].

[16] R. Vaidyanathan et al., "A Dual Mode Human-Robot Teleoperation Interface Based on Airflow in the Aural Cavity," *The International Journal of Robotics Research*, vol. 26, no. 11-12, pp. 1205-1223, Nov. 2007.

[17] G. M. Friehs, V. a Zerris, C. L. Ojakangas, M. R. Fellows, and J. P. Donoghue, "Brain-machine and brain-computer interfaces.," *Stroke; a journal of cerebral circulation*, vol. 35, no. 11 Suppl 1, pp. 2702-5, Nov. 2004.

[18] K. I. Mitsuru Takahashi, Manabu Gouko, "Electroencephalogram (EEG) and Functional Electrical Stimulation (FES) System for Rehabilitation of Stroke Patients," *21st IEEE International Symposium on Computer-Based Medical Systems*, pp. 53-58, 2008.

[19] L. R. Hochberg, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164-71.

[20] D. A. Winter, *Biomechanics and Motor Control of Human Movement*, Third. Waterloo, Ontario, Canada: John Wiley & Sons, Inc., 2005.