Adaptive Robotic Socket

A prosthetic socket for above-knee amputees that adapts to the user's environment

A Major Qualifying Project Submitted to the Faculty of the WORCESTER POLYTECHNIC INSTITUTE in partial fulfillment of the requirements for the Degree of Bachelor of Science

By

Shane Fagan

Tynan MacLeod

Ryan Moran

Alex Rutfield

Alexandra Shea

Submitted to

Project advisor: Professor Gregory Fischer

Project co-advisors: Professor Edward Clancy & Assistant Professor Cagdas Onal

Abstract

Although prosthetic socket technology has been rapidly advancing, amputees are still concerned that their socket and prosthetic are not adaptive to their current activity or consistently comfortable. The goal of our project was to design a prototype of an above-knee adaptive prosthetic socket that responded to an amputee's movement and physiological signals to alleviate pressure on the residual limb during daily activities. The device adjusted multiple air-filled bladders appropriately when the user was standing, sitting, and walking.

Disclaimer: All data collection throughout this project was done ONLY with members of the project team, and therefore, IRB approval was not required for any experiments performed.

Executive Summary

Nearly 2 million people are living with limb loss in the United States each year, and over 180,000 new amputations are performed from either disease or trauma (Cooper). With the help of prosthetic devices, lower limb amputees are able to regain their mobility and return to a mostly average lifestyle. However, many amputees experience complications with their devices that can have adverse effects to their health, particularly at the socket-residual limb interface. Amputees commonly experience blisters, cysts, edema, skin irritation, and pressure sores as a result of poor fitting and unnatural external forces on the residual limb (Mak, Zhang and Boone). Proper socket fitting is one strategy used to eliminate such problems, but a static fitting alone cannot completely reduce the risk. Because the residual limb may deform, swell, and experience forces differently during higher impact activities, the socket therefore needs to be dynamically fitting.

The group has designed a trans-femoral prosthetic limb socket system that uses four air-filled bladders to automatically provide the appropriate support and fit to the residual limb depending on the state of the amputee. These states include sitting, standing, and walking. The goal of our project was to design a first prototype of the system that can read a person's state, inflate and hold the bladders at specific pressures, and allow for appropriate testing that would be helpful for teams who work on this project in the future.

The layout of the team's design incorporates three main systems: the mechanical/pneumatic system, the electrical system, and the control system. The mechanical/pneumatic system includes four air-filled bladders that are located around the mid-perimeter of the socket at the anterior, posterior, medial and lateral locations. Air is supplied to the bladders by three reservoir tanks which cumulatively hold 450mL of air at 150psi. It is released from the tanks by a pressure regulator before entering a series of components that lead to each individual bladder. This series includes an

on/off valve, a pressure sensor, and a blow-off valve. The valves and pressure sensors are included in our electrical system and control system, which uses an Arduino Mega as a microcontroller. The pressure sensor is used to read the pressure within each bladder, and the control unit is told what pressure the bladder needs to hold depending on the person's state. Finally, the valves control the air flow to inflate or deflate the bladders to the desired states. An accelerometer is interfaced with the control unit which uses the sensor data to detect the activity the person is doing. It is attached to a box on the side of the socket which also serves as a user interface. This box includes an on/off switch as well as LEDs which indicate which state the person is in. All components except for the accelerometer and the user interface box are located in a backpack.

Because this is the first prototype, it would have been difficult to justify any clinical testing on actual amputees. The group's primary objective was to design a proof of concept system that functions properly so that future projects can continue with clinical testing. Preliminary testing was performed using a testing rig designed for able-bodied subjects based upon the same configuration designed to fit inside the socket. This rig helped us to determine if each individual system was working properly. We looked at response time for reading changes in states, lifespan of reservoir tanks before having to refill, the effect of longer tubing on the rest of our system, and verification of ideal pressures for each bladder.

Additionally, we were able to simulate the use of the socket on an actual amputee using an industrial robotic arm in place of a residual limb, the ABB Robot. We filled the inside of the socket with foam and dragon skin molding to act as a residual limb. Then, we used a robot arm to move the socket in such a way that simulated sitting, standing and walking. Just as we acquired from our testing rig results, our system proved to correctly determine the user's activity when the ABB Robot simulated sitting, standing, and walking on the socket, but not as well because the robot did not mimic or simulate walking as well.

This is only the first iteration of what could eventually be a marketable product that could help thousands of amputees. To ensure the success of future projects, we came up with recommendations that we believe could improve the design. One idea is for the socket to be able to sense more states than just sitting standing and walking. Other useful states that could be sensed might be jogging and walking up or down stairs. Another recommendation we have is to use reservoir tanks that can hold a larger volume, a larger pressure, or both. This would decrease the amount of times the user would have to fill up the tanks. Future work on this project may also include incorporating more bladders into the socket. This could potentially benefit both the comfort level and the stability of the socket. Lastly, this design is intended to end up working on actual patients with prosthetic legs. Therefore, because our prototype was intended as a proof-of-concept design, there are still many safety features that could be added to ensure the safety of the patients, such using more precise blow-off valves and electrical protection components such as diodes and fuses.

Acknowledgements

The Adaptive Robotic Socket team would like to thank the following individuals and organizations for their guidance and help throughout the completion of this project in order to make it successful:

- Dr. Gregory Fischer for being the head advisor, proposing the idea of the Adaptive Robotic Socket, donating AIM lab space, providing various resources, and in general leading our team to the completion of a successful project.
- Co-advisors Edward Clancy and Cagdas Onal for providing helpful feedback throughout the project.
- PhD candidates Michael Delph and Christopher Nycz for assisting the team in various aspects throughout the project such as prototyping and report writing.
- Joseph St. Germain for his generous donation of mechanical components as well as his advice on technical aspects of the project.
- Robert Boisse for his generous donation of electrical components as well as his advice on technical aspects of the project.
- Professor Craig Putnam for his helpful assistance with operating the ABB robot in the Washburn Robotics Laboratory, which was necessary for testing our prototype.
- Ottobock for their very generous donation of a sample trans-femoral prosthetic socket that was incorporated into our final design.

Table of Contents

Authorship

Section	Written by	Edited by
Abstract	Ryan	Alex
Executive Summary	Ryan	Alex
Acknowledgements	Shane	Ryan
Authorship	Shane	Tynan
1. Introduction	Lexi	Ryan
2. Literature Review	See below:	See below:
2.1 Socket	Lexi	Ryan
2.2 Medical Effects of Lower Limb Prosthetics	Lexi	Tynan
2.3 Qualities of Existing Prosthetic Sockets	Ryan	Lexi
2.4 Bladder	Ryan	Lexi
2.5 Kinematics	Ryan	Tynan
2.6 Sensors	See below:	See below:
2.6.1 EMG	Shane	Tynan
2.6.2 Pulse Oximetry	Shane	Alex
2.6.3 Galvanic Response	Tynan	Shane
2.6.4 Pressure Sensors	Tynan	Alex
2.7 Software Interfaces	Alex	Shane
2.8 Digital Signals	Alex	Tynan
2.9 Prosthetic Control and Power Systems	Alex	Shane
2.10 Power Sources	Alex	Tynan
2.11 Vacuum Pumps	Ryan	Lexi
3. Project Strategy	See below:	See below:
3.1 Initial Client Statement	Lexi	Ryan
3.2 Objectives	Shane	Tynan
3.3 Constraints	Tynan	Shane
3.4 Revised Client Statement	Lexi	Alex
3.5 Project Approach	Ryan	Shane
4. Project Design	See below:	See below:
4.1 Mechanical System	Ryan/Lexi	Shane
4.1.1 Force Plate Testing	Ryan	Lexi
4.1.2 Estimating Pressure Values in Each State	Ryan	Shane
4.1.3 Socket	Lexi	Ryan
4.1.4 Valves	Ryan	Shane
4.1.5 Bladders	Lexi	Ryan
4.1.6 Bladder Holder	Ryan	Tynan
4.1.7 Air Compressors/Supplies and Air Tanks	Ryan	Shane
4.1.8 Pneumatic System Overview	Ryan	Alex
4.1.9 Testing Rig	Ryan	Lexi
4.1.10 Mechanical System Summary	Ryan	Lexi
4.2 Electrical and Control System	See below:	See below:
4.2.1 Accelerometer	Shane	Alex

4.2.2 Pressure Sensors	Tynan	Alex
4.2.3 Valves	Tynan	Shane
4.2.4 Microcontroller	Shane	Tynan
4.2.5 Complete Circuit Diagram	Shane	Tynan
4.2.6 Power System	Alex	Tynan
4.2.7 Control System Breakdown	Alex	Shane
4.2.8 Alternative Designs	Shane	Tynan
4.3 Testing Rig Experiments and Results	See below:	See below:
4.3.1 Detecting the Correct State	Alex	Shane
4.3.2 The Impact of the Length of Pneumatic Tubing	Tynan/Lexi	Ryan
4.3.3 Fluid Analysis of Pressure and Air Flow Rate	Lexi	Ryan
5. Final Design	See below:	See below:
5.1 Mechanical System	Ryan	Lexi
5.1.1 Socket	Lexi	Ryan
5.1.2 Backpack/Bag	Lexi	Ryan
5.1.3 Air Reservoir	Ryan	Shane
5.1.4 Pressure Regulator	Lexi	Ryan
5.1.5 Valves	Lexi	Ryan
5.1.6 Bladders and Bladder Holder	Ryan	Shane
5.1.7 Molded Leg	Ryan	Lexi
5.2 Electrical and Control System	Shane	Alex
5.2.1 Sensors and Valves	Tynan	Shane
5.2.2 Interface Control Board	Shane	Tynan
5.2.3 Microcontroller	Shane	Alex
5.2.4 Power Supply	Alex	Shane
5.2.5 Whole System Circuit Diagram	Shane	Tynan
5.2.6 Control System	Alex	Shane
5.3 Testing and Analysis	See below:	See below:
5.3.1 Reservoir Life Analysis	Ryan	Lexi
5.3.2 Detecting the Correct State	Alex	Shane
6. Discussion and Recommendations	Ryan	Shane
7. References	Alex	Ryan
Appendices	Everyone	Shane

List of Figures

List of Tables

1. Introduction

Each year in the United States, nearly two million people are living with limb loss and over 180,000 new amputations are performed, either because of disease or trauma (Cooper). As a result, prostheses are constantly being updated to keep up with the high demand for both appearance and functionality. Prosthetic devices can be separated into two distinct groups: passive and active.

Passive devices do not require energy to enter the system. Most rely on pressure adjustment or the gait to create motion within the limb. These devices are generally better for keeping up appearances of intact limbs and joints, as opposed to replacing the limbs' original functions. Most of them include springs, hooks, or small mechanical additions, but overall they are mainly for cosmetic purposes and only retain some of the original functionality of the limb. The sockets of these devices are usually custom-molded to the specific user, paying close attention to the initial cosmetic appearances rather than the varied pressures and forces applied through physical activities such as standing and walking. Especially for gait, the angles and mechanical forces usually stay relatively constant over a variety of gait speeds and activities (Au). The passive prosthetic systems usually have very little functionality built into them, however they have been the decisive choice when the user is given the decision between passive and active devices. Some passive devices are able to perform basic functions but they seldom have the same abilities as an active prosthetic.

Active devices are actuated in response to external forces and a host of sensors and circuitry. These prosthetic systems focus more on functionality than cosmetic appearance. The patient usually has some control over the device and consequently could flip a switch or kick a ball using a system of sensors and buttons. Active prosthetic devices can furthermore be divided into two categories: body-powered and myoelectric.

Body-powered devices use the patient’s body movements, like flexing the thigh muscles or shifting the hip, to control the prosthetic device, usually through a system of cables or electrical wires (Miller, Stubblefield and Lipschutz). The benefits of body-powered prosthetics are the accuracy of the signaling, and the reliability between the intended movement and the result. The disadvantages of this type of device are the unnatural body movements sometimes needed to produce the intended movement, large amounts of energy expended by the user to learn and carry through with the action, and poorer cosmetic appearance relative to passive or myoelectric devices (Micera, Carpaneto and Raspopovic).

Myoelectric prosthetics analyze the user’s electromyography (EMG) to find their intended motion. These devices capture an electrical signal from the muscles, process or analyze the signal, and use the result to control when and how the prosthetic moves and acts. These types of devices allow a patient to flex or extend their knee, rotate their ankle, have fingers that contract and change orientation, as well as perform a greater variety of functions. Myoelectric devices have been more researched for their uses in upper-limb prosthetics within the area of dexterity and degrees of freedom, but they have their uses for lower limb prosthetics as well. Since the legs are the main load-bearing units of the human body, myoelectric devices can be used to adjust forces and pressures on the prosthetic itself and aid in gait form (Sansam).

Whether they are meant for the upper-body or lower-body, prosthetics can create several challenges when being fitted to a user. They can have a discrepancy in weight and balance compared to the residual limb in that they are much lighter. They often cause osteoarthritis in the remaining limbs, instability, and joint or back pain in the movements of the limbs and prosthetic devices. Lower-body devices can create changes in a person's gait, especially if there is a height difference between the old limb and the new device. This may force the user to relearn how to move their limbs so the prosthetic device can respond appropriately. As a result, there is a steep learning curve for users which can cause frustration and rejection of the new device. Another factor in dealing with weight-bearing prosthetics such as trans-femoral devices is the fit and alignment of the new prosthetic, which involves the flexibility of the socket around the stump, the amount and direction of the force being applied from the device and socket to the residual limb, and the positioning of the sensors and electrical components within the socket (Gailey).

Because of all of these issues with fit and performance of lower limb prosthetic sockets, this research group aims to design and build a socket that will adapt to the user's movements. It will use a variety of sensors in order to determine the position of the residual limb, determine the activity level of the user, and adapt in order to keep the socket on the leg and the user comfortable.

2. Literature Review

Each year in the United States alone, nearly two million people are living with limb loss and over 180,000 new amputations are performed, either because of disease or trauma (Cooper 2007). Active powered devices, either powered by the user's own movements or a series of motors and pumps, give an amputee the opportunity to regain much of the original function of the limb, especially if on the lower-limb. These types of prosthetics read a variety of signals, such as EMG, and have a feedback system to aid in the adaptive movement and function of the limbs. Legs especially need a type of adaptive system that will dynamically adjust to the movements and loads presented upon it. The goal of this project was to evaluate all of these approaches and incorporate the most viable of these elements to design and develop a lower limb prosthetic socket that dynamically adapts to the user's environment and provides stability to the user.

2.1 Socket

As mentioned, amputees who receive new prosthetic limbs may reject the device. Many of the complications of this are attributed to poor socket fit (Backus). The socket of a prosthetic device directly interacts with the stump of the amputation by encapsulating it and providing the first set of load-bearing mechanisms. The encapsulation and loads can create many concerns relating to the fit and stability of the device.

If the socket does not fit the user properly, then he or she can develop numerous skin problems including blisters, cysts, edema, skin irritation, pressure sores, and dermatitis. Because of the nature of an amputation, soft tissues which usually receive little load-bearing are now subject to high pressures, forces, and shear motions. These deformations and stresses will affect cellular metabolisms and functions, including blocking blood vessels from delivering nutrients and oxygen to the cells of the surrounding tissue (Mak, Zhang and Boone). Any aid a prosthetic can give in terms of fit would be beneficial for the skin. If the socket fit is too snug on the surface of the skin, then it will trap oxygen, cellular waste, and water. This could cause bacterial growth, pain, and possible infections. Also, heat will be generated from the abrasive nature of the tissue interaction within the socket. If the socket is too loose, this will cause instability in movement of the limb and cause more work for the user to operate the new prosthetic device.

To remain stable, the socket needs to be able to bear and distribute the loads given by the residual limb and the rest of the body. It must also be able to tolerate changes in the loads, as they will generate forces with different distributions and directions. To do this, the socket must either distribute the load over the entirety of the stump, or be designed so the load can be spread more heavily in areas with higher load-bearing. This force distribution will also aid in lessening tissue deformation and inflammation. While cyclic loading can intensify this issue, the tissue can also adapt to these forces by changing its shape and structure (Zhang).

The socket must fit the user almost exactly to avoid instability from looseness and skin irritations from tightness. If the socket is too tight, the amputee will have a greater risk of experiencing discomfort, skin abrasions, and pressure sores. If it is too loose, the amputee may be in danger of slipping out of the socket and falling or experiencing pain and discomfort due to unwanted shifting of the residual limb inside the socket. One way to help with this is to line the socket with soft, elastic polymers that adapt to the loadings given by the residual limb and cause less irritation with the underlying tissues. Another way to address this issue is to use thermoplastic or thermoset polymers to mold the socket around the user’s residual limb to aid with the fit (Mak, Zhang and Boone). Finally, a bladder, or series of bladders, may be used to dynamically adapt to the external loadings and movements of the user as well as assist with both fit and stability.

2.2 Medical Effects of Lower Limb Prosthetics

There are numerous medical problems associated with the use of prosthetic limbs involving the residual limb-socket interface. If the residual limb of an upper-leg amputee is subjected to unendurable loads, various complications with both the skin tissue and residual limb can develop. For this reason, it is important for the designers of prosthetic limb sockets to minimize the negative effects of these loads as much as possible. In that way, people who are concerned about sustaining injuries from their prosthetics may be more active, and in turn healthier, with more comfort.

2.2.1 Ischemia and Pressure Sores

Ischemia is the condition in which blood flow is insufficient, which can result in discolored skin as well as pressure sores (also known as pressure ulcers). Pressure sores are perhaps the most commonly cited medical ailment with respect to lower-limb prosthetics. This insufficient blood flow is caused by external forces that unnaturally act upon the residual limb. Studies have shown that as external loading on the residual limb increases, the rate of blood flow decreases (Mak, Zhang and Boone).

Pressure sores are also very common for people who live more sedentary lives, mostly elderly people who have lost the ability to walk with ease. Sitting or lying down for an extended period of time can cause ulcers from prolonged pressure on particular locations. This is different from the forces experienced by an amputee because amputee’s loads are often repeated in movement, such as walking or jogging. Still, the tissue of the residual limb is not designed for such concentrated continuous loading, so it is important to design an interface that distributes pressure on the leg appropriately to alleviate pressure sores.

2.2.2 Lymphatic Supply and Metabolites

External loads also have a negative effect on the lymphatic system at the residual limb. The lymphatic system, which is associated with the circulatory system, is responsible for transporting excess fluid, protein, and metabolic waste from tissues to the circulatory system. The level of metabolites in sweat can indicate the tissue's current lymphatic supply. Studies show that there is a significant increase in sweat lactate during loading and a decrease in sweat volume during ischemia (Mak, Zhang and Boone).

2.2.3 Skin Abrasion

Frictional rubbing occurs on a residual limb while in a prosthetic leg. Continuous or aggressive rubbing may lead to any number of skin lesions, including calluses, thickening, abrasions, and blisters. Repetitive rubbing also produces heat, which may only add to an already uncomfortable leg-device interaction. Studies reveal that there are two main types of skin reactions to repeated rubbing (Mak, Zhang and Boone). The first is skin thickening, which occurs when the skin is rubbed lightly but continuously. The second is blistering, which occurs if there is a large load on the limb. Jogging would be an example of an exercise in which blisters may form due to large impact and frictional forces.

2.3 Qualities of Existing Prosthetic Sockets

One of the areas of focus in the design was what the leg-device interface would look like. The team investigated the best way in which the residual limb would physically fit into the device. Different prosthetic devices use different materials to cushion the residual leg, but none of them have been able to adapt to the user in the way we envisioned. It was important to see what materials were used in similar devices to determine if they should be used in the design or if the team should have used a completely different material.

2.3.1 End-Bearing Socket

The end-bearing socket incorporates both a casted layer and a fluid layer. There are many steps involved in making this type of device. The first is to make a cast of the residual limb, then coating the predetermined area with a material able to form a liner to conform to the contours of the residual limb. This liner is then placed in the shell of the prosthetic device, and sits on top of a pocket of fluid. The patent (Kandel) from which this information was taken did not specify what the fluid was or what materials were used for the cast or the liner (see Figure 1).

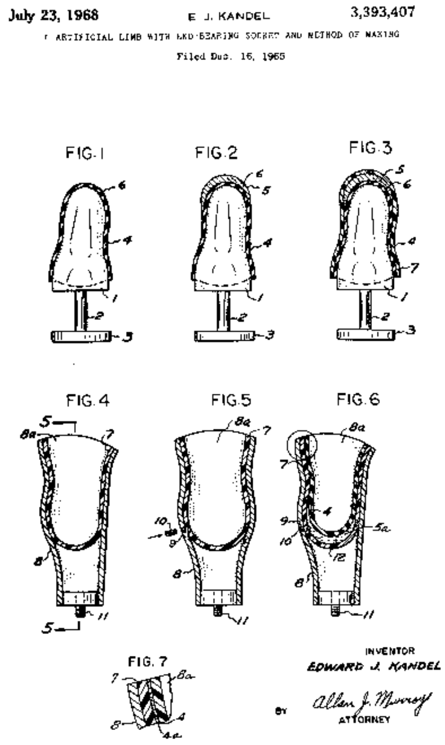


Figure 1 - Patented design for artificial limb with end-bearing socket (Kandel)

2.3.2 Suction Socket

The suction socket design comprises a tubular body that has an open proximal end and a closed distal end. A valve is secured to the socket, allowing air to escape. Unlike the previous end-bearing socket design, this device allows the user to manually adjust the pressure within the device using a valve. This is a one-way valve which expels air from the inside of the socket to the outside, creating a vacuum in the space between the interior of the socket and the patient's leg (see Figure 2). The leg stays in place within the socket until air is released from the pressure valve (Lenze and Rossi).

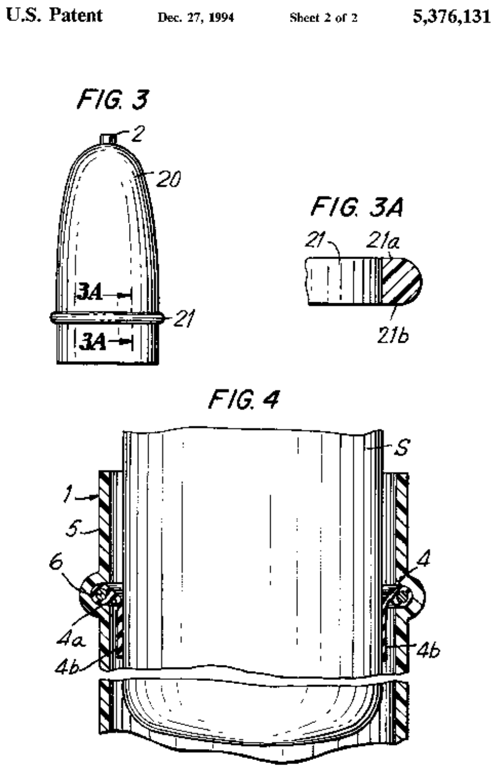


Figure 2 - Suction socket for artificial limb (Lenze and Rossi)

As shown in Figure 2, this suction socket design does have some significant disadvantages. One of the disadvantages is its ineffective seal at the proximal open end of the socket. Although the socket may momentarily be perfectly fitted to the stump, this will not last long since the stump undergoes shrinkage and swelling regularly. Shrinkage allows air to leak into the socket and reduces the differential air pressure that is keeping the stump in the socket. One technique that is utilized to solve this problem is having the person wear special socks to fill the residual space, but it becomes rather inconvenient when the user has to continuously be removing and applying the leg and socks (Lenze and Rossi).

2.3.3 Hypobarically-Controlled Socket

This design is very similar to the suction socket mentioned above, with regard to its intent. It utilizes a similar pressurized vacuum seal concept. One difference is that this design claims to contain a semi-compressible molding material that contours to the residual limb under the influence of the vacuum, as shown in Figure 3. The device can be hooked up to a vacuum source and a vacuum regulator that can control the tightness of the vacuum.

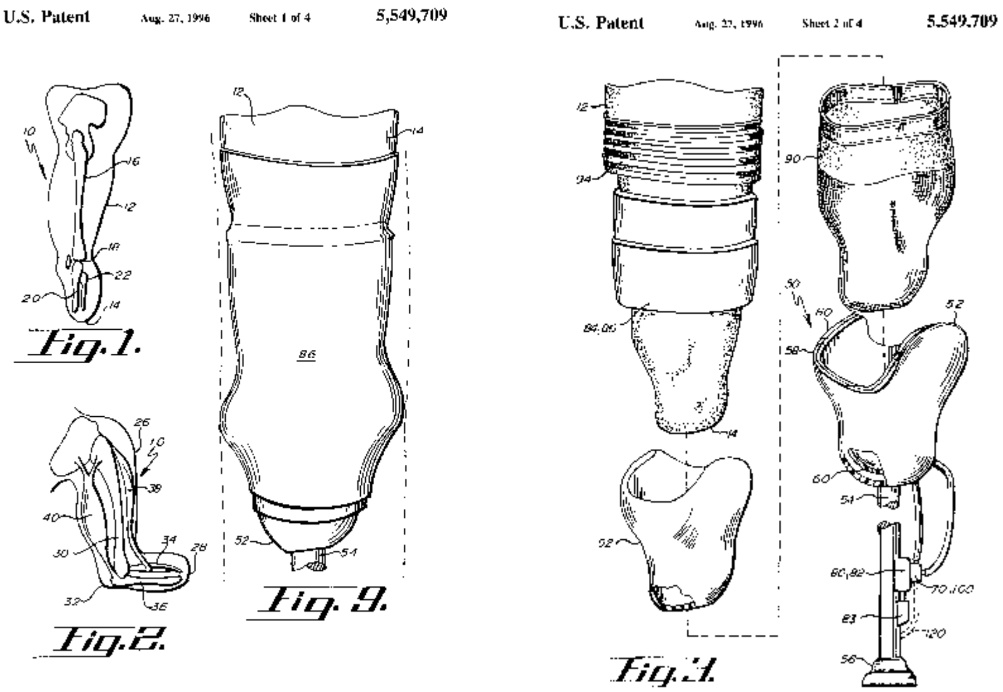


Figure 3 - Hypobarically-Controlled artificial limb for amputees (Caspers)

Although this device does give the user control over the tightness of the socket, it does not do so automatically and requires manual adjustment (Caspers).

2.3.4 Smart Variable Geometry Socket (SVGS)

The problem with the above methods is their inability to automatically manage the volume of the socket. The Smart Variable Geometry Socket (SVGS) is a device that automatically and continuously maintains the appropriate fit for the socket. The SVGS has three major components: a mechanical control circuit, a fluid reservoir, and several discrete bladders inside the socket. No external power is needed, as all the power comes from the movement of the amputee. This is possible due to a “natural pumping cycle” that occurs in the socket. As the user walks or runs, the suction is created in the socket during the swing phase. Also during this phase, fluid is drawn into the bladder. The socket then compresses in the stance phase when the user’s foot touches the ground, and the fluid is then distributed among the bladders to provide support. Throughout this process, only gravity and dynamic forces control the level of suction. This design provides the user with an automatically changing internal pressure and suction that is driven entirely by the dynamic forces acting on it (Greenwald).

2.4 Bladder

Prosthetic limbs are usually rigid, giving them little area to move around and adapt to changes in body position or gait. This characteristic benefits the patient's stability during movement and reduces rubbing, but it can also be harmful because it gives the user less flexibility. Because the movement is unnatural for the user, it can require a significant effort to learn how to walk and move again with this new limb. This can cause frustration and discourage patients from using their prosthetic. In addition, as it was mentioned before, direct contact between the friable skin and the stiff prosthetic causes inflammation, swelling, and irritation. Studies have shown that the residual limb volume can vary between -11% and 7% of its original volume in a single day, depending on activity levels and weight (Montgomery). If the prosthetic is already fitting too tightly around the skin, it can cause more pain and irritation, which forces the user to compensate by putting more weight on their remaining leg. This is a harmful cycle that causes more harm to the user and is an important area to consider when fitting bladders into a socket.

A bladder may be used to help with fit and stability. One or more bladders may be placed into the socket between the residual limb and the prosthetic socket. These bladders can be used to help fit a variety of users, meaning the same socket and prosthetic can be used by multiple people of varying shapes and sizes by inserting these bladders to adapt with the shape changes.

Bladders can also be used for adapting to dynamic movement. They are normally either filled with air or a liquid, both of which are able to adapt to changes in pressure and different forces. These bladders can be a buffer between the limb and the socket, allowing distribution of loads throughout the joint during dynamic and static activity. One way to improve this situation is to use a combination of fluid-filled and pneumatic-actuated bladders, along with sensors at the bladder and skin surfaces, such as in the device shown in Figure 4. As the user moves, the sensors detect if any of the bladders are in contact with the skin surface. If either of the bladders is not in contact, it would send a signal to a reservoir of liquid or gas. This, in turn, would cause that bladder to fill with the fluid. Then, when the bladders have been in contact with the surface of the skin, a signal can also be sent to allow for some removal of the gas and liquid from the bladder (U.S. Patent No 13/506, 471, 2012). This allows the fit and stability of the prosthetic socket to be dynamically adjustable during daily motions (see Figure 4).

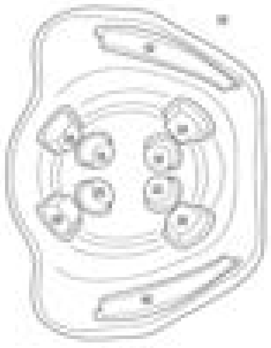


FIG. 3

Figure 4 - Prosthetic Limb Bladder System

Figure 4 shows a top-down view of the socket. Within the socket are 8 small bladders around the circumference and length of the residual limb. Along the interior of the socket, there will be wires and electronics that would control the bladders.

2.5 Kinematics

Kinematics is the study of motion and position in a system without regard for the dynamics. In biomechanical research, it is used to measure location-related properties as well as predict them based on an object's size and joint angles.

Kinematic signals measure characteristics such as position, velocity, and acceleration of a device. The kinematics of the human body can be detected using wearable health monitoring systems (WHMS). These sensors are usually connected to a microcontroller. One common kinematic sensor is the accelerometer, which measures an object's acceleration in 3D space (Pantelopoulos and Bourbakis). Researchers have performed studies involving accelerometers to observe human movement with and without lower-limb prosthetics.

One study, performed by the MIRA Institute for Biomedical Technology and Technical Medicine, involved measuring the upper-leg activity while a person is walking to observe the difference between amputee and non-amputee movement. The researchers noted that trans-femoral amputees would use their healthy leg for support as long as possible, creating an asymmetrical gait (Wentink). This indicates that amputees are not as comfortable adding loads to their prosthetic legs as they are to their healthy ones. It also reinforces the idea that many amputees relearned how to walk after attaching a prosthetic.

In addition to modeling of gait, kinematic models have also been used to study the motion of upper limbs. Engineers used a 3D local coordinate system to develop a simulated version of a person's shoulder in the form of a kinematic chain. The model was divided into four separate segments: thorax, clavicle, humerus, and scapula. The kinematic chain included joints connecting these sections, allowing the program to model various different positions and movements of the shoulder (Jackson). This project showed how kinematics can be used to model the actions of the human body.

2.6 Sensors

The adaptive prosthetic should be able to determine the user's current activity as well as ensure that he or she is healthy and that the device is fitting properly. Sensing technologies to measure various biological signals include electromyography (EMG), pulse oximetry, galvanic response, and pressure on the leg.

2.6.1 EMG

When a muscle contracts or moves, the muscle is responding to nerve stimulation sent from the central nervous system. Several medical tests can be performed on a patient to help detect muscle movement and activation. One test is an electromyography, or EMG. An EMG “measures muscle response or electrical activity in response to a nerve’s stimulation of the muscle” (John Hopkins Medicine). When measuring EMG, the patient is usually asked to contract his or her targeted muscle to get an appropriate electrical signal reading (MedlinePlus) and then the signals can be logged as data.

When utilizing EMG signals, there are two main methods. One method is intramuscular while the other method is on the surface of the body. Intramuscular EMG is measured by inserting small needles or wires into the muscle and detecting electrical activity. These EMGs will not be further researched or exploited throughout this paper. Incorporating an intramuscular EMG that involves needles being put on the socket of a lower-limb prosthetic is extremely complex. Also, the needles must be applied by a professional and only work for short periods of time, making them impractical for a prosthetic that must be worn for several hours. Surface EMGs, or sEMGs, are the most feasible and practical for prosthetic applications as they can stick to the skin, instead of piercing through the skin into the muscle. However, with this in mind, more signal processing is sometimes needed for sEMGs, as they may pick up more noise. Surface EMGs have a more clinical and practical use, but because the sensor is not directly connected to the muscle, the signal may not be quite as accurate. The sensor sometimes suffers from detecting other movements or muscle action potentials, such as muscle crosstalk (Kamavuako, Scheme and Englehart). This is important to keep in mind when integrating the sEMGs into the socket.

Surface EMGs may not be as reliable as intramuscular EMGs, but they still exhibit valuable characteristics. Prosthetic control using EMG signals is a field that has been growing, and sEMG has the advantage because no harm is done on the body and it has many potential applications. Surface EMGs have been used in multiple cases for prosthetics. In one experiment, the signal processing system used with sEMGs successfully distinguished noise from movements and the actual muscle signal (Ryu and Kim, 2014). This shows how with correct signal processing, sEMGs can be used to collect muscle activity data and recognize the user's intent.

An important aspect of sensors in the use of lower-limb prosthetics is predicting the user’s gait phase within their gait cycle. The gait cycle consists of steps and strides in a repetitive manner, and is an important aspect characterizing the walking of humans (PhysioPedia, 2014). The gait cycle consists of two phases, the swing phase and the stance phase, as shown in Figure 5 (Ryu and Kim, 2014). Through an experiment done in South Korea, researchers Ryu and Kim discovered that one can not only determine the current step of a human’s gait phase, but help predict the next step in the phase of the gait cycle. Also, they determined a correct prediction of the gait cycle could greatly contribute to intelligent prosthetics and rehabilitation (Ryu and Kim, 2014). In other words, incorporating sEMG sensors into a socket of a lower limb prosthetic might be used to predict the next step of the user’s gait cycle.

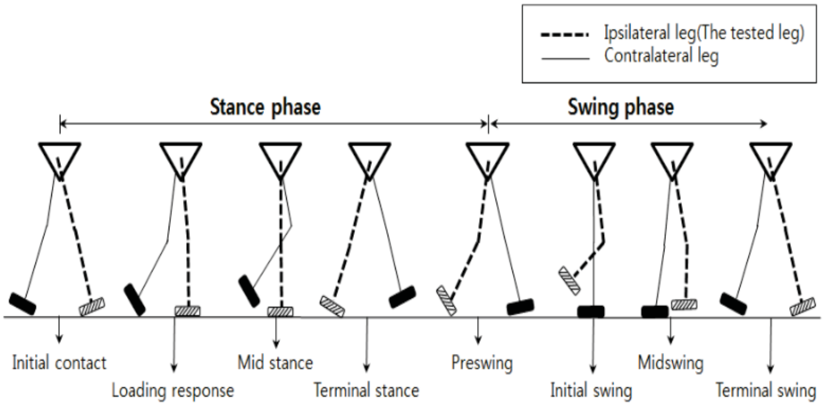


Figure 5 - Gait phases of a human walking (Ryu and Kim, 2014)

Including a surface EMG signal can be very beneficial to an adaptive robotic socket used for prosthetics. For instance, using the EMG signal from a person's residual limb could help predict what the user is trying to do. Also, intramuscular EMGs have shown that they can predict upper limb movements with respect to force (Kamavuako, Scheme and Englehart). EMG-sensing in a prosthetic socket can help communicate with the control aspect of the prosthetic to predict the user's movements. For instance, if there is no EMG signal, the user may be sitting or lying down and the prosthetic could loosen.

2.6.2 Pulse Oximetry

To determine the amount of oxygen saturation in the blood, a non-invasive method known as pulse oximetry can be used. Pulse oximeters emit both infrared and red light into tissue. Hemoglobin in blood that contains oxygen absorbs more infrared light while hemoglobin without oxygen absorbs the red light. The sensors then receive the light and determine the amount of oxygen in that area based on the remaining light. Pulse oximetry determines the oxygen saturation of the blood in tissues, such as the finger, earlobe, and nose (John Hopkins Medicine).

One of the major complications associated with lower-limb prosthetics are pressure sores. Pressure sores develop on the residual limb when the socket and prosthetic are too tight. This is because there is too much force on a given area, which in turn results in insufficient oxygen levels in the blood of the tissues. Utilizing oximetry sensors can determine if tissues in the body are receiving enough oxygenated blood (John Hopkins Medicine). However, most pulse oximeters are used on the finger, ankle, foot, or earlobe where there is little body mass. It is assumed to be more difficult to measure the oxygen-saturation of the blood accurately on the upper-leg area of a residual limb. The team could not find any research or prior work of scientists or engineers trying to incorporate a pulse oximeter inside a socket. However, if executed correctly, the pulse oximeter could detect the possibility of low oxygen levels in the blood and alert the user. Also, a release or decrease in pressure on that spot of the body could return the necessary flow of oxygenated blood to the tissues.

The most widely, clinically used pulse oximeters are of type transmission. In other words, the “emitting” and “detecting” parts of the sensor are on opposite sides of the tissue, and therefore only able to be used on thin body tissues such as an earlobe or a finger. However, another type of pulse oximetry sensing is available and known as reflection pulse oximetry. This type of pulse oximeter has its “emitting” and “detecting” aspects of the sensor on the same side of the body tissue being measured (Mendelson). Therefore, reflective pulse oximeters can be used on any part of the body, including residual lower limbs.

2.6.3 Galvanic Response

Galvanic skin response, also known as the electrodermal response, indicates the changes in electrical properties of a person’s skin. The electrical properties of skin change when there are changes in the wetness on the skin surface. This response is often used in psychological studies because the amount of sweating that occurs on the skin correlates with emotional changes that could not be sensed otherwise (Montagu, 1966). When a person is uncomfortable or emotional, it can cause them to sweat. When they sweat, their skin becomes more conductive, which can be measured by a galvanic skin response sensor. This can even detect subconscious changes in a subject's mood or emotion that they might not even realize (McCleary, 1950).

A galvanic skin response sensor applies a constant voltage across the skin. The voltage is very small and cannot be felt by the subject. The sensor then measures the current that flows through the skin as the voltage is applied. Using the known voltage and the changing current, the device can determine the skin conductance in microSiemens. This is done using Ohm's Law, which is *Voltage = Current * Resistance*. Using this law, resistance is calculated as *Voltage/Current*. Since resistance is the reciprocal of conductance, the conductance can then be calculated as 1/*Resistance*.

There are two types of skin conductance, tonic and phasic. Tonic skin conductance is the baseline conductance of the subject. This is the conductance when there are no environmental stimuli. People can have different tonic skin conductance values, but they typically range from 10-50 microSiemens (Vijaya, 2013). Phasic skin conductance is the conductance that changes when events occur. Several types of stimuli can evoke emotional or physical stress on an individual, which increases sweating and increases conductivity.

2.6.4 Pressure Sensors

Pressure sensors can be attached to a prosthetic to determine if there is an appropriate amount of pressure between the user and the device. If there is too much or too little pressure, the user may not be wearing the device properly. There are a variety of pressure sensors capable of being used in a prosthetic, though there are a few major sensor types that are typically used. The first sensor is the strain gauge. Strain gauges are used in circuits known as Wheatstone bridges. The change in pressure on the gauge will affect the resistance and the voltage across the sensor, which can be detected and calculated using the bridge circuit. The size of the change in voltage can determine how much pressure the sensor is experiencing. Another type of pressure sensor is the capacitive pressure sensor. These sensors consist of two metal plates with a dielectric material between them. Under pressure, one of the plates will move so that the capacitance between the plates changes. These sensors are not accurate at high temperatures, but due to their linear output, they are widely used at ambient temperatures. A third type of sensor is a piezoelectric pressure sensor. These sensors have piezoelectric crystals that develop a voltage difference when they are under pressure. The change in voltage of the crystal indicates the change in pressure (Bao).

Strain Gauges

Strain gauge sensors show the change in resistance of a metal foil or wire when it experiences some kind of deformation. The increase in resistance is a result of an increase in the length of the metal and a decrease in its cross sectional area, since the electrons in the metal have a more difficult time flowing through a thinner piece of metal. When a metal is compressed, the resistance decreases since the electrons can move more freely in a wider piece of wire or foil. These strain gauges are often bonded into a Wheatstone bridge arrangement to create pressure sensors.

Under extremely heavy loads, the strain gauge sensors may become unusable because it could reach its elastic limit, or maximum strain. If this limit is exceeded, the gauge becomes permanently deformed and changes some of its material properties (Craddock). For this reason, strain gauges should be used with caution since exceeding the elastic limit could damage the sensor or produce inaccurate results.

Piezoresistive

Commercially, piezoresistive strain gauge pressure sensors are successful due to their high sensitivity and uniformity. These sensors are made up of a square, thin, and deformable sheet of single-crystal silicon with piezoresistors embedded in the center four edges (see Figure 6). These piezoresistors are placed at the areas on the film that experience the most strain when the sheet is subjected to external loading.

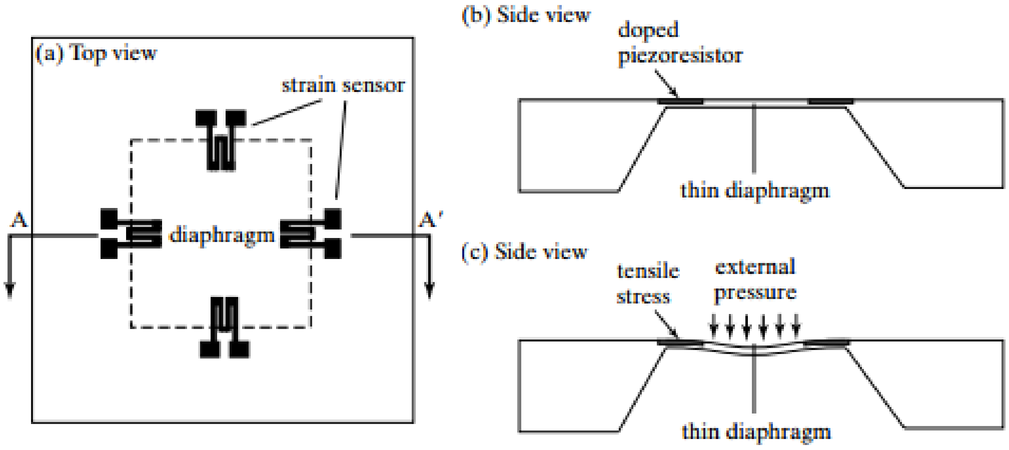


Figure 6 - Bulk Piezoresistive Strain Gauge Pressure Sensor

There are two main types of piezoresistive pressure sensors, bulk and surface pressure sensors. The most significant difference between the two is the size of the sensor and the controllability of the sensor. Surface micro-machined piezoresistive pressure sensors are smaller and offer more controllability than bulk micro-machined pressure sensors. Figure 7 shows a diagram of a surface sensor.

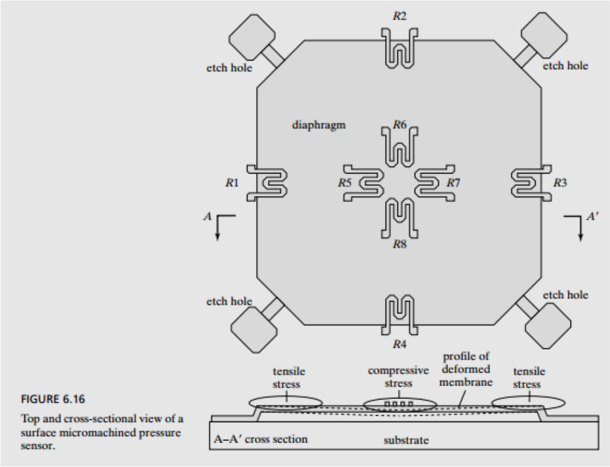


Figure 7 - Surface Piezoresistive Pressure Sensor

Eight piezoresistive sensors are involved in the design of these complete sensor assemblies. There are four resistors located in the middle of four membrane edges, which are suspended from a small gap in the substrate. Four of the eight sensors experience tension when the membrane is deflected downward, while the other four experience compression when the membrane is deflected downwards. (Chapter 6: Piezoresistive Sensors)

An advantage to using these sensors is the high output levels of 20mV/V. However one drawback is that one needs to correct the temperature coefficient of the piezoresistors and the long term drift of the resistors to eliminate 0.1% per year drifts, which will skew data in future uses of these devices (Craddock).

Quartz Sensors

Quartz is a material that is interesting to mechanical engineers because of its brittle nature. It can be capacitive, much like the silicon in the piezoresistive devices, or it can be based on the Bourdon tube principle. This principle involves a coiled tube of fused quartz, which is pressurized to create a rotational force that will cause the tube to try to unroll. This can then be measured using capacitive or inductive techniques. The drawback for these types of sensors is the size and fragility of the sensor, which only makes it useful and practical for precision instruments (Craddock).

Force-Sensing Resistors

Perhaps the most useful pressure sensor for our application is a Force Sensing Resistor (FSR). When a force is applied to this resistor, the electrical resistance decreases. Thus the greater the force applied, the more conduction, which allows more current to flow. This type of sensor would allow us to determine how much force is experienced at various locations in the socket during the specific states of the amputee (standing, sitting, and walking).

2.7 Software Interfaces

Researchers have programmed integrated circuits involved in prosthetics primarily to perform three different functions: creating a graphical user interface (GUI), analyzing the shape of a person's stump, and processing digital signals from the biosensors.

2.7.1 Graphical User Interfaces in Prosthetics

Many prosthetic devices have been accompanied with GUIs that allow the doctors and patients to observe characteristics and test different abilities. A team of engineers designed a multi-fingered prosthetic hand with 14 degrees of freedom (DOF). Since a robot with 14 DOF could have various different poses, the researchers created a GUI that allowed the user to control each DOF individually, as well as multiple fingers at once. It also includes two buttons for grasping and releasing objects (Bahari). The i-LIMB prosthetic hand, created by Touch Bionics, includes GUI software called BioSim, which allows the prosthesis to analyze myoelectric signals and edit the controls and triggers of the device. BioSim also includes software called BioSim Patient, which allows the patient to view input signals and change the triggers themselves (Waryck). Another myoelectric hand, called the Bebionic v2, uses Bebalance software as a GUI and communicates with it via a USB dongle or RF module. The RF module is built into the hand; allowing the prosthesis to edit the setup without disconnecting the prosthetic from the patient. The GUI can set the system into one of five operating modes, change the input signals, or edit the default grip (Waryck).

2.7.2 Analyzing Shape

Programming software has also been utilized in analyzing the shape of a patient's residual limb. In 2009, an interface was designed to accurately find millions of points on a person's knee using C++ programming, OpenGL, and a laser scanner. The interface was capable of generating coordinate data for each point and combining the points into a structure. The result was then read by CAD software which would quickly design a socket to fit the patient (Lai).

2.8 Digital Signals

Prosthetics that read biological signals must process the values from the sensors for them to be read in an identifiable form. An effective way of analyzing these signals is to convert them into digital signals, for digital processing can be more efficient and flexible than analog. However, according to the sampling theorem, the digital signal must have a sampling frequency of at least twice the bandwidth of the analog signal in order to measure the actual frequency accurately. Otherwise, aliasing can occur, in which the system will believe the frequency to be lower than it actually is (Liang, Bronzino and Peterson).

Digital signals must also be processed effectively, as they can contain inaccurate values, or noise. There are various methods for filtering out noise on a discrete signal, such as averaging a fixed set of values (Liang, Bronzino and Peterson) and using the Discrete Fourier Transform

(Das). Averaging can separate signal from noise if a system output repeats similar values at every iteration. The equation for averaging signals is shown below:

$$y_t(n) = \left(\frac{1}{N}\right)\sum_{i=0}^{N-1} y_i\,(n-i)$$

According to the equation, y_i is the recorded signal and y_t is the output signal after averaging N values. The signal-to-noise ratio is directly proportional to the square root of N, so if the true signal value never changes, then the signal will become more accurate when more values are used for each average (Liang, Bronzino and Peterson).

The Discrete Fourier Transforms (DFT) can convert digital signals from the time domain into the frequency domain, allowing them to be affected by bandwidth filters (high-pass, low-pass, bandpass, and bandstop) which can eliminate frequencies that are too extreme to be realistic. A DFT uses the following formula:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} x(n)e^{-j\omega n}$$

In the above equation, *X(k)* is the result of the transforming recorded signal *x(n)* being converted into angular frequency ω. The angular frequency is also equivalent to *2πk/N*. The variable N is the number of samples recorded, while the variable k increments from 0.

The DFT will be able to take any finite set of data values and convert it into a set of discrete frequencies (Das).

2.9 Prosthetic Control and Power Systems

Prosthetic devices require control systems which allow them to work efficiently and accurately when the user wears them. Many of these devices, particularly those meant for the upper-limb, use myoelectric signals as part of their control system. The signal acquisition and interpretation for these devices can be separated into three steps: acquiring the signals, reducing their dimensionality, and, for recent prosthetics, classification (Pietka and Kawa).

2.9.1 Signal Acquisition and Interpretation

Myoelectric signals can be measured in two ways: having many electrodes in various locations, or only using a few in carefully chosen locations. Using only a few electrodes reduces the calculations required by the device's control system, though the sensing itself would require smaller quantization errors to achieve the same precision and accuracy (Graupe, Beex and Monlux). The system may include an EMG preamplifier to increase the signal's strength.

These signals are recorded as discrete values, allowing the system to process them more efficiently. The quality of the signal is dependent on the signal-to-noise ratio, so removing as much noise as possible is essential to understanding the user's intent. Noise generally appears as undesired frequencies, which can be removed from the system using a Butterworth bandpass filter (Graupe, Beex and Monlux). Once the noise is removed, the control system can sort the signals based on patterns within them, such as sudden amplitude changes. The system will then use these patterns to understand the user's current activity and instruct the device to perform an appropriate action (Pietka and Kawa).

2.9.2 Control Algorithms

The control algorithm for a prosthetic device can sometimes be separated into higher and lower controllers. A powered trans-tibial prosthetic device created by Jinming Sun and Philip

Voglewede was separated in such a way. This device had a higher level finite state controller while the lower level controller operates the motors (Sun and Voglewede). The higher level controller would process digital signals and determine if the device was in "stance phase or swing phase based on if the prosthesis foot has contact with the ground” (Sun and Voglewede). This information was later used by the lower level controller to move the motors appropriately using Proportional-Integral-Derivative control. This setup improved the speed of the control system as it was able to perform multiple functions at once.

Controllers sometimes need to use serial communications to send messages between different parts of the device. This is the case for a two-DOF prosthetic wrist, designed by a team at the Institute of Biomedical Engineering at the University of New Brunswick (Brookeshaw, Bush and Goudreau). The wrist received inputs from the arm, such as myoelectric signals and inputs from switches, via a Controller-Area Network (CAN) bus, after being processed in a translator board. This also allowed the wrist to connect to other serial communication systems.

2.10 Power Sources

It is essential that prosthetic devices are powered properly, ensuring that they remain fully operational as long as possible. Most powered prosthetic devices are connected to a rechargeable battery, which helps to increase the portability of the system.

Two common types of rechargeable batteries are lithium-ion and lithium-polymer. Lithium-polymer batteries are more sensitive to abuse and easier to damage (Slee). Rechargeable batteries can also be built into the prosthetic or be easily replaceable. Replaceable batteries allow the user to carry a backup battery and exchange it as needed without returning to the charger immediately (Liberating Technologies, Inc.). Built-in batteries have the benefit of being concealed more easily, allowing the prosthetic to look more like a human body part. Upper-body prosthetics typically use 7.2-Volt batteries to control their motors. The prosthetic designed by Jinming Sun and Philip Voglewede for below-knee amputees uses a 14.8-Volt battery for the motors and a 5-Volt battery for the microcontroller.

2.11 Vacuum Pumps

Prosthetics with bladders often use pumps to actuate them. These pumps, which can be either electrical or mechanical, can be used for vacuum-assisted suspension, or changing the pressure between the residual limb and the interior of the socket. Electric pumps require a power source, such as a battery, but they are able to monitor pressure and reactivate when the pressure drops below a certain threshold (Komolafe, Wood and Caldwell). Many prosthetics use lithium-ion batteries to power a motor, which will run an electric pump connected to the bladders. Mechanical pumps, which do not require an external power source, are capable of pulling air from the socket into a separate chamber.

A study performed at the Northwestern University Prosthetics-Orthotics Center tested several vacuum pumps, some electrical and some mechanical. The two electrical pumps tested were the Harmony e-Pulse and LimbLogic VS. The LimbLogic was capable of performing more evacuations, or removing the air from a socket, before its battery was depleted, despite having less battery energy stored. It was also able to perform evacuations faster than the Harmony e-Pulse. The three mechanical pumps tested were the Harmony P2, Harmony HD, and Harmony P3, all created by Ottobock. The Harmony HD was intended for heavier users (up to 150 kg), while the other two pumps were better suited for lighter users (up to 100 kg). The mechanical testing results were very similar for all three pumps in terms of evacuation time, but the Harmony HD required the most force while the Harmony P3 required the least.

3. Project Strategy

Although prosthetic socket technology has been rapidly advancing, amputees are still concerned that their socket and prosthetic are not adaptive to their current activity or consistently comfortable. The goal of our project was to design a prototype of an above-knee adaptive prosthetic socket that responded to an amputee's movement and physiological signals to alleviate pressure on the residual limb during daily activities. The device adjusted multiple air-filled bladders appropriately when the user was standing, sitting, and walking.

3.1 Initial Client Statement

Develop a smart robotic socket for prosthetic limbs. In particular, a socket for lower limb prostheses that dynamically adapts to the user, environment, and intended motion to allow for a stable, but comfortable fit. This could incorporate integrated physiologic sensing, including electromyography (EMG), oximetry and perfusion, and galvanic sensors. This device must also include mechanical sensing, such as pressure sensors or an Inertial Measurement Unit (IMU) sensing. It should contain controllable air-filled bladders for adapting fit and perhaps other means. This socket would incorporate user intent modeling and all of this must be packaged into a portable, battery-powered unit.

3.2 Objectives

After receiving our initial client statement, the team listed various objectives and goals for our project. After finding over 15 different objectives, we put them in a hierarchical structure by creating an objectives tree (see Figure 8). We determined our five higher level objectives to be user-friendly, safe, reliable, cost-effective, and adaptable to the environment, as well as finding several lower-level objectives.

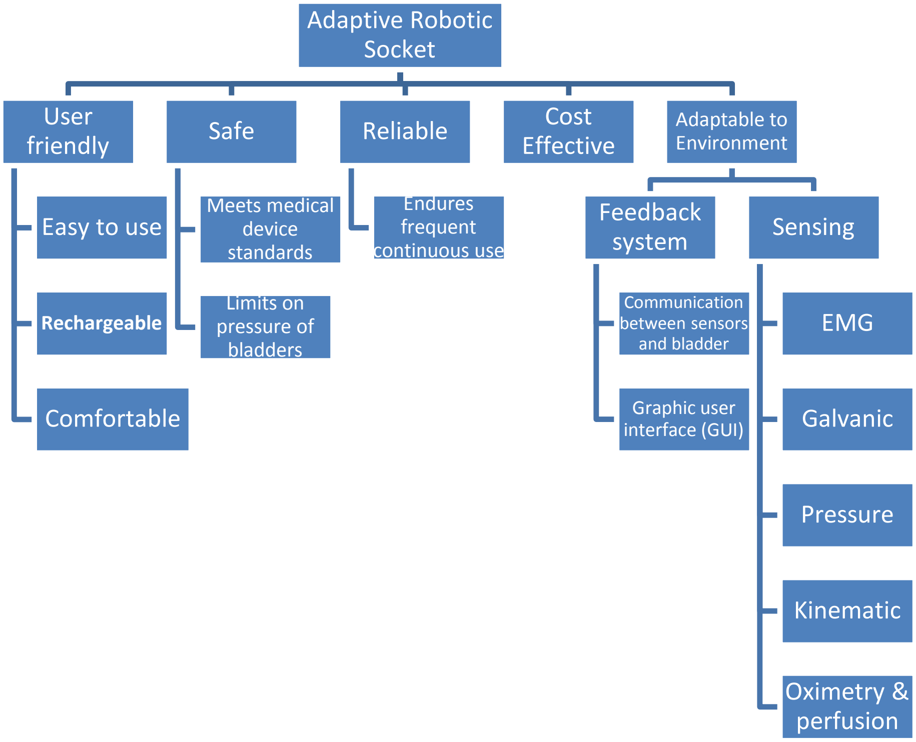


Figure 8 - Objectives Tree

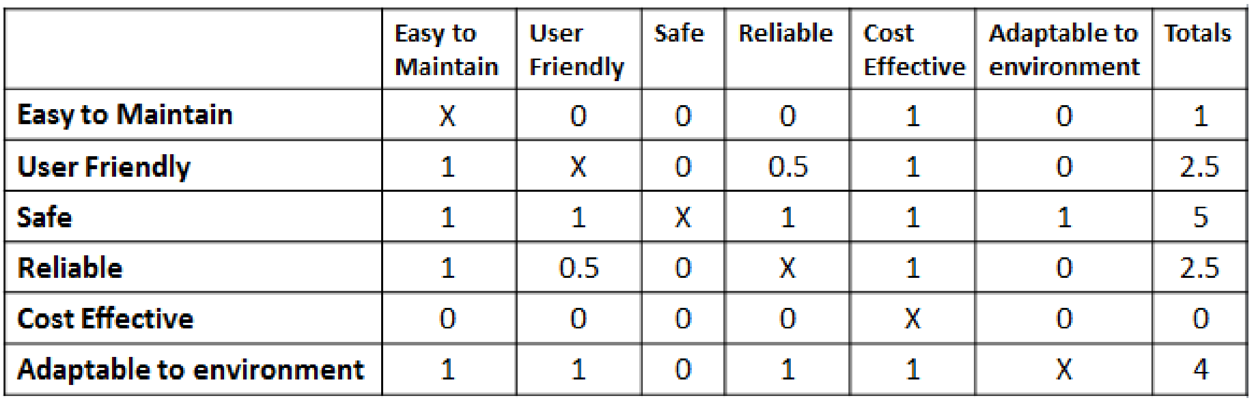
The adaptive robotic socket must be user friendly. This emphasizes that the socket should be easy to use, rechargeable (enough for a charge to last a full day), and comfortable. The socket should not be complicated and should be straightforward in terms of understanding how to use it. It also should avoid any external power supplies and exploit the use of rechargeable batteries. The socket itself should not cause the user too much irritation or be too uncomfortable.

Since the adaptive robotic socket is a medical device being used on humans, it must be safe and reliable. According to the FDA, external limb prosthetic components, such as sockets, are considered Class I medical devices. That means these sockets do not need to submit a premarket notification, which is a document that is requested by the FDA to show that the device is safe and effective (U.S. Food and Drug Administration). Also, we must make sure the pressure of the bladders do not cause any pain or harm to the user's residual limb. These values were researched and tested and shown in section 4.1.4. In terms of reliability, the socket must not fail after being continuously used over time.

One of the main goals of our project was to make a prosthetic device that is adaptive to the environment. The socket would be able to do this with data collection from various sensors. The signals we were hoping to acquire were EMG, galvanic, pressure, kinematic, and oximetry. We also hoped to use these biological signals to communicate with the bladders inside the socket to react to the user's environment. If time permitted and if we saw it to be necessary, we hoped to include a graphic user interface (GUI) to display to the user, and possibly their physician, their biosignals.

To determine the importance of our objectives, we created a pair-wise comparison chart (see Table 1). This chart was used to rank our objectives, in which a higher number indicates higher importance. We determined that our most important objective was to make the adaptive robotic socket safe. The following objectives in order of importance are: adaptive to the environment, reliable, user friendly, easy to maintain, and cost effective.

Table 1 - Pair-Wise Comparison Chart



	Easy to Maintain	User Friendly	Safe	Reliable	Cost Effective	Adaptable to environment	Totals
Easy to Maintain	X	0	0	0	1	0	1
User Friendly	1	X	0	0.5	1	0	2.5
Safe	1	1	X	1	1	1	5
Reliable	1	0.5	0	X	1	0	2.5
Cost Effective	0	0	0	0	X	0	0
Adaptable to environment	1	1	0	1	1	X	4

3.3 Constraints

The following constraints are conditions for our project that the overall design must meet in order to be successful.

Budget: The budget we are given by Worcester Polytechnic Institute is $160 per student. This allows us to use up to $800, which will be contributed by the school. These resources will be used towards the entire project. This includes the initial design, prototyping, and troubleshooting, as well as the fabrication of the final socket and any other purchases we may have to make throughout the project. In the case that additional funding is needed, we are expecting to use some of our own personal money or possibly requesting additional funds from Worcester Polytechnic Institute.

Time: The device must be completed by the end of the academic year 2014-2015 (by May 2015).

FDA Standards: A commercial version of the proposed socket design must adhere to all FDA regulations because it is considered a medical device. These standards were discussed in section 3.2.

Biocompatibility: The device must be biocompatible and non-invasive in order to be considered successful. It must not cause unwanted responses to the patient's body. This means that it must be built using biocompatible materials, which will not adversely affect the skin of the individual.

Portability: The device must be portable. This means that it cannot be too large, since it must be part of a prosthetic leg. It must be usable within a prosthetic limb and allow the user to complete normal tasks such as sitting, standing, and walking. It must also be able to power itself for an extended period of time in order for the socket to be considered a feasible solution to the issues it is addressing.

3.4 Revised Client Statement

Design and develop a lower-limb prosthetic socket to be used by above-knee amputees. The socket is required to be safe and easy to use for the user. It must be usable for the following activities: standing, sitting, and walking. The socket must incorporate signals from sensors that may include mechanical, electrical, and physiological sensors as well as able to adapt dynamically to the user's environment and motion while providing comfortable stability using some form of bladder system. This socket must use a control system that takes accurate feedback from the sensors, determines the amputee's current state, and adjusts the bladders and the socket accordingly.

3.5 Project Approach

The purpose of this project was to design a proof of concept prosthetic leg socket which will attach above the knee and actively adjust to the appropriate pressure on the residual limb of an amputee as he or she is sitting, standing, or walking. To do this, we would utilize pressure sensors and kinematic sensors. We also investigated incorporating EMG, oximetry, temperature, and galvanic response sensors into the design. This socket would use a control system that takes feedback from the sensors, determines the amputee's current state, and adjusts multiple fluid or air filled bladders accordingly within the socket.

3.5.1 Internal Physiological Sensing

Using these types of sensors, we would be able to gather necessary information from the user and communicate it to our control system. We intended to investigate four types of internal physiological sensors: EMG, oximetry, temperature, and galvanic response. Each sensor would gather a different piece of important information that would tell the control system how the bladders need to be adjusted.

An EMG sensor, along with an accelerometer, could be used to determine intended movement and general muscle activity. This is important information because it describes to the control system what kind of activity the user is performing, such as standing, sitting, or walking.

Oximetry sensors would give us more descriptive information about the status of the residual limb with regard to oxygen saturation. When these sensors detect inadequate oxygen saturation (less than 96%) to a specific location on the residual limb, the control system would then be told that the bladders need to adjust to alleviate the pressure on those problem areas.

Galvanic response sensors would also be useful to our design because they would allow us to determine the user's physical activity level by how much they are sweating. Temperature sensors could also be used to serve a similar purpose.

3.5.2 External Physical Sensing

Aside from measuring internal physiological information from the user, it would also be important that we sensed external mechanical signals. The two external signals that we would be sensing were pressure on the residual limb and kinematic signals.

A pressure sensor is necessary to ensure that the user is never experiencing too much pressure on any location for an extended period of time. It would also help prevent the bladders from experiencing pressure that would cause them to break. When too much pressure would be sensed in any location, the control system would know to adjust the bladders accordingly to alleviate the pressure. Kinematic sensors, much like the EMG sensor, would give us information regarding the intended movement of the user.

3.5.3 System

All of the above signals would be converted into digital signals and interpreted by our control system. The readings would be presented in an understandable format when connected to a PC. The control system would be programmed to understand how to inflate and deflate each air-filled bladder based on the intended movement and the physiological response of the residual limb.

3.5.4 Adjustable Fitting

The residual limb would be rested within the socket on an adaptive pad of independently inflating and deflating air-filled bladders. The amount of air within each bladder would be controlled by an air pump that follows instructions from our control system. Each bladder would be adjusted independently to allow for pressure relief on a multitude of spots on the residual limb. If time permitted, we would even look into creating a massage setting for when the user is at rest, to not only make the device more comfortable, but also to increase blood flow in the residual limb.

4. Initial Design

This chapter will discuss the steps we took toward developing our first prototype. The mechanical system will be discussed first, followed by the electrical system, which includes the control system and power requirements.

4.1 Mechanical System

The overall objective of this project was to develop an adaptive trans-femoral prosthetic leg socket that could automatically adjust its interior pressures on the residual limb based on the user's activity to help prevent limb or socket complications such as pressure sores. We knew early on that we could approach this challenge with a hydraulic system or a pneumatic system. Ultimately, we chose to build a pneumatic system since hydraulic systems are known to be very difficult and sometimes messy to work with. This section describes not only the components and setup of our pneumatic system, but it shows how we chose these components through force testing and research into each part's specifications and functionality. Additionally this section will describe the testing rig that will be used in the next stages of the project.

4.1.1 Force Plate Testing

The purpose of this experiment was to determine the maximum pressure experienced at the residual limb location during each state of motion; standing, sitting, and walking. The maximum pressure results from this test helped to determine what valves, sensors, and air supply were needed in our system. This experiment was tested on able-bodied subjects.

Materials:

- AMTI AccuSway 6-axis Force Plate
- AMTINetForce (Force Plate Software)

- Anatomical markers
- Wooden walkway
- Video camera

Setup:

Walk/Jog: The 6-axis force plate is placed on the floor with square wooden boards of the same height placed on both sides of the force plate on the x-plane.

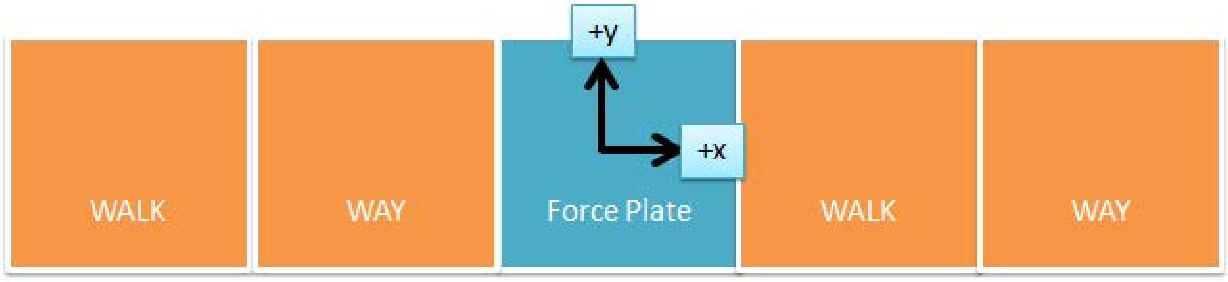


Figure 9 - Walk/Jog force plate setup

Sit/Stand: The 6-axis force plate is placed on the floor in front of a stationary chair. A wooden plate of the same height is placed next to the force plate.

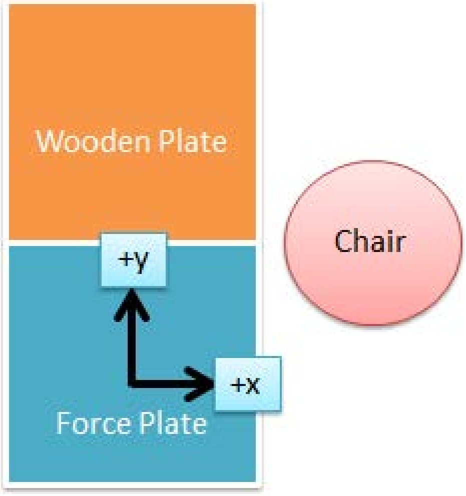


Figure 10 - Sit/Stand force plate setup

Procedure:

<u>AMTINetForce</u>

1. Zero the device force plate
2. Set sampling rate to 1000 Hz
3. Collect data for 5 seconds (Manually press start and stop for as close to 5 seconds as possible)
4. Save data files

<u>Camera</u>

1. Place a video camera at a perpendicular position to the center of the force plate.
2. Record each of the subject’s motions. These visuals will be important for further analysis and calculations.

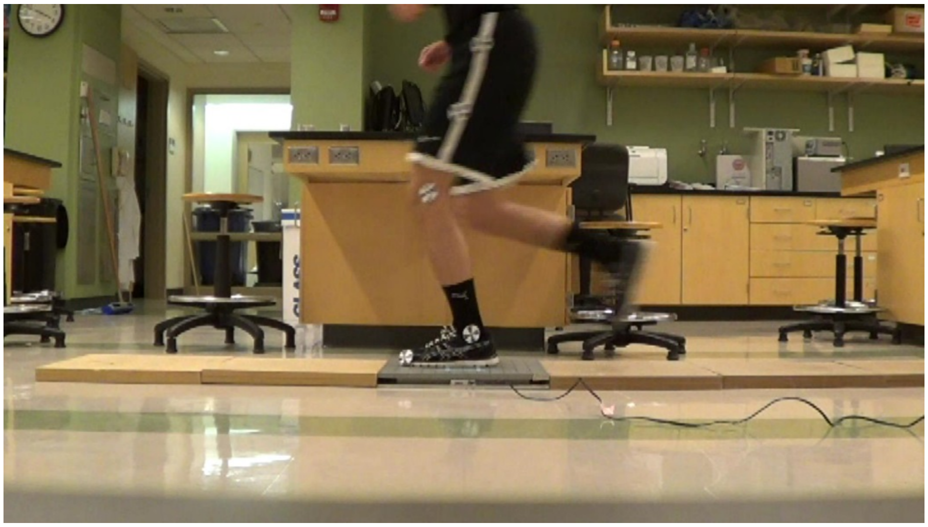


Figure 11 - Force Plate Test Subject Jogging

Walk/Jog

1. Subject starts by standing on one side of the wooden walkway
2. When 5 second timer starts, the subject walks onto and then past the force plate so that one foot strikes near the center of the plate. Data are then collected from that foot strike and saved to file.
3. Repeat step two by having the subject jog instead of walk.

*Note: Be sure that the subject starts far enough away from the force plate so that there is no unnatural alteration in their walking or jogging stride.

Sit/Stand

1. The subject starts in a standing position with the foot closest to the camera on the force plate and the other foot on the wooden plate.
2. When the 5 second timer starts, the subject sits down into a chair which is positioned next to the force plate. Data is then collected and saved to file.

3. Repeat step 2 except start in a sitting position and stand up.

Mathematical Theory:

The purpose of this experiment was to find the maximum possible pressure experienced by any one of the bladders during any of the specified states: sitting, standing, walking and jogging (for extra data). To find this value, only three data points needed are the maximum forces in the x, y, and z directions during the activity which produced the highest forces overall. It is intuitive and clear in the resulting data that jogging produced the highest forces overall.

The magnitude of the resultant force, $F_{R_{max}}$, of the three directional components can be calculated using the Pythagorean Theorem.

$$\sqrt{{F_{x_{max}}}^2 + {F_{y_{max}}}^2 + {F_{z_{max}}}^2} = F_{R_{max}}$$

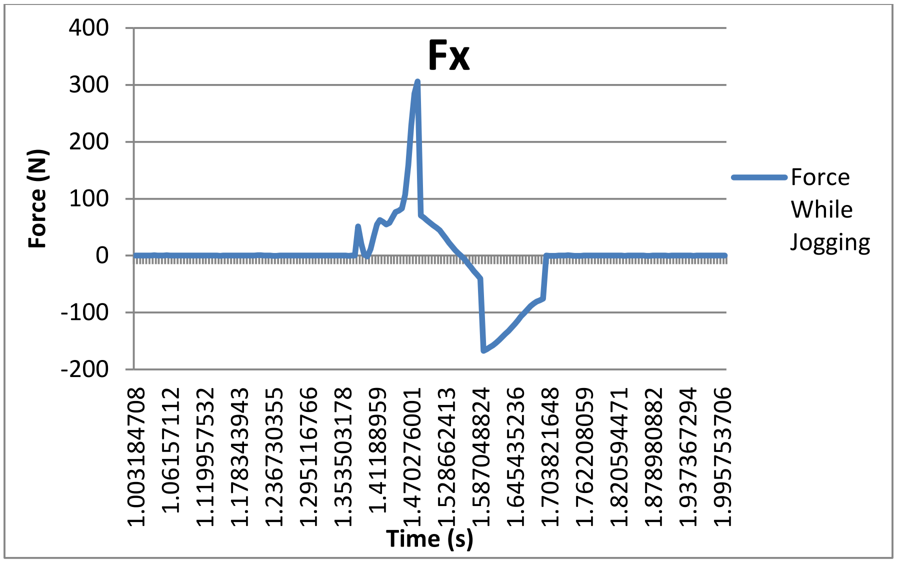
This $F_{R_{max}}$ value represents the highest possible force that can be exerted on one bladder at any time. The maximum pressure on one bladder can then be calculated using the relationship among pressure, force and surface area.

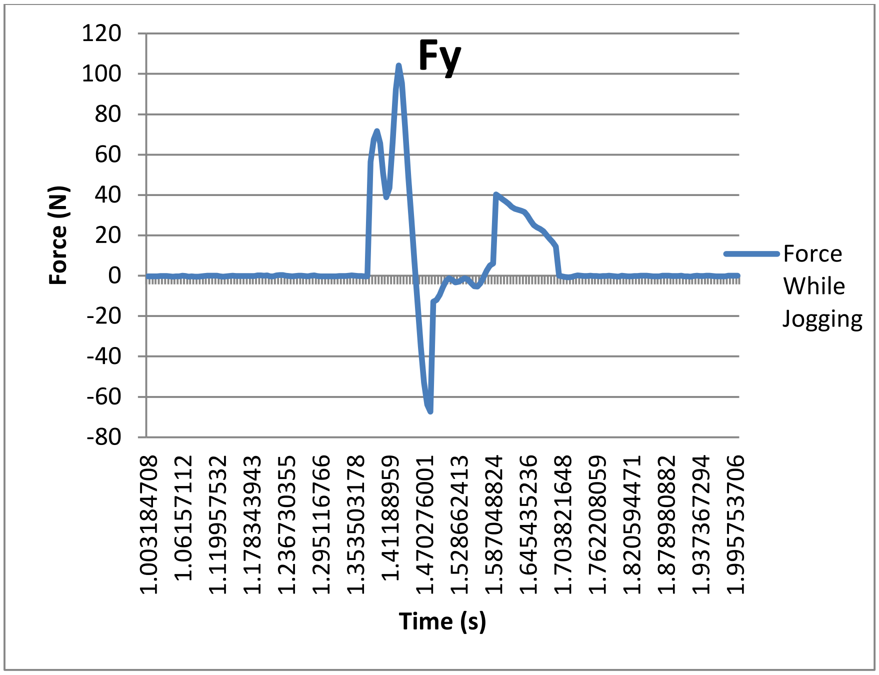
$$P_{max} = \frac{F_{R_{max}}}{A}$$

The area, A, represents the surface area of one of the bladders.

Results:

The following force-time graphs show the maximum forces in the x, y, and z directions while the subject is running. These values, along with a known bladder area of 0.0176m^2 were imputed into the above equations.


Fx
Force (N)
400
300
200
100
0
-100
-200
1.003184708
1.06157112
1.119957532
1.178343943
1.236730355
1.295116766
1.353503178
1.41188959
1.470276001
1.528662413
1.587048824
1.645435236
1.703821648
1.762208059
1.820594471
1.878980882
1.937367294
1.995753706
Time (s)
Force While Jogging


Fy
Force (N)
120
100
80
60
40
20
0
-20
-40
-60
-80
1.003184708
1.06157112
1.119957532
1.178343943
1.236730355
1.295116766
1.353503178
1.41188959
1.470276001
1.528662413
1.587048824
1.645435236
1.703821648
1.762208059
1.820594471
1.878980882
1.937367294
1.995753706
Time (s)
Force While Jogging

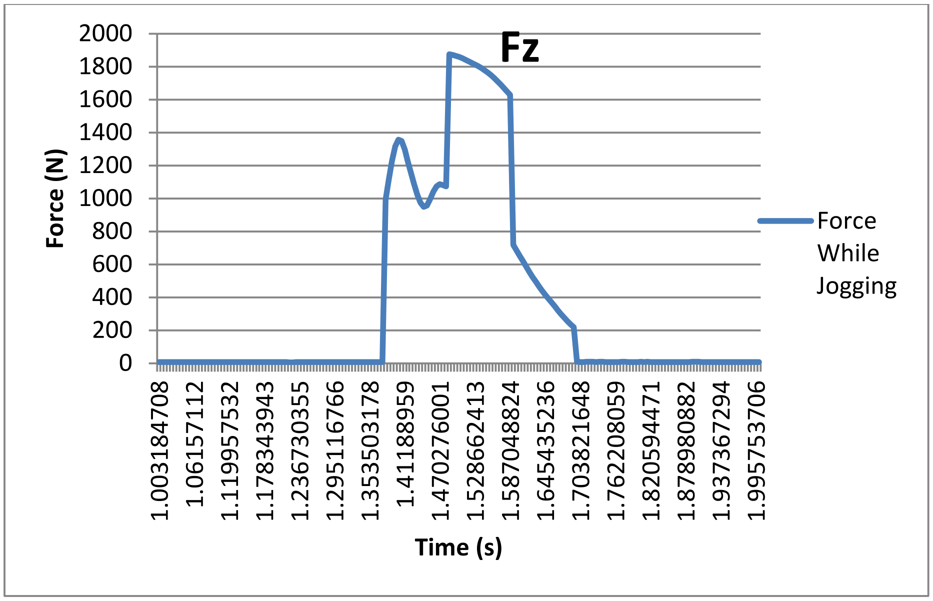


Figure 12 - Force-Time plot in x, y, and z directions for jogging

After completing these calculations, we found that the maximum pressure that could be felt by one of the bladders is 15.6 psi. This value was crucial in helping us to determine what components to include in our pneumatic system. It had to exist within the ranges specified by the valves, transducers, and bladders that we ordered. Since 15.6 psi is a maximum value, it is likely that the air-filled bladders will not usually need to compensate for such high forces during regular use, especially during the lower-impact states. However, this pressure calculation is crucial because it allows us to determine a worst-case-scenario and make educated decisions based on this information.

It is also important to note that the data collected are useful for various calculations other than those for maximum pressure. We also used these data to calculate the average ideal forces within each bladder for each specified state.

4.1.2 Estimating Pressure Values in Each State

The force plate experiment shown above also helped to determine what pressures are experienced at the residual limb-socket interface during the four states described. This data, along with the data collected in a 1979 study (Redhead, 1979), allowed us to make educated conclusions about what pressure values exist at this interface. Finding the normal pressure that the residual limb exerts on specific locations within the socket helps us to make reasonable estimations for how much counter pressure the bladders need to exert in order to alleviate pressure on the interior of the socket.

One way to measure these the pressures, or stresses, at specific locations inside the socket is to put pressure transducers at these locations. That is what was done in the 1979 study *Total surface bearing self-suspending above-knee sockets* (Redhead, 1979). An amputee was given a socket lined with pressure transducers at the locations shown in Figure 13.

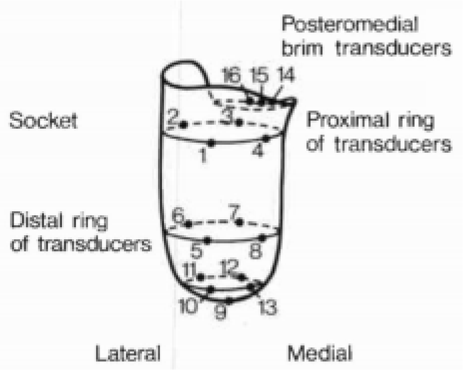

Posteromedial brim transducers
16 15 14
Socket
2
3
1
4
Proximal ring of transducers
Distal ring of transducers
6
7
5
8
11
12
10
9
13
Lateral
Medial

Figure 13 - Pressure Transducer Locations

The amputee then stood on an experimental rig which included an overhead hoist and harness with weights attached. The purpose of this setup was to subject the amputee to higher loads and study the effect on the internal socket stresses as these loads increase. The setup for this experiment is shown below.

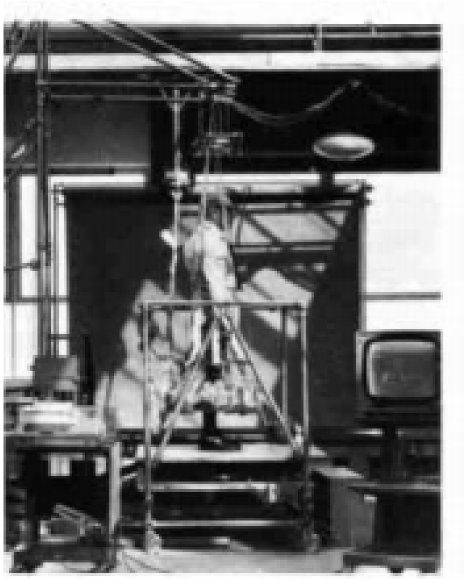


Figure 14 - 1979 Study: Experimental Setup

The results of this experiment are shown in Figure 14. The bottom x-axis shows the number assigned to each transducer. The transducers at the locations that we are focusing on for our project are 5 (lateral), 6 (posterior), 7 (medial), and 8 (anterior). These are close to the locations where we put the air filled bladder within our socket. The axis on the right-hand side shows the load applied on the amputee (kg) and the left-hand axis shows the pressure value at each transducer (MPa).

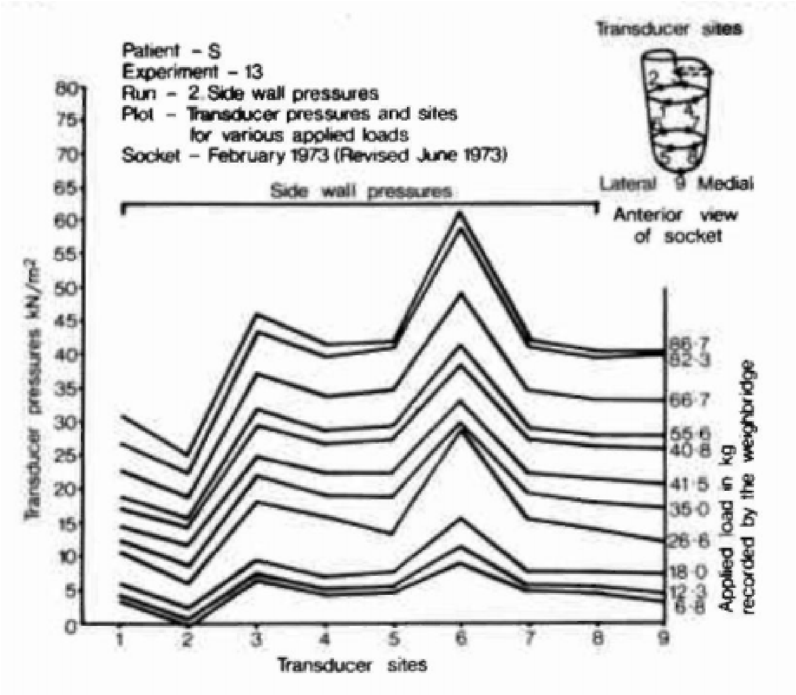
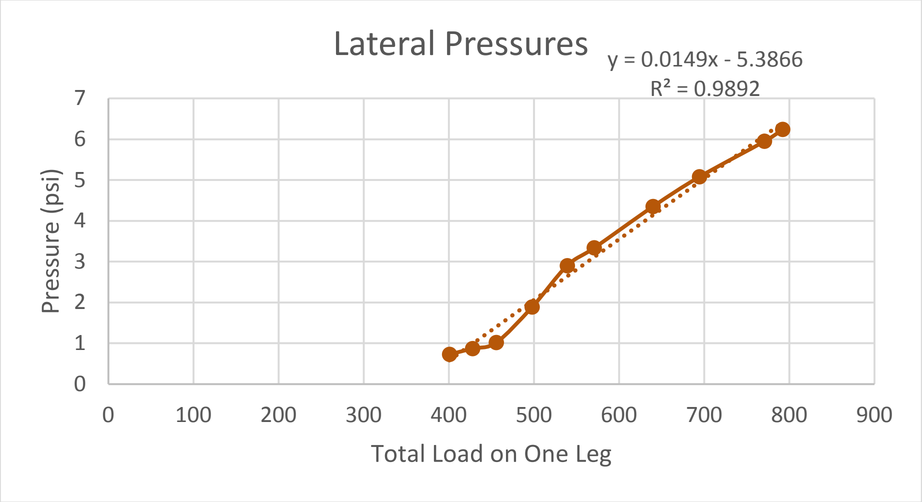


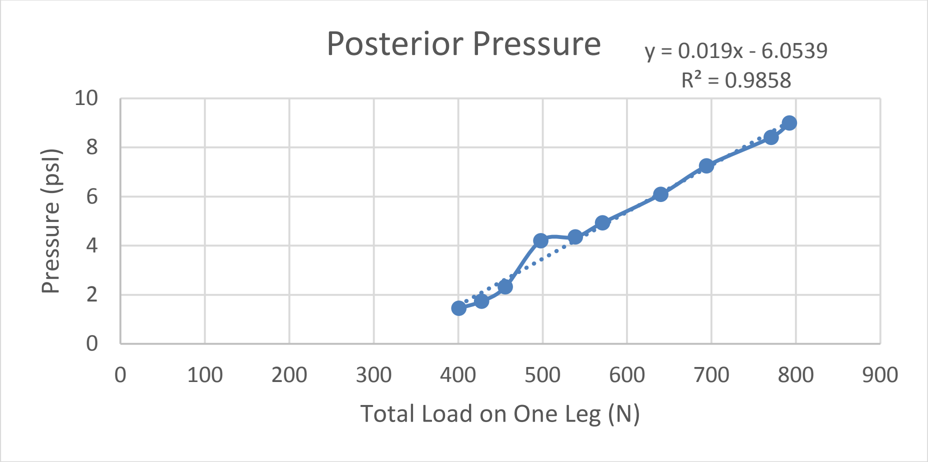
Figure 15 - Graph of Transducer Pressures at Each Socket Location

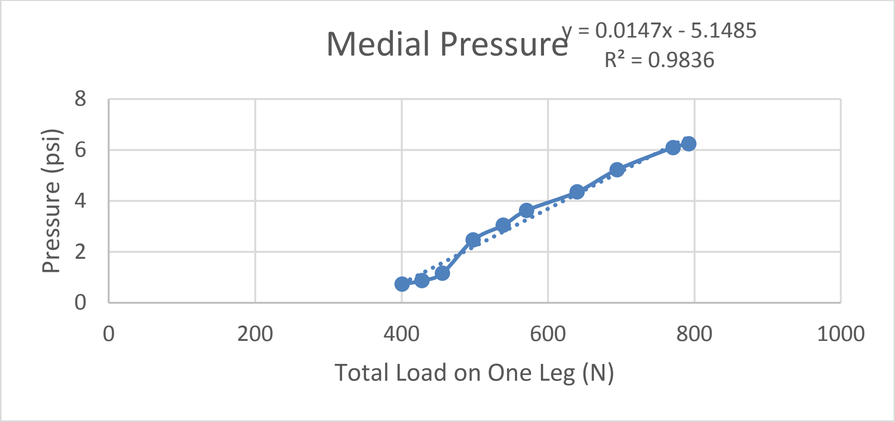
Because we didn't have pressure transducers or a trans-femoral amputee (let alone approval to run tests on an amputee), using the data from this study was our best method of finding ideal pressure values at each interior location for each state.

We reasoned that for each activity that we were studying, there is a downward load that is applied to the leg. If we know what these loads are, and we can find a predictable relationship between load and transducer pressures, then we can find the theoretical bladder pressure values for each of our subjects.

We plotted applied load on just one leg with the transducer pressure experienced at each of the four interior socket locations. The results are the following graphs.


Lateral Pressures
y = 0.0149x - 5.3866
R² = 0.9892
Pressure (psi)
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
900
Total Load on One Leg


Posterior Pressure
y = 0.019x - 6.0539
R² = 0.9858
Pressure (psi)
0
2
4
6
8
10
0
100
200
300
400
500
600
700
800
900
Total Load on One Leg (N)


Medial Pressure
y = 0.0147x - 5.1485
R² = 0.9836
Pressure (psi)
0
2
4
6
8
0
200
400
600
800
1000
Total Load on One Leg (N)

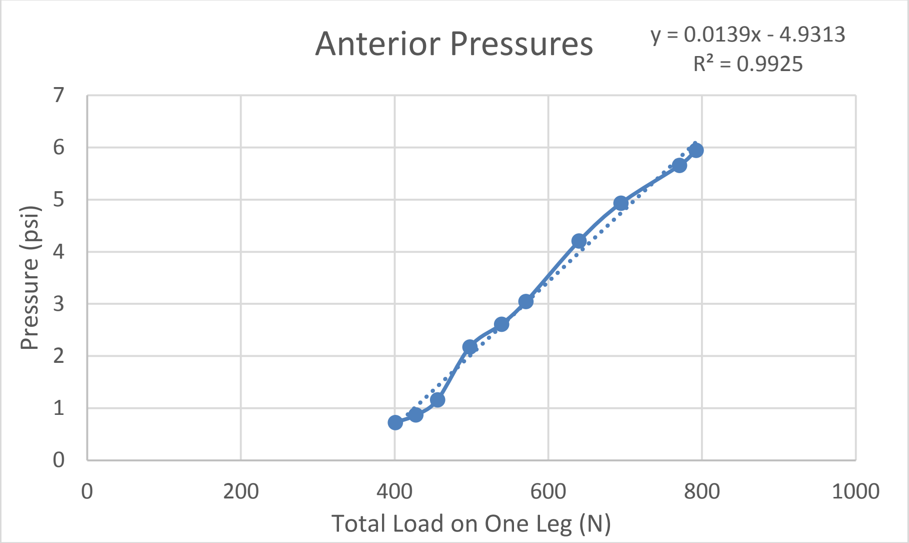
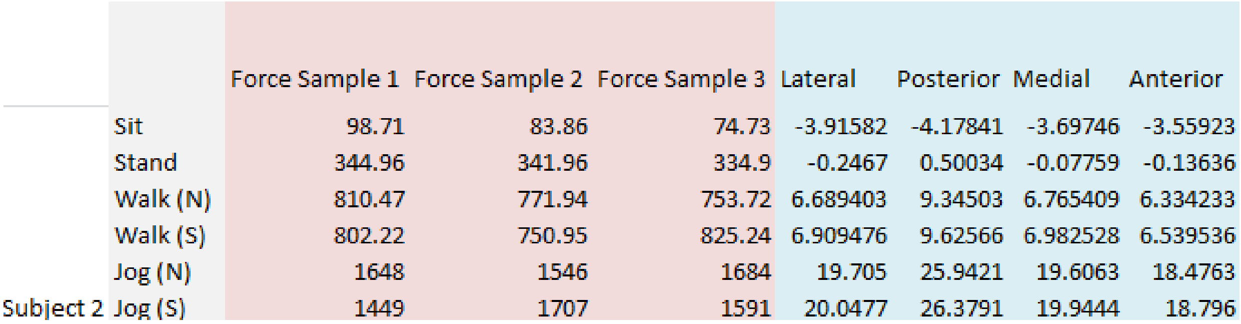


Figure 16 - Interior Socket Pressure Values Based on Research Study

With these relationships between applied load and experienced pressure at these locations, we could estimate the values of sitting, standing, walking and jogging from the data collected in our force plate experiment. We took the maximum downward force experienced in three samples from each of our five subjects and substituted them in for “x” in each of the linear equations in the graphs shown above. The results are shown in Table 2 which includes individualized pressure values at each interior location during each state. Walking and jogging were both performed with the subject moving normally and again with a straight leg, similar to the way an amputee would walk or jog without a knee joint. There was little significant difference between walking and jogging normally or with a straight leg. The pressures are all in psi.

Table 2 - Specific Pressure Values for Subject 2



		Force Sample 1	Force Sample 2	Force Sample 3	Lateral	Posterior	Medial	Anterior
	Sit	98.71	83.86	74.73	-3.91582	-4.17841	-3.69746	-3.55923
	Stand	344.96	341.96	334.9	-0.2467	0.50034	-0.07759	-0.13636
	Walk (N)	810.47	771.94	753.72	6.689403	9.34503	6.765409	6.334233
	Walk (S)	802.22	750.95	825.24	6.909476	9.62566	6.982528	6.539536
	Jog (N)	1648	1546	1684	19.705	25.9421	19.6063	18.4763
Subject 2	Jog (S)	1449	1707	1591	20.0477	26.3791	19.9444	18.796

Notice how the vales for sitting and standing are virtually zero. During sitting position, the leg experiences very small forces. The highest force is likely to be experienced at the posterior location since that is where the person's weight will most likely be applied. This position of center of pressure as well as the magnitude of the pressure is very dependent on the sitting posture of the subject. For the purpose of our experiment, we concluded from these results that only enough pressure to keep the socket on is necessary for sitting position, and posterior position may require a little extra pressure to support the person's weight.

Standing position is also virtually zero because a downward force would exert very little normal stress on the sides of the leg. Once again, we determined that only enough pressure needed to keep the socket on the leg is needed for when the subject is in standing position.

This study was most useful for determining the pressure values experienced at each location for walking. The medial and lateral values were very similar at around 6-7psi for all subjects. The posterior value was consistently the highest value, reaching the 8-10psi range for all subjects, as the anterior vale was typically similar to medial and lateral. These values were very useful for our determination of how much to inflate each of the bladders because they showed how much reaction pressure is needed to compensate for the interior pressures experienced during walking.

Jogging showed extremely high interior pressures due to high downward forces. All pressures were well above 15psi, which is the safe maximum limit of our air-filled bladders. Continuing testing on bladders at such high pressure would surely pop our bladders and therefore be very dangerous for the test subject. We therefore chose to limit the focus of our study to sitting, standing, and walking positions for the sake of safety.

The team used the study's findings, their force plate data, and the analysis described above, to assign ideal pressure value for each bladder for each state. The result is shown in Table 3 below which shows these pressures in psi.

Table 3 - Assigned Pressure Values for Each State

	Posterior	**Anterior**	**Medial**	**Lateral**
Sitting	4.0psi	2.0psi	2.0psi	2.0psi
Standing	2.0psi	2.0psi	2.0psi	2.0psi
Walking	8.0psi*	8.0psi*	6.7psi*	6.7psi*

The walking values vary depending on the person's weight and gait. The data for each subject can be found in Appendix E, and additional analysis and calculations can be found in spreadsheets which were submitted with this report.

4.1.3 Socket

The prosthetic socket is an integral part to our project because it is the interface between the user's residual limb and the prosthetic itself. For this project, we received a trans-femoral socket from Ottobock that was used for demonstration purposes, as shown in Figure 17. From previous anthropometric data, it was found that the average mid-thigh circumference for males 30-50 years of age is 55 cm (US Department of Health and Human Services). This means that the mean diameter of these thighs is 15 cm, or 6.89 inches. To have enough clearance on either side of the thigh for the bladders, wires, and other parts, the socket should be about 9-10 inches in diameter. The Ottobock socket is approximately this distance, but it tapers downwards. Unfortunately, we were unable to secure any amputees to test the socket in its full functionality. Therefore, our team constructed a testing rig in which to test all of the necessary components and their interactions with one another. This rig is discussed further in Section 4.1.9, but the rig attaches on the thigh like the prosthetic socket would and is open-ended so it can be worn and tested by able-bodied subjects.

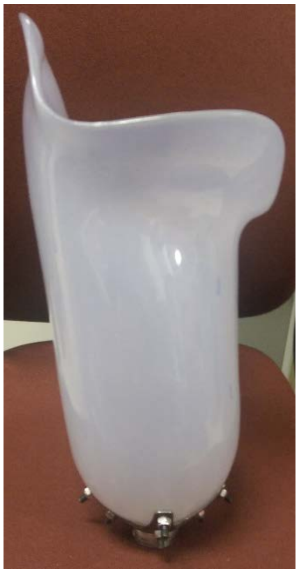


Figure 17 - Ottobock Socket

4.1.4 Valves

Pneumatic valves are important to our overall design because they are involved in the pressure regulation control of each bladder. During each state of motion, the control unit tells the valves to either open or close depending on whether the pressure within each bladder needs to be increased or relieved. This was the general function that we needed the valves to fulfill. Two possible valve configurations were considered.

The first configuration involves a single, three-way valve in which air flow would be controlled in three directions. This valve would have three possible settings; closed, open, and release. When the valve is closed (no air flow), the desired pressure in the bladder has been reached and no further air is needed. When the valve is in the open state, air flows into the bladder from the reservoir. During this state, the mechanism inside the valve opens a path from the inflow tube to the outflow tube, which is connected to the bladder. When the bladder pressure needs to decrease, the valve will be set to release mode, closing off the inflow and opening the third port, leading out into the atmosphere. A diagram of the system is shown below.

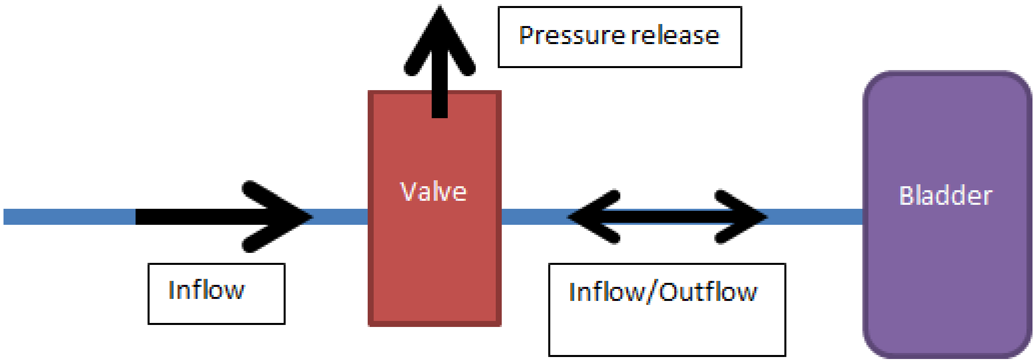


Figure 18 - Pneumatic System Option 1

The second valve configuration that we considered is a series of two 2-way valves which serve the same function as the single-valve system previously shown. These are simple On/Off valves which each can exist in two possible states: open and closed. Two independently controlled valves allow us to create several desired results based on the states of each valve.

Possible states of valves with results

1. Both valves closed:

No air is let into the bladders.

2. Valve 1 open/Valve 2 closed:

   Air flow into the bladder but no pressure release.

3. Valve 1 closed/Valve 2 open:

   No inflow, and pressure is released from the bladder into the atmosphere

4. Both valves open:

   Simultaneous inflow and pressure release. (Obviously this state would not be that useful).

This 2-way valve system is shown in the diagram in Figure 19.

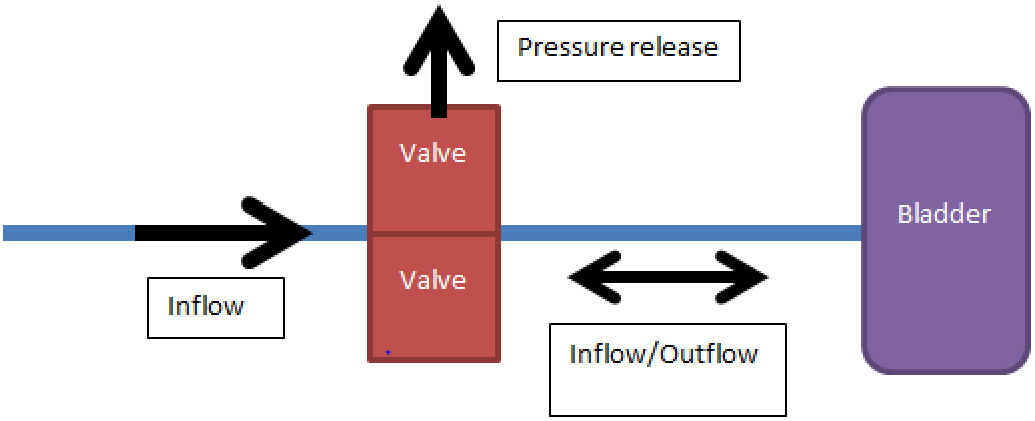


Figure 19 - Pneumatic System Option 2

There are several advantages and disadvantages to each system, all of which were considered as we were deciding on a final design. Since option 1 only requires one valve, it would therefore be easier and require less power to control. One valve per bladder means that there would be four valves total in the system, as opposed to the eight required for Option 2. However, the most significant drawback to purchasing electronic three-way valves is the price. Electronic three-way valves that could fulfill our desired operations were consistently more expensive than simple 2-way On/Off valves which would serve the same function if placed in series. The 2-way valves we purchased were ten dollars each, which came to a total of 80 dollars. Three way valves of the same quality are about 40 dollars each, which would have made a total of about 160 dollars for four valves. For this reason, we chose to purchase eight two-way valves (two per bladder) in order to keep the cost lower.

Figure 16 shows the valves that we selected. The assembly displayed in the picture shows two valves connected by a manifold which allows for one to serve as a pressurizing valve and the other to serve as a release valve.

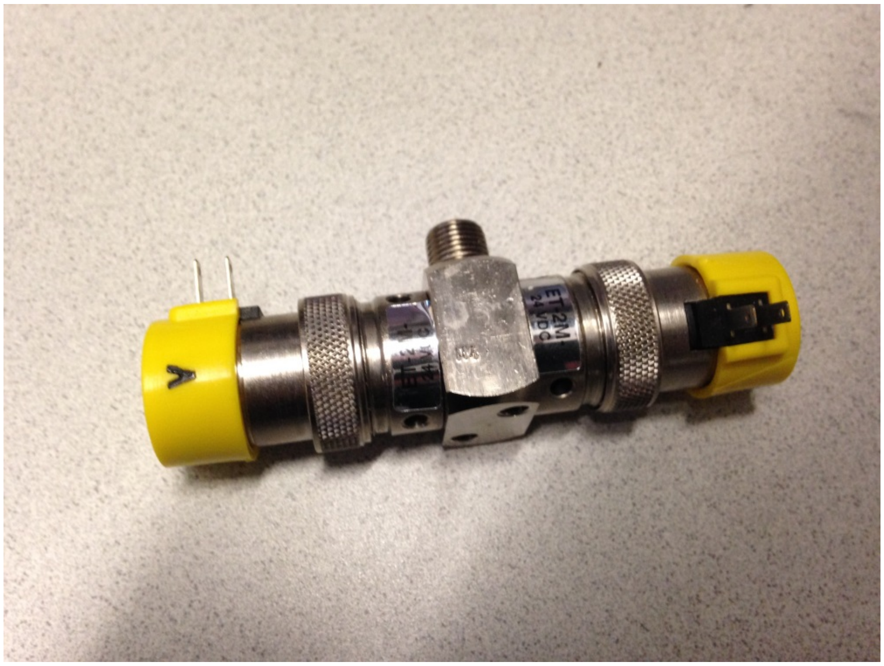


Figure 20 - Clippard 2-way Manifold Mount Solenoid Valve Model ET-2M-24

Section 4.3.4 describes the specifications for these valves as well as their integration into the electrical system.

4.1.5 Bladders

The bladders are one of the most important parts of the socket system. They are used to help adapt to the various movement-related tasks of daily living including standing and walking. The bladders inflate and deflate depending on the activity that is being performed and the pressure that is being placed on the bladder. When brainstorming ideas for this adaptability system, there were several different variables considered. These variables include what the actual adaptive system would be, what each bladder would be filled with, how each bladder would be actuated, and how many bladders would be used in this system. Several design matrices were produced in which different objectives such as comfort and durability were weighted. Then each of the variables were scored, multiplied by the weight, and the variable with the highest score is the one that was integrated in our bladder design.

Table 4 - Design Matrix for Bladders

		Straps (For Tightening)		Suctioning		Bladder	
	Weight (%)	Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Comfort	15	2	30	2.5	37.5	4	60
Cost	10	4	40	2	20	4	40
Ease of Use	15	2	30	3	45	4	60
Ease of Implementation	25	3	75	2.5	62.5	3	75
Durability	5	3.5	17.5	2	10	3.5	17.5
Pressure Limits	5	3	15	2	10	3.5	17.5
Controllability	25	1.5	37.5	2.5	62.5	3.5	87.5
TOTAL	100		245		247.5		**357.5**

Tables 4 through 8 show this design matrix process. The metrics for these matrices are scored from 1-5; 1 meaning that the design did not exemplified/fit the requirement at all/rarely, and 5 meaning that the design exemplified/fit the requirement the most/always.

Table 5 - Design Matrix for Method of Socket Adaptability

		Straps (For Tightening)		Suctioning		Bladder	
	Weight (%)	Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Comfort	15	2	30	2.5	37.5	4	60
Cost	10	4	40	2	20	4	40
Ease of Use	15	2	30	3	45	4	60
Ease of Implementation	25	3	75	2.5	62.5	3	75
Durability	5	3.5	17.5	2	10	3.5	17.5
Pressure Limits	5	3	15	2	10	3.5	17.5
Controllability	25	1.5	37.5	2.5	62.5	3.5	87.5
TOTAL	100		245		247.5		**357.5**

Table 6 - Design Matrix for Filling the Bladder

		Air-Filled Bladder		Fluid-Filled Bladder		Combination of Air and Fluid-Filled	
	Weight (%)	Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Comfort	15	3	45	4	60	3.5	52.5
Cost	5	4	20	4	20	4	20
Durability	10	2.5	25	3.5	25	3	20
Pressure Limits	30	3.5	**105**	2.5	75	3	90
Controllability	40	4	160	3	120	2	80
TOTAL	100		**355**		300		262.5

Table 7 - Design Matrix for Actuating the Bladder

		Powered by Pneumatics		Powered by Hydraulics		Fixed Bladders (No Power)	
	Weight (%)	Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Comfort	10	3	30	4	40	3	30
Cost Effective	15	3	45	2	30	4	60
Ease of Implementation	15	2.5	37.5	2.5	37.5	4	60
Pressure Limits	25	3.5	87.5	2.5	62.5	1.5	37.5
Controllability	35	3.5	122.5	2	70	1	35
TOTAL	100		**322.5**		240		222.5

Table 8 - Design Matrix for Number of Bladders Used

		One Bladder filling Entire Socket		Multiple Bladders	
	Weight (%)	Score	Weighted Score	Score	Weighted Score
Comfort	10	3.5	35	3	30
Cost	5	4	20	4	20
Ease of Use	15	2	30	3	45
Durability	10	3	30	3.5	35
Pressure Limits	30	2.5	75	3.5	105
Controllability	30	3	90	2	60
TOTAL	100		280		**295**

In this socket system, multiple bladders will be placed on four different parts of the thigh's circumference, equidistant apart. These locations were selected because they are common areas in anatomical studies (Redhead 1979). The configuration also allows for counteracting pressures which help to keep the socket secure on the leg. These bladders would be powered by pneumatics, filling with air as it adapts to the environment. Bladders were ultimately chosen because of their flexibility in material, so it will adapt to the process of inflation and deflation, it can help fit a variety of body sizes, and it is comfortable for the users.

Once we decided on the method of the adaptability system, we performed initial testing on the amount of pressure that these bladders would exert on the human leg. The methods and results of this experiment are explained below. One example of the bladder and pneumatic system set up is shown in Figure 21. These bladders are suitable for use in the testing rig, but were found to be too large for the socket that was donated by Ottobock. Instead, Air Jack Twin Bladders (Part # AETTAW) will be used in the final prototyping, which is further discussed in Chapter 5.

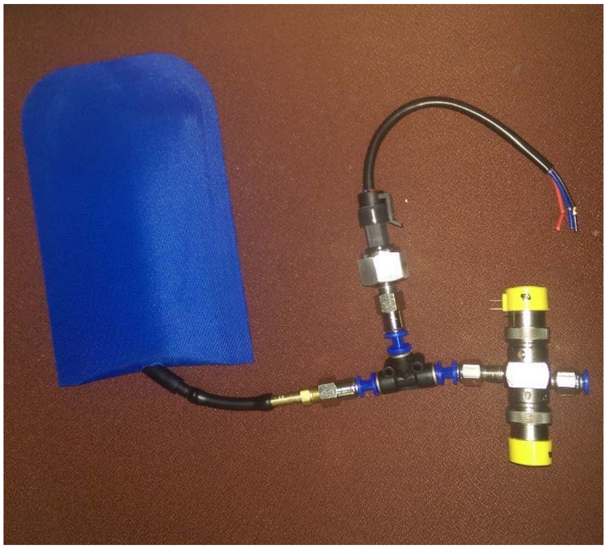


Figure 21 - Bladder and Pneumatic System Set-Up

This testing was used to determine the maximum pressure the bladders can exert on a human thigh without causing pain or restricting blood flow. These pressures are needed to determine the types of bladders that the socket should contain. This experiment involved placing blood pressure cuffs around a person’s thigh and increasing pressure until the user experiences a light tingling sensation indicating a momentary loss of blood flow. The purpose of this was to determine a maximum pressure before the subject experiences any form of discomfort. It is important to note that all tests were performed only on team members, therefore no IRB was required. Also, no subjects underwent any discomfort more intense than the initial minor discomfort from a traditional blood pressure cuff's constriction within the typical clinically used operating range. The pressure that is read on the blood pressure cuff is the pressure the cuff is exerting on the limb. This test will help determine the pressure the bladder should exert on the thigh and still be comfortable for the user.

Materials

- Blood pressure cuff
- Test subjects

Methodology

While the subject is seated, place the blood pressure cuff around the thigh 1-2 inches above the knee. Slowly increase the pressure of the cuff in increments of 10 mmHg and wait 3-4 seconds for a response from the subject. Use zero as a starting pressure. Repeat this process until the subject feels a minor tingling sensation in the constricted area. Record the pressure from the pressure sensors. Repeat this entire process two more times for precision. Repeat the entire process for three total subjects, each having three trials of testing.

Results:

Table 9 - Initial Blood Pressure Testing

Subject #	**Maximum Pressure- Test 1 (mmHg)**	**Maximum Pressure- Test 2 (mmHg)**	**Maximum Pressure- Test 3 (mmHg)**	**Average Maximum Pressure (mmHg)**
1	130	100	80	110
2	120	105	110	111.67
3	110	120	115	115

Table 9 shows the results from the test. For each of the three users, the maximum pressure is shown. This pressure is the maximum pressure the user allowed before they claimed they were uncomfortable with the pressure. The results from these tests indicate that an average of 112.2 mmHg of pressure was placed onto the leg. This converts to 14.96 kPa or 2.17 psi.

4.1.6 Bladder Holder

A significant design challenge that we faced was how we were going to attach the bladders to the interior of the socket. Various adhesives were considered, but we were wary of using them since when the bladder inflates it will bulge and lose some contact with the interior surface of the socket. Additionally, if the bladders ever need to be replaced, it would be tedious and inefficient to reattach the bladder with some kind of glue.

For these reasons, we decided to design a bladder holder that the bladders could slide into, as shown in Figure 22. The initial prototype bladder holder that we designed was made out of a white spandex material which is fairly elastic to coincide with the bladders' expansion. The holder is formed to the interior of the socket, and bladder-sized pockets are sewn onto the spandex material at the appropriate locations. The holder is held secured on the socket with bolts that go through the fabric and socket. The fabric is protected from the forces at its contact with the bolt with eyelets on both sides of the spandex.

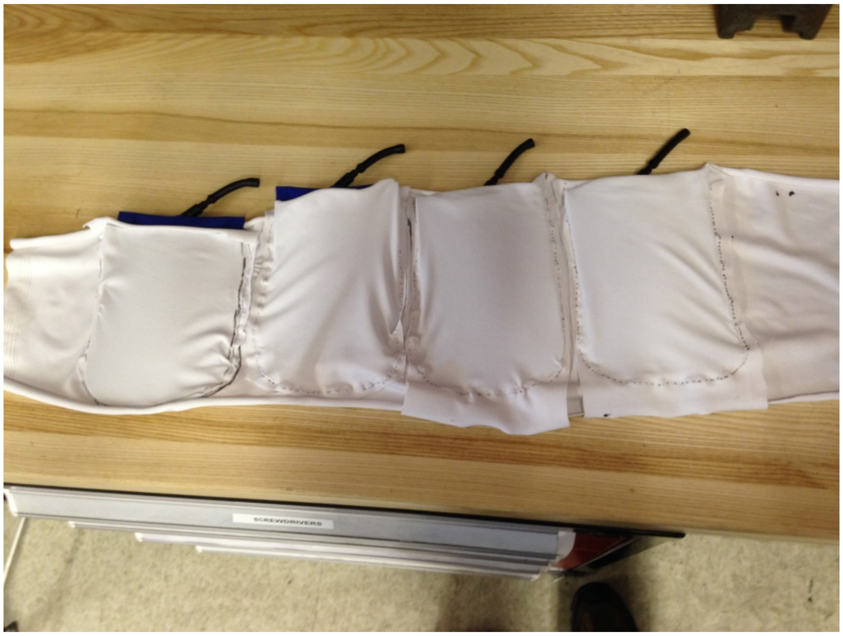


Figure 22 - Testing Rig Bladder Holders

This bladder holder is roughly 32.5inches long and 9.5 inches wide. Each bladder has roughly the dimensions of 8in x 5.25in and is separated by 0.4715 inches. Extra space was provided on either side of the holder to allow for the possibility of sewing the ends together later on.

A similar bladder-holder design was used for our testing rig, as described in Section 4.2.9. The difference is that there is a hole in the bottom of the spandex to allow for the able-bodied person to slide her or his leg through.

4.1.7 Air Compressors/Supplies and Air Tanks

One of the most challenging problems of this design is the issue of air supply. We contemplated several designs and methods for supplying air to our pneumatic system. The first design includes a small but compressed air tank that is attached to a pressure regulator, which then feeds into the rest of the pneumatic system. Since the pneumatic system involves pressure release, this reserve air tank will eventually need to be refilled. When the tank is low on pressure, the user will take out a portable miniature air compressor and manually refill the tank. The portable air compressor is included in the design despite not being attached to the rest of the pneumatic system.

This design would be useful for someone who is not very active on a daily basis. However, our objective is to design an adaptive trans-femoral socket for someone who frequently sits, stands, walks, and occasionally jogs. From the user's perspective, it may be very tedious to constantly need to refill their tank. Ideally, the user would not have to think about their device at all and simply trust that it will work properly.

For this reason it seems more appealing to have a miniature portable air compressor built into the socket. This compressor will then be able to fill in the tank on its own when the pressure within the tank runs low. It will do this through communication with the control unit. Figure 23 shows a picture of an appealing miniature air pump.

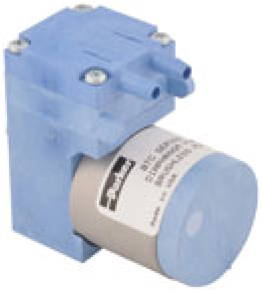


Figure 23 - Hargraves BTC Series Miniature Air Pump

This air compressor provides high performance with compact size. Its dimensions are 2.13" x 1.18" x 2.15" and it weighs only 4.5 oz. The maximum flow is 3.5 liters per minute and the pressure range is 0-30 psi, both of which are within our design parameters. Additionally, the compressor contains an electrical component which operates under 6, 12, and 24 V and has a current range of 50-900mA. The largest issue with this air compressor is the price. It costs around $200 which would push the limits of our budget. Since the primary goal of this prototype was to prove the concept, it was decided that an external compressor to pressurize the air reserve was a reasonable alternative.

The team had several possible options for what to use as the air reserve. The first option is a miniature and lightweight paintball gun compressed air tank. This tank would be advantageous because it only weighs about 3-4lb. However it is too bulky and mounting it to the socket without it being obtrusive would be very challenging.

Another option is to build a reserve “tank” out of coiled tubing that is wrapped around the outside of the socket. A tank does not necessarily need to be a typical metal canister of air. It only needs to be able to hold a desired volume and withstand a considerable amount of pressure. For this reason, a copious amount of tubing would be sufficient. The tubing would be visible on the outside of the socket, but a plastic covering over the final model would be able to hide it.

Choosing which air compressor and tank to use was less of a priority for the purposes of testing the pneumatic system and building our first prototype than the other hardware incorporated in our pneumatic system. In our testing laboratory, we have access to a continuous air supply with pressure regulators included. This setup allowed us to test our pneumatic system without yet having to commit to buying a specific air compressor. It was important for us to determine that the system works properly and that the valves, sensors, and bladders respond to pressures the way we want them to before purchasing a fairly expensive miniature portable air compressor.

4.1.8 Pneumatic System Overview

Show in Figure 24 is a diagram of the entire pneumatic system followed by a summary of the system. Part numbers of each component, including the pneumatic components below, are shown in Appendix G.

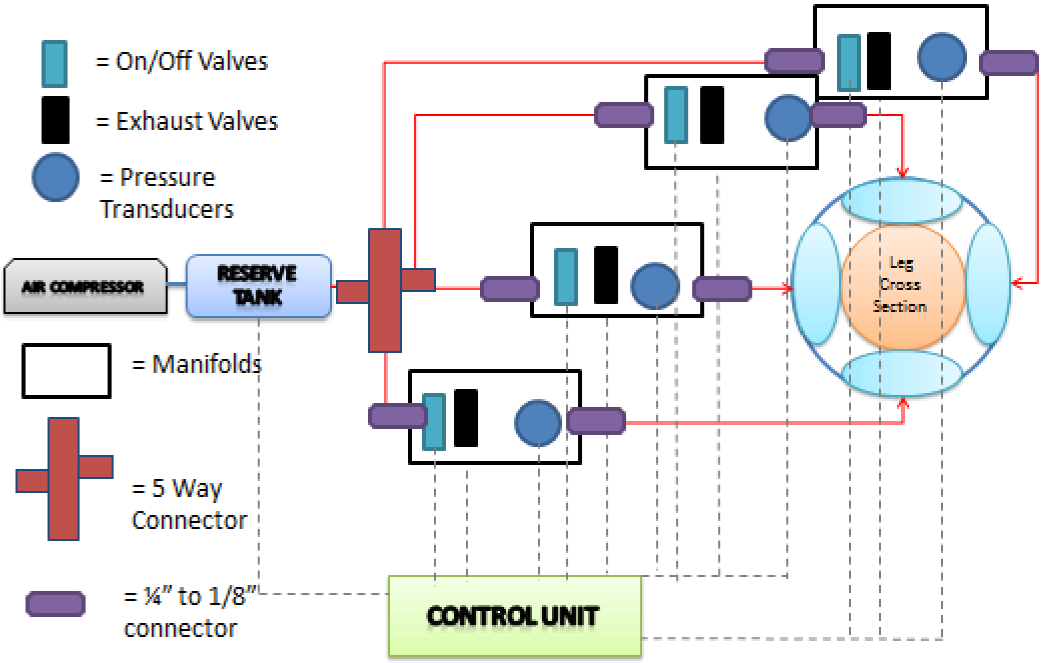


Figure 24 - Pneumatic System Diagram

Air is initially fed into the system by an air compressor, which may either be portable or built into the socket depending on further experimentation and calculations. The compressor fills a reserve tank with air when the tank runs low (see Section 5.1.3 Air Reservoir for more tank information). This tank is also equipped with a pressure regulator to allow for control over pressure release. A signal from the control unit allows air to flow from the tank to the system when air is needed. Air is released and is then split four ways by two 3-way push connect fittings. This sends streams of air into the directions of each of the four bladders.

Once each individual stream of air reaches each manifold, the first On/Off valve is opened to let air into a bladder, which may be located on the front, back, or sides of the residual limb. A pressure transducer is connected to the line leading into the bladder by a 3-way push connect fitting. This transducer measures the pressure within this line (and therefore the pressure within the bladder) and communicates that pressure back to the control unit. The control unit then makes a decision based on pre-programmed logic whether to feed the bladders more pressure or release pressure. If the pressure gets too high inside the bladders, the control unit will compensate by opening the second valve mounted on the manifold, which lets the air bleed out into the atmosphere. The first valve is closed during this action. Exhaust (or blow-off) valves automatically limit the maximum pressure in each bladder.

This process will take place constantly so that the bladders are always at the appropriate pressure throughout the user's predetermined states.

4.1.9 Testing Rig

To test the general function of our system, we built a testing rig which an able-bodied person could wear to simulate the experience inside of the socket. There are several objectives to testing on this rig:

1. Show that the bladders will never become too loose and cause the rig (socket) to fall off.
2. Place pressure sensors under the bladders to measure actual, rather than theoretical, pressures at these specific locations.
3. Show the general locations of all included components.
4. Show that the components are secure, even during motion.

The rig consists of a metal tube which is meant to simulate the side walls of the socket. There is an opening in the side of the tube to allow for the user to wrap the rig around his or her quadriceps. On both sides of the tube are slits for backpack straps to loop through to close the ring and provide a desired, customized fit. The metal tube is sharp at the top and bottom, so those edges are covered with an insulating foam material to prevent possible injury to the test subject. The bladders are held in place by the bladder holder described in the previous “Bladder Holder” section. The only difference between this bladder holder and the one that will be used in the actual socket is that the test rig bladder holder has a hole through the bottom to allow the able-bodied person’s leg to go through.

Between each bladder transversely about the socket are carefully shaped solid pieces of foam which help to hold the leg in place and provide cushion for the users. Even when the bladders are in a relaxed state, it is still important to sustain the structural integrity of the interior of the socket.

All electrical components are mounted to the outside front of the testing rig, as the four pneumatic assemblies are mounted to the remaining exterior spaces around the ring. The pneumatic components are screwed into the metal and zip-tied for extra security, as shown in Figure 25.

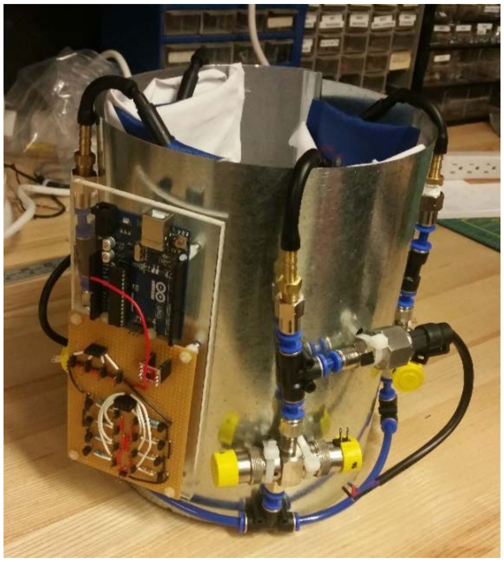


Figure 25 - Testing Rig

4.1.10 Mechanical System Summary

The mechanical system is composed of two major sub-assemblies, the structural sub-assembly and the pneumatic sub-assembly. The structural sub-assembly refers to the components which contribute to the structural makeup of our design, including the socket, bladder holders, and the hardware and attachments that keep the rest of the system on the socket. The pneumatic sub-assembly includes the valves, sensors, bladders, and air supply. This assembly, however, is not only a part of our mechanical system, but our electrical system as well. The next section will discuss how we relate the mechanical inputs and output of our pneumatic system to the electrical ones.

4.2 Electrical and Control System

The electrical system is designed to gather data from the accelerometer, fluid pressure, and contact pressure sensors. It is also responsible for powering each of the sensors as well as the microcontroller and pumps. The microcontroller will receive all the sensor data, perform calculations to determine the user's current activity, and adjust the pressure in the bladders to ensure the amputee remains comfortable. This section will describe the sensors used in this design as well as the circuits, power system, and control system.

4.2.1 Accelerometer

One of the main components of our design is the accelerometer. This gives us data that can help to determine the user's current state (sitting, standing, and walking). The accelerometer we used for testing is the Analog Devices ADXL362 SparkFun Triple Axis Accelerometer Breakout board.

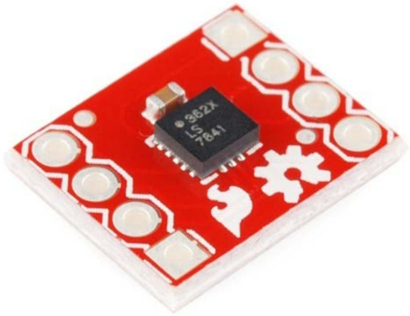


Figure 26 - Accelerometer Breakout Board ADXL362

This breakout board measures acceleration in all three Cartesian planes and serializes the data to be sent to a separate microprocessor. To test this accelerometer, we connected it to the Arduino Uno as shown below. The Arduino Uno was chosen as the microprocessor for the initial prototype (see Section 4.2.4 Microcontroller for more information). The team wanted to determine if we could accurately evaluate an individual's current state using the accelerometer, so we strapped the circuit onto a person's upper leg to mimic the positioning of the device on our final product.

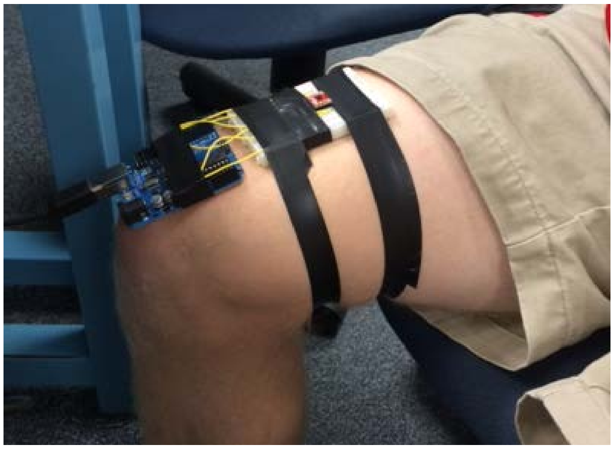


Figure 27 - Accelerometer Testing Setup

We then collected data on the individual at 10Hz while they sat, stood, walked, and jogged for about 10 seconds each. The resulting data are shown in Figure 28.

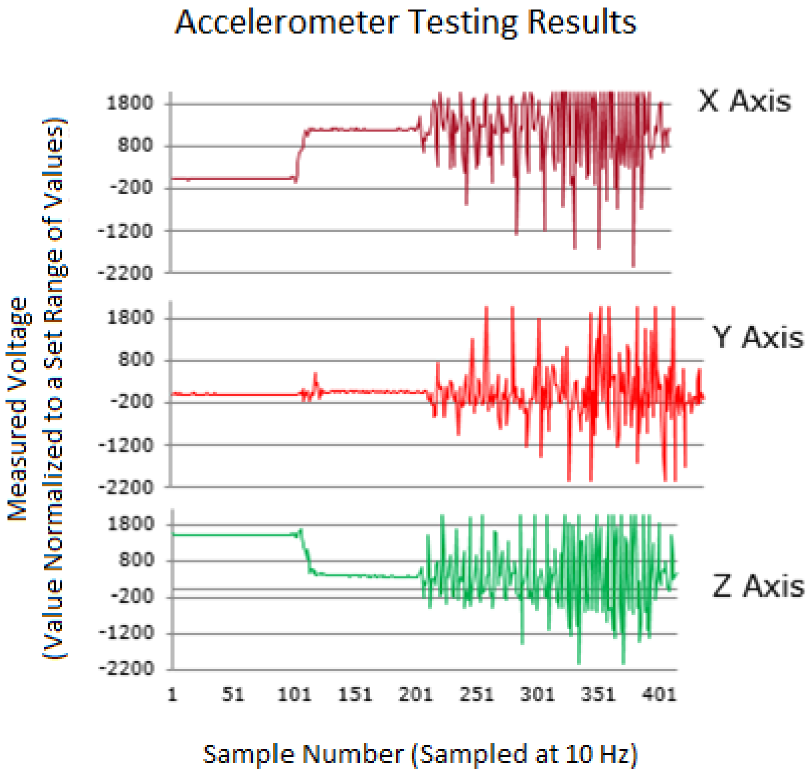


Figure 28 - Accelerometer Testing Results

Following this data collection, we developed a method to determine the individuals current leg position based on the X-axis values, which was the axis parallel to the hip joint's axis. We created a MATLAB code, which is in the appendix of this report, which converts the x-axis value to an angle relative to the torso. The formula used can be found in Section 4.2.7. The accelerometer and formula had a very high success rate and we decided that this method would be suitable for determining the state of an individual.

4.2.2 Pressure Sensors

As part of the control system, the microprocessor must receive an input signal proportional to the pressure inside each bladder. This is a need to determine whether the bladder should be inflated or deflated based on the current state. To receive this signal, we decided to purchase four pressure transducers. The transducer we used is a 0-15 psi stainless steel pressure transducer that is connected in line with the bladder.

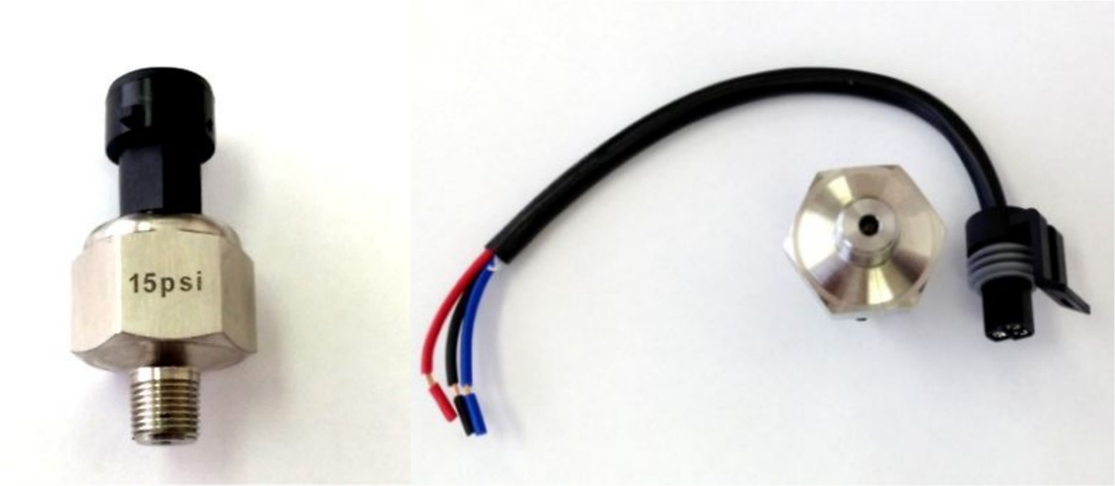


Figure 29 - Pressure Transducer

This transducer has a 1/8"-27 NPT thread to connect to the air line on the bladders. It also has three wires attached to the opposite side. The three wires are used for ground, a 5V input to power the transducer, and a variable voltage output that connects to the microprocessor. The variable output is 0.5V to 4.5V that correlates linearly with the pressure in the bladder. An output of 0.5V corresponds to 0 psi, while 4.5V corresponds to 15 psi as shown in the graph below. It has an accuracy of ±2 percent of the full scale reading.

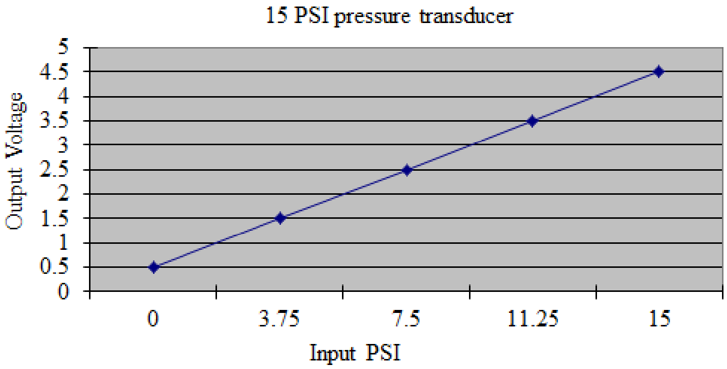


Figure 30 - Pressure Transducer Correlation

To make sure that these transducers were correctly calibrated and that they would work as we needed them to for our design, we tested them at varying pressures. One of the transducers was connected to an air line between an air compressor and a bladder. We also connected the wire leads correctly to measure the output values using an Arduino Uno. Then, we adjusted a pressure regulator to allow varying levels of pressure into the bladder, creating the results shown in Table 10. We plotted the values in the as shown in Figure 31 and fit a calibration curve to the values we obtained. The curve is given by the following equation: Voltage = 53.391*Pressure + 117.73 with an R^2 value of 0.9945.

Table 10 - Pressure Transducer Calibration Test Results

Pressure (PSI)	Digital Value	Analog Voltage (V)	Expected Voltage* (V)
0	105	0.515	0.5
4	316	1.548	1.5
5	373	1.828	
6	455	2.230	
7	524	2.568	2.5
8	540	2.646	
9	628	3.077	
10	640	3.136	
11	712	3.489	3.5
12	755	3.700	
13	802	3.930	
14	844	4.136	
15	923	4.523	4.5

*Source only gives 5 expected analog output values for 5 different pressures

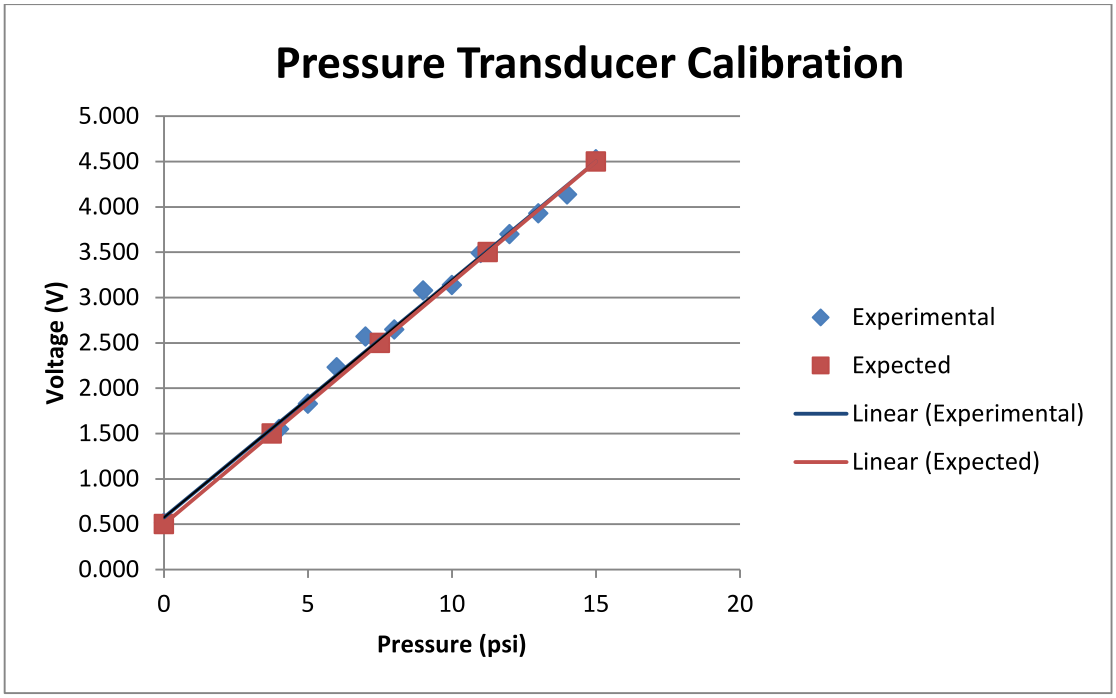


Figure 31 – Graph of Pressure Transducer Calibration Results

4.2.3 Valves

In our design, we included valves to control the air flow in and out of the bladders. The valves are used to maintain the desired pressure in each bladder. For each of the four bladders, there are two valves. One is to allow air to enter the bladder from a compressed air tank while the other is used as an exhaust to allow air to exit the bladder. We decided to use eight valves from Clippard Minimatics. These valves are the ET-2M-24V manifold mount solenoid valves. They are normally closed two way valves and are only open when 24 volts are applied across the two terminal leads. Two valves connect to the bladder using a manifold mount as shown in Figure 32.

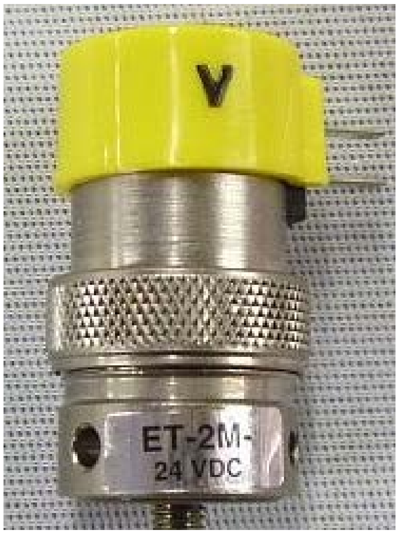


Figure 32 - ET-2M-24V Valve

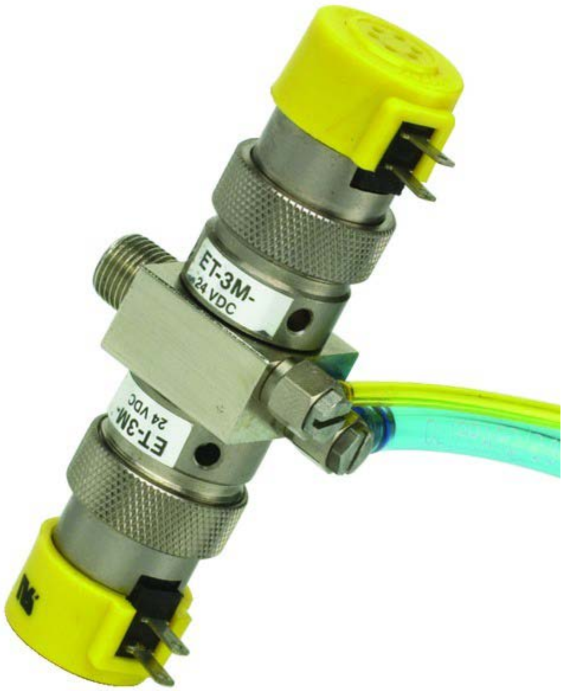


Figure 33 - Manifold Mount Example

These valves are ideal for our system because they are low power (0.67 watt), have a very fast response time (5-10 ms), have a large enough flow rate (0.6 standard cubic feet per minute at 100 psi), and can handle up to 105 psi. Since these valves are powered by 24 volts and we need to control them using a microcontroller, we created a transistor circuit to operate them easily, which is also shown in Figure 36. The transistors we used were made by STMicroelectronics with model number STPSA42. These transistors worked very well with our system because of its very low power consumption due to its low collector cut-off current at 100nA. The circuit allows for a microcontroller to send a low voltage signal to each transistor, which will then switch to allow the 24-volt difference across the valve terminals, opening the valves independently.

4.2.4 Microcontroller

For all of our initial testing and experiments up to this point, we have used the Arduino Uno (see Figure 34). This development board is useful because it is very user-friendly and simple to use. The Arduino Uno uses a microcontroller called the ATmega328. This development board has 14 digital input/output pins (depending on what the user assigns the pins to be), six analog inputs, and a clock speed of 16 MHz. Also, its maximum sampling rate is 10 kHz and the resolution of the Analog to Digital Converter (ADC) is 10 bits, in other words returning digital values between 0-1023.

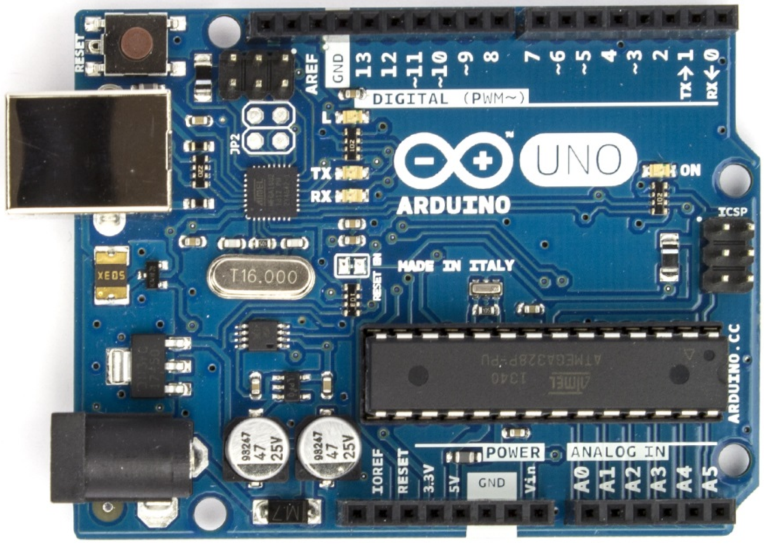


Figure 34 - Arduino Uno R3

The Arduino Uno has been very helpful in completing our preliminary testing and analysis of our prototype. However, we plan to use an alternative microcontroller for our final design that is more complex rather than relying on the Arduino Uno.

4.2.5 Complete Circuit Diagram

The diagram shown below is the electrical schematic for our prototype design. This initial design utilizes an Arduino Uno for all of the signal processing and communication within the system. The Uno powers the four pressure sensors, each of which sends an analog signal back to the Uno. The Arduino is also connected to the accelerometer breakout board to receive acceleration data from the sensor. Lastly, the Uno has 8 digital outputs to transistor circuits. Each output connects to a transistor that then connects to a valve. All of the valves are also connected to a 24-volt power source.

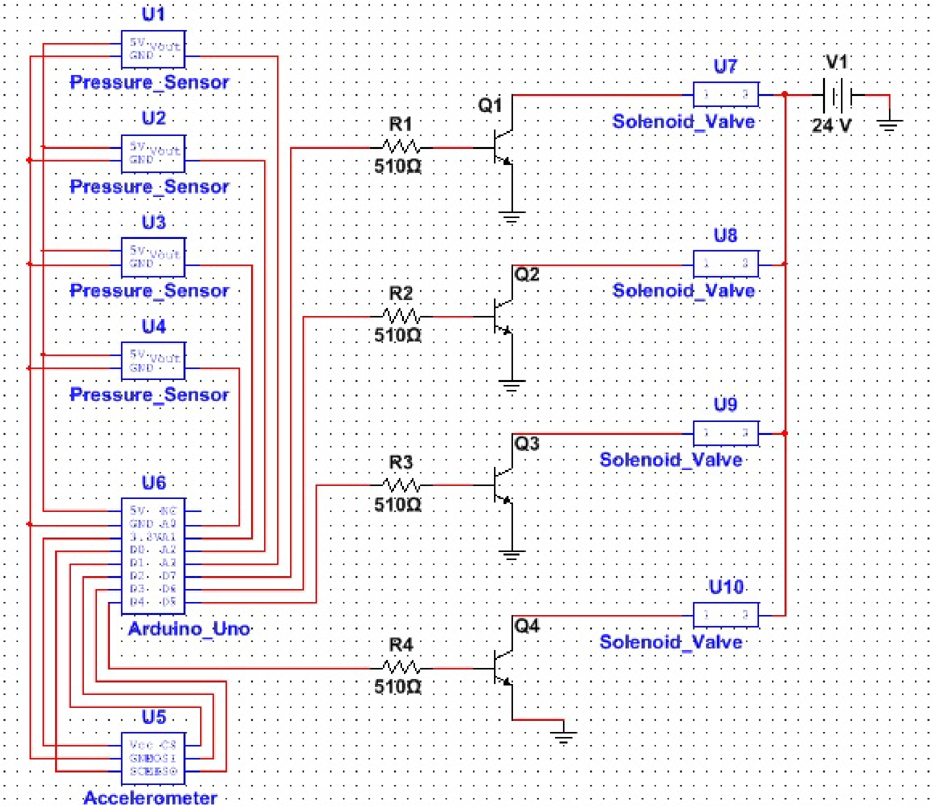


Figure 35 - Prototype Electrical Diagram

4.2.6 Power System

The prosthetic device requires a portable power supply, such as a battery, because the user will be moving around while wearing it. Each valve, contact pressure sensor, fluid pressure sensor, and accelerometer will require a power source. The required voltage for each sensor, the microprocessor, and the valves are shown in Table 11. Based on these, the maximum voltage any part of the system will require is 24 volts, which will be for the valves. The team is also aware that the sensors would require very little current (less than 1 mA), as would the microcontroller. The microcontroller will be connected to the main power source via a voltage regulator, allowing only 3.3 volts into the board. The battery used would also have to be rechargeable, as regularly disconnecting and replacing it could be difficult and expensive for the user.

The team decided to use a lithium-polymer battery, as these are very robust and are commonly used in cell phones and other electronic objects that may need to endure an occasional jolt or strong force. Compared to lithium-ion, lithium-polymer batteries are more lightweight and resistant to leakage, which makes them more ideal for prosthetics involved in motion.

The battery chosen for this design was a 25.9-volt rechargeable lithium-polymer battery pack with a maximum discharge rate of 7 amps, more current and voltage than the system requires (Part number: PL-5545135S7WR). It weighs 1.6 lbs (726 grams) and has a volume of 21.28 in^3 (354.4 cm^3), making it easy to carry on a prosthetic system.

Table 11 - Components and Required Voltage

Component	Voltage Required (Volts)
Valve	24
Accelerometer	3.3
External Pressure	5
Fluid Pressure	5
Microcontroller	3.3

4.2.7 Control System Breakdown

The control system of the final design must be able to read signals and perform simple, quick calculations to improve the user's comfort and safety as efficiently as possible. It consists of three main stages: reading the signals, determining the user's current activity, and adjusting the bladders appropriately. This section discusses the initial plan for the system and is represented in Figure 36, in which blue indicates the first state, gold the second, and gray the third.

Reading the Signals

The initial design for the prosthetic consisted of two different types of sensors: accelerometer and air pressure. The timer on the Arduino would allow the program to read the accelerometer values at 50 Hz. This frequency was chosen because the accelerometer would sample at this rate during our initial tests, and the pressure sensors could be read later at a different rate. The system began by reading and storing 100 samples from the accelerometer, taking approximately two seconds. Two seconds was long enough for the user to make a detectable gate motion if they were walking but short enough to minimize the risk of injury from inappropriately pressurized bladders. Also, having 100 samples stored at a time was enough to stop a few inaccurate readings from misinterpreting the results, but few enough samples for calculations to be performed quickly. The sensor would return four values, indicating its rotation around the X, Y, and Z axis as well as the temperature. The air pressure sensors would not be read during this stage.

Determining the User's Activity

The control system believes that the user is performing one of five activities: sitting, transitioning from sitting to standing, standing, or walking. To determine the current behavior, the system focuses on the rotation around the X axis (the axis of rotation of the hip joint) as well as the external pressure on the prosthetic, which our experiments show change significantly depending on the activity. The system begins by converting each of the X-axis rotation readings (X_r) into a degree angle measurement (Θ_x), using the following formulas:

$$\frac{|X_{stand} - X_{sit}|}{90} = d_1$$

$$d_0 = 2 * X_{sit} - X_{stand}$$

$$\frac{|d_0 - X_r|}{d_1} = \theta_x$$

X_{stand} and X_{sit} are the average X-axis readings for standing, and sitting respectively; d_0 is the average reading for if the leg is pointing directly upwards; and d_1 is the change in the sample values for every degree the upper leg rotates.

The initial plan was to write a program that would analyze the amplitude and frequency of the leg angles and use those to determine the user's current state. However, writing a program for finding the amplitude and frequency of a signal with little initial knowledge proved to be difficult and inaccurate. It was important that the system would be able to respond quickly to any change in the user's activity, so performing heavy, inaccurate calculations could force the socket to wait several seconds before adapting properly. Instead, we decided to consider the mean and standard deviation of the angles, as these are relatively quick to calculate. Based on our experiments, we discovered that when a person's legs are still, such as if they are sitting or standing, the upper leg angle has a predictable average and small standard deviation. However, if the legs are in motion, such as walking, then the mean is less predictable while the standard deviation is significantly higher, as the leg is constantly changing relative to the ground. These observations implied that the mean and standard deviation of the leg's angles could be used to determine a person's activity. Once the system knows these based on each X-axis sample, it finds the mean and standard deviation of the sampled angle values over the past two seconds. Then, the program compares these values to known thresholds for each activity, which were found through our experiments. The minimum and maximum thresholds for the mean and standard deviation of each action can be seen in Table 12.

Table 12 - Mean and standard deviation thresholds for activity (Rotation around X-axis)

Activity	**Mean (degrees)**		**Standard Deviation (degrees)**	
	Minimum	**Maximum**	**Minimum**	**Maximum**
Sitting	45	160	0	6
Transition	110	160	10	No maximum
Standing	170	200	0	6
Walking	150	210	0	30
Jogging	140	210	30	No maximum

The program compares the thresholds for the measured samples and finds which activity it qualifies for. For example, if the average X-axis angle is 80 with a standard deviation of 2, it will believe that the person is currently sitting because 80 is between 45 and 130 while 2 is between 0 and 5. However, it is possible for a mean and standard deviation to not match any of these activities. In this case, the system will simply repeat the process of reading the samples and performing the calculations without adjusting the bladders.

Adjusting the Bladders

Once the system knows the person's activity, it can adjust the bladder pressures appropriately. Initially, we intended to use a PID control algorithm, which adjust the valves based on the difference between the current pressure and intended pressure. However, it was very difficult to adjust the bladder size quickly with this control algorithm, as the valves could only turn on and off at a 100 Hz rate, and could not control how much air was flowing when they were on. Since the valves only had two settings, we decided to use a bang-bang control algorithm, in which the system takes an objective pressure reading as well as the current fluid pressure. Then it lets air into the bladder if the pressure is too low or lets air out because it is too high. The system will continuously let air in and out of the system until it reaches the proper pressure value. To avoid sending excessive messages, the system will only attempt to turn the voltage to a valve on or off if its state needs to change. Once the pressure is at the appropriate value, the valves are turned off and the program will immediately start again by reading the sensors to determine the activity.

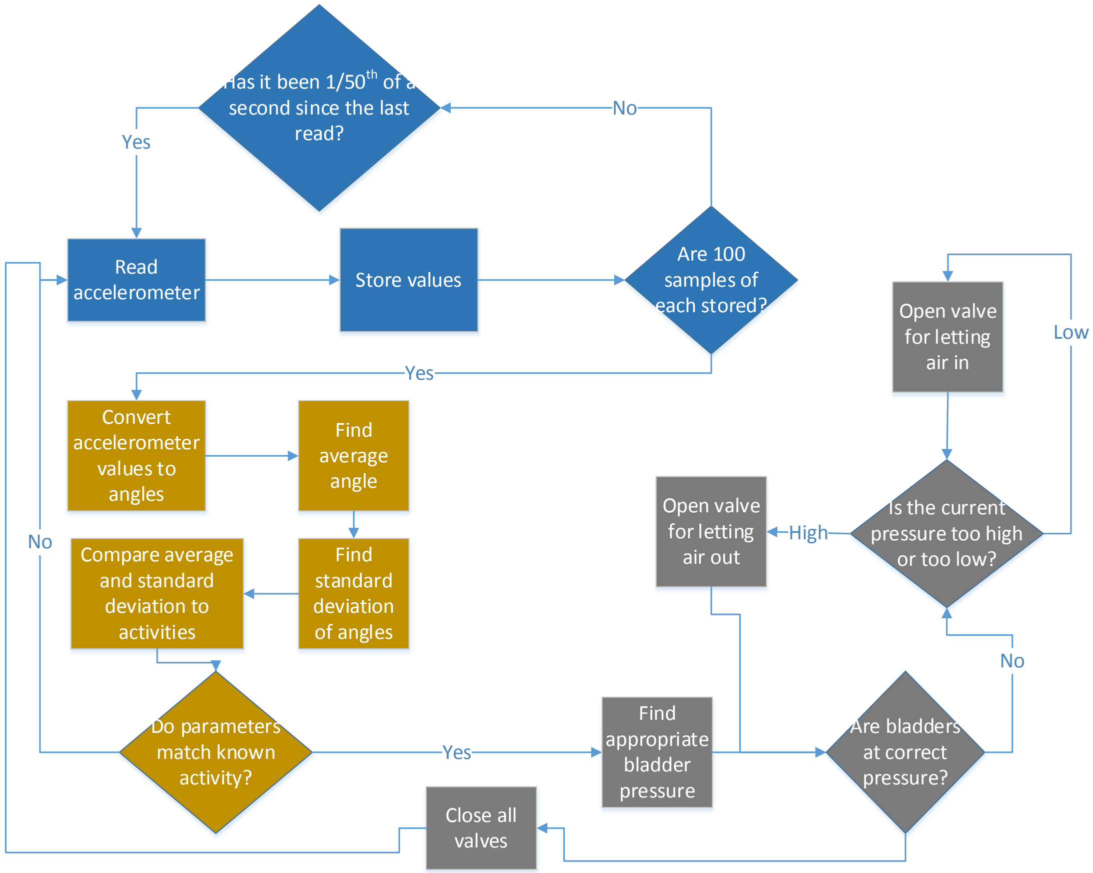

Has it been 1/50th of a second since the last read?
Yes
No
Read accelerometer
Store values
Are 100 samples of each stored?
Yes
Convert accelerometer values to angles
Find average angle
Find standard deviation of angles
Compare average and standard deviation to activities
No
Do parameters match known activity?
Yes
Find appropriate bladder pressure
Are bladders at correct pressure?
No
Is the current pressure too high or too low?
High
Low
Open valve for letting air in
Open valve for letting air out
Close all valves

Figure 36 - Initial Control System

4.2.8 Alternative Designs

In the beginning of the project, the team considered including multiple sensors in the system. Besides accelerometer and pressure, we were also deciding on whether or not to include pulse oximetry, galvanic response, and EMG. Through further research, testing and analysis, and experiments, we decided to only utilize accelerometer and pressure sensors.

Pulse oximetry is a very useful biological signal for medical device design. It can not only provide a patient's pulse, but it can also detect the patient's oxygen saturation in their blood. However, after further research, pulse oximetry is rarely used in lower limb applications. The team thought out first, throughout their background research, that utilizing pulse oximetry on the leg may be possible. However, pulse oximeters are generally only attached to a person’s finger, ear, toe, or nose. These areas are utilized because of the slight interference between the emitting diode, the blood being targeted, and the detecting diode.

Galvanic skin response (GSR), also known as electrodermal response, can be useful in medical devices as it can measure the resistance of a person’s skin. Therefore, GSR could detect if a patient is sweating, and potentially help interpret if they are comfortable or uncomfortable. However, this requires a skin-tight connection between the sensing electrode and the person’s skin. This would be a very difficult task for our prototype as we would need to always ensure a consistent connection. This is much different than the accelerometer because the accelerometer only needs to be part of the socket; it does not need to be directly connected or touching the skin.

Electromyography, or EMG, is further being utilized in new technological medical devices, especially those with robotic applications. For testing and analysis, we were fortunate enough to have a prototyping board already configured to measure body signals, including EMG. We used the BITalino prototyping board (see Figure 37) and the accelerometer to simultaneously collect data of the user sitting, standing, walking and jogging.

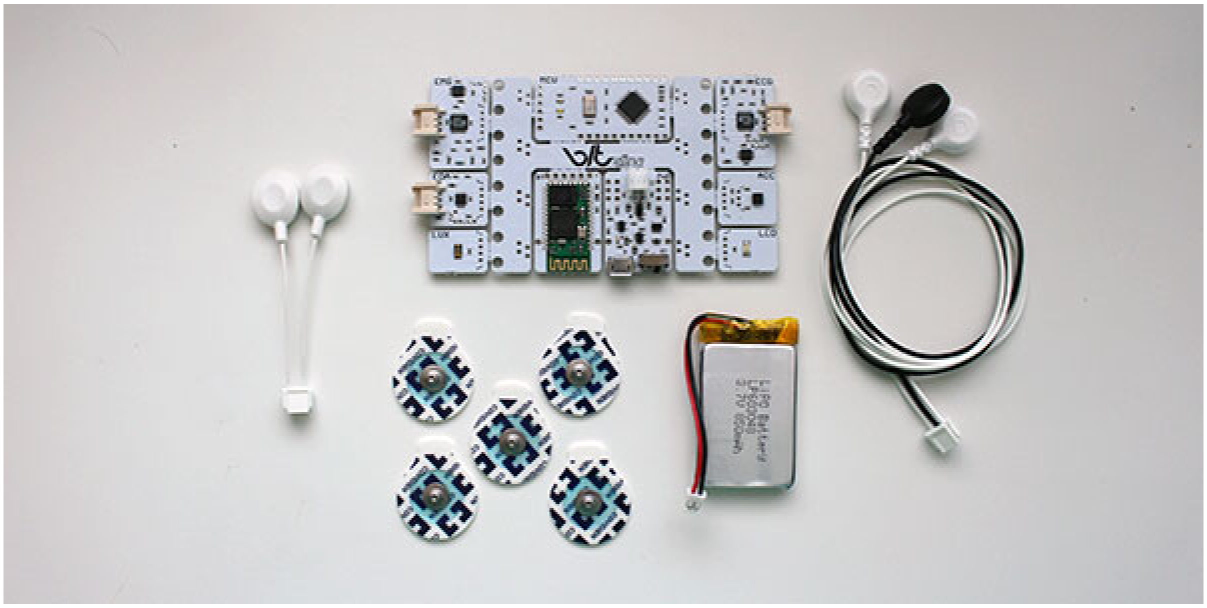
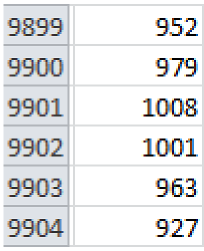


Figure 37 - BITalino BioSignals Prototyping Board

Using the BITalino BioSignals board, we utilized the EMG circuit that detected the muscle activity from our own legs. We performed the experiment with the electrodes on the front of our leg (quadriceps). The accelerometer was connected to a breadboard and an Arduino which both supplied voltage to the accelerometer and collected data from the accelerometer.

We recorded data while we were sitting down, standing, walking, and jogging. We executed each activity for 10 seconds. We then looked at patterns and specific characteristics of the EMG and the accelerometer data that related to the specific activity. After this, we wrote MATLAB code to display data acquisitions, accelerometer and EMG, on the same time scale. The purpose of this was to see if both sensors are necessary or if they convey the same information.

The EMG sensor on the BITalino board samples at a rate of 1,000 Hz, or 1,000 samples per second, and therefore we collected approximately 40,000 EMG data points. The BITalino automatically offsets the digital output of the EMG to 512 with a minimum possible value of 0 and a maximum possible value of 1023 (10-bit). A snapshot of the EMG data can be seen in Figure 38, where the first column is the sample number and the second column is the digital EMG sensor value.



9899	952
9900	979
9901	1008
9902	1001
9903	963
9904	927

Figure 38 - EMG Sensor Data

After collecting data from both the accelerometer and EMG, the team developed a MATLAB code to display both sensors on the same time scale (see Figure 39). As one can see, the graph of the EMG looks similar to the other three graphs showing the X, Y and Z axis of the accelerometer. It does appear that the EMG starts to pick up the activity of the user nearly a second before the accelerometer. However, it also picks up muscle twitches and movements that may not always suggest the correct activity. Therefore, this proves that only one of the sensors is needed, the accelerometer.

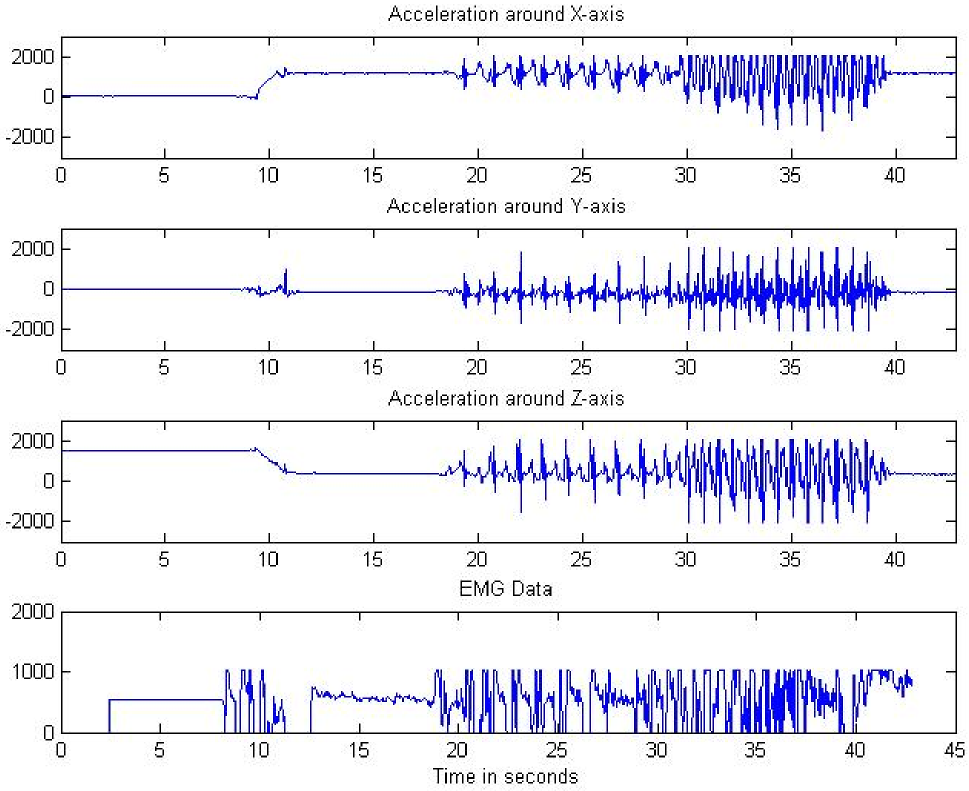


Figure 39 - Simultaneous Accelerometer and EMG Data

Since the EMG would need a very close connection from the electrode to the skin, it could possibly become inconsistent. This proved to be true in our testing as many of our values randomly went high (to 1024) or low (to 0). Another concern with using EMG is that it can be far more consequential. For instance, some upper-limb prosthetic designers are already trying to utilize EMG in their prosthetics. A consequence could be a malfunction and lead to something similar to someone dropping their coffee. However, a malfunction of a lower limb prosthetic using EMG could be as unforgiving as someone falling down the stairs because their leg muscle had a twitch or the connection between the electrode and skin became loose. One advantage that the EMG seems to have is that it can detect a movement slightly before the accelerometer. With proper development and usage, EMG could possibly be exploited in the future for lower limb prosthetics (see Chapter 6. Discussion and Recommendations). However, in terms of this project, EMG as well as pulse oximetry and galvanic skin response will no longer be a part of the team's design.

4.3 Testing Rig Experiments and Results

The team performed various experiments to help determine the final design. These experiments including inflating and deflating the bladders to determine how long the reservoir tank could last, wearing the rig and seeing if the control system could determine the correct state, testing the response time of the system, and the impact of the length of tubing used.

4.3.1 Detecting the Correct State

This section includes an experiment with the testing rig worn by three members of the team, Shane, Tynan, and Alex. We performed the three actions stated throughout this paper (sitting, standing, and walking) for three different trials to see if the sensors and microcontroller could correctly detect the state.

Materials

-Testing rig (equipped with Arduino and accelerometer)

-PC (connected to Arduino)

Goal: To find how quickly and accurately the control system can determine the user's current activity

Methodology

One subject will attach the testing rig to his leg using a Velcro strap. The strap should be enveloped around the testing rig as tightly as possible. The subject will then wrap the remainder of the strap over his shoulder and hold it to ensure that the rig does not fall. Another subject will hold the computer and upload a program which will be determining the user's current state. This program will output a number that represents what it believes the current activity to be. The subject wearing the testing rig will then sit for fifteen seconds, with either the PC holder or third subject keeping track of the time. Then, the subject will slowly stand up and wait fifteen seconds before walking for another fifteen seconds.

Once this is complete, the subject holding the PC will disconnect the Arduino to stop serial port communication and will store the output results in a separate file. After three trials, the first subject will disconnect the testing rig. The procedure will be repeated with the other two subjects as well.

Results

For each trial, the state was determined at a frequency of five Hz. Each subject's state determinations were averaged together, creating one vector of state values for each subject. These mean samples were then plotted in MATLAB along with the subject's actual state, assuming a sitting to standing transition period of one second. These MATLAB plots are shown in Figure 40.

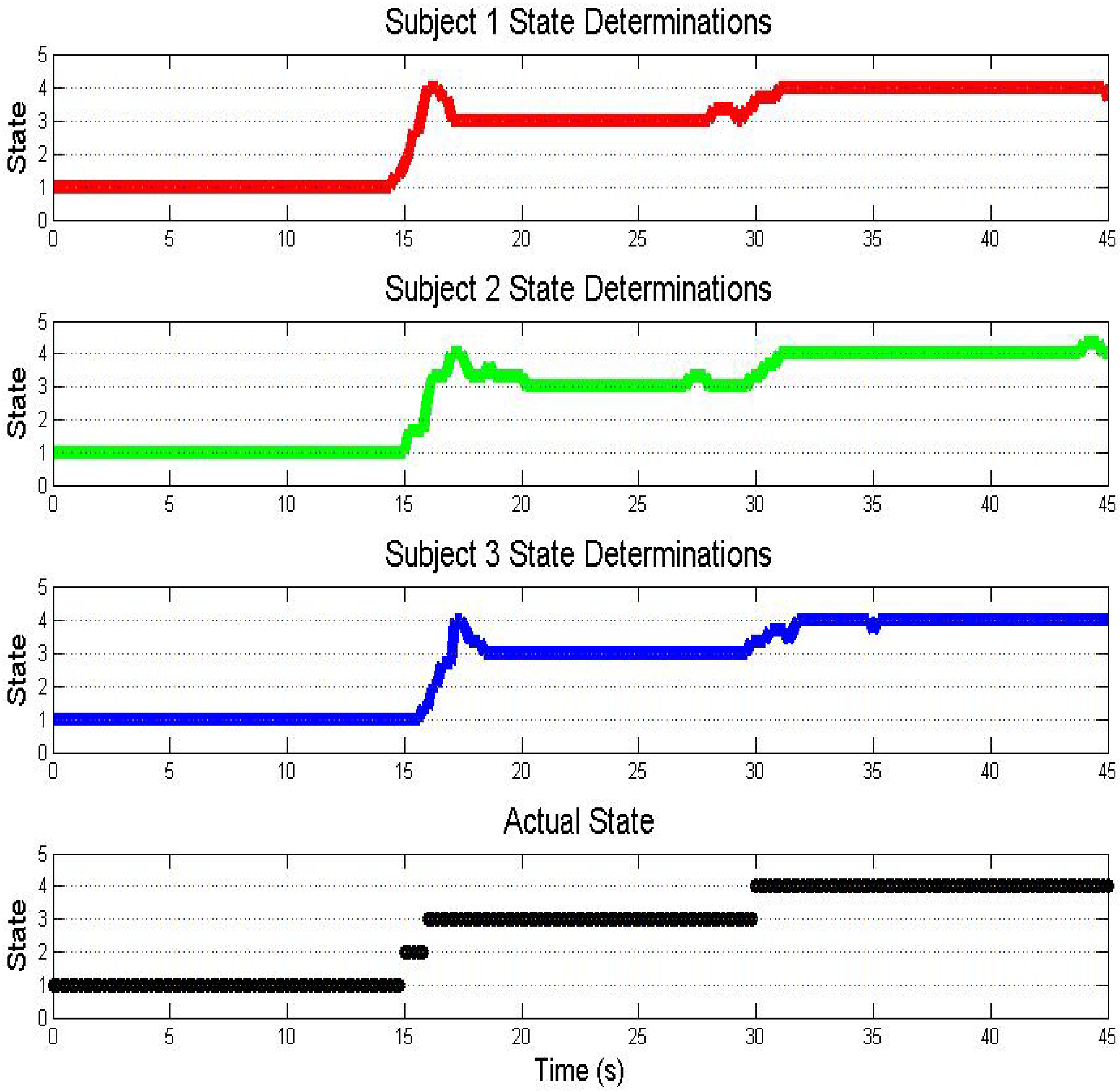


Figure 40 - State Determinations

For these plots, the values one, two, three, four, and five indicate sitting, sitting-standing transition, standing, walking, and jogging respectively. Although the actual state graph only allows for integer values, each subject had three trials and therefore the average of the three trials was plotted for each subject. Based on these plots, the system had the most difficulty with transitioning between states. During the sitting to standing transition, it would briefly believe that the user is walking. However, we do not view this as a major issue because whether the user is walking or transitioning from sitting, the bladders will increase in pressure. If the pressure is slightly higher than it should be, it will not cause an issue if the system is able to rectify the mistake quickly. The second transition was from standing to walking. For this test, it would commonly result in a delay of a second or two before determining the new state. One possible explanation for this is that the subject would commonly lead the walk with the leg that did not have the testing rig. This would delay the rig from entering the swinging gait, making it more difficult for the system to realize the person is walking.

However, once the system is past the transition, it is very consistent and accurate. Apart from a few scattered incorrect samples, the system was able to correctly recognize when the state did not change for several seconds. This indicates that the wearer should be able to remain in a state for a long period of time with little fear of the system adjusting improperly.

This data was also analyzed to find the overall accuracy, or percent of samples for each trial that gave the correct state. Table 13 displays this information, also giving the average percent correct for each subject's trials.

Table 13 - State Determination Accuracy

Subject 1			**Subject 2**			**Subject 3**		
Trial 1	**Trial 2**	**Trial 3**	**Trial 1**	**Trial 2**	**Trial 3**	**Trial 1**	**Trial 2**	**Trial 3**
94.25%	95.13%	96.02%	86.73%	94.69%	91.15%	91.59%	92.48%	90.27%
Average: 95.13%			Average: 90.86%			Average: 91.45%		

There were a number of factors that could have affected these results. For example, there was often a delay between when the wearer was told to change activities, and when they reacted to the instructions of the timekeeper. The graph of the actual state assumes an immediate transition between activities, save for the one second transition between sitting and standing. The second subject’s first trial was noticeably lower as he attempted to adjust the rig during the sitting-standing transition, which affected the accelerometer readings. The remaining eight runs all had an accuracy of over 90%, with the best trial having an accuracy of 96.02%.

4.3.2 The Impact of the Length of Pneumatic Tubing

This section goes into detail about how important the length of tubing is from the tank to the actual bladders. An experiment involving using different lengths of tubing and the response time of filling up the bladders to the ideal pressure was tested. This experiment was performed to see if components of the socket could possibly be moved away from the socket, and into another sealed compartment such as a backpack.

Goal

The goal of this experiment is to determine the effect that the tubing length in the system has on two different parameters. These parameters are the difference in pressure at either end of the tube, and the amount of time the bladder will take to reach an equilibrium pressure.

Methodology

This test will be set up using various lengths of tubing, two pressure sensors, a pressure switch, a bladder, a pressure regulator, and the compressed air output in the lab. We attached the pressure regulator to the compressed air to monitor the amount of pressure allowed into our system. We then attached a pressure switch and one pressure sensor right at the regulator. We then attached a length of tubing to the sensor. We attached the other sensor to the other end of the tubing and then attached the bladder to the second sensor. This setup is shown in Figure 41.

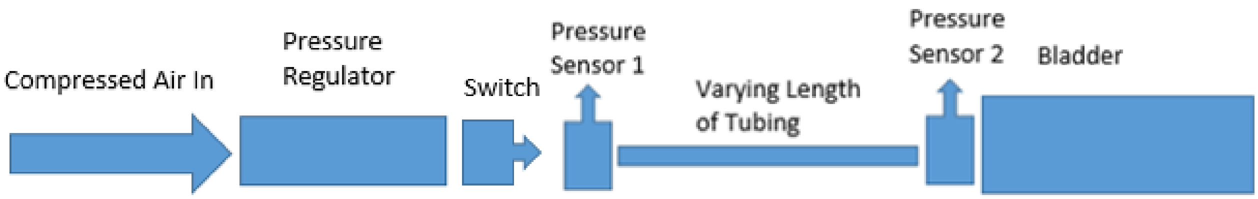


Figure 41 - Tube Length Experimental Setup

To perform the experiment, we cut lengths of tubing to 2, 6, 12, 18, 24, 36, and 48 inches long. We attached the 2 inch piece of tubing to the experimental set up and then allowed the regulator to allow 5 PSI into the tubing and bladder. We opened the pressure switch and constantly measured the pressure sensor outputs at about 20 Hz, approximately 20 times per second. We stopped recording once the bladder had reached an equilibrium pressure. We then could determine the difference between the two pressure sensor readings and the time they each took to reach equilibrium. We performed this test at 5, 10, and 15 PSI for each length of tubing and recorded all results.

Results

The results of the experiment showed us that the length of tubing between the two sensors does not have a significant effect on the readings of the sensors. For every length of tubing and every pressure we tested, both sensors read the exact same pressure values at every single sample. This means that within the selection of tubing for these pressure ranges, there was no more than a 50ms difference in response time. This gives us confirmation that moving the pressure sensor further away from the bladder will not significantly affect the readings that we obtain.

The second parameter we wanted to test was how long the bladder would take to reach an equilibrium pressure depending on the input pressure and tubing length. Table 14 shows the amount of seconds the bladder took to reach equilibrium for each test.

Table 14 - Tube Length Experiment Results

Tube Length Between Sensors (inches)	Input Pressure (PSI)	Sensor 1 Raw Reading (bits)	Sensor 1 Reading (PSI)	Sensor 1 Time to Equilibrium (seconds)
2	5	389	5.207823961	15.105
	10	725	11.36919315	14.501
	15	949	15.47677262	12.588
6	5	409	5.574572127	15.171
	10	707	11.0391198	14.023
	15	942	15.34841076	13.401
12	5	404	5.482885086	13.544
	10	715	11.18581907	14.836
	15	960	15.67848411	13.162
18	5	405	5.501222494	14.645
	10	677	10.48899756	13.162
	15	966	15.78850856	11.057
24	5	387	5.171149144	15.841
	10	697	10.85574572	14.166
	15	948	15.45843521	13.975
36	5	392	5.262836186	15.889
	10	684	10.61735941	14.501
	15	930	15.12836186	15.41
48	5	405	5.501222494	19.189
	10	714	11.16748166	15.601
	15	966	15.78850856	14.31

Although the results may not intuitively make sense, this could be due to the differences being within the noise or margin of error. The results of the experiment show that the length of tubing does not have a very large effect on the time to equilibrium. As shown in the table and Figure 42, there is no significant difference in filling times for any tubing length.

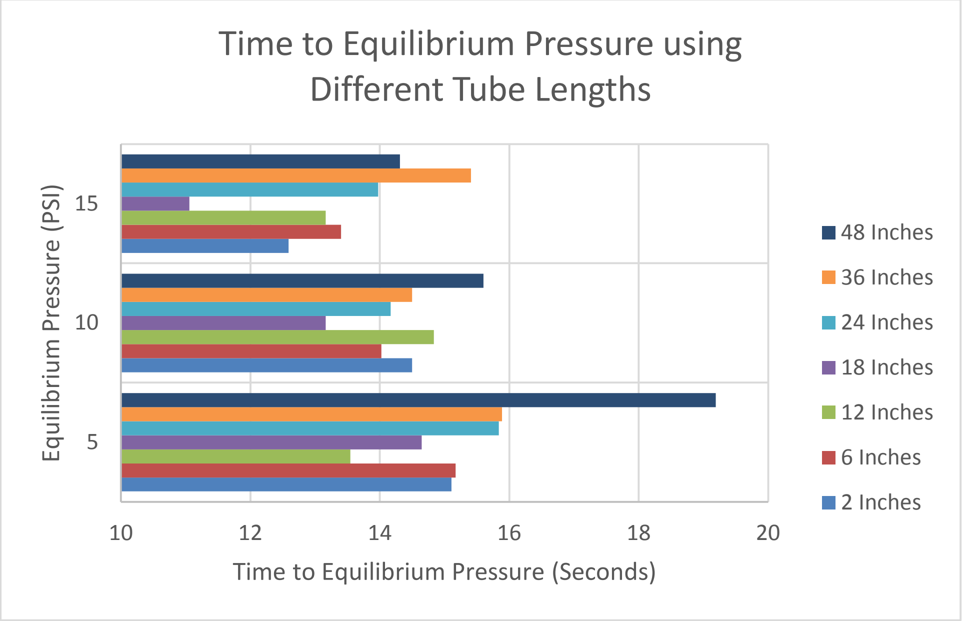


Figure 42 - Equilibrium Times Graph

When tested at 5, 10, and 15 psi, the filling times did not have any strong correlation to tubing length. Once the tubing was about 48 inches long, we finally started to see some delay and the bladder took a bit longer to fill, but every other tubing length had no significant difference. Since we will definitely not be utilizing any tubing longer than 36 inches in our system's design, this test confirmed that we can move the air supply and sensors further away from the bladders without any error resulting from this change.

4.3.3 Fluid Analysis of Pressure and Air Flow Rate

The purpose of this analysis was to determine and verify whether or not the length/diameter of the tubing would affect the pressures within the bladders.

Equation for Determining Pressure Drop for Compressed Air in Pipe Line

Our team first determined that we would be able to use an equation for determining the pressure drops in compressed air lines. The team decided to use this equation because our air is being compressed and pushed through the tubing into the various valves and bladders. It could not be assumed that air is incompressible, so we could not use equations such as the Hagen-Pousille equation. Instead, the equation for determining pressure drops in compressed air lines is shown below (Engineering Toolbox):

$$\Delta P = \frac{7.57 * q^{1.85} * L * 10^4}{p * d^5}$$

In this equation, ΔP is the pressure drop between the ends of the tubing (given in kg/cm^2), p is the initial absolute pressure of the system (converted into kg/cm^2), L is the length of the tubing (converted into meters), d^5 is the diameter of the tubing to the fifth power (converted into mm to the fifth power), and q is the air volume flow rate at atmospheric conditions (converted into m^3/min). This initial absolute pressure was found by adding the standard atmospheric pressure, which is 14.7 psi (1.03 kg/cm^2), added to the gage pressure, which is the amount we are pressurizing the air to which is 10 psi (0.7 kg/cm^2). This means that the initial absolute pressure was 1.73 kg/cm^2. Everything besides the volumetric flow rate was given or derived experimentally, as shown earlier in Section 4.3.2. Therefore, our first step was to solve for the air volume flow rate and determine how the length and diameter of the tubing affected this variable.

Air Volume Flow Rate vs Length of Tubing

This is the pressure drop equation for compressed air lines solved for the air volume flow rate, q:

$$q = \left(\frac{\Delta P * p * d^5}{7.57 * L * 10^4}\right)^{\frac{1}{1.85}}$$

Then we solved for the volumetric flow rate for each of following tube lengths as derived from the above experiment: 2 inches, 6 inches, 12 inches, 18 inches, 24 inches, 36 inches, and 48 inches. This was tested with a 4 mm inner diameter tubing, which is the size tubing used in most our pneumatic system. Below is the volumetric flow rate equation solved for the 2 inch long tubing:

$$q = \left\{ \frac{\left[(5.208 - 5.0\,psi) * \left(\frac{0.0703\,\frac{kg}{cm^2}}{1\,psi} \right) \right] * \left(1.73\,\frac{kg}{cm^2} \right) * [4\,mm^5]}{7.57 * \left[(2\,in) * \left(\frac{0.0254\,m}{1\,in} \right) \right] * 10^4} \right\}^{\frac{1}{1.85}}$$

$$q = 0.0670\ \mathrm{m}^3/\mathrm{min}$$

This process was repeated for each of the stated tubing lengths. Table 15 below shows the summary of all of the calculated volumetric flow rates.

Table 15 - Air Volume Flow Rates for Changes in Tube Length

Tube Length (in)	Air Volume Flow Rate, q (m^3/min)
2	0.0670
6	0.0370
12	0.0254
18	0.0204
24	0.0175
36	0.0140
48	0.0120

This table was then converted into a graphical representation in order to visualize the data. Figure 43 shows the calculated air volume flow rates versus the tubing length.

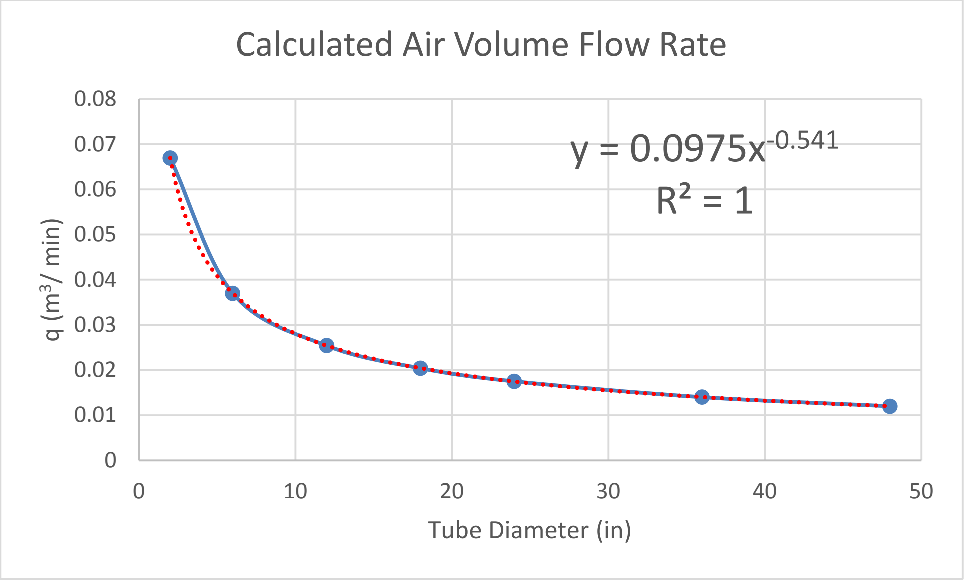


Figure 43 - Volumetric Flow Rates vs. Tube Length

The data points were then fitted to a power trend line using the given equation we calculated above and R^2 value was shown as well. This graph shows a steep decrease in volumetric flow rate from 2 until about 20-24 inch tube length and then the rate seems to level off to around 0.015 m^3/min. All of this data shows that after about 20 inches to tubing, there is not a significant amount of change in the flow rate of air within the tubing. Most of this is intuitive in that as you lengthen the tubing, the speed of flow should decrease, with a sharp amount at the beginning and then leveling off near the end. This is mostly due to friction between the flow and the walls so as the air flows through more tubing, it encounters more friction, which would cause the air volume flow rate as the length of tubing increases.

Air Volume Flow Rate vs Diameter of Tubing

This process was then repeated for the changes in the diameter of the tubing. The volumetric flow rate was calculated for the following inner diameters of tubing: 2 mm, 4 mm, 6 mm, 8 mm, and 10 mm. This was tested against the 24 in long tubing, since this is the value that would most likely take place within our pneumatic system and it was the length at which the earlier air volume flow rate data started to stabilize. Below is the volumetric flow rate equation solved for the 2 mm wide tubing:

$$q = \left\{ \frac{\left[(5.208 - 5.0\,psi) * \left(\frac{0.0703 \frac{kg}{cm^2}}{1\,psi} \right) \right] * \left(1.73 \frac{kg}{cm^2} \right) * [2\,mm^5]}{7.57 * \left[(24\,in) * \left(\frac{0.0254\,m}{1\,in} \right) \right] * 10^4} \right\}^{\frac{1}{1.85}}$$

$q = 0.0027\ \mathrm{m}^3/\mathrm{min}$

This process was repeated for each of the stated tubing diameters. Table 16 shows the summary of all of the calculated volumetric flow rates.

Table 16 - Calculated Volumetric Flow Rates

Tube Diameter (mm)	Air Volume Flow Rate, q ($\mathrm{m}^3/\mathrm{min}$)
2	0.0027
4	0.0175
6	0.0523
8	0.1139
10	0.2081

This table was then converted into a graphical representation in order to visualize the data. Figure 44 shows the calculated volumetric flow rates versus the tubing diameter.

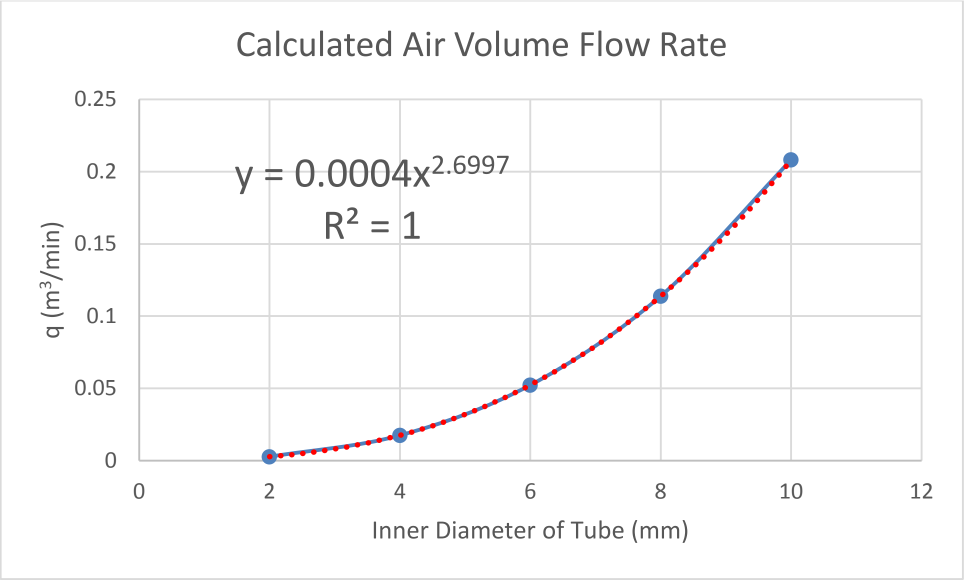


Figure 44 - Calculated Flow Rate vs. Tube Diameter

The data points were then fitted to a power trend line with the exact equation that was calculated above and R^2 value is shown as well. This curve shows a gradual increase of volumetric flow rate as the diameter of the tubing increases. This graph is logical in the sense that as you increase the diameter of the tubing, more air will be able to pass throughout the system, which should increase the volume of air that passes through the tubing within a given time frame. This could be applied to our system in that if we have an issue of air being able to get to the bladders in a given time frame, the diameter of the tubing can be increased. The diameter of the standard tubing that was used in our system is 4 mm, and the middling value within this graphical representation was 6 mm. 6 mm is the point at which the exponential curve starts to sharply increase. Therefore, if we are able to stay around the range of 4-6 mm tubing, there would not be a sharp volumetric flow rate increase of air into the bladders, which could lead to drastically high pressures in a short period of time. This would initially cause the valves and bladders to initially react more rapidly, however, with more use, this could cause the system to not be able to react in time. This could result in instability, blown bladders, or too much pressure on the residual limb of the user, instigating irritation and loss of feeling within the limb. Therefore, it was decided to keep the tubing diameter between 4-6 mm.

Change in Volume Related to Tubing Length and Diameter

The next thing that was analyzed was how the change in length/diameter of the tubing would affect the change in volume within the tubing. This was done in order to show the maximum range in volume of air the tubing can hold from the shortest length to the longest length, and from the smallest diameter to the largest diameter. Therefore we used the following equation:

$$V = A * L$$

This is an intuitive equation in the volume is derived from the cross sectional area of the object times the length. We then applied this equation to the change in volume shown below:

$$\Delta V = \ (A_1 * L_1) - (A_2 * L_2)$$

For this equation, we kept the diameter consistent at 4 mm, like above, and studied the change in volume within the tubing from the shortest length of tubing we tested, 2 inches, to the longest length, 48 inches. This is what resulted:

$$\Delta V = (\pi * \frac{d^2}{4} * L_1) - (\pi * \frac{d^2}{4} * L_2)$$

$$\Delta V = (\pi * \frac{0.15748\ in^2}{4} * 48\ in) - (\pi * \frac{0.15748\ in^2}{4} * 2\ in)$$

$$\Delta V = 0.896\ in^3$$

Next, we tested the same equation, but this time we kept the length consistent at 24 inches, like above, and studied the change in volume within the tubing from the smallest diameter of tubing, 2 mm, to the largest diameter, 10 mm. This is what resulted:

$$\Delta V = (\pi * \frac{d_1^2}{4} * L) - (\pi * \frac{d_1^2}{4} * L)$$

$$\Delta V = (\pi * \frac{0.3937\ in^2}{4} * 24\ in) - (\pi * \frac{0.07874\ in^2}{4} * 24\ in)$$

$$\Delta V = 2.805\ in^3$$

As shown above, there is an approximate difference of 1-3 cubic inches in tubing volume from the shortest tubing to the longest. The volume of the each of the 4 bladders when fully inflated is approximately 50 cubic inches, so this is significantly small amount of difference in volume. This shows that there would not be a big difference in the volume of air that would be contained within the tubing whether it was right next to the socket at 2-6 inches or back in the back pack at 30-35 inches, or whether or not the tubing is relatively small or large in diameter. This shows that there would not be a significant difference in the air pressure between the valves within the backpack and the bladders within the socket. The pressure would be lower since, in ideal cases, pressure is inversely proportionate to volume, however the difference would be between one-third and half a psi. This is the mean difference in pressure within the experimental pressure data shown in the previous section(s). This is not a significant change in pressure, especially in the working system that we have. We used the changes in pressure that we experimentally calculated, which validates all of our comparisons and models. We therefore conclude from all of these analyses that changing the positioning of all of the non-bladder components to the backpack instead of on the socket will not significantly affect the flow rate or air pressure within the bladders during states of activity.

5. Final Design

This chapter will discuss the steps we took toward developing our final prototype. The mechanical system will be discussed first, followed by the electrical system, which includes the control system and power requirements, and then lastly testing experiments. Figure 45 shows the system at the high-level, including the Arduino Mega, circuit interface board, main circuit board, and computer.

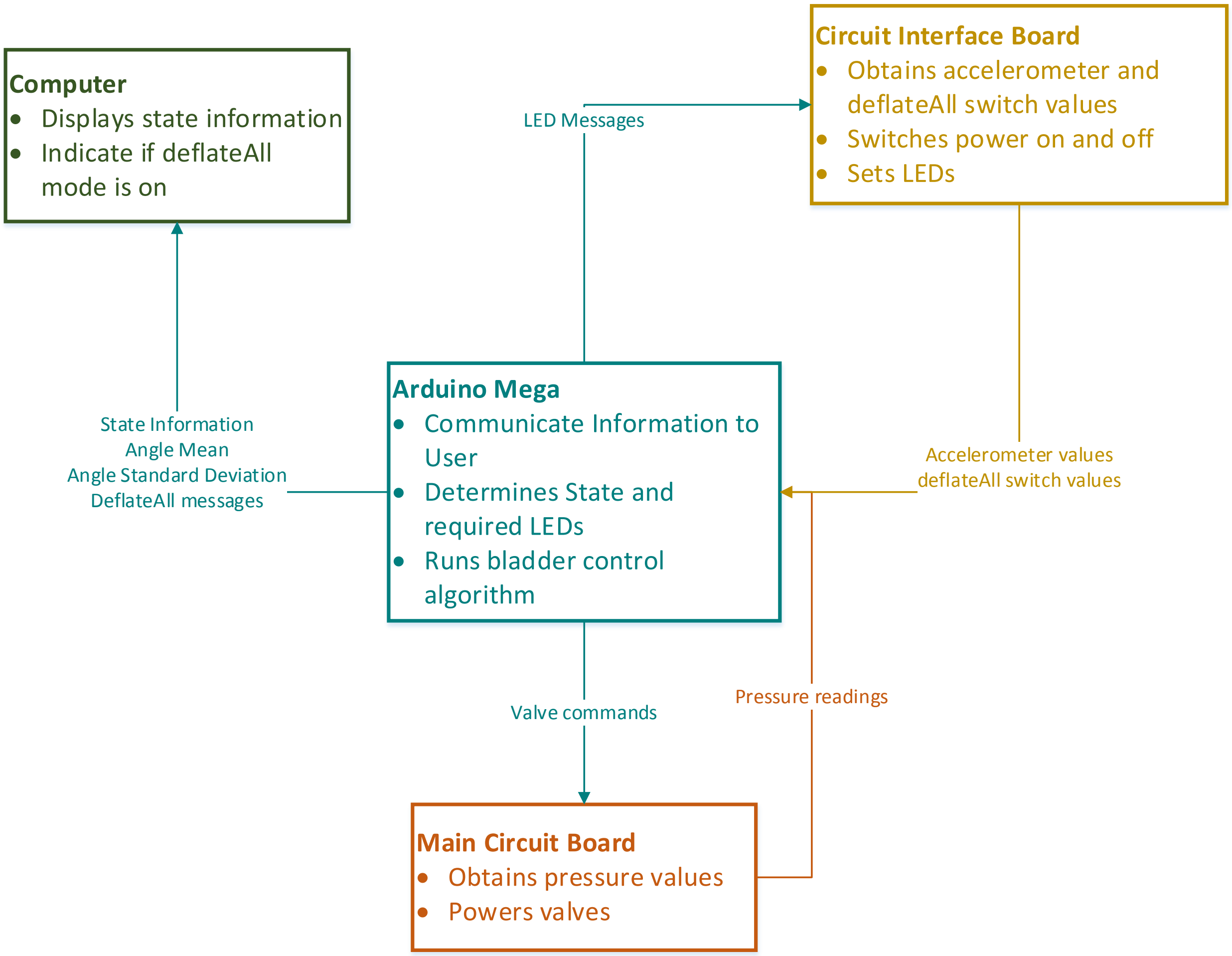


Figure 45 - Upper Level System Diagram

5.1 Mechanical System

The basic layout of our mechanical/pneumatic design did not change too drastically, but there were some important changes that were made. This section will describe what these changes were and why we decided to choose each sub-design and component for our final mechanical and pneumatic assembly.

The most significant addition to our design is the consolidation of all of our components into a backpack. In order to secure every component, we built a structure to hold everything in place. This structure, shown in Figure 46, is designed to fit inside of a backpack. This figure also shows how the entire system is connected to the bladders within the socket. Figure 47 shows the structure with the backpack.

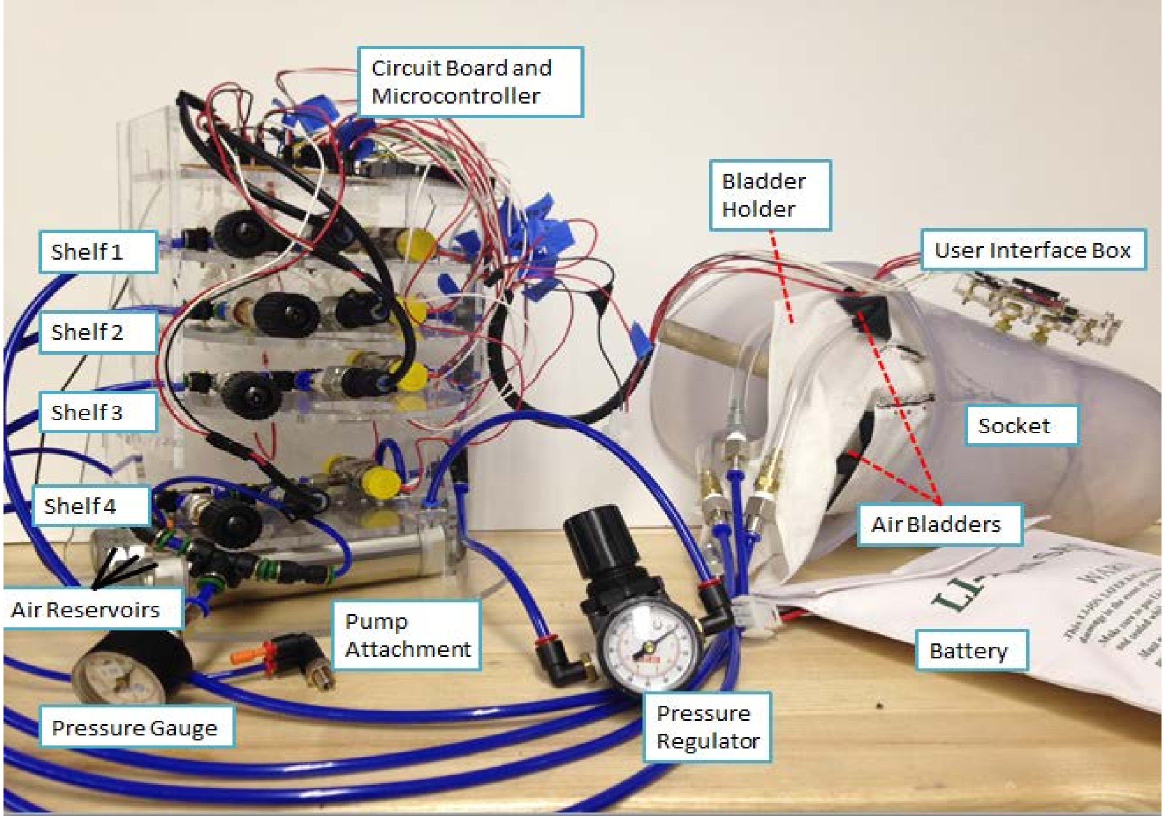


Figure 46 - Mechanical Setup

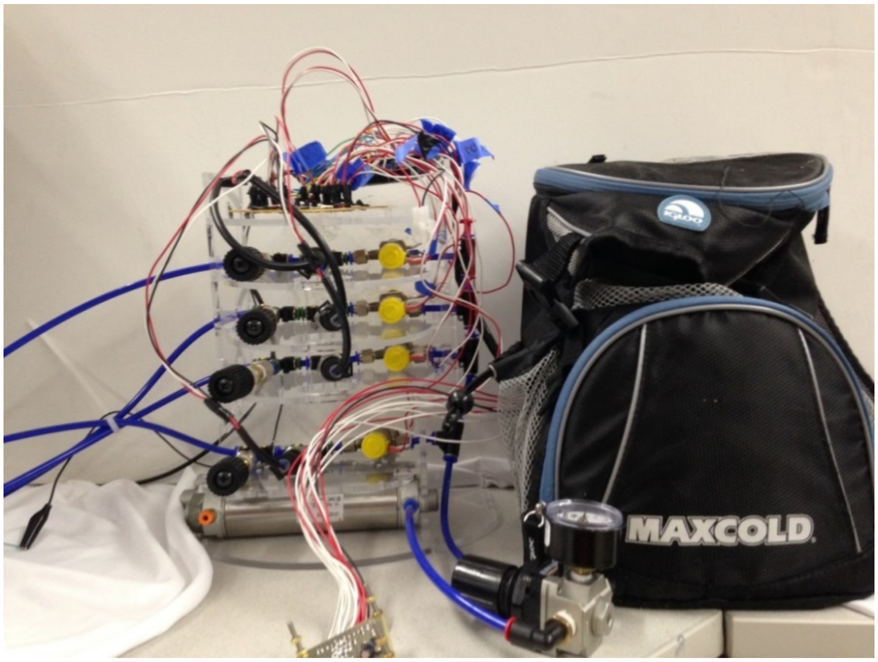


Figure 47 - Backpack Structure

Figure 48 is a close-up image of a shelf. Each shelf contains an On/Off valve, a pressure sensor, and a blow-off valve in series.

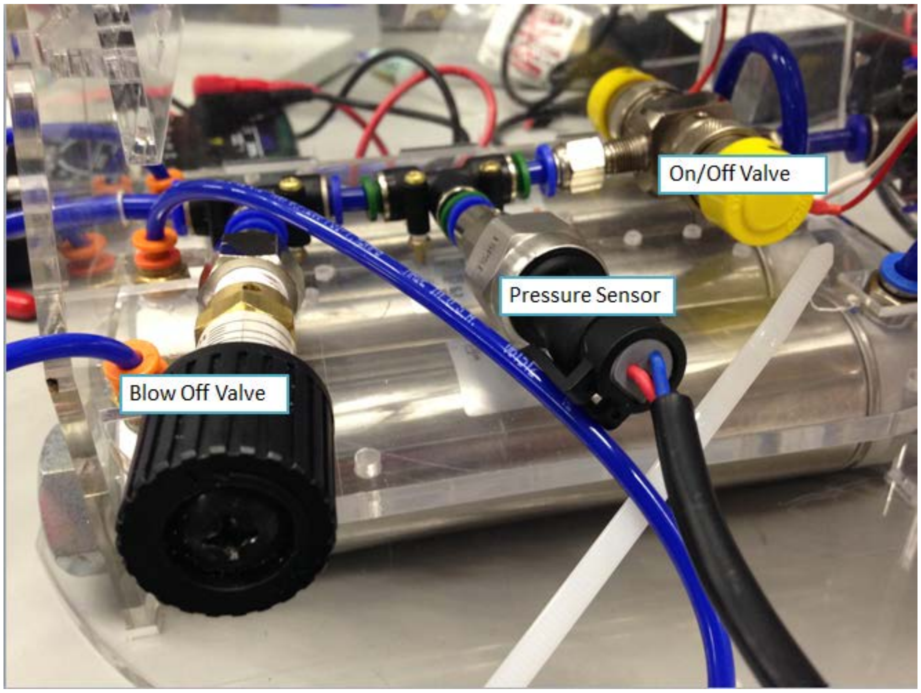


Figure 48 - Shelf Setup

The pneumatic system underwent some significant changes and improvements. Figure 49 shows a simplified overview of the entire pneumatic system.

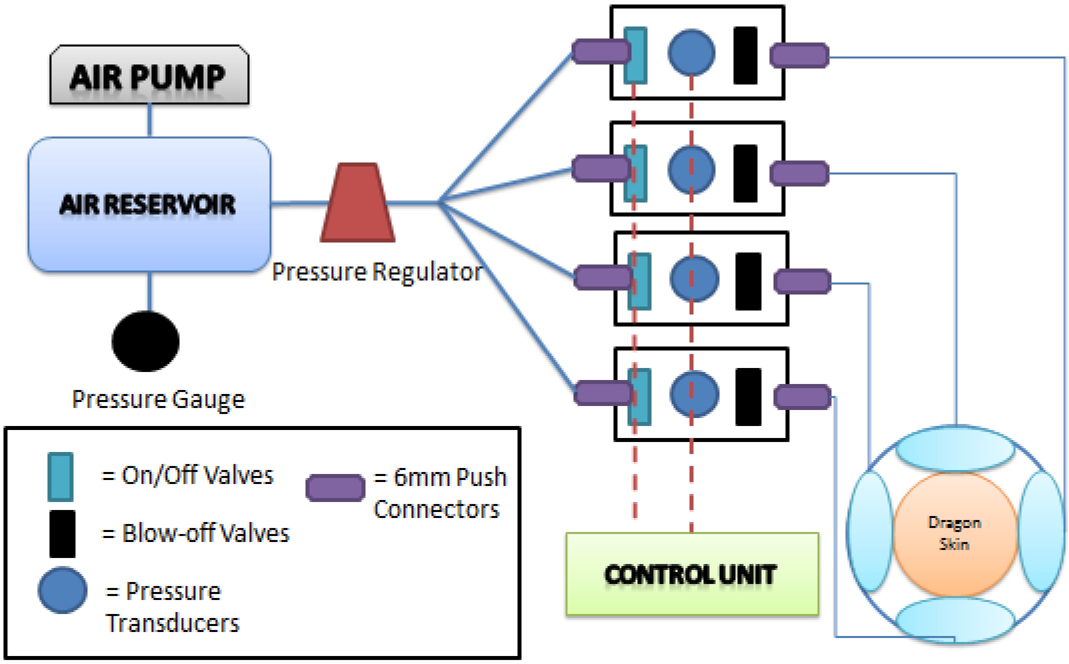


Figure 49 - Final Pneumatic System

There are several significant changes to this layout than the previous layout shown in Chapter 4. Starting at the source of air, our final design is using a pump to pre-pressurize the air reservoir rather than a continuously operating integrated air compressor. Note that later versions of the project may choose to include an air compressor, but that is an easy addition.

The air reservoir has changed from a single tank to three separate tanks. This was done to maximize the volume of the reservoir (still pressurizing to 150 psi). Attached to the tank is a pressure gauge which sticks out of the backpack. This allows the user to see how much air is in the tank at any time.

Air then flows from the reservoir to a pressure regulator which is always set to 10psi. The regulator controls the amount of air flow coming from the tanks. Tubing coming from the regulator is then split using T-shaped push connectors and fed into the shelves of the structure which hold the on/off valves, the pressure transducers, and the blow-off valves. The on/off valves and the pressure transducers are all connected to our control unit and are used to regulate the pressures within each bladder. The blow-off valves are there for safety, as they are designed to release air when the pressure in the bladders gets too high (10 psi).

Finally, air is let into the bladders which are located in the pockets of the bladder sleeve. This sleeve is slipped onto a mold of a residual limb which sits inside the socket.

5.1.1 Socket

As stated earlier, the prosthetic socket is an integral part to our project because it is the interface between the user's residual limb and the prosthetic itself. For this project, we received a testing socket made out of a hard plastic material from Ottobock, as shown in Figure 50. To have enough clearance on either side of the thigh for the bladders, wires, and other parts, the socket should be about 9-10 inches in diameter. The Ottobock socket is approximately this distance, but it tapers downwards. Unfortunately, we were unable to secure any amputees to test the socket in its full functionality. Instead, after removing all the components besides the bladders from the testing rig and placing them in the backpack, we created a molded leg. This leg is placed inside of the socket along with the bladder and its holders in order to simulate a leg. As will be discussed later, we attached this system to a robotic system that simulates all of our necessary activities.

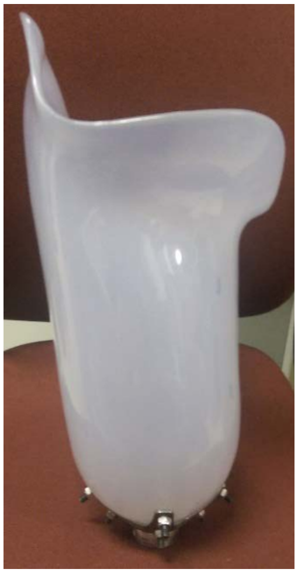


Figure 50 - Ottobock Prosthetic Socket

5.1.2 Backpack/Bag

As it was mentioned earlier, all of the mechanical and control components except for the bladder and the associated tubing and holders will be placed inside of the backpack. This was to eliminate most of the weight placed on the testing rig/socket for ease of use and testing. The components held in the backpack include the battery, pressure regulator, pressure valve, all three tanks, all four sets of valves and pressure transducers, electrical board, Arduino board, and all associated wiring and tubing. Therefore, a shelving structure had to be made in order to hold all of these items for ease of use and access. After several designs and modifications, the final design of the shelving structure is shown in Figure 51.

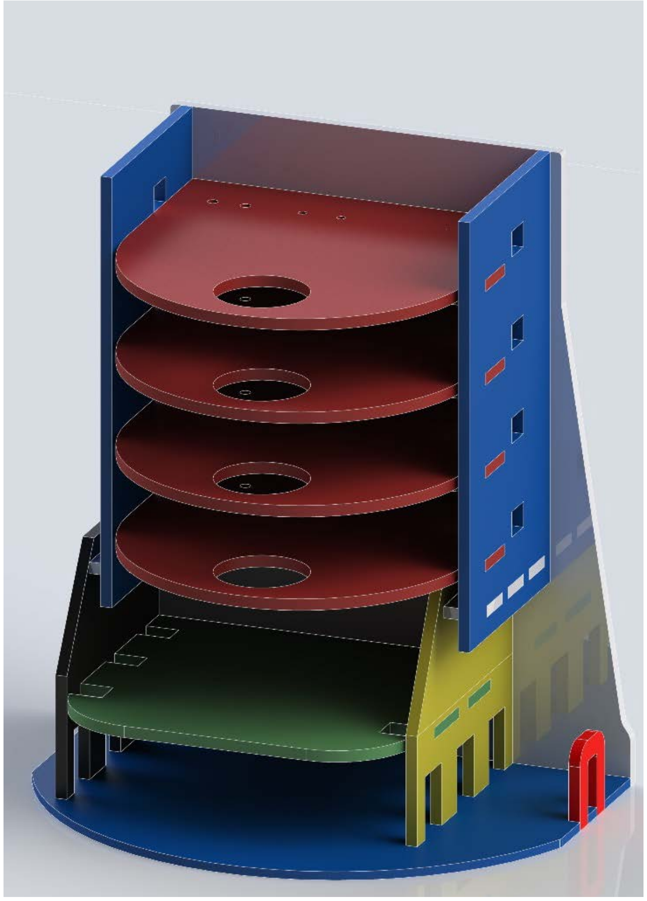


Figure 51 - CAD Model of Backpack Structure

This structure is comprised of a series of five shelves that attach to the back of the structure, two covers that attach to the base, and a small ring holder on the right part of the base. All of these pieces are made out of 0.212" thick acrylic. The two covers on the base are used to hold the three tanks and also attach to the back of the structure for stability. The ring at the right part of the base is supposed to hold the pressure regulator to the base. Unfortunately, due to the positioning of the tubing from the tanks to the T-connectors, the pressure regulator was moved to the outside of the backpack and is no longer attached to the base.

The five shelves attached to the sides and back of the structure are used to hold the electrical board, Arduino board, as well as each set of valves and pressure sensors. The Arduino and electrical board are placed on the top shelf for ease of access and attachment of wires. Each of the four sets of on/off valve, pressure transducer, and blow off valve for each bladder was placed on its own shelf. On the bottom shelf, there are cut-outs on the sides of the shelf close to the end. These are for the tubing connectors between each of the valves. On each of the shelves, there are holes drilled into them to screw in the electrical/Arduino boards and T-connectors between the valves and transducers. On the sides of all of the shelves, there are support pieces to help stabilize and secure all of the shelves. In the sides of these shelves are cut-outs for the tubing on each side. The tubing on the right side goes to the pressure regulator. The tubing on the left side goes straight to the bladders in the socket.

Most of these pieces have slots on the ends of them to be able to slide into all of the other components, so everything fits together securely as one entire unit. This whole structure fits within a backpack donated by one of our group members. Within the backpack, there is the shelving structure in the back, the pressure regulator outside in one of the side pockets, and the battery is located in the front pocket of the backpack. This system is relatively light and easy to transfer between packs if necessary (see Figure 52).

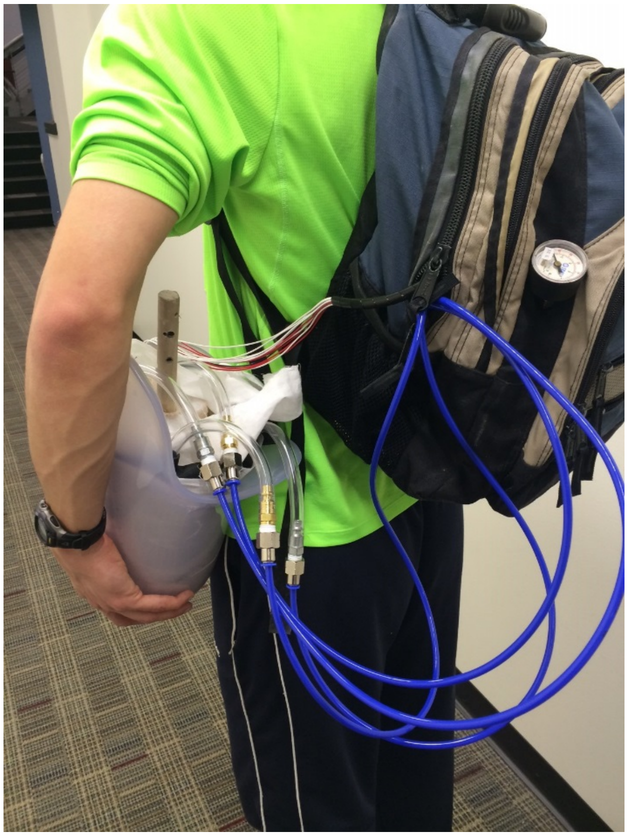


Figure 52 - Socket Connected to Backpack

5.1.3 Air Reservoir

The air reservoir tanks used in our final design come from a pneumatic kit assembled by VEX Robotics (see Figure 53). The tanks themselves come from SMC. We included three of these tanks in our final prototype. The tanks are connected in series with tubing. An end of the first tank leads out to a standard pneumatic fitting which can be attached to most bicycle pumps. One end of the last tank leads to the pressure regulator which controls the airflow coming out of the tanks.

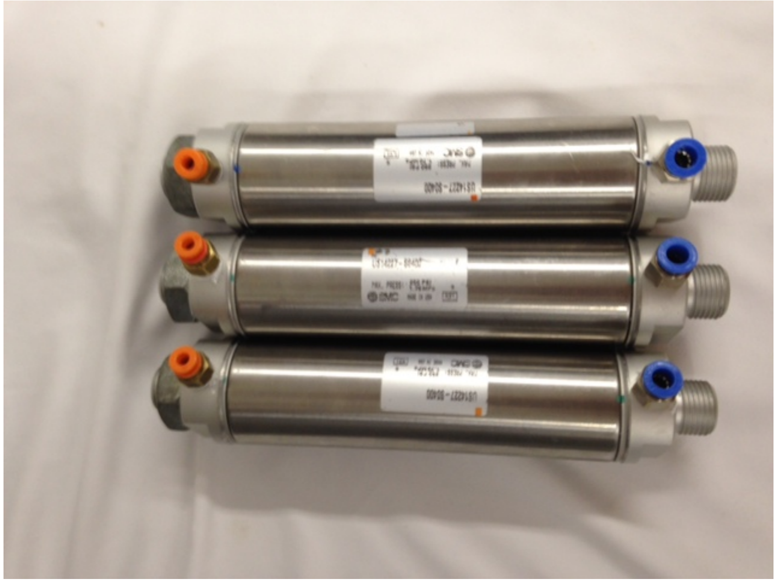


Figure 53 – Air Tanks

Each tank can hold 150mL of air, so including three tanks increased our storage volume to 450ml. Having a larger reservoir will allow more air storage and therefore the tanks will not need to be refilled as frequently.

Each tank can hold a maximum pressure of 250psi. For safety, as well as our lack of an adequate air compressor, we only fill our reservoir to 150psi. Our current method of pumping air into the tank is to use a standard bicycle pump which connects to one of the tanks. The bike pump's gauge only goes this high and the pump is not designed to provide much higher pressure than 150psi. Like increasing the tank volume, providing more pressure into the tanks would significantly increase the amount of air storage and require fewer refills when the socket is being used.

A pressure gauge is also attached with tubing between the tanks so the pressure within the tanks is always known. This gauge sticks out of the backpack for the user to easily read.

5.1.4 Pressure Regulator

The pressure regulator that was used in the final design came from the AIM lab in which we did all of our work. It is a SMC Air Pressure Regulator with Gauge, Model NAR2000 with a Nordston EFD 100 psi Pressure Gage. We have one of these regulators on the side of the backpack. It originally was supposed to be located on the left part of the backpack shelving base, but it was unable to fit there while still receiving the tubing from the other components within the shelving unit. Instead, the pressure regulator is located on a side pocket of the backpack in order to be visible and easily adjusted. A visual of the pressure regulator is shown in Figure 54.

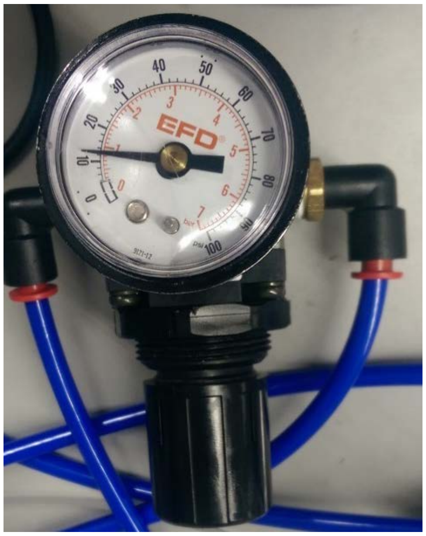


Figure 54 - Pressure Regulator

On the pressure regulator, there are two corner connectors on either side of the center of the regulator. Tubing attaches on the left corner connector and originates from one of the tanks. Tubing attaches on the right corner connector and originates from one of the T-connectors next to the valves. The air enters the regulator from the tank and then the regulator dictates the amount of pressure that leaves the regulator and enters each set of valves. The pressure regulator has been determined to always be set to ten psi, because the max pressure setting (walking) is set to at eight psi.

5.1.5 Valves

Our final prototype incorporates the same on/off valves as in our initial design, as shown in Figure 55. These valves are controlled by our microcontroller and are told to simply open and close to either let air in or prevent air from going though. They also are crucial for deflating the bladders because they allow air to purge out into the atmosphere. This is important for allowing us to maintain specific pressures within each bladder.

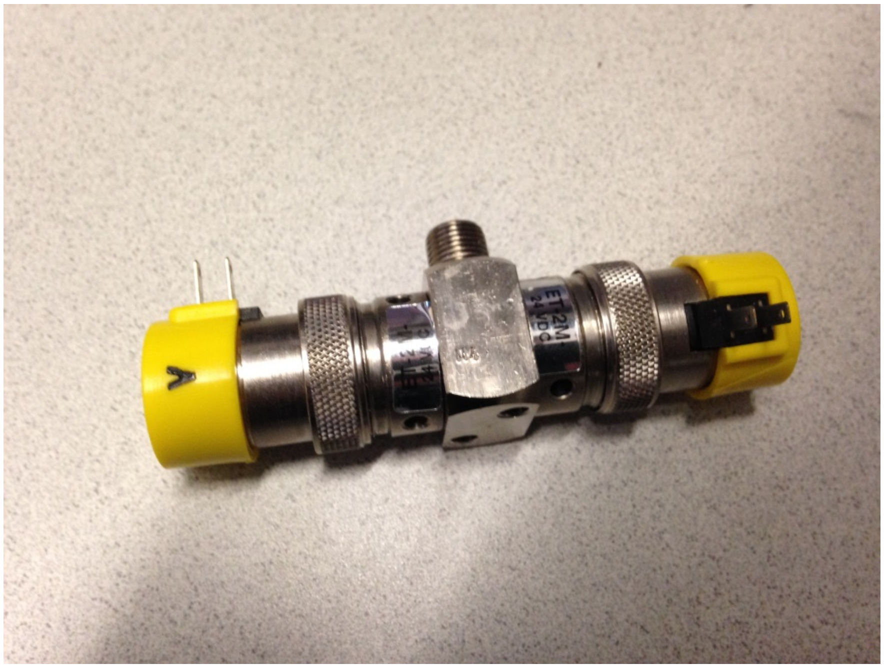


Figure 55 - Clippard 2-way Manifold Mount Solenoid Valve Model ET-2M-24

In addition to these valves, we added blow-off valves to our system. These were essential to have before we performed any testing. The blow-off valves are located in the backpack but are the components right before the bladders. Without these valves there would be a risk of letting too much air into the bladders and inadvertently cutting off the circulation in the leg. These valves, shown in Figure 56, act as a safety so to allow for comfortable, safe use.

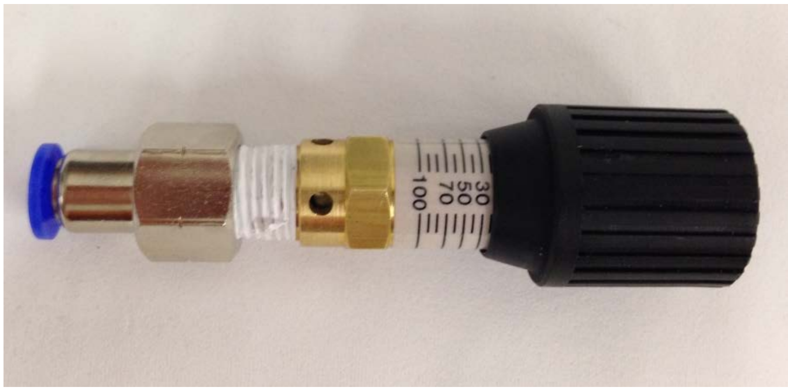


Figure 56 - Blow-off Valve

5.1.6 Bladders and Bladder Holder

One of the most significant changes to our mechanical/pneumatic design was the change in the bladders themselves. We chose to purchase air wedges that are similar to those used in our initial design, only much smaller (see Figure 57). The new bladders are 2.5inches wide and 4.5 inches long, whereas the old ones were 6.46inches by 5.98 inches (manufacturer part # TAJW). This change in dimensions made a big difference in our overall air usage. With the new bladders we are able to pull less air from the tanks which means the user will not need to refill as often.

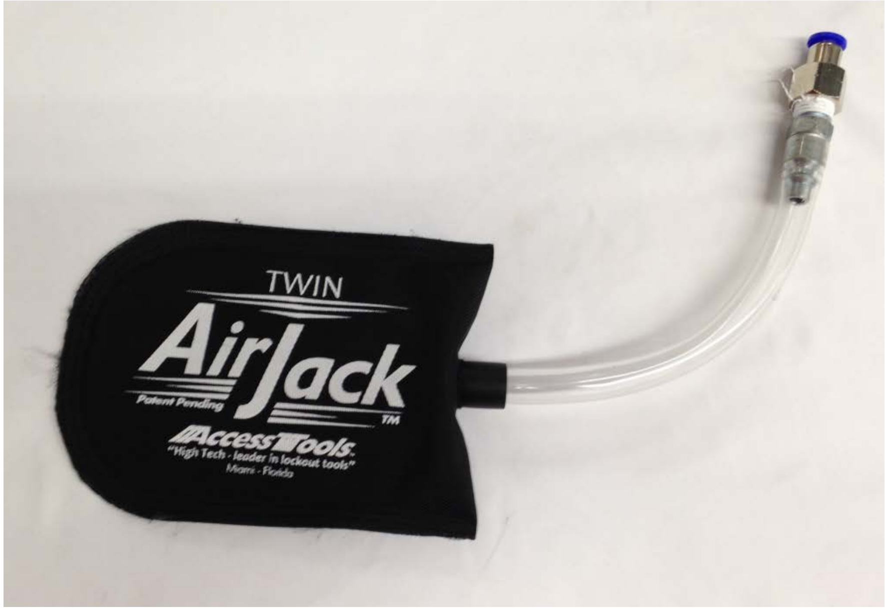


Figure 57 – Air Jack Bladders

These bladders inflate to a maximum pressure of 15 psi which is well above our functional limit of 10 psi. They are also the perfect size such that they are all able to fit on the residual limb sleeve. The previous bladders would have been much too large.

The bladder sleeve for the final prototype, shown in Figure 58, is designed very similarly to that which was used on the testing rig. However because this sleeve is actually designed for use by a trans-femoral amputee, it is stitch where the bottom of the residual limb would be. The sleeve fits nicely around the mold and inside the socket. Pockets holding the bladders were stitched on the outside of the sleeve at the medial, lateral, anterior and posterior locations.

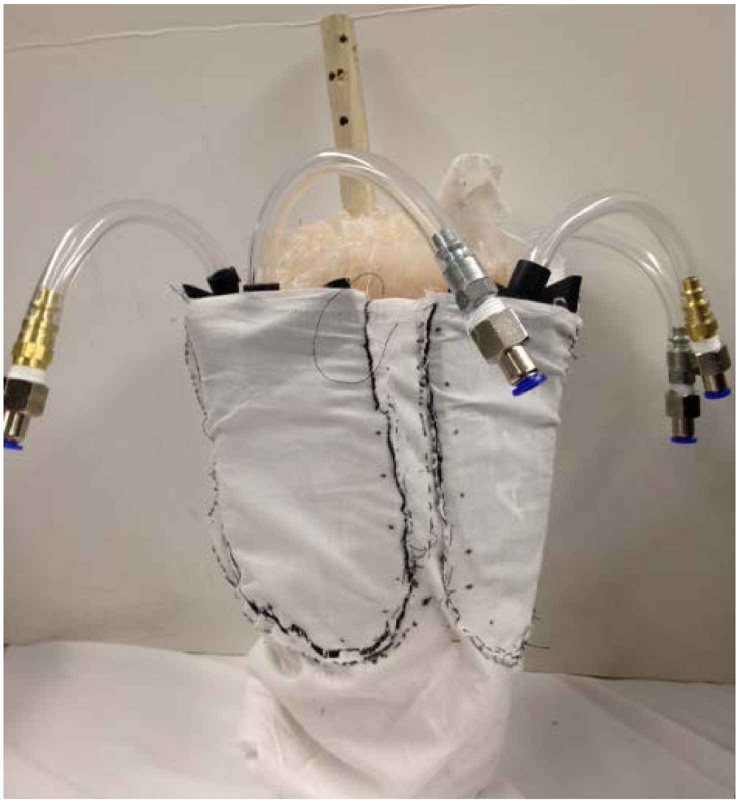


Figure 58 - Bladder Sleeve

This sleeve was also made from a different material than the original design. Initially we thought that a more elastic material would be better suited for our design, but while sewing the testing rig bladder sleeves we realized that elastic fabric was very difficult to handle. It stretched so much that it was difficult to get truly accurate measurements, and sewing it was also very difficult. Therefore we decided to use a less elastic fabric for the final bladder sleeve.

5.1.7 Molded Leg

In preparation for experimentation with the ABB robot arm (discussed in Section 5.3.2), we created a mold of a residual limb to be placed inside the socket.

The first step to creating this mold was to create the outer mold that the material will be poured into. We used papier-mâché to create this outer mold around an able-bodied person's leg.

Once the outer mold was dry it was placed inside the socket. The entire interior was lined with a water resistant plastic. A foam cylinder was then placed inside the socket before the interior molding material was applied. This material, called Dragon Skin® comes in two viscous liquid parts which when combined create a solid rubber mold within a few hours. In order to create a more realistic residual limb we added a skin-colored pigment to our mold before it cured.

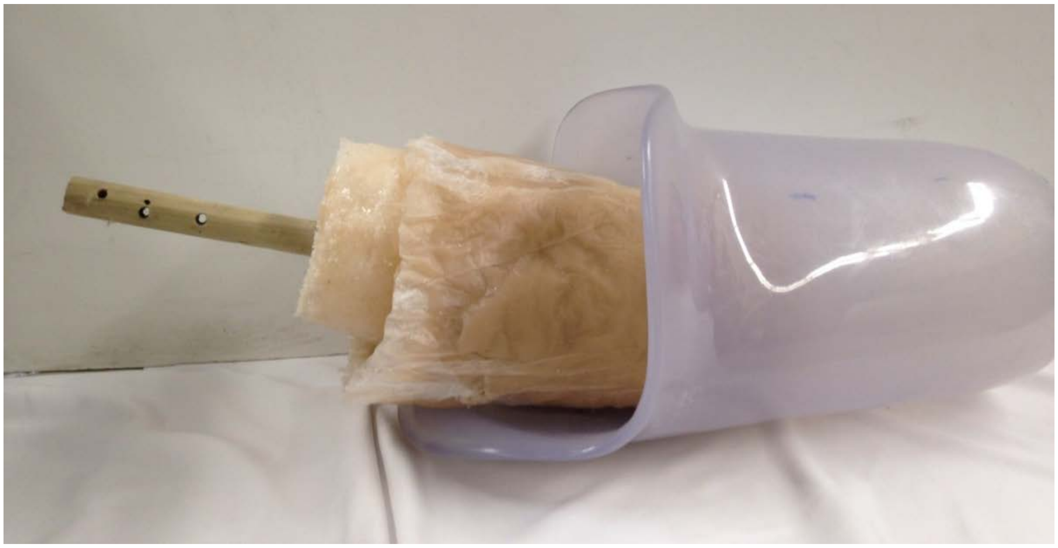


Figure 59 - Residual Limb Mold with Socket

Once the mold was finished curing we detached it from the socket and cut away the plaster and plastic. The result fits into the socket quite well, just enough to allow the bladders to provide a cushion.

5.2 Electrical and Control System

The electrical system is designed the same way as the initial prototype; the system gathers data from the accelerometer and pressure sensors. It is also responsible for powering each of the sensors as well as the microcontroller and pumps. The microcontroller will receive all the sensor data, perform calculations to determine the user's current activity, and adjust the pressure in the bladders to ensure the amputee remains comfortable. This section will describe the sensors used in this design as well as the circuits, power system, and control system.

5.2.1 Sensors and Valves

In our final design, the sensors and valves had only one substantial modification. The pressure sensors and valves that we are using on the final design are the same as our prototype, but our accelerometer breakout board has changed. The pressure sensors are 0-15 PSI stainless steel pressure transducers that are connected in line with each bladder. These are the same transducers described in Section 4.2.2.

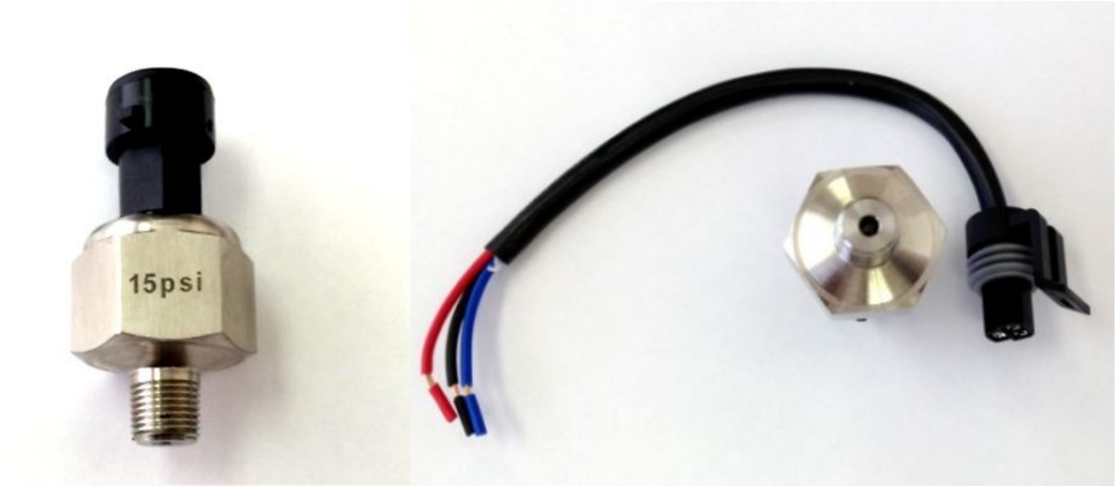


Figure 60 - Pressure Transducer

The valves we used in the final design are also the same as the valves we used in the prototype, described in Chapter 4 of this report. We are using 8 Clippard Minimatic ET-2M-24V valves. They are manifold mounted solenoid valves that are normally closed and will open with a voltage of 24V across the terminals. The valves are described in more detail in Section 4.2.3.

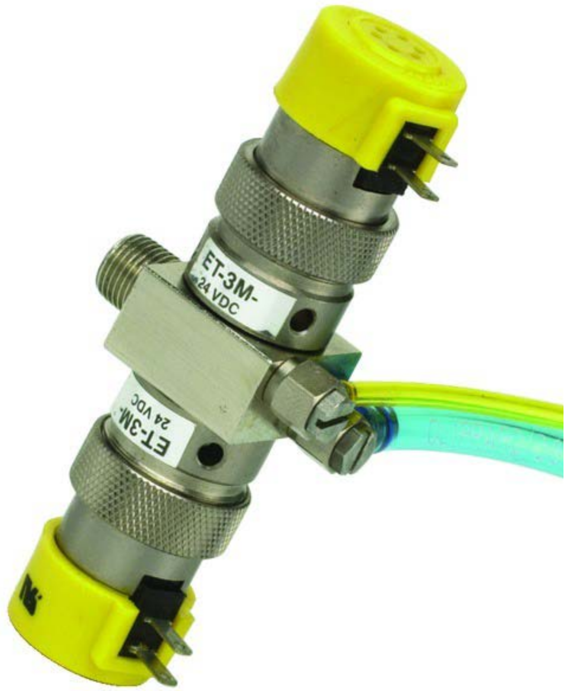


Figure 61 - Clippard Minimatic Valves on Manifold

The main difference in the sensing of the final design is in the accelerometer. Initially, we were using the Analog Devices ADXL362 Triple Axis Accelerometer Breakout Board. This board serialized all three Cartesian planes in order to send the signal to the microprocessor, where it is deserialized. We used a serialized board because our Arduino Uno had a limited number of inputs, so using just one of them for all three planes was beneficial to us. Since we designed the initial prototype, we have changed microcontrollers and are now using the Arduino Mega, as described in Section 5.2.3. This new board has many more input and output pins, which allowed us to use an accelerometer with a pin for each individual plane instead of serializing the data. We decided to use this type of output because it is easier to work with when it comes to signal processing. The new accelerometer board that we chose is the Analog Devices Triple Axis Accelerometer Breakout – ADXL335, shown in Figure 62.

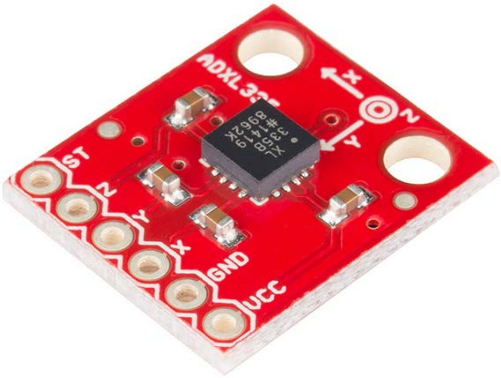


Figure 62 - ADXL335 Triple Axis Accelerometer

5.2.2 Interface Control Board

Since most of the components were moved to the backpack, we had to design a new board that would be on the socket. Not all of the components could move to the backpack because the accelerometer had to stay on the quadriceps of the user and the user had to be able to control the socket. Besides the accelerometer, we brainstormed some ideas about what we wanted to be on this board. These ideas are shown below:

- Two interface switches:
  - A power switch (to turn on the whole system)
  - A “DeflateBladders” switch (to release all the air in the bladders)
- A voltage regulator (to convert the 26V from the battery to 12V to power the Arduino)
- Five indicator Light Emitting Diodes (LED):
  - Power LED (turns on when user switches on corresponding switch)
  - DeflateBladders LED (turns on when user switches on corresponding switch)
  - Sitting LED (turns on when user is sitting)
  - Standing LED (turns on when user is standing)

- o Walking LED (turns on when user is walking)

Before constructing the board, we drew out on paper the schematic and then wired the schematic using Multisim, shown in Figure 63.

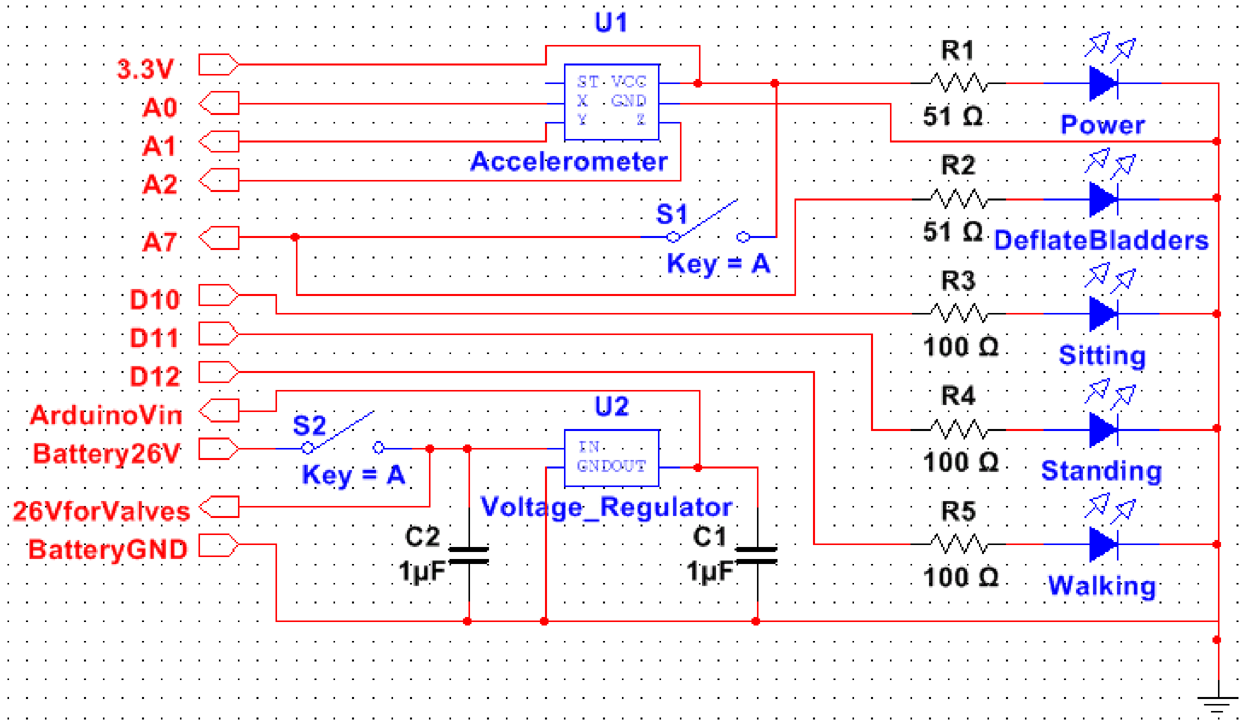


Figure 63 - Interface Control Board Schematic

There are twelve input/output pins connected to this board. Coming from the battery in the backpack is the 26V pin and the GDN pin. Once the power switch is turned on, current flows to the pin that provides 26V for the valves. The voltage regulator converts the 26V down to 12 V that then goes to the Vin pin to power the microcontroller. Also, 1 micro Farad capacitors are used at both the input and output pins of the voltage regulator in order to improve the stability of the system. Three separate digital pins turn on their respective LEDs to indicate sitting, standing, or walking. The 3.3 Volt analog pin from the microcontroller powers the accelerometer. Three analog pins are connected from the accelerometer to the microcontroller that indicate the user's actions from the x, y, and z axis.   Lastly there is an analog pin connected from the user interface board to the microcontroller that tells the microcontroller to deflate all bladders if the pin is high, which is triggered by the “DeflateBladders” switch. A picture of the actual board is shown in Figure 64. The LED indicator lights from right to left designate power, deflate all bladders, sitting, standing, and walking. Figure 65 shows the interface control board enclosed and labeled.

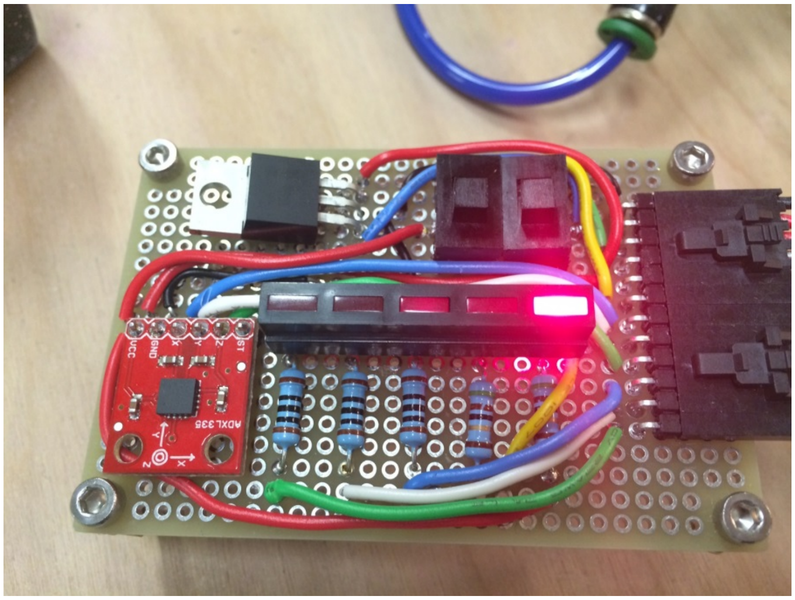


Figure 64 - Interface Control Board

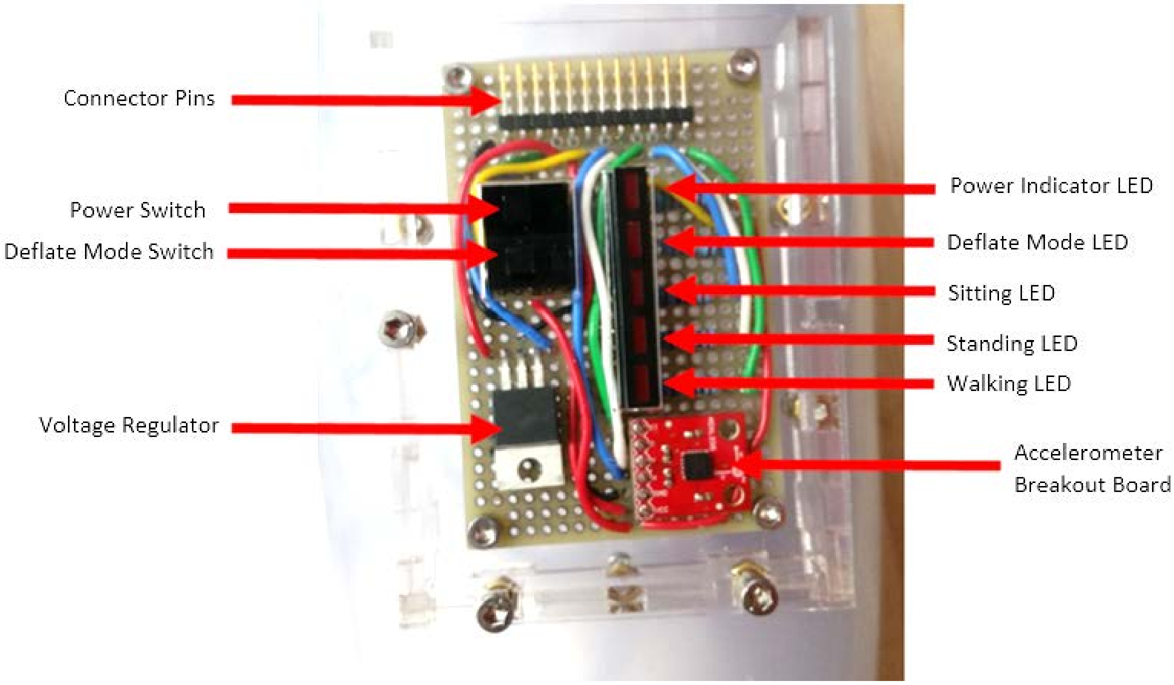


Figure 65 - Interface Control Board Labeled

5.2.3 Microcontroller

As mentioned in Chapter 4, all of our initial testing and experiments used the Arduino Uno. For our final design, we needed a microcontroller with more inputs and outputs to fit the needs of our new desired components such as the new accelerometer and the interface control board. A few basic requirements for our microcontroller were:

- Must be able to supply 3.3 V (to accelerometer) and 5 V (to pressure sensors)
- Must have at least eight analog input pins (for pressure sensors, accelerometer, and interface control board)
- Must have at least 11 digital output pins (valves and interface control board)
- Extra general purpose input/output pins for future work

After investigating microcontrollers, the team decided on the Arduino Mega 2560 (see Figure 66). This made a very smooth transition from our Arduino Uno because each of them uses the same software environment. Also, because the speed and processing capability of the Arduino Uno was enough, we just needed a microcontroller with the same specifications, just with more input and output pins. Just like the Arduino Uno, the Arduino Mega 2560 has a CPU speed of 16MHz, a maximum sampling rate of 10 kHz, and the resolution of the ADC is 10 bits, in other words returning digital values between 0 and 1023. However, the Arduino Mega 2560 uses the ATmega560 processor instead of the ATmega328 processor. This processor allows the Arduino Mega 2560 to have 16 analog inputs and 54 digital input/output pins, certainly meeting our requirements.

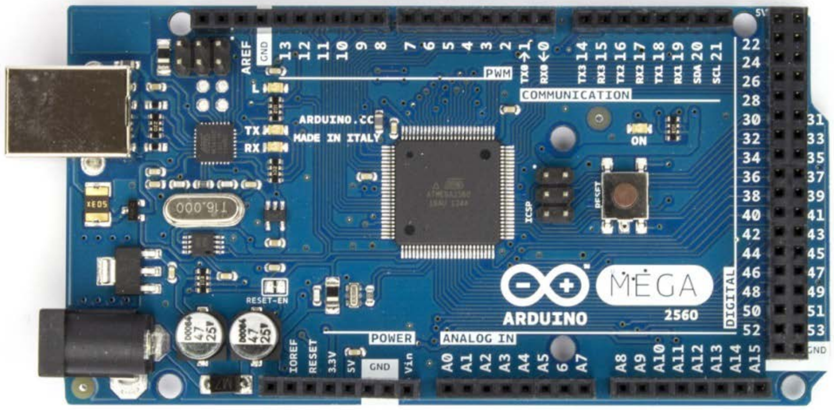


Figure 66 - Arduino Mega 2560

5.2.4 Power Supply

As stated in Section 4.2.6, we chose to use a 25.9-volt, rechargeable, lithium-polymer battery pack to power the system. This battery is small and light enough to fit in the backpack without risking injury to the user. Once we finalized our components, we needed to calculate how long the battery could last on a single charge. To do this, we added together the required currents for each electrical component and the maximum amount of charge (in milliamp hours) to estimate the battery life of the system in the worst case scenario. Table 17 shows the current and voltage required for each electrical component of the final system. It also shows the power required in milliwatts by multiplying the current by the voltage:

Table 17 - Component Current Consumption

Component	**Required current (mA)**	**Required voltage (V)**	**Required Power (mW)**	**Number of components**
Pressure sensor	3	5	15	4
Valve	28	26	728	8
Accelerometer	0.350	3.3	1.155	1
Digital Output	40	5	200	11
3.3 Volt output	50	3.3	165	1
5 Volt output	50	5	250	1

The chosen battery has a maximum charge of 3650 milliamp-hours and energy of 95.4 watt-hours. Due to the control system, a maximum of only four valves would be operating at once while the prosthetic is in use. Also, only up to two of the LEDS can be lit simultaneously, so even if the control system is using the most current possible, only six digital outputs, four valves, the 3.3-volt source, and two LEDs will require current. In the following formula I_p, I_v, I_a, I_{out}, $I_{3.3}$, I_5 are the currents for a single pressure sensor, valve, accelerometer, digital output, 3.3-volt source, and 5-volt source. By dividing the maximum current usage *f*rom the battery's charge, B_c, the result is the worst-case lifespan of the battery, t_i, based on the charge alone.

$$4I_p + 4I_v + I_a + 6I_{out} + I_{3.3} + I_5 =$$

$$(4\ sensors)\left(\frac{3\ mA}{sensors}\right) + (4\ valves)\left(\frac{28\ mA}{valve}\right) + 0.350\ mA + (6\ pins)\left(40\frac{mA}{pins}\right) + 50\ mA$$

$$+ 50\ mA = 464.35\ mA = I_{total}$$

$$t_i = \frac{B_c}{I_{total}} = \frac{3650\ mA \cdot hr}{464.35\ mA} = 7.8\ hr$$

Based on these calculations, the system could run continuously on the battery for at least 7.8 hours at full charge. The user would certainly be able to perform day-to-day activities such as going for walks and running errands without a concern of the battery dying during operation. However, rechargeable batteries tend to lose maximum capacity over time, so if the user is going to be away from home most of the day, it is recommended that they bring the charger or a spare battery, especially if they have charged the battery several times prior.

However, the battery will still run out if it loses energy before losing charge. So we must perform similar calculations using the power of each component to see how long the battery will last in terms of energy (t_p). For this calculation, the battery's energy (B_p) is divided by the maximum power that could be used at any time. In the following equations, P_p, P_v, P_a, P_{out}, $P_{3.3}$, and P_5 represent the power of the pressure sensors, valves, accelerometer, 3.3-volt output, and 5-volt output respectively.

$$4P_p + 4P_v + P_a + 6P_{out} + P_{3.3} + P_5 =$$

$$(4\ sensors)\left(\frac{15\ mW}{sensors}\right) + (4\ valves)\left(\frac{728\ mW}{valve}\right) + 1.155\ mW + (6\ pins)\left(200\frac{mW}{pins}\right)$$

$$+ 165\ mW + 250\ mW = 4588.155\ mW = 4.588\ W = P_{total}$$

$$t_p = \frac{B_p}{P_{total}} = \frac{94.5\ W \cdot hr}{4.588\ W} = 20.6\ hr$$

These calculations show that, if battery charge was not a factor, the system would have enough power to last 20.6 hours. Since the power would last longer than the charge, the battery charge will run out first, giving the system a worst-case lifespan of 7.8 hours.

5.2.5 Whole System Circuit Diagram

The diagram shown in Figure 67 is the electrical schematic for our final design. This design utilizes an Arduino Mega 2560 for all of the signal processing and communication within the system.

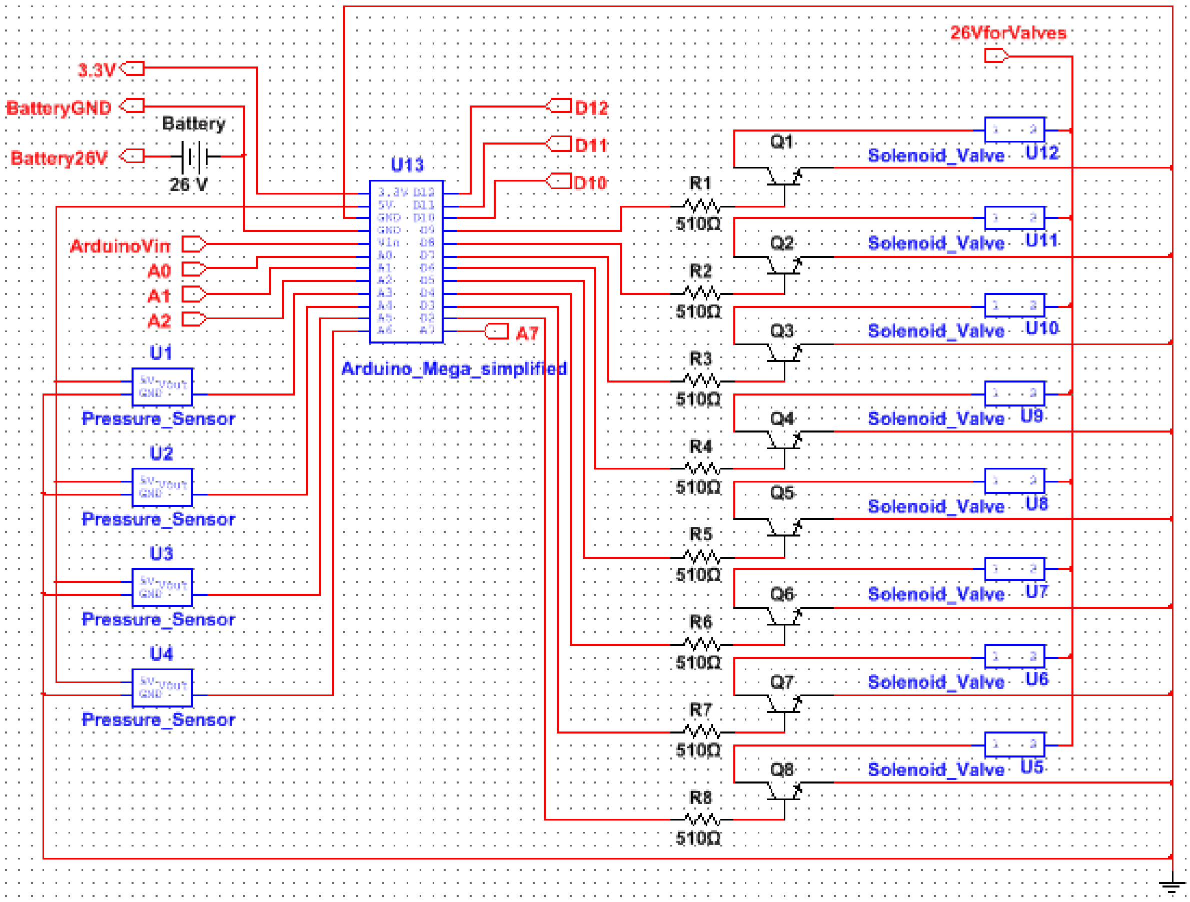


Figure 67 - Whole System Circuit Diagram

The Arduino Mega powers the accelerometer with 3.3 Volts and powers the four pressure sensors with 5 Volts. The 26 Volts from the battery goes to the interface control board where the switch is. When the power switch is turned on, the 26 Volts gets converted to 12 Volts through the regulator which is then connected to the Vin pin of the Arduino Mega, while 26V from the battery powers the valves. The ground of the battery is connected to the Arduino Mega while the other ground pin on the Mega is connected to the rest of the main board. The x, y, and z-axis of the accelerometer is connected to the analog input pins of A0, A1, and A2, respectively. The outputs of the four pressure sensors are connected to analog input pins A3, A4, A5, and A6.

When the “DeflateBladders” switch is turned on on the interface control board, a 3.3 Volt signal is sent to the Arduino Mega analog input pin A7, which triggers the microcontroller to deflate all four of the bladders. Digital output pins trigger the eight valves through pins D2 through D9. Digital output pins D10, D11, and D12 light up the three LEDs on the interface control board when the system detects if the user is sitting, standing, or walking, respectively. The 12 red connectors, such as “26VforValves”, “BatteryGND”, etc., are all connected to the connectors on the interface control board that have the same name. These connectors on the control board are shown in Figure 59 in section 5.2.3 Microcontroller.

5.2.6 Control System

The final control system is very similar to the initial design, for it has three states: read sensor data, determine the activity, and control bladder pressure. However, there were some significant alterations done to the final design that improved the system as a whole and made the device more comfortable for the user.

The program starts by reading the accelerometer data at a rate of 100 Hz; twice as fast as the initial control algorithm. The reason for the increased reading speed is due to the Arduino Mega being capable of reading analog values at a rate of 10 kHz. However, if these values are read too quickly, it would be more difficult to determine the user’s state, as his or her leg would not have time for significant movement, regardless of the current activity. The readings for the x-axis rotations are converted into angles relative to the torso, using the same formulas as the initial system mentioned in Section 4.2.7. However, the final system does not include external force sensors, so these were removed from the design.

Once the system has stored the initial 100 angle values (which takes approximately one second), it uses their mean and standard deviation to determine the activity (see Table 9 in Section 4.2.7). If the program cannot determine the activity from the values, the Arduino will continue to read the accelerometer data until it is able to do so. When it is successful, it will set the "goal" pressures, or bladder pressures that are most comfortable for the given activity, and the LEDs accordingly. The first LED indicates sitting, the second standing, and the third walking. If the first two LEDs are lit, this indicates a transition between sitting and standing.

After the system recognizes the user's state, it attempts to set the four bladders to their respective goal pressures. The goal pressure for walking were found from our force plate tests and are shown in Appendix H. Since these experiments gave negative values for sitting and standing, we set the remaining states arbitrarily (all bladders 1 for sitting, 2 for standing, and 2.5 for transition). Since the ADC converts voltages from 0 to 5 volts to a 0 to 1023 reading and the voltage output of the pressure sensor ranged from 0.5 to 4.5 volts for 0 (P_{min}) to 15 (P_{max}) psi, we used the following formulas to convert the desired psi (P) into a goal pressure reading (R):

$$R_{min} = 0.5 * \frac{1024}{5} = 102.3$$

$$R_{max} = 4.5 * \frac{1024}{5} = 921.6$$

$$\frac{R_{max} - R_{min}}{P_{max} - P_{\min}} = \frac{921.6 - 102.4}{15 - 0} = 54.613 = a$$

$$R = aP + b = 54.613P + 102.4$$

As before, the program uses a bang-bang controller, which lets air in when the pressure is too low or out if too high. This controller runs at 100 Hz, as that is the maximum frequency in which the valves are able to change. The controller runs on all bladders simultaneously, ensuring that the socket fit avoids uneven pressures during operation. In the initial design, the system would remain in the control loop until all bladder pressures were set correctly. However, the team found a flaw in this system. If the program determined the activity incorrectly, it would set the bladders to the wrong pressure and not correct the mistake until after they were set. If the bladders were set to the wrong pressure, the user could become uncomfortable and risk injury. To avoid this scenario, the Arduino will continue reading the accelerometer during the control loop. When the system has stored twenty new angle values, it will briefly exit the control loop and determine the activity again using all the values in the array. Then it will reenter the control algorithm and continue adjusting the bladders using goal pressures based on this newly determined activity. Since the program is single-threaded, it is unable to assess the activity and control the bladders simultaneously. Fortunately, it is able to find the activity quickly enough that the brief exit has minimal effect on the pressure control. If the system is unable to determine the activity after interrupting the control loop, then it will assume the previous activity is still ongoing and reenter the loop without changing the goal pressures. This exit leads to another reason for having the accelerometer read at only 100 Hz: the faster the accelerometer is read, the sooner it would exit the control loop to identify the activity again. It was crucial that the system spent enough time controlling the bladders so they would reach the proper pressures quickly.

Another issue with the initial algorithm was that the system would “flip” rapidly between switching valves on and off, trying to adjust the pressure to a very precise pressure. This would occasionally slow the system down. To solve this problem, we added a deadband, which represents the maximum difference between the current and desired pressure reading required for the controls to believe a bladder is at the correct pressure. Giving the control system a small range of desired values instead of a precise one allowed the system to adjust much more quickly. The small drawback to this is that the pressure may not be as accurate. However, using a mini tolerance of four, the pressure error would only be 0.06 psi at most (*4*15/1024*), which is not a significant enough difference to be noticed by the user.

If the emergency “DeflateBladders” switch is flipped, the control algorithm will not run. Instead, the system will immediately turn on all the valves for releasing air and turn off the remaining ones. The program will continue to determine the activity, but the control algorithm will not run until the switch is flipped back to its return position.

Once all the bladders are at the appropriate pressure, the system will exit the control loop and determine the activity again, therefore continuously alternating between the second and third states of activity determination and pressure control. The final system is shown in the flowchart in Figure 68, in which blue blocks are the initial accelerometer readings, gold are activity determination, and gray are the control loop.

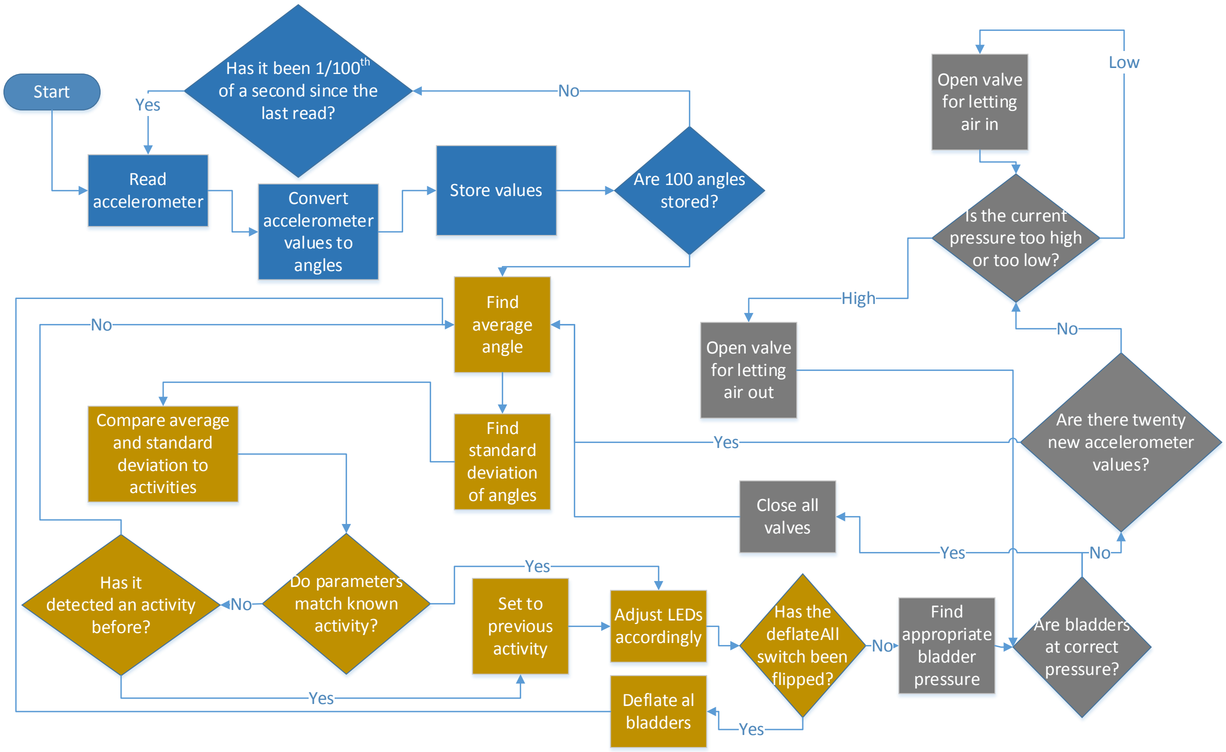

Start
Yes
Has it been 1/100th of a second since the last read?
No
Read accelerometer
Convert accelerometer values to angles
Store values
Are 100 angles stored?
Find average angle
No
Compare average and standard deviation to activities
Find standard deviation of angles
Has it detected an activity before?
No
Do parameters match known activity?
Yes
Set to previous activity
Adjust LEDs accordingly
Has the deflateAll switch been flipped?
Yes
Deflate al bladders
Yes
No
Find appropriate bladder pressure
Are bladders at correct pressure?
No
Yes
Close all valves
Are there twenty new accelerometer values?
Yes
No
Open valve for letting air out
High
Is the current pressure too high or too low?
Low
Open valve for letting air in

Figure 68 - Final Control Algorithm

5.3 Testing and Analysis

After assembling the final prototype, we modeled the air reservoir life and we performed a very similar experiment as we did in section 4.3.1. However, this time we used a robot to simulate the activities of sitting, standing, and walking with the socket instead of wearing the testing rig.

5.3.1 Reservoir Life Analysis

The purpose of this mathematical analysis is to test how long the system's air reservoir tanks will last before having to refill. To measure the lifespan of air within these reservoirs, we chose to quantify it based on one typical "cycle" of state changes; one that may be common is a person's daily life. One cycle is equal to the progression of state changes as shown below.

Sit → Stand → Walk → Stand → Sit

Because the forces at the stump-socket interface vary depending on what state the person is in, each bladder will inflate differently for each. The amount of air needed from the reservoirs was calculated for each state change. These values were totaled and then divided into the total amount of air initially in the tank in order to determine the number of cycles one could endure before having to refill the reservoir.

It is important to note that the amount of air in the bladders and therefore the amount of air that is drawn from the reservoirs would vary with different weight, gait, and posture. For the purpose of this experiment, we chose to only use the data from our heaviest subject to reduce variability and provide the worst-case scenario our data could produce.

Before conducting this experiment, we performed some mathematical calculations in order to determine the number of cycles required to exhaust the reservoirs. To do this, we used the Ideal Gas Law.

$$PV = nRT$$

The Ideal Gas Law could be applied to our system because air behaves enough like an ideal gas at room temperature.

The first step in the calculation is to determine how much air can be held in the reservoirs as a molar quantity, N_R. The equation is then rearranged to

$$N_R = \frac{PV}{RT}$$

Where P is equal to the reservoir pressure, V is the reservoir volume, R is the Gas Constant, and T is temperature. Table 18 shows the values used for each of these variables.

Table 18 - Reservoir Values for Ideal Gas Equation

Variable	**Value with practical units**	**Value with converted units**
P	150 psi	150psi
V	450ml	0.051892ft^3
R	10.73159 ft^{3*}psi/R*lb-mol	10.73159 ft^{3*}psi/R*lb-mol
T	25 degrees C	536.4R

The result is an N_R value of 4.14x10^{-4} lb-mol. A lb-mol is simply a unit signifying the amount of air.

Next we calculated the molar value for the amount of air in all bladders, n, when set to the appropriate pressure for each bladder in each state. The values used for each of these variables are listed in Table 19.

Table 19 - Bladder Values for Ideal Gas Equation

Variable	**Value**	**Units**
P_b	Depends on bladder position	Psi
V_b	~44in^3	0.0255 ft^3
R	10.73159 ft^{3*}psi/R*lb-mol	10.73159 ft^{3*}psi/R*lb-mol
T	25 degrees C	536.4R

Table 20 shows reasonable pressure values for each bladder position in each state. The Sitting and standing values were determined based on simple experimentation with our testing rig. 2psi is a comfortable pressure that provides enough fit on the leg without constricting at all. There are low loads on the leg at these states, and therefore the socket simply needs to fit well enough to not fall off. Notice how the posterior value for sitting is slightly higher at 4psi. This is to provide some pressure relief for the amputee while the posterior section of their residual limb is pressing down on his/her seat. The walking value was determined experimentally from our heaviest test subject. See the section about the force plate testing (4.1.1) to see how these values were found.

Table 20 - Pressure Values for Each State in Each Location

	Posterior	Anterior	Medial	Lateral
Sitting	4psi	2psi	2psi	2psi
Standing	2psi	2psi	2psi	2psi
Walking	8.21psi	8.21psi	6.71psi	6.71psi

The following equations were used to determine the amount of air in each bladder in each state

$$n_{post} = \frac{P_{post}V_b}{RT}$$

$$n_{ant} = \frac{P_{ant}V_b}{RT}$$

$$n_{med} = \frac{P_{med}V_b}{RT}$$

$$n_{lat} = \frac{P_{lat}V_b}{RT}$$

Since $\frac{V_b}{RT}$ is a constant value, we can plug in the values of each variable to get 4.41x10^{-6} lb-mol/psi.

The next step was to calculate the amount of air lost from the reservoir during each transition.

<u>Sitting (Turning the socket on)</u>

This value of air lost from the tanks is simply the value of air needed for sitting position. The calculation is shown below.

$$n_{post} = (4psi)\left(4.41\text{x}10^{-6}\,\frac{\text{lb}-\text{mol}}{psi}\right) = 1.76x10^{-5}\,lb - mol$$

$$n_{ant} + n_{lat} + n_{med} = 3 * (2psi)\left(4.41\text{x}10^{-6}\,\frac{\text{lb} - \text{mol}}{psi}\right) = 5.29x10^{-5}\,lb - mol$$

$$Total\ air\ tank\ expenditure\ for\ sitting = n_{post} + (n_{ant} + n_{lat} + n_{med})$$

$$= 7.05x10^{-5}\,lb - mol$$

Sitting-Standing

To stand, only the posterior bladder changes from 4psi to 2psi. Since no air is being added to the system, the tank does not need to supply any air. In this pressure release however, 8.82x10^{-6} lb-mol are lost.

Standing-Walking

To calculate the amount of air loss in the tank during the transition from standing to walking, we simply subtracted the amount of air needed for walking by the value for standing. The total amount of air that the reservoir loses is the sum of all of these differences.

$$n_{post-walk} - n_{post-stand} = n_{post-loss}$$

$$n_{ant-walk} - n_{ant-stand} = n_{ant-loss}$$

$$n_{med-walk} - n_{med-stand} = n_{med-loss}$$

$$n_{lat-walk} - n_{lat-stand} = n_{lat-loss}$$

$$Total\ air\ loss\ from\ reservoir = \Sigma n_{loss}$$

The result for the total air expenditure from the reservoir is 9.64x10^{-5} lb-mol.

Walking-Standing

Air is released back to standing pressures. The reservoir does not lose any air.

Standing-Sitting

The tank will supply another 8.82x10^{-6}lb-mol to the posterior bladder (4psi) to return to sitting position.

All of the losses in each transition are then added together to get total air, $N_{L,}$ loss during one cycle of state changes.

$$N_L = 7.05x10^{-5} \ + 9.64\text{x}10^{-5} + 8.82\text{x}10^{-6} = 1.757x10^{-4}lb - mol$$

Finally, in order to find the number of times N_R lb-mol will provide N_L lb-mol to the system, we simply divide them.

$$\# \; of \; refills = \frac{N_R}{N_L} = \frac{4.14\text{x}10^{-4}}{1.757x10^{-4}} = 2.36cycles$$

The result is 2.36 cycles before needing to refill the reservoir. This value is acceptable for a person who does not move around very much throughout the day, but for people who sit, stand, and walk around rather frequently this number will cause inconveniences. They will have to refill their tanks very often in order to sustain a working socket.

One can easily and drastically improve the life of the reservoirs by increasing the volume of the tanks, and/or the pressure within them. We were limited with only 150psi of pressure and 450mL of volume. One can also purchase an electronically controllable air compressor that recognizes when the tanks are empty and automatically refills them. This kind of compressor, however, was very expensive and did not fit into our budget. This is perhaps a possible addition to the design for future iterations of this project.

5.3.2 Detecting the Correct State

This section includes an experiment with the ABB Robot, as seen in Figure 69. We will be performing a very similar experiment as we did in Section 4.3.1, except instead of subjects actually sitting, standing and walking, the ABB Robot will be simulating these actions.

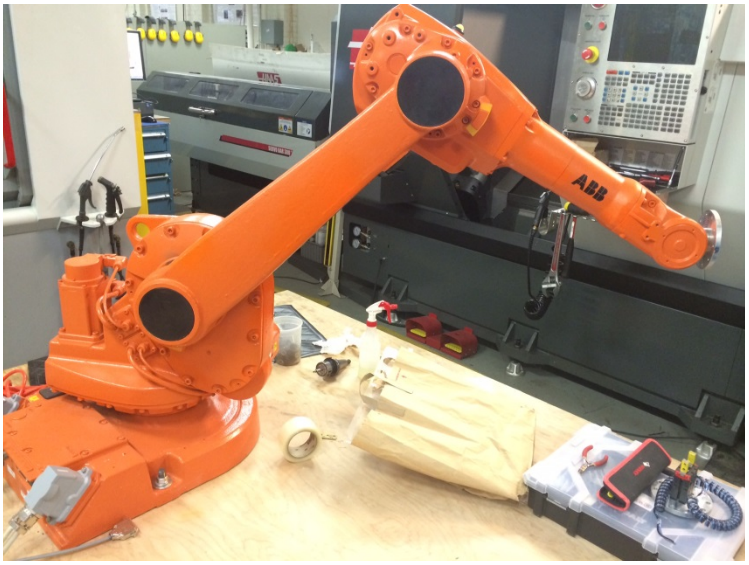


Figure 69 - ABB Robot

Materials

- Final prototype
- ABB Robot

Goal: To find how quickly and accurately the control system can determine the user's current activity.

Methodology

The team first made sure the molded leg was inside the socket and secure. Then the team attached the final prototype to the ABB Robot. After this, the ABB Robot simulated sitting for ten seconds (see Figure 70), standing for ten seconds (see Figure 71), and walking for ten seconds (see Figure 72).

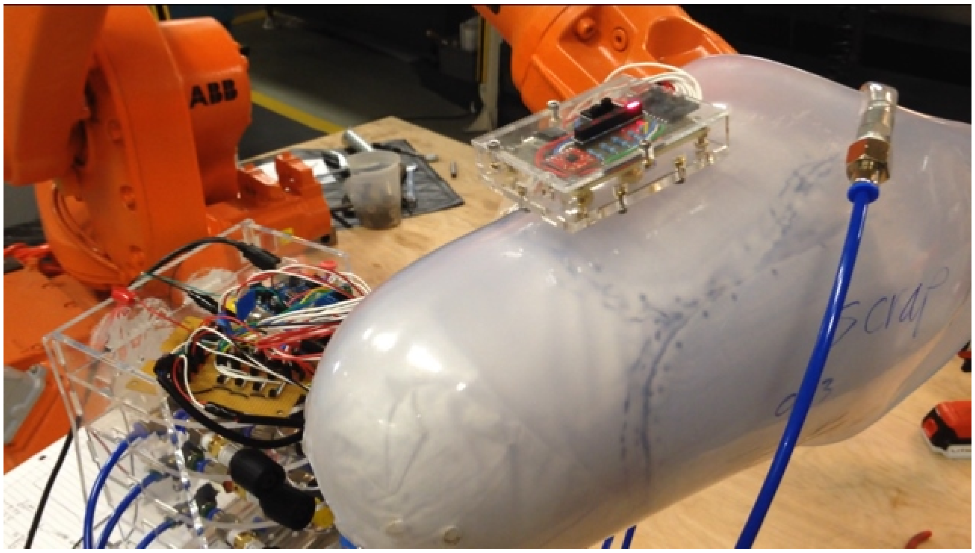


Figure 70 - ABB Robot Simulating Sitting

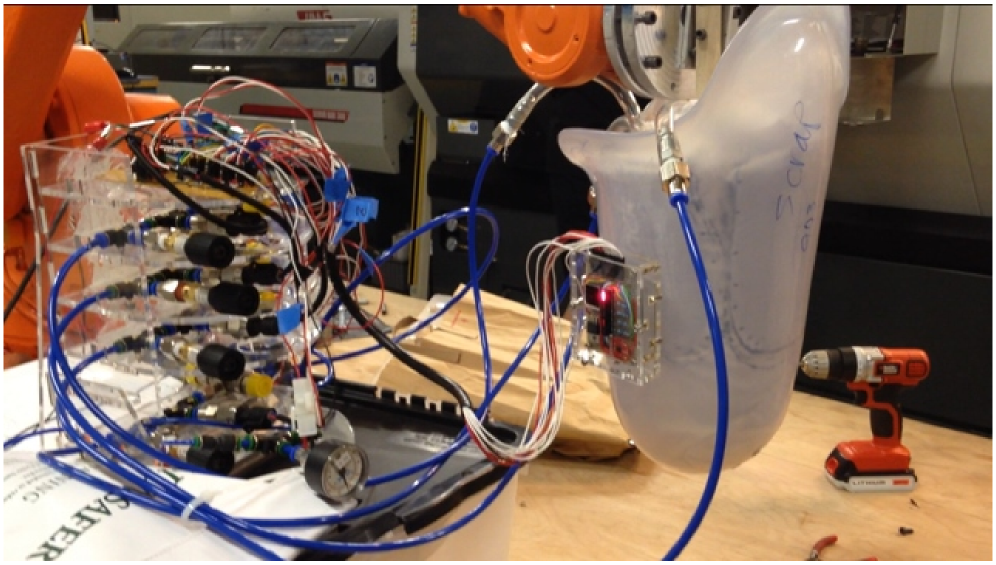


Figure 71 - ABB Robot Simulating Standing

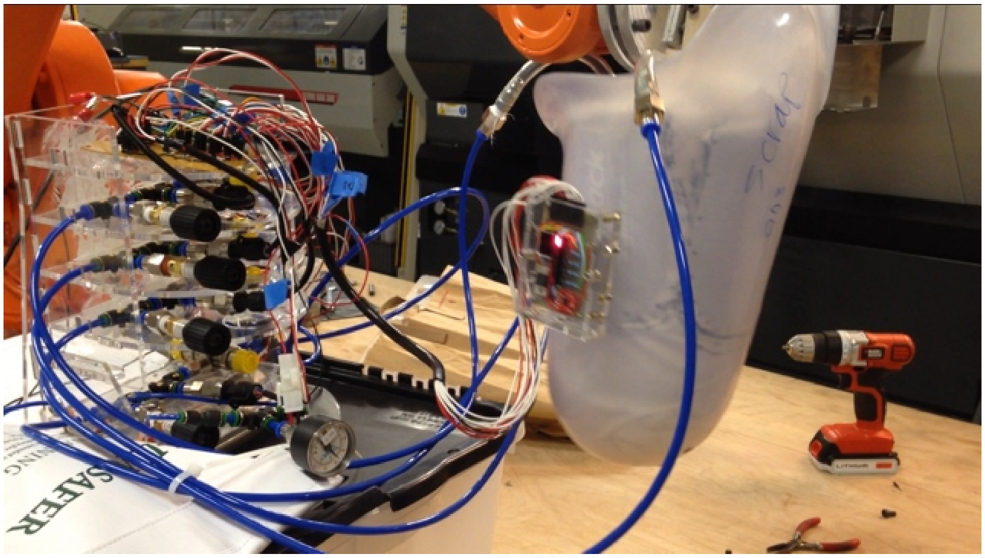


Figure 72 - ABB Robot Simulating Walking

We recorded data visually using iPhone cameras and digitally using a serialized output to an excel sheet which displays the user’s current state. This was then repeated for another trial. This same experiment was then completed another day except with three trials.

Results

With the excel sheet of the user’s states, we analyzed the data using MATLAB. We imported the data and then graphed it, as seen in Figure 73. The x-axis is in sample points, which were taken at a rate of 5 Hz (five samples per second). The y-axis represents the user’s state: 1, 2, 3, and 4 which represents sitting, transitioning, standing, and walking, respectively.

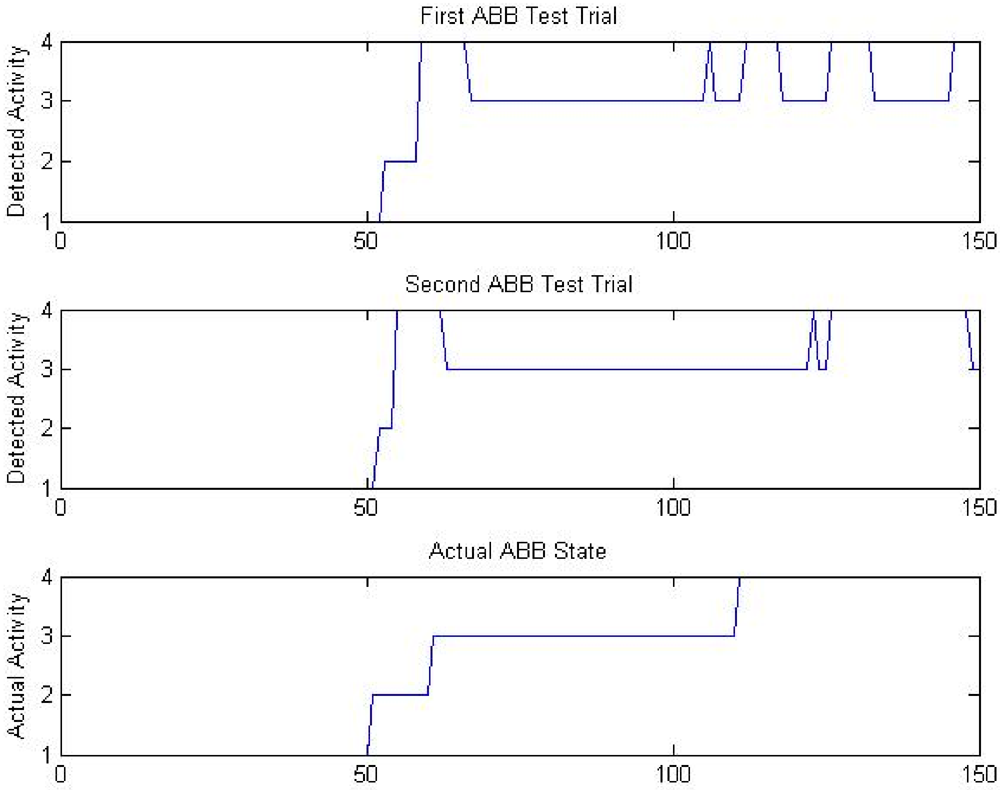


Figure 73 - ABB Robot Simulation

The first ten seconds (approximately 50 data points) represent when the robot was simulating standing (detected activity #1), which the microcontroller accurately determined in both experiments. The system then correctly detected a transition between sitting and standing, and then detected standing (activity #3) for the next 10 seconds or so which was correct. When the robot simulated walking, the system did correctly pick up walking, but only about half the time. The other half of the time the system picked up standing instead. This is most likely due to the fact that the robot was moving slowly and simulated walking at a very slow pace, and the system thought the socket was standing instead of walking. Also, the robot's movements were not as smooth as someone walking. It would sometimes freeze for a split second while simulating walking, which is another reason the system could have picked up standing instead of walking. This 30 second experiment was done twice in which each of the graphs are shown in Figure 73. Since the interface board was not attached to the socket during this test and the robot was unable to simulate walking accurately, the team decided to perform this experiment again. This time, the interface board was attached to the socket and the robot moved slightly faster when it simulated walking. The results are shown in Figure 74.

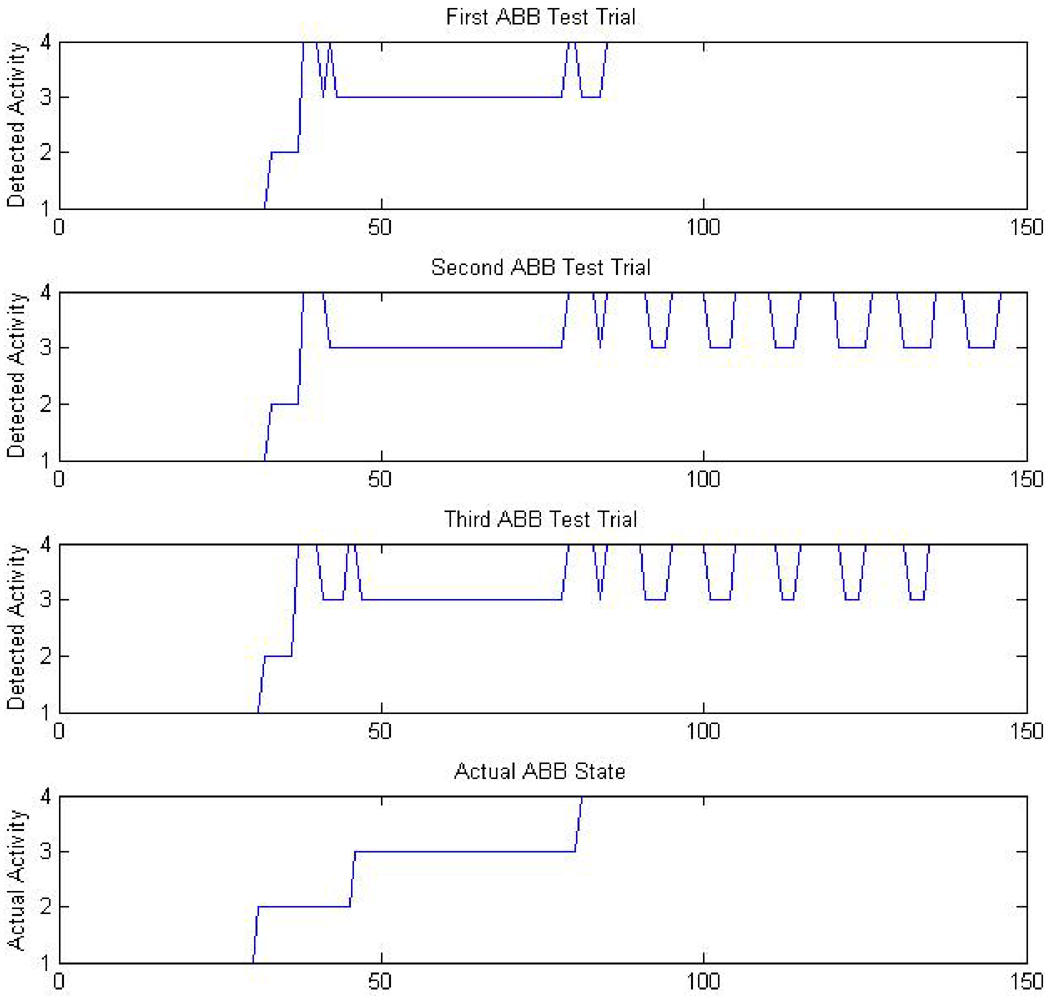


Figure 74 - States for Second ABB Robot Simulation

For this test, the group did not enforce the ten seconds of each state, allowing the timing to vary slightly. The robot was in the sitting position for seven seconds, took about three seconds to transition to standing, stood for five seconds, and then walked for about fifteen seconds. Based on the actual state of the robot, shown in Figure 74, the program was capable of accurately determining the robot's state with the same consistency as the initial state detection tests mentioned in Section 4.3.1. The exception, again, was walking. The robot was more accurate in performing the walking state than the previous ABB test, though the brief stops still occurred, creating the temporary state determination that the user was standing.

6. Discussion and Recommendations

Overall, this project was a success. We successfully created a functional adaptive robotic socket prototype for lower limb prosthetics that:

- Senses if the user is sitting, standing or walking through the use of an accelerometer
- Lights up the appropriate LED on the interface control board for the current state
- Fills up air-filled bladders to their appropriate pressures depending on the state
- Is portable and comfortable
- Has an on/off switch and a “DeflateBladders” switch.

This is only the first iteration of what could eventually be a marketable product that could help thousands of amputees. In order to ensure the success of future work on this project, we have created a list of recommendations that we believe could improve the design.

1. **Read more user states than just sitting, standing, and walking.** Include more states that are common in everyday life such as jogging, walking up and down stairs, and walking sideways. For these states, one would also need to come up with a way to determine appropriate pressure values for each bladder in each state.
2. **Include an air compressor as part of the system or get a tank with larger volume and/or pressure.** Currently the user needs to manually inflate the tanks every 2-3 times they get up to walk around, and for some people this is far too few. The user also needs to currently inflate the bladders with a bike pump, which is cumbersome to carry around. We did our best considering our budget, but with next year’s budget it may be worth it to buy a more expensive miniature air compressor that can be electronically controlled and can fit into the actual system.

3. **Build a system that is safe to test on actual amputees.** One of the most difficult challenges of our project was to test a prototype designed for amputees without actually testing on them. Because this is such an early stage of the project, it would have been unethical to do so without knowing that it is safe for them to use. There are also several factors that are necessary for socket-fitting that would have made clinical testing very difficult. The students to work on this project in the future should prepare the system for clinical testing.
4. **Add more bladders to cover more of the socket's interior.** Currently there are only four bladders on the interior of the socket which cover a large surface area and are sufficient in alleviating some of the internal stresses experienced on the residual limb. The rest of the socket is lined with a layer of foam which should still relieve some stress. However, the brim of the socket is also a location where high stress exists, so it may be beneficial to further investigate that location to determine if bladders are needed there.

The team believes that they have successfully created a prototype that shows much promise for the future of medical devices, specifically prosthetic sockets.

7. References

Au, Samuel K., Paolo Bonato, and Hugh Herr. "An EMG-positioncontrolled system for an active ankle-foot prosthesis: an initial experiemental study." *9th International Conference on Rehabilitation Robotics*. Chicago, IL: IEEE, 2005. 375-379. Web.

Backus, Byron. "Problems in Lower Limb Socket Fit and Present Clinical Solutions." *The Academy Today* (2005). Document.

Bahari, M. Saiful. "Design and Development of a Multifingered Prosthetic Hand." *International Journal of Social Robotics, Vol. 4, Issue 1* (2011): 59-66. Web.

Bao, Minhang. *Piezoresistive sensors*. Lausanne: Elsevier-Sequoia, 1991. Web.

Bergmann, J.H.M. and A.H. McGregor. "Body-Worn Sensor Design: What do Patients and Clinicians Want?" *Annals of Biomedical Engineering, Vol. 39, No. 9* (2011): 2299-2312. Web.

Brookeshaw, Marcus, et al. "Two-degree-of-freedom powered prosthetic wrist." *Journal of Rehabilitation Research & Development, Vol. 48, No. 6* (2011): 609-618. Web. 26 September 2014. <http://go.galegroup.com/ps/i.do?action=interpret&id=GALE%7CA263880120&v=2.1&u=mlin_c_worpoly&it=r&p=HRCA&sw=w&authCount=1>.

Caspers, Carl A. United States: Patent 5,549,709. 1995. Web.

Chapter 6: Piezoresistive Sensors. 25 July 2005. 30 September 2014. <http://www.mech.northwestern.edu/FOM/LiuCh06v3_072505.pdf>.

Cooper, Rory A., Hisaichi Ohnabe, and Douglas A. Hobson. *An Introduction to Rehabilitation Engineering*. Boca Raton, FL: Taylor & Francis Group, 2007. Web.

Craddock, Russell & Peter Kinnell. *Trench Etched Resonant Pressure Sensor: TERPS*. Company Report. Groby, UK: GE Measurement & Control Solutions, n.d. Web.

Das, Apurba. "Chapter 7: Discrete Fourier Transform." Das, Apurba. *Signal Conditioning: An Introduction to Continuous Wave Communication and Signal Processing*. Berlin: Springer-Verlag, 2012. 159-192. Web.

De Koster, Karsten. *Gait Cycle*. 1 January 2014. Web. 30 September 2014. <http://www.physio-pedia.com/Gait_Cycle>.

Gailey, Robert, et al. "Review of secondary physical conditions associated with lower-limb amputation and long-term prosthesis use." *Journal of Rehabilitation Research and Development, Vol. 45, No. 30* (2008): 15-30. Web.

Galea, Anna M., Kristen LeRoy, and Thieu Q., Truong. Active Prosthetic Socket. US Patent Application: Patent 506,473. 2012. Web.

Graupe, Daniel, et al. *A Multifunctional Prosthesis Control System Based on Time Series Identification of EMG Signals using MicroProcessors*. Fort Collins: Colorado State University, 2013. Web.

Greenwald, Richard M. "Volume Management: Smart Variable Geometry Socket (SVGS) Technology for Lower-Limb Prostheses." *JPO, Vol. 15, Num. 3* (2003): 107-112. Web.

iWorx. "Galvanic Skin Response (GSR) and Investigation into 'Cheating'." 2013. *Rogers State University*. Web. 17 September 2014. <http://www.rsu.edu/BehavioralSciencesLab/docs/Labs/TheGalvanicSkinResponse(GSR)InvestigationCheating.pdf>.

Jackson, M., et al. "Improvements in Measuring Shoulder Joint Kinematics." *Jounal of Biomechanics, Vol. 45, Issue 12* (2012): 2180-2183. Web.

John Hopkins Medicine. *Oximetry*. 2014. Web. 17 September 2014. <http://www.hopkinsmedicine.org/healthlibrary/test_procedures/pulmonary/oximetry_92,P07754/>.

—. *What is EMG?* 2014. Web. 17 September 2014. <http://www.hopkinsmedicine.org/healthlibrary/test_procedures/neurological/electromyography_emg_92,p07656/>.

Kamavuako, Ernest N., Erik J. Scheme and Kevin B. Englehart. "On the Usability of Intramuscular EMG for Prosthetic Control: A Fitts' Law Approach." *Journal of Electromyograhpy and Kinesiology Vol. 24* (2014): 770-777. Web.

Kandel, E. J. Artificial Limb with End-Bearing Socket. United States: Patent 3,393,407. 1965. Web.

Komolafe, Oluseeni, et al. "Methods for characterization of mechanical and electrical prosthetic vacuum pumps." *JRRD Volume 50 Number 8* (2013): 1069-1078. Web.

Lai, C.W., et al. "An interface system to aid the design of rapid prototyping prosthetic socket coated with a resin layer for transtibial amputee." *13th International Conference on Biomedical Engineering Volume 34 Number 5* (2009): 37-45. Web.

Lenze, J. F. and J.D. Rossi. Suction socket for artificial limb. United States: Patent 5,376,131. 1994. Web.

Liang, Hualou, Joseph D. Bronzino and Donald R. Peterson. *Biosignal Processing: Principles and Practices*. Hoboken: CRC Press, 2012. Web.

Liberating Technologies, Inc. *Built-in & Removable Prosthetic Batteries*. 2013. Web. 26 September 2014. <http://www.liberatingtech.com/>.

Mak, Arthur F.T., Ming Zhang and David A. Boone. "State-of-the-art research in lower-limb prosthetic biomechanics-socket interface." *Journal of Rehabilitation and Development Volume 38 Number 2* (2001): 161-174. Web.

MedlinePlus. *Electromyography*. 10 September 2012. Web. 17 September 2014. <http://www.nlm.nih.gov/medlineplus/ency/article/003929.htm>.

Mendelson, Y., Ochs, B. "Noninvasive Pulse Oximetry Utilizing Skin Reflecance Photoplethysmography." *IEEE Transactions on Biomedical Engineering, 35* (1988): 10. Web.

Micera, S., J. Carpaneto and S. Raspopovic. "Control of Hand Prostheses Using Peripheral Information." *Biomedical Engineering, IEEE Reviews in* (2010): 48-68. Web.

Miller, L.A., et al. "Improved Myoelectric Prosthesis Control Using Targeted Reinnervation Surgery: A Case Series." *Neural Systems and Rehabilitation Engineering, IEEE Transactions on, vol. 16, no. 1* (2008): 46-50. Web.

Montgomery, John T., Meagan R. Vaughan, and Richard H. Crawford. "Design of an actively actuated prosthetic socket." *Rapid Prototyping Journal, Vol. 16, Iss. 3* (2010): 194-201. Web.

Pantelopoulos, Alexandros and Nikolaos G. Bourbakis. "Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 40, no.1* (2010): 1-12. Web.

Pietka, Ewa and Jacek Kawa. "Control System of Bioprosthetic Hand Based." *Information Technologies in Biomedicine*. Gliwice, Poland: Springer, 2012. 211-220. Web.

Redhead, R. G. "Total surface bearing self suspending above-knee sockets." *Prosthetics and Orthodics International* (1979): 126-136.

Ryu, J., & Kim, D. *Multiple Gait Phase Recognition Using Boosted Classifiers Based on sEMG Signal and Classification Matrix*. New York, NY: ACM, 2014. Web.

Sansam, Kate, et. al. "Predicting walking ability following lower limb amputation: asystematic review of the literature." *Journal of Rehabilitation Medicine, Vol. 41* (2009): 593-603. Web.

Slee, Daren. *Lithium Ion & Lithium Polymer Batteries*. July 2012. Web. 26 September 2014. <http://ewh.ieee.org/r6/ocs/pses/PSESOCChapterJuly2012PresentationReleasesmallerifle.pdf>.

Sun, Jinming and Philip A. Voglewede. "Powered Transtibial Prosthetic Device Control System Design, Implementation, and Bench Testing." 6 December 2013. *The American Society of Mechanical Engineers.* Web. 24 September 2014. <http://medicaldevices.asmedigitalcollection.asme.org/article.aspx?articleid=1763767>.

Waryck, Brian. "Comparison of two myoelectric multi-articulating prosthetic hands." *MyoElectric Controls/Powered Prosthetics Symposium*. Fredericton, New Brunswick, Canada: IEEE, 2011. Web.

Wentink, E.C. "Intention Detection of Gait Initiation using EMG and Kinematics." *Gait & Posture Vol. 37* (2013): 223-228. Web.

You, B., Wang, H., & Huang, L. "The System of sEMG Recognition for Prosthetic Hand Control." *Strategic Technology (IFOST), 2010 International Forum on*. Ulsan: IEEE, 2010. 44-49. Web.

Zhang, M, et al. "Frictional action at lower limb/prosthetic socket interface." *Medical Engineering & Physics, Vol. 18, No. 3* (1996): 207-214. Web.

Appendices

Appendix A-MATLAB Code for Accelerometer Testing

```
%Model of lower torso and leg based on x-value of accelerometer data
%Assumes person's back is straight and lower leg is always perpendicular to
%ground
clear; clc; close all;
pause(1);
accel_data=xlsread('accel_testing.xlsx');
accel=accel_data(:,1);

radius=12;%assumed length of quad (code is more for finding angles)

%sitting values
sitting=mean(accel(1:90));

%standing values
standing=mean(accel(110:190));

one_deg=abs(standing-sitting)/90; %values/degree
zero_ang=2*sitting-standing;%if leg is straight up
figure(1);
time_vect=0:0.01:(length(accel)-1)/100;
plot(time_vect, accel);
xlabel('Time (s)');
ylabel('Value');
title('Accelerometer');
figure(2);
l_out=line([0 0], [0 radius]);
l_extra=line([0 0], [0 radius]);
line([0 0], [0 radius], 'LineWidth', 5);%model torso
title('Leg Orientation with Upright Torso');
for v=1:length(accel)
    axis([-2*radius 2*radius -radius radius])
    value=accel(v);
    curr_ang=abs(zero_ang-value)/one_deg;%(value-offset)/(value/degree)
    delete(l_out);
    delete(l_extra);
    r_sin=radius*sind(curr_ang);
    r_cos=radius*cosd(curr_ang);
    l_out=line([0 r_sin], [0 r_cos]);
    l_extra=line([r_sin r_sin], [r_cos r_cos-12]);
    pause(0.05);%allows viewer to see motion on plot
end
```

Appendix B - MATLAB Code for State Determination Results and Calculations

```
clear; clc; close all;
xldata=xlsread('statechangedata.xlsx');
data=xldata(1:226,:);
[rows columns]=size(data);
for c=1:rows
    ShaneAvg(c)=mean(data(c,1:3));
    AlexAvg(c)=mean(data(c,4:6));
    TynanAvg(c)=mean(data(c,7:9));
end

time=0:1/5:45;
actualState=zeros(1,length(ShaneAvg));
for v=1:length(ShaneAvg)
    if v<76
        actualState(v)=1;
    else
        if v<81
            actualState(v)=2;
        else
            if v<151
                actualState(v)=3;
            else
                actualState(v)=4;

            end
        end
    end
end

%figure(3);
figure1 = figure;
title('State Determinations over time');
 % Create subplot
subplot1 = subplot(4,1,1,'Parent',figure1,...
    'YTickLabel',{'0','1','2','3','4','5','3','','4','','','',''},...
    'YTick',[0 1 2 3 4 5]);
box(subplot1,'on');
hold(subplot1,'all');
plot(time, ShaneAvg, 'red', 'LineWidth', 6);
hold on;
plot(time, 1);
plot(time, 2);
plot(time, 3);
plot(time, 4);
plot(time, 5);
ylabel('State', 'FontSize', 14);
title('Subject 1 State Determinations', 'FontSize', 16);
axis([0 time(end) 0 5]);

subplot2 = subplot(4,1,2,'Parent',figure1,...
    'YTickLabel',{'0','1','2','3','4','5','3','','4','','','',''},...
    'YTick',[0 1 2 3 4 5]);
box(subplot2, 'on');
```

```
hold(subplot2, 'all');
plot(time, AlexAvg, 'green', 'LineWidth', 6);
hold on;
plot(time, 1);
plot(time, 2);
plot(time, 3);
plot(time, 4);
plot(time, 5);
ylabel('State', 'FontSize', 14);
title('Subject 2 State Determinations', 'FontSize', 16);
axis([0 time(end) 0 5]);

subplot3 = subplot(4,1,3,'Parent',figure1,...
    'YTickLabel',{'0','1','2','3','4','5','3','','4','','','',''},...
    'YTick',[0 1 2 3 4 5]);
box(subplot3,'on');
hold(subplot3,'all');
plot(time, TynanAvg, 'blue', 'LineWidth', 6);
hold on;
plot(time, 1);
plot(time, 2);
plot(time, 3);
plot(time, 4);
plot(time, 5);
ylabel('State', 'FontSize', 14);
title('Subject 3 State Determinations', 'FontSize', 16);
axis([0 time(end) 0 5]);

subplot4 = subplot(4,1,4,'Parent',figure1,...
    'YTickLabel',{'0','1','2','3','4','5','3','','4','','','',''},...
    'YTick',[0 1 2 3 4 5]);
box(subplot4,'on');
hold(subplot4,'all');
scatter(time, actualState, 'filled', 'black', 'LineWidth', 0.5);
hold on;
plot(time, 1);
plot(time, 2);
plot(time, 3);
plot(time, 4);
plot(time, 5);
ylabel('State', 'FontSize', 14);
title('Actual State', 'FontSize', 16);
axis([0 time(end) 0 5]);
xlabel('Time (s)', 'FontSize', 14);

scores=zeros(9,1);
for d=1:length(ShaneAvg)
    for k=1:9
        if data(d,k)==actualState(d)
            scores(k)=scores(k)+1;
        end
    end
end
percent=scores/length(ShaneAvg)
avgPercent=[mean(percent(1:3)) mean(percent(4:6)) mean(percent(7:9))]
```

Appendix C - MATLAB Code for ABB Robot Initial Test

```
clear;clc; close all
%get data from sheet
fulltable=xlsread('abbtest.xlsx');
fulltableData=[fulltable(:,1) fulltable(:,3)];
actions=0:5;
spot1=1;
spot2=1;
for v=1:length(fulltable(:,1))
    value1=fulltableData(v,1);
    %filter out non-numbers (means, sds, bad prints, etc.)
    if ismember(value1, actions) && ~isnan(value1)
        data(1, spot1)=value1;
        spot1=spot1+1;
    end
    value2=fulltableData(v,2);
    if ismember(value2, actions) && ~isnan(value2)
        data(2, spot2)=value2;
        spot2=spot2+1;
    end
end
maxlength=150;%5 read/sec * 10 seconds/state * 3 states = 150 readings
figure(1);
subplot(3,1,1);
%adjust for data not starting at same time
shift=abs(find(data(1,:)~=1,1)-find(data(2,:)~=1,1));
plot(data(1,shift:maxlength+shift-1));%slight shift due to delayed start
title('First ABB Test Trial');
ylabel('Detected Activity');
subplot(3,1,2);
plot(data(2,1:maxlength));
title('Second ABB Test Trial');
ylabel('Detected Activity');

%add plot for actual state
actualState=zeros(1,maxlength);
for v=1:maxlength
    if v<51
        actualState(v)=1;
    else
        if v<61
            actualState(v)=2;
        else
            if v<111 || v>161
                actualState(v)=3;
            else
                actualState(v)=4;
            end
        end
    end
end
subplot(3,1,3);
plot(actualState);
title('Actual ABB State');
ylabel('Actual Activity');
```

Appendix D – MATLAB Code for Final ABB Robot Test

```
clear;clc; close all
%get data from sheet
fulltable=xlsread('finalStateData.xlsx');
fulltableData=[fulltable(:,1) fulltable(:,2) fulltable(:,3)];
%possible action values
actions=0:5;

spot1=1;
spot2=1;
spot3=1;
started1=0;
started2=0;
started3=0;
for v=1:length(fulltable(:,1))
    value1=fulltableData(v,1);
    value2=fulltableData(v,2);
    value3=fulltableData(v,3);
    %ignore all values until sitting appears (start of test)
    if value1==1
        started1=1;
    end

    if value2==1
        started2=1;
    end

    if value3==1
        started3=1;
    end

    %filter out non-numbers (means, sds, bad prints, etc.)
    if started1 && ismember(value1, actions) && ~isnan(value1)
        data(1, spot1)=value1;
        spot1=spot1+1;
    end

    if started2 && ismember(value2, actions) && ~isnan(value2)
        data(2, spot2)=value2;
        spot2=spot2+1;
    end

    if started3 && ismember(value3, actions) && ~isnan(value3)
        data(3, spot3)=value3;
        spot3=spot3+1;
    end
end
%test lasted approximately 30 seconds. 30 seconds*5 samples/sec=150 samples
maxlength=150;

%plot all data
figure(1);
subplot(4,1,1);
plot(data(1,1:maxlength));
```

```
title('First ABB Test Trial');
ylabel('Detected Activity');
subplot(4,1,2);
plot(data(2,1:maxlength));
title('Second ABB Test Trial');
ylabel('Detected Activity');
subplot(4,1,3);
plot(data(3,1:maxlength));
title('Third ABB Test Trial');
ylabel('Detected Activity');

%plot actual state
actualState=zeros(1,maxlength);
for v=1:maxlength
    if v<31%~6 seconds sitting
        actualState(v)=1;
    else
        if v<46%~3 seconds transition
            actualState(v)=2;
        else
            if v<81%~5 seconds standing
                actualState(v)=3;
            else
                actualState(v)=4;
            end
        end
    end
end
subplot(4,1,4);
plot(actualState);
title('Actual ABB State');
ylabel('Actual Activity');
```

Appendix E – Final Code for Operation

```
#include <SPI.h>
/*THIS IS THE MAIN CODE FOR OPERATION
sets the pump pressures based on the user's activity
also looks for switch to be flipped for emergency deflateAll
if accelerometer needs to be configured, use calibrateAngles script
*/
const int pump1InPin = 3;
const int pump1OutPin = 2;
const int pump2InPin = 4;
const int pump2OutPin = 5;
const int pump3InPin = 6;
const int pump3OutPin = 7;
const int pump4InPin = 8;
const int pump4OutPin = 9;
const int sitLED = 10;
const int standLED= 11;
const int walkLED = 12;

const int accelXPin = A0;
const int accelYPin = A1;
const int accelZPin = A2;
const int pressure1Pin = A6;
const int pressure2Pin = A5;
const int pressure3Pin = A4;
const int pressure4Pin = A3;
const int deflatePin = A7;

int pumpStatus[8];
int origPumpIn=0;
int origPumpOut=0;
int withinTolerance[4];
bool changeAct = false;
int XValue, YValue, ZValue;
const int numSensors = 4;
const int arraySize = 100;
int accel_XInt[arraySize];
int accel_YInt[arraySize];
int accel_ZInt[arraySize];
float meanAccelX;
float sdX;
float sdY;
float meanAccelY;
float xAngles[arraySize];
//change these values for configuration
float accelXSit = 346;
float accelXStand = 413;

float oneDegX=abs(accelXStand-accelXSit)/90;
float zeroAngleX=2*accelXSit-accelXStand;
float accelYStraight = 319;
float accelYSideways = 268;
float oneDegY=abs(accelYStraight-accelYSideways)/90;
float zeroAngleY=2*accelYStraight-accelYSideways;
float yAngles[arraySize];
int arrayCount=0;
int checkActivity=0;
int act=0;
bool readyToRead = false;
bool resetActivity = false;
int pp1, pp2, pp3, pp4;
int objectives[4];
bool deflateAll = 0;
void setup() {
  //setup valve outputs
    pinMode(pump1InPin, OUTPUT);
         pinMode(pump1OutPin, OUTPUT);
    pinMode(pump2InPin, OUTPUT);
         pinMode(pump2OutPin, OUTPUT);
         pinMode(pump3InPin, OUTPUT);
         pinMode(pump3OutPin, OUTPUT);
         pinMode(pump4InPin, OUTPUT);
         pinMode(pump4OutPin, OUTPUT);

//ensure all valves are off
         digitalWrite(pump1InPin, LOW);
         digitalWrite(pump1OutPin, LOW);
         digitalWrite(pump2InPin, LOW);
         digitalWrite(pump2OutPin, LOW);
         digitalWrite(pump3InPin, LOW);
         digitalWrite(pump3OutPin, LOW);
         digitalWrite(pump4InPin, LOW);
         digitalWrite(pump4OutPin, LOW);

         cli();//stop interrupts
         Serial.begin(9600);

         //set timer1 interrupt at 100.1Hz (if needed)
         TCCR1A = 0;// set entire TCCR1A register to 0
         TCCR1B = 0;// same for TCCR1B
         TCNT1  = 0;//initialize counter value to 0
         OCR1A = 155;// = (16*10^6) / (100.2*1024) - 1 (must be <65536)
         TCCR1B |= (1 << WGM12);
         TCCR1B |= (1 << CS12) | (1 << CS10);
         TIMSK1 |= (1 << OCIE1A);

         sei();//allow interrupts*/
}//end setup

void updateAnalog(){
 if(readyToRead){//don't need pressure values at beginning
   pp1=analogRead(pressure1Pin);
   pp2=analogRead(pressure2Pin);
   pp3=analogRead(pressure3Pin);
   pp4=analogRead(pressure4Pin);
 }
  xAngles[arrayCount]=abs(zeroAngleX-analogRead(accelXPin))/oneDegX;
 arrayCount++;

  if(arrayCount==100){
   arrayCount=0;
   readyToRead=true;
 }
 checkActivity++;
 if(checkActivity==20){
   checkActivity=0;
   resetActivity=true;
 }
  if(analogRead(deflatePin)>600)
   deflateAll=1;
 else
   deflateAll=0;
}

//find average of data
float average(float data[], int len){
        float sum=0;
        for(int i=1; i< len ; i++){
                sum+=data[i];
     //   Serial.println(data[i]);
        }
        return sum/(len-1);
}

//find standard deviation of data
float standard_dev(float data[], int len, float mean){
        float std_dev;
        float square_dev=0.0;
        for(int i=1; i<len; i++){
                square_dev+=(data[i]-mean)*(data[i]-mean);
       // Serial.println(data[i]);
        }
        std_dev=sqrt(square_dev/(len-1));
        return std_dev;
}



///determine users current activity based on mean and sd
int findActivity(float avg, float st_devx, float st_devy){
        if(avg>45 && avg<160 && st_devx<6)
                return 1;//sitting
                if(avg>110 && avg<160 && st_devx>=10)
                        return 2;//transition
                        if(avg>170 && avg<200 && st_devx<6)//st_devy indicates swinging leg to the sides, such as a
buckling knee
                                return 3;//"Standing";
                        if(avg>150 && avg<210 && st_devx<=30)
```

```
                                    return 4;//"Walking";
                           if(avg>140 && avg<210 && st_devx>30)
                                    return 5;//"Running";
                           else
                                    return 0;//"Undetermined";

}

//convert given psi to integer reading from pressure sensor
int psiToValue(float psi, int pumpNum){
 if(pumpNum==1)
  return psi*749.2/15+172.4;
 else
  return psi*819.2/15.0+102.4;
}

/*bladders in code
1: front
2: left
3: back
4: right*/
void loop() {
    int tolerance = 10;
    while(!readyToRead){
     updateAnalog();
     delay(10);
    }
    if(readyToRead){
     updateAnalog();
     meanAccelX = average(xAngles, arraySize);
     sdX = standard_dev(xAngles, arraySize, meanAccelX);
         int previousAct=act;
     act=findActivity(meanAccelX, sdX, sdY);
     //prevent undetermined activity from keeping it out of control system too long
     if(act==0)
      act=previousAct;
      Serial.print("Mean: ");
     Serial.println(meanAccelX);
     Serial.print("SD: ");
     Serial.println(sdX);
     Serial.println(act);
     switch (act){
          case 1://sit
                  objectives[0]=psiToValue(1,1);
        objectives[1]=psiToValue(1,2);
        objectives[2]=psiToValue(1,3);
        objectives[3]=psiToValue(1,4);
        digitalWrite(sitLED, HIGH);
        digitalWrite(standLED, LOW);
        digitalWrite(walkLED, LOW);
                  break;
          case 2://transition
                  objectives[0]=psiToValue(2.5,1);
        objectives[1]=psiToValue(2.5,2);
        objectives[2]=psiToValue(2.5,3);
        objectives[3]=psiToValue(2.5,4);
        digitalWrite(sitLED, HIGH);
        digitalWrite(standLED, HIGH);
        digitalWrite(walkLED, LOW);
                  break;
          case 3://stand
                  objectives[0]=psiToValue(2,1);
        objectives[1]=psiToValue(2,2);
        objectives[2]=psiToValue(2,3);
        objectives[3]=psiToValue(2,4);
        digitalWrite(sitLED, LOW);
        digitalWrite(standLED, HIGH);
        digitalWrite(walkLED, LOW);

                  break;
          case 4://walking
                  objectives[0]=psiToValue(8.03,1);
        objectives[1]=psiToValue(6.86,2);
        objectives[2]=psiToValue(8.03,3);
        objectives[3]=psiToValue(6.49,4);
        digitalWrite(sitLED, LOW);
        digitalWrite(standLED, LOW);
        digitalWrite(walkLED, HIGH);

                  break;
          case 5://run (don't have these values yet, so just keep them where they are)
                  objectives[0]=pp1;
        objectives[1]=pp2;
        objectives[2]=pp3;
        objectives[3]=pp4;
        digitalWrite(sitLED, LOW);
        digitalWrite(standLED, LOW);
        digitalWrite(walkLED, LOW);
                  break;
          default:
        objectives[0]=pp1;
        objectives[1]=pp2;
        objectives[2]=pp3;
        objectives[3]=pp4;
        digitalWrite(sitLED, LOW);
        digitalWrite(standLED, LOW);
        digitalWrite(walkLED, LOW);
         break;
         }

    if(deflateAll){//emergency deflate
         Serial.println("deflate");
          digitalWrite(pump1InPin, LOW);
          digitalWrite(pump1OutPin, HIGH);
          digitalWrite(pump2InPin, LOW);
          digitalWrite(pump2OutPin, HIGH);
```

```
                digitalWrite(pump3InPin, LOW);
                digitalWrite(pump3OutPin, HIGH);
                digitalWrite(pump4InPin, LOW);
                digitalWrite(pump4OutPin, HIGH);

         for(int p=0; p<8; p++){
          if(p % 2==0)
            pumpStatus[p]=0;
          else
            pumpStatus[p]=1;
         }
      }
               if((abs(pp1-objectives[0]) >= tolerance || abs(pp2-objectives[1]) >= tolerance || abs(pp3-objectives[2]) >=
tolerance || abs(pp4-objectives[3]) >= tolerance) && !deflateAll){
                adjustAllPressures(objectives, tolerance);
       }
    }
}


/*adjust pressures in bladders by opening and closing valves
uses bang-bang control
(too high)-let air out
too low- let air in
exit if all pressures are within miniTolerance
or if resetActivity flag is set
When pressure is set, does not set again until exits and reenters
delay(10) exists b/c valves can only flip at 100 Hz
print statements commented but not deleted for troubleshooting purposes

*/
void adjustAllPressures(int values[], int tolerance){

               int miniTolerance=4;
               int channel=0;
               int pumpPressureValue1=pp1;//analogRead(pressure1Pin);
               int pumpPressureValue2=pp2;//analogRead(pressure2Pin);
               int pumpPressureValue3=pp3;//analogRead(pressure3Pin);
               int pumpPressureValue4=pp4;//analogRead(pressure4Pin);
               int pumpInOn;
               int pumpInOff;
               int pumpOutOn;
               int pumpOutOff;

               //assume no pumps are ready
               boolean pump1Set=0;
               boolean pump2Set=0;
                boolean pump3Set=0;
                boolean pump4Set=0;

               //find current pump status
               int value1 = values[0];
               int value2 = values[1];
         int value3 = values[2];
         int value4 = values[3];

    int pumpIn;
         int pumpOut;
         while(!pump1Set || !pump2Set || !pump3Set || !pump4Set){

        //check pump1
                  if(!pump1Set){
                           channel=0;
                           pumpIn=pumpStatus[channel*2];
                           pumpOut=pumpStatus[channel*2+1];
                           origPumpIn=pumpIn;
                           origPumpOut=pumpOut;

                           if(value1>pumpPressureValue1+tolerance*withinTolerance[0]+miniTolerance){//pressure
too low
                                    pumpIn=HIGH;
                               pumpOut=LOW;
                               withinTolerance[channel]=0;
                           //Serial.println("Set pumpin 1 High");
                           }
                           else{
                                    if(value1<pumpPressureValue1-tolerance*withinTolerance[0]-
miniTolerance){//pressure too high
                                             pumpIn=LOW;
                                             pumpOut=HIGH;
                                             withinTolerance[channel]=0;
                                             //Serial.println("Set pumpin 1 Low");
                                    }
                                    else{
                                             withinTolerance[channel]=1;
                                             pumpIn=LOW;
                                             pumpOut=LOW;
                                             pump1Set=1;
                                             //Serial.println("pump1 good");
                                    }
                           }
                           if(origPumpIn != pumpIn){//pump in status needs to change
                                    if(pumpIn==LOW){//pump needs to be turned off
                                             digitalWrite(pump1InPin,LOW);//PORTB &= pumpIn1Off;
                                    }
                                    if(pumpIn==HIGH){//pump need to be turned on
                                             digitalWrite(pump1InPin, HIGH);//PORTB |= pumpIn1On;
                                             //Serial.println("pump1In officially set");
                                    }
                                    pumpStatus[channel*2]=pumpIn;

                           }
                           if(origPumpOut != pumpOut){//pump out status needs to change
                                    if(pumpOut==LOW){
                                             digitalWrite(pump1OutPin,LOW);//PORTB &= pumpOut1Off;
                                    }
```

```
                if(pumpOut==HIGH){
                    digitalWrite(pump1OutPin,HIGH);//PORTB |= pumpOut1On;
                }
                pumpStatus[channel*2+1]=pumpOut;

            }
        }

        //check pump2
        if(!pump2Set){
            channel=1;
            pumpIn=pumpStatus[channel*2];
            pumpOut=pumpStatus[channel*2+1];
            origPumpIn=pumpIn;
            origPumpOut=pumpOut;

            if(value2>pumpPressureValue2+tolerance*withinTolerance[1]+miniTolerance){//pressure
too low
                pumpIn=HIGH;
              pumpOut=LOW;
              withinTolerance[channel]=0;
            //   Serial.println("Set pumpin 2 High");

            }
            else{
                if(value2<pumpPressureValue2-tolerance*withinTolerance[1]-
miniTolerance){//pressure too high
                    pumpIn=LOW;
                    pumpOut=HIGH;
                    withinTolerance[channel]=0;
                    //    Serial.println("Set pumpin 2 LOW");

                }
                else{
                    withinTolerance[channel]=1;
                    pumpIn=LOW;
                    pumpOut=LOW;
                    pump2Set=1;
                    // Serial.println("pump2 good");

                }
            }
            if(origPumpIn != pumpIn){//pump in status needs to change
                if(pumpIn==LOW){//pump needs to be turned off
                    digitalWrite(pump2InPin,LOW);//PORTB &= pumpIn2Off;
                    //Serial.println("pump2in LOW officially set");

                }
                if(pumpIn==HIGH){//pump need to be turned on
                    digitalWrite(pump2InPin, HIGH);//PORTB |= pumpIn2On;
                    //Serial.println("pump2in HIGH officially set");
                }
                pumpStatus[channel*2]=pumpIn;
            }
            if(origPumpOut != pumpOut){//pump out status needs to change
                if(pumpOut==LOW){
                    digitalWrite(pump2OutPin, LOW);//PORTB &= pumpOut2Off;
                    // Serial.println("pump2out LOW officially set");
                }
                if(pumpOut==HIGH){
                    digitalWrite(pump2OutPin, HIGH);//PORTB |= pumpOut2On;
                    // Serial.println("pump2out HIGH officially set");
                }
                pumpStatus[channel*2+1]=pumpOut;
            }
        }

        //check pump3

        if(!pump3Set){
            channel=2;
            pumpIn=pumpStatus[channel*2];
            pumpOut=pumpStatus[channel*2+1];
            origPumpIn=pumpIn;
            origPumpOut=pumpOut;

            if(value3>pumpPressureValue3+tolerance*withinTolerance[2]+miniTolerance){//pressure
too low
            pumpIn=HIGH;
            pumpOut=LOW;
            withinTolerance[channel]=0;
             //Serial.println("Set pumpin 3 High");

            }
            else{
                if(value3<pumpPressureValue3-tolerance*withinTolerance[2]-
miniTolerance){//pressure too high
                    pumpIn=LOW;
                    pumpOut=HIGH;
                    withinTolerance[channel]=0;
                    //Serial.println("Set pumpin 3 LOW");

                }
                else{
                    withinTolerance[channel]=1;
                    pumpIn=LOW;
                    pumpOut=LOW;
                    pump3Set=1;
                    // Serial.println("pump3 good");
                }
            }
            if(origPumpIn != pumpIn){//pump in status needs to change
                if(pumpIn==LOW){//pump needs to be turned off
                    digitalWrite(pump3InPin,LOW);//PORTB &= pumpIn2Off;
                    //  //Serial.println("pump3in LOW officially set");
                }
                if(pumpIn==HIGH){//pump need to be turned on
                    digitalWrite(pump3InPin, HIGH);//PORTB |= pumpIn2On;
         //Serial.println("pump3in HIGH officially set");
                }
                pumpStatus[channel*2]=pumpIn;
            }
            if(origPumpOut != pumpOut){//pump out status needs to change
                if(pumpOut==LOW){
                    digitalWrite(pump3OutPin, LOW);//PORTB &= pumpOut2Off;
                     //Serial.println("pump3out LOW officially set");
                }
                if(pumpOut==HIGH){
                    digitalWrite(pump3OutPin, HIGH);//PORTB |= pumpOut2On;
                    //Serial.println("pump3out HIGH officially set");
                }
                pumpStatus[channel*2+1]=pumpOut;
            }
        }

        //check pump4
        if(!pump4Set){
            channel=3;
            pumpIn=pumpStatus[channel*2];
            pumpOut=pumpStatus[channel*2+1];
            origPumpIn=pumpIn;
            origPumpOut=pumpOut;

            if(value4>pumpPressureValue4+tolerance*withinTolerance[3]+miniTolerance){//pressure
too low
                pumpIn=HIGH;
            pumpOut=LOW;
            withinTolerance[channel]=0;
       //Serial.println("Set pumpin 4 High");

            }
            else{
                if(value4<pumpPressureValue4-tolerance*withinTolerance[3]-
miniTolerance){//pressure too high
                    pumpIn=LOW;
                    pumpOut=HIGH;
                    withinTolerance[channel]=0;
                    //Serial.println("Set pumpin 4 LOW");

                }
                else{
                    withinTolerance[channel]=1;
                    pumpIn=LOW;
                    pumpOut=LOW;
                    pump4Set=1;
                    //Serial.println("pump4 good");
                }
            }
            if(origPumpIn != pumpIn){//pump in status needs to change
                if(pumpIn==LOW){//pump needs to be turned off
                    digitalWrite(pump4InPin,LOW);//PORTB &= pumpIn2Off;
                    //Serial.println("pump4in LOW officially set");
                }
                if(pumpIn==HIGH){//pump need to be turned on
                    digitalWrite(pump4InPin, HIGH);//PORTB |= pumpIn2On;
                    //Serial.println("pump4in HIGH officially set");
                }
                pumpStatus[channel*2]=pumpIn;
            }
            if(origPumpOut != pumpOut){//pump out status needs to change
                if(pumpOut==LOW){
                    digitalWrite(pump4OutPin, LOW);//PORTB &= pumpOut2Off;
                    //Serial.println("pump4out LOW officially set");
                }
                if(pumpOut==HIGH){
                    digitalWrite(pump4OutPin, HIGH);//PORTB |= pumpOut2On;
                    //Serial.println("pump4out HIGH officially set");
                }
                pumpStatus[channel*2+1]=pumpOut;
            }
        }

        delay(10);
         updateAnalog();
        if(resetActivity){
            resetActivity=false;
            break;//exit control loop
         }
        pumpPressureValue1=pp1;//analogRead(pressure1Pin);
        pumpPressureValue2=pp2;//analogRead(pressure2Pin);
        pumpPressureValue3=pp3;//analogRead(pressure3Pin);
        pumpPressureValue4=pp4;//analogRead(pressure4Pin);
    }
}
```

Appendix F – Code for Configuring Accelerometer

```
#include <SPI.h>
/*this script is used to calibrate motion angles
//using mean and std
To calibrate accelerometer, have prosthetic as perpendicular to
ground as possible and set accelXStand to the average reading.
Then have it parallel to ground and set accelXSit to the average reading
The internal calculations will do the rest, assuming a linear relationship
between the value and angle relative to torso
*/
int XValue, YValue, ZValue, Temperature;
int stopAll = 0;
int readValue = 0;
int counter = 0;
const int numSensors = 4;
const int arraySize = 100;
int accel_XInt[arraySize];
int accel_YInt[arraySize];
int accel_ZInt[arraySize];
int allAccel[numSensors][arraySize];
float meanAccelX;
float meanAccelY;
float meanAccelZ;
float sdX;
float sdY;
float sdZ;
float sitMean = 90;
float standMean = 180;
float walkMeanX = 176.5;
float runMeanX = 168.5;
float sitSDX = 0.25;
float standSDX = 0.25;
float walkSDX = 27.45;
float runSDX = 73.36;
float xAngles[arraySize];
float yAngles[arraySize];
float accelXSit = 346;
float accelXStand = 413;
float accelYStraight = 319;
float accelYSideways = 268;
float oneDegX=abs(accelXStand-accelXSit)/90;
float oneDegY=abs(accelYStraight-accelYSideways)/90;
float zeroAngleX=2*accelXSit-accelXStand;
float zeroAngleY=2*accelYStraight-accelYSideways;
bool resetActivity=false;

void setup(){

  cli();//stop interrupts
  Serial.begin(9600);

//set timer1 interrupt at 49.92Hz
```

```
  TCCR1A = 0;// set entire TCCR1A register to 0
  TCCR1B = 0;// same for TCCR1B
  TCNT1  = 0;//initialize counter value to 0
  OCR1A = 155;// = (16*10^6) / (49.92 Hz*1024) - 1 (must be <65536)
  TCCR1B |= (1 << WGM12);
  TCCR1B |= (1 << CS12) | (1 << CS10);
  TIMSK1 |= (1 << OCIE1A);

  sei();//allow interrupts
}//end setup
int arrayCount=0;
bool readyToRead=false;
int checkActivity=0;

//timer reads accelerometer at 100.2 Hz rate
ISR(TIMER1_COMPA_vect){
  int x = analogRead(A0);
  Serial.println(x);
  accel_XInt[arrayCount]=x;
  xAngles[arrayCount]=abs(zeroAngleX-accel_XInt[arrayCount])/oneDegX;
  arrayCount++;
  if(arrayCount==100){
    arrayCount=0;
    readyToRead=true;
  }
}

float average(int data[], int len){
  float sum=0;
  for(int i=1; i< len ; i++){
    sum+=data[i];
  }
  return sum/(len-1);
}

float averageFloat(float data[], int len){
  float sum=0;
  for(int i=1; i< len ; i++){
    sum+=data[i];
  }
  return sum/(len-1);
}

//find standard deviation of values
float standard_dev(int data[], int len){
  int mean=average(data, len);
  float std_dev;
  float square_dev=0.0;
  for(int i=1; i<len; i++){
    square_dev+=(data[i]-mean)*(data[i]-mean);
  }
    std_dev=sqrt(square_dev/(len-1));
   return std_dev;
```

```
}
//read sensors outside of timer
void readSensors(){
 int16_t accelXArray[arraySize];
 int16_t accelYArray[arraySize];
 int16_t accelZArray[arraySize];
 int pumpPressureValue = 0;
 int externalPressureValue = 0;
 int pumpPressureArray[arraySize];
 int externalPressureArray[arraySize];
 int arrayCount = 0;
 int sum=0;
  while(arrayCount<arraySize){

   if(!readValue){
    XValue = analogRead(0);
    YValue = analogRead(1);
    ZValue = analogRead(2);
     accel_XInt[arrayCount] = XValue;
     accel_YInt[arrayCount] = YValue;
     accel_ZInt[arrayCount] = ZValue;
     arrayCount++;
     delay(10);

   }
  }
  Serial.println("Finished Reading.");
  for(int j=0; j<arraySize; j++){
   xAngles[j]=abs(zeroAngleX-accel_XInt[j])/oneDegX;
   yAngles[j]=abs(zeroAngleY-accel_YInt[j])/oneDegY;
   Serial.print(xAngles[j]);
   Serial.print(",");
   Serial.println(accel_XInt[j]);

  }

}

void loop(){
if(readyToRead){
 meanAccelX=average(accel_XInt, arraySize);
 sdX=standard_dev(accel_XInt, arraySize);
 Serial.print("Mean: ");
 Serial.println(meanAccelX);
 Serial.print("Standard dev: ");
 Serial.println(sdX);
}
}
```

Appendix G – Major Components in Final Prototype

Part	Manufacturer Part Number	Price per Unit	Qty	Brand	Bought from
Socket	N/A	(Donated)	1	Ottobock	N/A
Air Jack Twin Bladders	AETTAW	$34.19 for two	2	Access Tools	National Tool Warehouse
6mm pneumatic tubing	021612 03	(Donated)	~15 ft	Nitra Pneumatics	Automation Direct
3 way‘T’ push connectors (4-pack)	B00G9CEU04	$5.70	4	Amico	Amazon
Polymer Li-Ion Battery Pack	PL-5545135S7WR	$129.95	1	Powerizer	AA Portable Power Company
0-15 psi Pressure Transducer	N/A	$26.00	4	N/A	ebay
Valve	ET-2M-24	$10.00	8	Clippard	Surplusgizmos
Manifold	15490-3-MR	$6.09	4	Clippard	Clippard
6mm push to 10/32 male	MTC 6-U10-1	$2.48	8	MettleAir	Amazon
1/8” NPT female to 6mm push	MTCF 6-N01	$5.98	9	MettleAir	ebay
Arduino Mega	A000067	(Donated)	1	Arduino	N/A
Air Tank	US14227-S0400	(Donated)	3	SMC	N/A
Pressure Regulator	NAR2000	(Donated)	1	SMC	N/A
Pressure Gauge	9143-06	(Donated)	1	SMC	N/A
Backpack	Max Cold model	(Donated)	1	Igloo	N/A
Dragon Skin (for molded leg)	B004BN9V7A	(Donated)	3 Packs	Smooth-On, Inc.	N/A
Bladder Holder Material (fabric)	N/A	$3 / yard	1	Jo-Ann Fabrics	Jo-Ann Fabrics
Triple Axis Accelerometer Breakout Board	ADXL335	$14.95	1	Analog Devices, Inc.	SpaprkFun

Appendix H – Pressures from Subject Tests

Subject	Activity	Lateral	Posterior	Medial	Anterior	Average of anterior and posterior	Average of medial and lateral
1	Sit	-4.0009	-4.2869	-3.7814	-3.6386		
	Stand	0.6926	1.6981	0.8491	0.7399		
	Walk	6.8612	9.5641	6.9349	6.4945	8.0293	6.89805
2	Sit	-4.31022	-4.68134	-4.08657	-3.92716		
	Stand	-0.59774	0.0527	-0.42392	-0.46384		
	Walk	5.443645	7.75648	5.536373	5.172083	6.464281	5.490009
3	Sit	-3.91582	-4.17841	-3.69746	-3.55923		
	Stand	-0.2467	0.50034	-0.07759	-0.13636		
	Walk	6.689403	9.34503	6.765409	6.334233	7.839632	6.727406
4	Sit	-3.52486	-3.67987	-3.31175	-3.19451		
	Stand	-0.20389	0.554921	-0.03536	-0.09643		
	Walk	4.915625	7.083165	5.01544	4.6795	5.881333	4.965532
5	Sit	-4.26697	-4.62618	-4.04389	-3.88681		
	Stand	-0.33868	0.38305	-0.16833	-0.22216		
	Walk	6.733985	9.40188	6.809393	6.375823	7.888852	6.771689