DESIGN OF A MUSCLE-POWERED WALKING EXOSKELETON FOR PEOPLE WITH SPINAL CORD INJURY

A PROJECT
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

VIKRAM VADIRAJ KATTI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

PROF. WILLIAM K DURFEE

JULY 2018

Acknowledgements

First and foremost, I would like to thank Prof. William Durfee for giving me the opportunity to work on this project. His patience, belief in me, and constant words of encouragement got me to this point. I thank him for always being there to answer my queries and to guide me through the project which was a great learning experience.

I would also like to thank Margaret, Justin, Kevin, and Allison for their contributions to the project. Ron Bystrom from the CSE shop and Peter Zimmerman from the machine shop have been immensely helpful in solving machining problems. I thank Eric Nickel from the VA for his help and guidance throughout the project. I thank Saeed, Jon, and Bhargavi for creating a friendly atmosphere in the lab.

Special mention to my best friend for just being there and pushing me to be the best version of myself. Finally, I would like to thank my parents and my sister for constantly motivating me and keeping my spirits high.

Dedication

To,

Mom and Dad

Abstract

Many people with spinal cord injury (SCI) use a wheelchair for mobility. This leads to prolonged periods of sitting, and over time can cause their muscles to atrophy. While there are several commercial exoskeletons for walking, there remains a need for a solution that allows a user with SCI to stand and walk under the power of their own muscles.

Our group is developing a device that combines a single channel of electrical stimulation with a lightweight, passive energy storing exoskeleton to provide short-range walking for a user with SCI. The muscle energy is stored in a gas spring during knee extension and is transferred to the hip and the knee to complete the gait cycle. A control system has been implemented to coordinate the movements of the legs with the stimulation of the quadriceps.

The first-generation prototype weighed 17.05 kg, and preliminary testing with a non-impaired person indicated that the walking speed could be around 0.27 m/s when using the device.

Preliminary tests indicated that the concept is feasible. The design needs modifications to prevent relative movement between the user's body and the structure of the exoskeleton so that a stable standing posture can be achieved. Optimizing the device for weight could potentially lead to a system that is significantly lighter than commercial motor-powered exoskeletons.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

The spinal cord is responsible for transmission of nerve signals from the motor cortex (part of the brain responsible for planning, control, and execution of voluntary movements) to all parts of the body, and from the sensory neurons to the sensory cortex (part of the brain involved in somatic sensation, visual stimuli, and movement planning).

Damage to any part of the spinal cord is known as spinal cord injury (SCI). It can lead to significant loss of sensory and motor functions below the level of injury (e.g. T5 injury – injury is around the T5 vertebrae). The severity of the injury is known as "the completeness" and can be categorized as complete if all sensory and motor function is lost below the level of injury or incomplete if there exists some sensory/ motor function below the level of injury.

Spinal cord injury can lead to paralysis. When a person's arms, hands, trunk, legs and pelvic organs are all affected, the condition is called tetraplegia or quadriplegia. When all or a part of the trunk, legs and pelvic organs are affected, the paralysis is called Paraplegia. The ASIA exam is a test used to define the extent and severity of a person's spinal cord injury [1]. The test categorizes people into grades based on the ASIA impairment scale. Grade A indicates complete lack of motor and sensory function below the level of injury, Grade B indicates some sensation below the level of injury, Grade C indicates some muscle movement below the level of injury but 50% of the muscles below the level of injury can't move against gravity, Grade D indicates that more than 50% of the muscles below the level of injury are strong enough to move against gravity, and Grade E is when all neurological function has returned.

Automobile accidents are responsible for most cases of spinal cord injury in the US (Table 1.1).

Cause	**Percentage**
Vehicular	39
Falls	28
Violence	15
Other	10
Sports	8

Table 1.1: Causes of SCI [2]

There are 282,000 people in the US who have spinal cord injury (SCI) and there are about 17,000 new cases each year [2]. Among these 21.3% have incomplete paraplegia and 20% have complete paraplegia [2]. The number of patients experiencing mobility impairment caused by SCI is increasing because of accidents and disease [3], [4]. Moreover, majority of the people with SCI are under the age of 30. SCI also leads to an increased risk of medical consequences of paralysis such as muscle atrophy, osteoporosis, obesity, coronary heart disease, diabetes, insulin resistance, impaired bowel and/or bladder function, and pressure ulcers [5], [6]. Among people with SCI, 51% identified walking as first choice for a technology application [7]. Therefore, there is a need for a device that can restore gait in people with SCI. The devices created to address this need can be broadly categorized into motor-powered exoskeletons, functional electrical stimulation (FES) based devices, passive orthoses, and hybrid systems.

1.2 Motor-Powered Exoskeletons

Ekso, Indego, and ReWalk are lower-limb exoskeletons with electric motors driving the hip and knee joints, a controller to initiate and control the stepping motions, and a lithium-ion battery pack for untethered power. These exoskeletons allow paraplegics to stand, sit, and walk for short distances.

In a clinical study with Ekso, the participants were able to walk with average speeds ranging from 0.11 to 0.21 m/s and were able to walk for times ranging from 28 to 94 minutes [8]. People using the Indego exoskeleton had a mean walking speed of 0.22 m/s, with speeds ranging from 0.22-0.45 m/s depending on the participants' level of injury [9]–[11]. Individuals using the Indego have been shown to have walking distances ranging from 64-121 m depending on the participants' level of injury [9]. A ReWalk study on individuals with injuries ranging from C8 to L1 showed that participants were able to walk at speeds from 0.25 to 0.48 m/s and distances from 91 to 170 m [12].

A full community ambulator is defined as someone able to maintain a speed of 0.8 m/s, while a limited community ambulator can walk a speed of 0.4 m/s [13]. The required velocity to cross a road safely is approximately 1.06 m/s [14], [15]. It is estimated that the walking distance of 342 m for some activities such as supermarket shopping is necessary for full community ambulation [14]. Users of exoskeletons are able to walk at average gait speeds of 0.26 m/s and a maximum distance of 170 m [14]. However, these speeds and distances do not meet the requirements for full community ambulation. Motorized lower limb exoskeletons are heavy. Reducing the overall weight of the exoskeleton can help preserve natural gait of users [16]. Ekso weighs 23 kgs, ReWalk weights 23.3 kgs, and

Indego is the lightest at 11.8 kg. Also, motorized exoskeletons do not help prevent muscle atrophy in people with SCI.

1.3 FES-based Devices

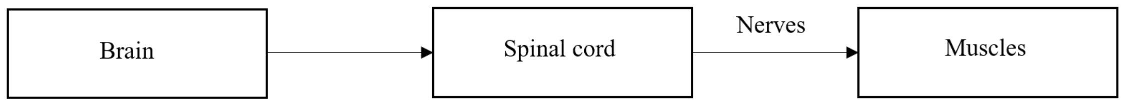

Figure 1.1: Information pathway before SCI

The brain sends signals that cause various muscles to contract or relax. These signals are in the form of electrical impulses that travel via nerves that originate from the spinal cord.

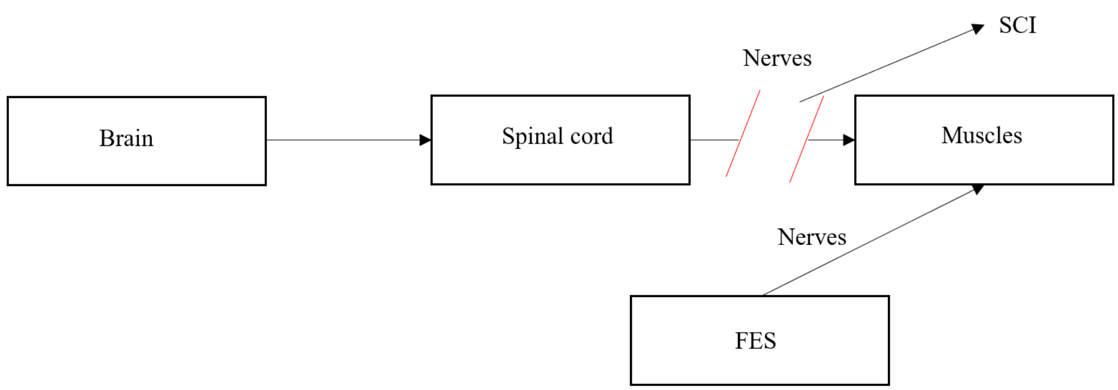

Figure 1.2: Information pathway after SCI

SCI leads to an interruption in this pathway. However, as long as the peripheral nervous system is intact (i.e. the connection between nerves and muscles), sending electrical impulses from an external source to generate nerve signals will make the muscles contract. FES works by utilizing this pathway; it stimulates the motor nerves to trigger muscle contractions [17]. The main components of a FES system are the electrodes, the stimulator, and the controller as shown in Figure 1.3 [18]. The electrodes may be applied to the surface of the skin or they may be implanted. The stimulator sends out electrical pulses of a specific strength with a particular frequency.

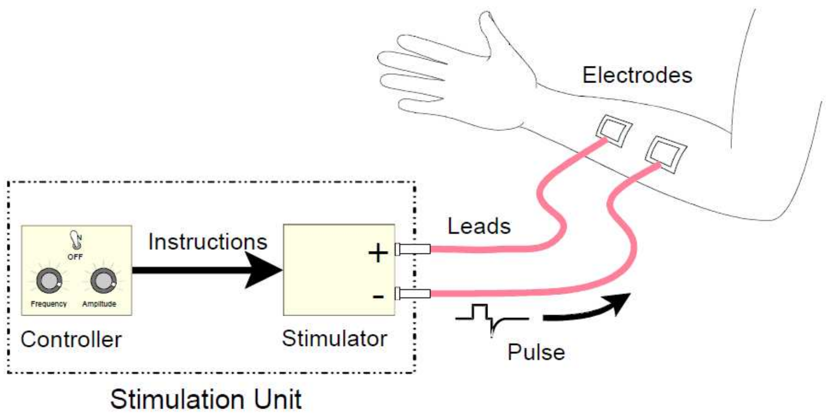

Figure 1.3: A typical FES system [18]

Presently, the only surface FES walking device with FDA approval is the Parastep. It can provide up to 6 channels of stimulation (3 muscles on each leg). Some individuals need only 4 channels where bilateral stimulation of the quadriceps results in knee extension, allowing the user to stand, stimulating the peroneal nerve causes a triple-flexor reflex response in the hip, knee flexors, and the ankle dorsiflexors resulting in a step. For some individuals, two additional channels (gluteal muscles) are used to enhance hip extension and stability during standing and walking [19]. The FDA approved Parastep-1 system represents technology that is more than four decades old. Surface FES systems have only been modestly successful in restoring upright stance and gait in people with SCI [20].

Researchers at Case Western Reserve University (CWRU) and the Department of Veterans Affairs (VA) developed an implanted FES system that activates the trunk, hip, knee, and ankle muscles to allow users to exercise, stand, and maintain an upright posture

[21]. Another study at the Cleveland FES Center used an implanted FES system (8-channel) to enable gait in an individual with incomplete SCI who could stand volitionally but not step. Post training, they achieved a walking speed of 0.2 m/s and a distance of 309 m thereby allowing limited community ambulation [22]. In another study, Kobetic et al. showed that a 16-channel FES system is feasible for repeatable short distance, independent, walker-support walking in paraplegia [23]. A study by Johnston et al. showed that an 8-channel implanted FES system provided enhanced functional abilities over traditional lower limb brace and reduced the need for physical assistance by a caregiver [24].

FES systems need good trunk control and a strong upper body because effort is required from the arms of the user to maintain balance. A challenge for all FES systems, with and without supplemental orthotics, is the necessity to generate torque about many lower limb joints using artificially activated muscle. The body uses sixty muscles acting on six multi degree-of-freedom joints during gait [25]. FES systems using implanted electrodes can activate a subset of these muscles, while systems using surface electrodes are limited to even fewer channels. Two other limitations of FES-aided gait systems are the rapid muscle fatigue that results from artificial muscle activation which in turn limits standing time and walking distance, and the inability to precisely control joint torques which leads to erratic stepping trajectories [26]. All FES systems require some sort of walking aid (rolling walker, parallel bars etc.) for walking and standing. The muscles that need to be stimulated should be strong enough to produce the required energy; this may require strengthening exercises or a training regimen.

1.4 Passive Orthoses

An orthosis is defined by the International Standards Organization as “an externally applied device used to modify the structural and functional characteristics of the neuromuscular and skeletal system”. Depending on the joints supported or covered the orthosis could be AFO (Ankle Foot Orthosis) or KAFO (Knee Ankle Foot Orthosis) or HKAFO (Hip Knee Ankle Foot Orthosis). Such orthoses provide assistance by locking the joints in stance or by coupling the movements of the two legs during swing phase.

The reciprocating gait orthosis (RGO) is an unpowered lower extremity rehabilitation mechanism [27], [28] that facilitates lower limb rehabilitation training for paraplegic patients. For RGO to work, the patients change the position of their centers of gravity with the swing of their bodies. The RGO is simple but cannot provide effective passive training to those who have almost totally lost motor ability. A study using isocentric reciprocating gait orthosis (IRGO), consisting of two knee ankle foot orthoses connected together with a pelvic band and the bilateral hip joints connected with a reciprocal link which causes hip flexion on one side and extension on the other side and vice versa, showed that intensive gait training with the IRGO improved walking speed and distance walked by paraplegics [29]. Another study showed that significant increases in walking speeds and step lengths occurred following gait training with IRGO in children with myelomeningocele [30].

Even though RGO’s facilitate rehabilitation in people with SCI, the resulting gait is slow and exhausting [31]. Improved RGO’s like advanced reciprocating gait orthosis

(ARGO) [32] show significant differences in joint range of motions, speed walking, cadence, and step length between normal walking and walking by those with SCI.

1.5 Hybrid Devices

Hybrid systems that combine electrical stimulation with a lower limb passive orthosis have been developed to overcome the problems associated with pure FES systems. In 1989, a hybrid assistive system (HAS) which combined FES and an externally powered brace was developed. The HAS used a self-fitting modular orthosis and cybernetic actuators [33]. In 1993, another hybrid FES system (HFES) consisting of a conventional FES system and an active orthosis was proposed [34]. More recently, a hybrid device that could provide more than 4 hours of continuous operation with standard camcorder batteries was developed by Kobetic et al. [35]. Our lab has developed the Controlled Brake Orthosis (CBO), a hybrid FES-aided standing and gait system that contains computer regulated friction brakes at the knee and the hip [36], [37]. A feasibility study done on a hybrid FES system developed by Kurokawa et al. introduced quasi-passive efficient walking into walk training using FES. After 2 months of training, the SCI participants in the study were able to walk for 2-3 minutes with hand rail support. In 2017, a hybrid neuroprosthesis developed at Case Western Reserve University restored stepping in three individuals with complete SCI. Walking speeds varied from 0.03 to 0.06 m/s and cadences from 10 to 20 steps/min [38].

The CBO was evaluated and compared to conventional four channel FES-aided gait using four subjects with paraplegia. The results demonstrated significant reduction in muscle fatigue and improvement in trajectory control when using the orthosis combined with FES compared to using FES alone [37]. The CBO activates the withdrawal reflex to generate hip and knee flexion. The electrically initiated reflex is highly variable from person to person and from day to day. Moreover, the neural system gradually adapts to accommodate the withdrawal reflex diminishing its usefulness to drive gait over time. The reflex also generates rapid swing phase motion that is difficult to control for gait [36], [39]. To overcome the limitations of CBO, an energy storage orthosis (ESO) was designed by Rivard [40]. The ESO stored energy during stimulation and then transferred it to the hip and knee joints to accomplish gait, eliminating the need for reflex activation. A single channel of stimulation of the quadriceps was used and the orthosis was a combination of gas spring and a pneumatic energy transfer system. The design was simulated on ADAMS and a preliminary prototype was bench-tested to verify the results of the simulation. Following this, a second generation ESO device was designed which replaced the gas springs with elastic rubber bands. Clinical testing on one SCI user was done, however there were issues while storing energy during stimulation [41]. In pneumatics, most of the energy is stored in the final stages of compression. Hence, a slight misalignment in the orthosis would lead to a minute amount of energy storage. To address this issue, constant force gas springs were used as energy storage elements in the third-generation design. Preliminary assessment, done using simulation and prototype testing, showed that concept is feasible [42].

1.6 Project Objectives

There remains a need for a muscle-powered device that uses surface stimulation to provide reliable short-range walking. The device in this project addresses this need. It builds on work by Boughner [42], Kangude [41], [43], [44],Rivard [40], and Goldfarb [36], [37].

The FES energy storing orthosis (FES-ESO), where muscle energy released by stimulation of quadriceps is stored in gas springs and transferred to knee and hip joints to accomplish gait. This approach combines electrical stimulation and an exoskeleton in a novel way that simplifies the electrical stimulation because only one muscle is stimulated and uses a passive exoskeleton that is projected to be lighter than current motor-driven exoskeletons. The muscle-powered approach may also help prevent muscle atrophy. The objectives of the project were to design, fabricate, bench test the FES-ESO, and to evaluate the performance of the FES-ESO in one volunteer with SCI using the six-minute walk test as the outcome metric

The FES-ESO is meant to be an exercise machine rather than a walking machine. The purpose of the device is to enable the user to exercise near a wheelchair. Standing can improve bone health and exercise can help prevent leg muscle atrophy. FES-ESO users likely exercise on an FES bike. The FES-ESO is another type of exercise machine that allows limited standing and walking.

2 Design Description

2.1 FES-ESO Concept

Figure 2.1: Rendering of the FES-ESO

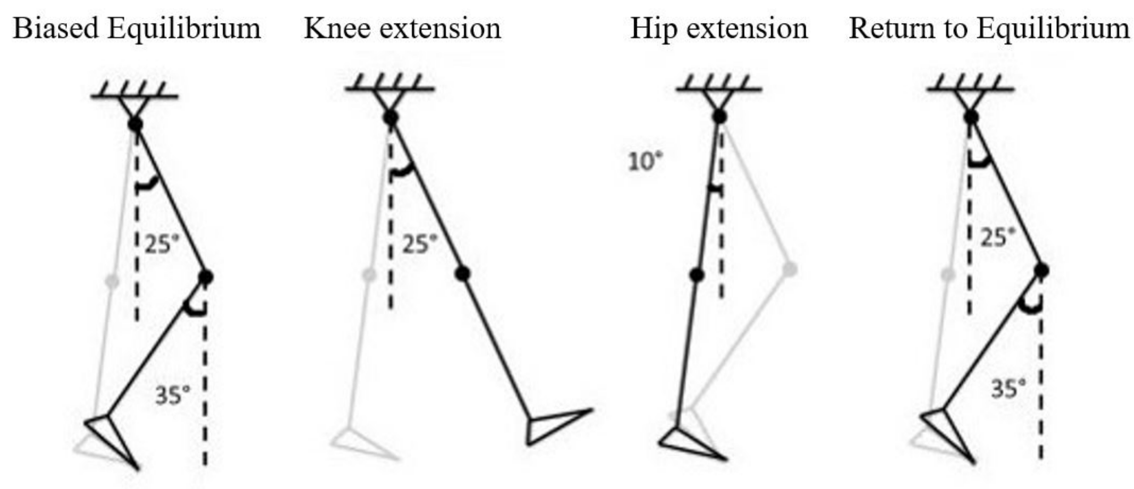

Figure 2.2: Phases of gait cycle of a user walking with FES-ESO

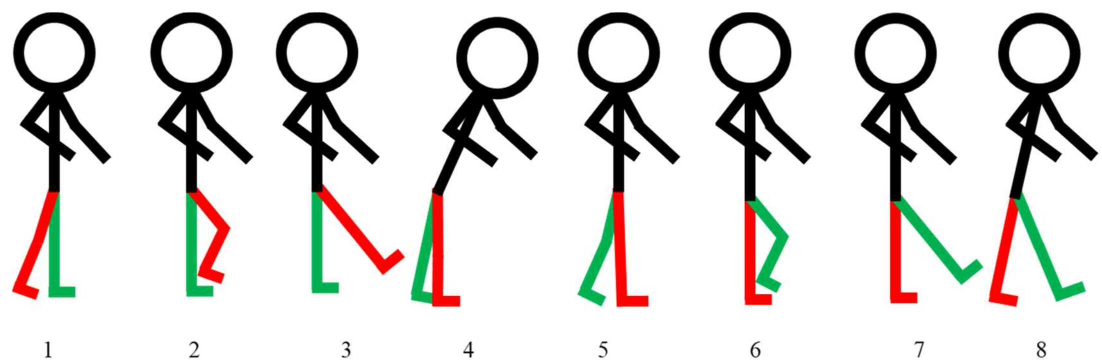

Figure 2.3: Stepping sequence of a user walking with FES-ESO

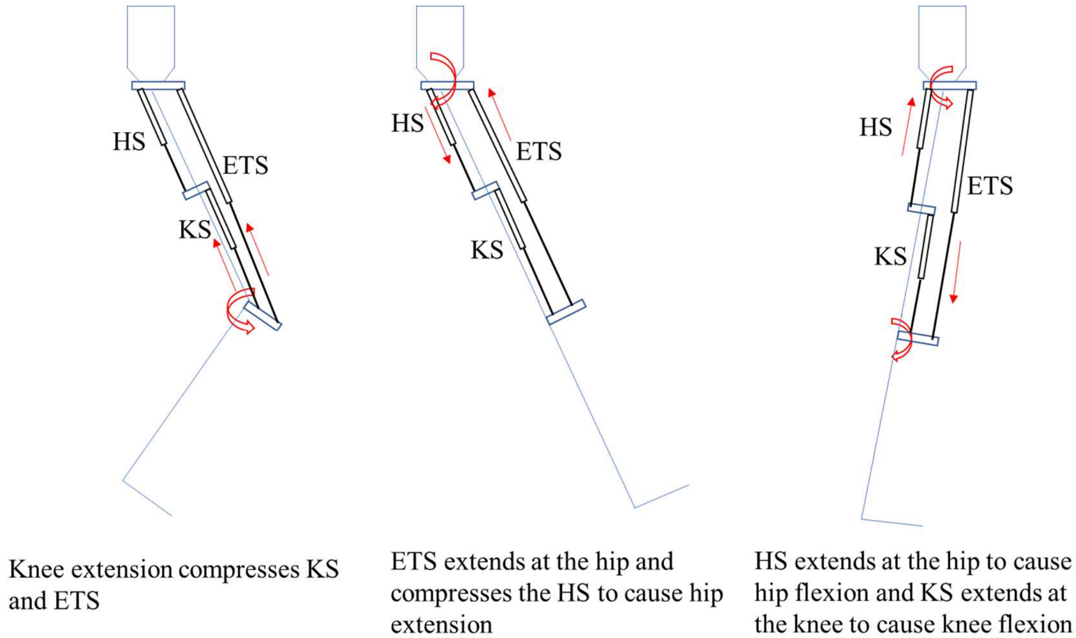

Figure 2.4: FES-ESO uses three gas springs (shown in black) - energy transfer spring (ETS), knee spring (KS), and hip spring (HS). The figure depicts how the gas springs store, transfer, and release energy to the joints

The FES-ESO is a muscle-powered exoskeleton that combines electrical stimulation with an energy-storing orthosis. The ESO uses gas springs as the energy storing elements and wrap spring brakes to lock and unlock the joints at specific angles. The device uses a single-channel of stimulation of the quadriceps as the energy source. When the quadriceps are stimulated, energy is stored in the gas springs which are a part of the ESO. This energy is then transferred to the hip and the knee joints in a coordinated manner to accomplish gait. S

The joint angles at each phase of gait using FES-ESO are shown in Figure 2.2. These angles are to serve as guidelines, they’re not meant to be achieved exactly. The step length would depend on the equilibrium state angles, where greater the flexion at the hip larger the step length.

The stepping sequence of a user walking with the device is shown in Figure 2.3. For the right leg (red color), when both the joints are unlocked, the device is in biased equilibrium where both the hip and the knee are flexed (shown as 2). At this stage, the quadriceps are stimulated resulting in knee extension (3). This is the phase where energy is stored in the gas springs. Following heel strike (4), energy is transferred to the hip causing it to extend (5-8). Finally, energy is transferred to the hip and the knee to bring it back to the equilibrium phase (2).

2.2 Design Requirements

Parameter	Value
Device weight	< 10 Kg
User height	1.6 m to 1.9 m
User weight	< 114 Kg
Body coverage	Over legs, around hips and lower torso
Hip range of motion	110 deg flex to 30 deg ext
Knee range of motion	95 deg flex to 2 deg ext
Holding torque hip	31 Nm
Holding torque knee	31 Nm
Don/Doff time	< 15 min

Table 2.1: Design Requirements

The critical requirements for the FES-ESO device are listed in Table 2.1. Indego is the lightest commercial exoskeleton and weighs 12 kg. The device must weigh less than 10 kg to be considered significantly lighter than the commercial exoskeletons. The height and restrictions were included to ensure a reasonable range of link lengths on the exoskeleton. The weight restriction ensured a reasonable structural strength of the exoskeleton.

Proper design of body attachment parts is crucial for the functioning of the exoskeleton. It is critical to limit the movement of the exoskeleton relative to the user's body. For this reason, coverage over legs, around hips and lower torso is essential.

The listed range of motion of hip and knee joints is required for the user to go from sit to stand and vice versa.

The knee joint needs to be locked upon heel contact after swing phase, and it needs to stay locked until the hip extends. While the hip is extending, the flexion of hip should remain locked; similarly, the hip joint should be locked in extension while flexing. 31 Nm is the amount of torque required to achieve this locking and was calculated by Kangude [41]. This holding torque at the hip and knee joints prevents collapse of the user. The brakes can hold when the user has some flexion at the hip and knee but are not designed to hold for a large flexion at the hip and the knee.

A don/doff time of 15 min is considered acceptable for a device at its early prototype stage.

2.3 Component Design

2.3.1 Wrap Spring Brake

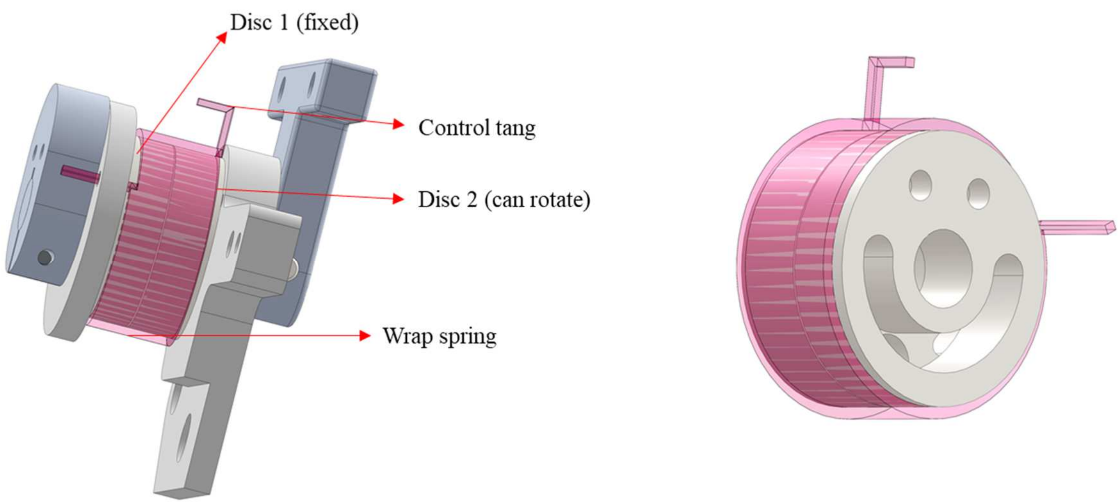

Figure 2.5: Working of a wrap spring brake

The ESO needs a braking mechanism at the hip and the knee joint to lock them at specific angles. Wrap springs were chosen as the braking component because of their high gripping torque to weight ratio. The wrap spring has two ends: a fixed end and a control tang. Consider two discs mounted on a fixed shaft where one of the discs (Disc 1 in Figure 2.5) is fixed and the other is free to rotate on the shaft (Disc 2 in Figure 2.5). The wrap spring is wound on the two discs. When disc 2 rotates in a direction that winds up the wrap spring on it, then friction between the discs and the spring creates a gripping torque that prevents rotation. If the disc rotates in the other direction the wrap spring will not wind on it, thus allowing rotation. Therefore, the wrap spring acts as a unidirectional brake.

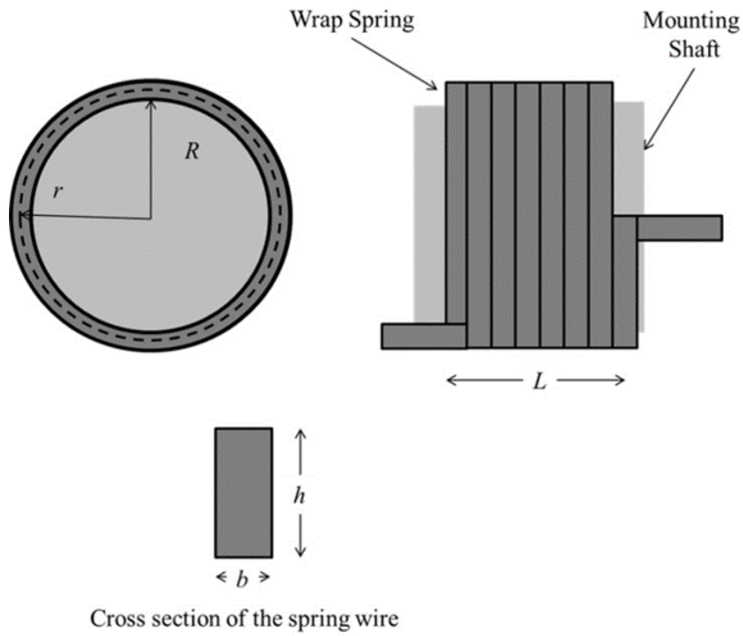

Figure 2.6: Wrap spring parameters

Gripping Torque Equation [45]

$$T_g = EIR\frac{r - r_0}{r_0 r^2}(\exp(2\pi N\mu) - 1)$$

Where R is the inner radius of mounted spring, r is the neutral axis of mounted spring, r_0 is the neutral axis of free spring, N represents the number of coils in length L, and μ is the coefficient of friction between spring and shaft.

The derivation of the equation is documented in Appendix A. The wrap spring parameters were optimized by Kangude [41]. The values of T_g for various values of coefficient of friction are listed in Table 2.2 where 0.6 is the μ for steel on steel, 0.4 is the μ for steel on aluminum, 0.1 is the μ for oiled steel on steel.

Coefficient of friction	T_g (Nm)
0.1	1.28E+01
0.2	8.56E+02
0.3	5.63E+04
0.4	3.70E+06
0.5	2.44E+08
0.6	1.60E+10
0.7	1.05E+12
0.8	6.93E+13
0.9	4.56E+15

Table 2.2: Exponential dependence of gripping torque (T_g) on coefficient of friction

Due to exponential dependence on the frictional coefficient (μ), the gripping torque is sensitive to friction between the discs and the wrap spring so much so that oil on the disc can cause the brake to fail as the T_g becomes 13 Nm.

Servo Motors

Figure 2.7: Hitec HS-35HD Ultra Nano Servo

The Hitec servo (Figure 2.7) was picked to control the locking and unlocking of the wrap springs. These servos are light (4.5g), compact, and they have a fast response time of 0.1 sec. When the servo is actuated it displaces the control tang of the wrap spring causing the joint to unlock. The force required to displace the control tang was measured using a force gauge and the average of 5 trials indicated a value of 9.6 N.

The moment arm required to displace the control tang was determined by using the simple torque equation: Moment arm = Torque / Force, where torque is taken as 80% of the servo's maximum torque rating and the force was experimentally determined to be 9.6 N. From this equation, it was found that the moment arm required to displace the control tang was 6.5 mm.

2.3.2 Gas Springs

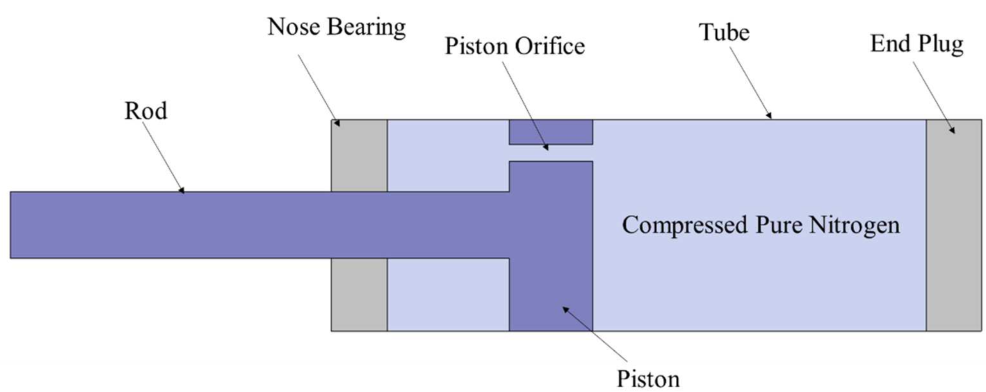

Figure 2.8: Components of a gas spring

A gas spring (Figure 2.8) is a system consisting of a pressure tube, a rod, and a piston. The energy for the spring is provided by gas at high pressure. The whole system is self-contained and sealed against loss. Gas springs expand and contract smoothly, they're compact and last several years without any maintenance. They also exert constant force which is desirable for an exoskeleton application. For these reasons, they were picked as the energy storing elements of the ESO.

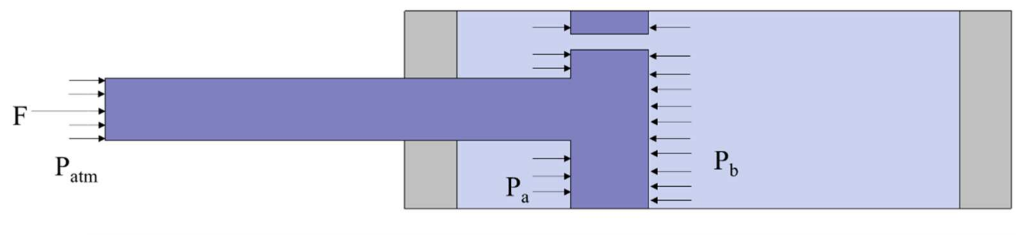

Figure 2.9: Forces in a gas spring

$$F = (P_b \times A) - (P_a \times A) + (P_a \times a) - (P_{atm} \times a)$$

In a standard gas spring, the pressure is the same on both sides of the piston

$$P_a = P_b = P$$

The equation can then be reduced to:

$$F = (P - P_{atm}) \times a$$

where F is the force expressed in Newton (N), P is the gas pressure (N/m^2), P_a is the gas pressure in chamber a (N/m^2), P_b is the gas pressure in chamber b (N/m^2), P_{atm} is the external pressure in Pascal (N/m^2), a is the rod cross section in square metres (m^2), A is the Piston cross section in square metres (m^2).

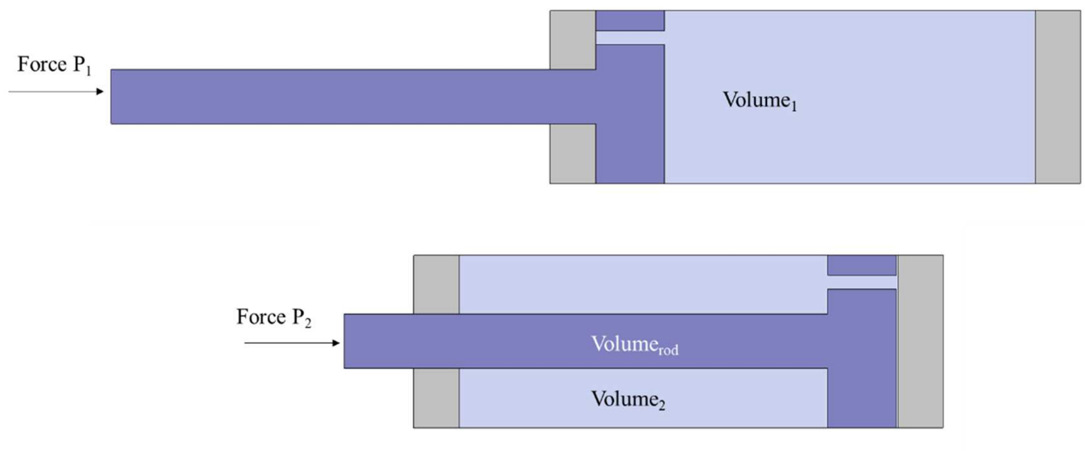

Figure 2.10: Force exerted by a gas spring is not perfectly constant

The force exerted by a gas spring at the beginning of compression is different from the force at the fully compressed state. The overall pressure in the cylinder increases during compression because of the additional volume of rod that enters the cylinder during compression. The smaller the cross section of the rod, the smaller the difference between the forces at the beginning and the end of compression. This difference in forces is defined by K factor.

$$Volume_2 = Volume_1 - Volume_{rod}$$

$$K\ factor = P_2/\ P_1 = Volume_1/\ Volume_2$$

Where P_1 is the force of a fully extended spring, P_2 is the force of the same spring when compressed. In general, the K factor for standard gas spring is between 1.2 and 1.4.

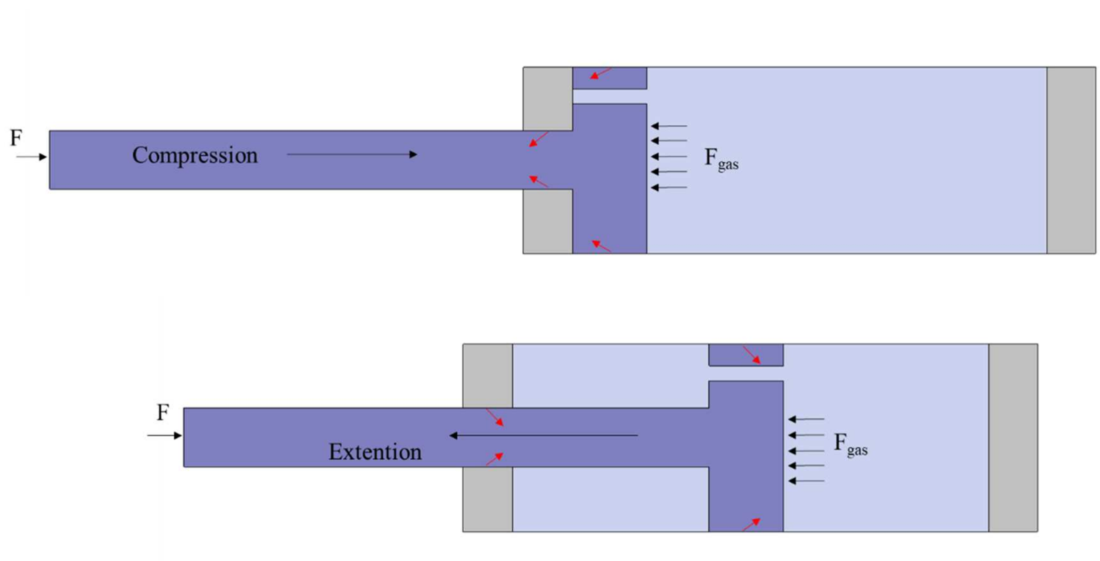

Figure 2.11: Force during compression and extension are different

When a gas spring is being compressed the frictional force due to rod seal is significant and works against the direction of motion when the piston is moving. This leads to a difference in force exerted during compression and extension for the same gas spring. For instance, a 70 lbs gas spring on McMaster Carr (9416K19) has a compression force rating of 91 lbs because of the need to overcome the friction. This needs to be factored into account during the design of the ESO as the springs cycle through extension and compression during different phases of gait.

The ESO uses 3 gas springs for each leg – energy transfer spring (ETS), hip spring (HS), and the knee spring (KS). Stimulation of quadriceps stores energy in the ETS and KS. The ETS then transfers the energy to the hip joint and compresses the HS causing hip extension. Finally, HS transfers energy to the hip joint and KS transfers energy to the knee joint causing both joints to flex and leading to the equilibrium state. A few conditions need to be satisfied for the system to work. The quadriceps should be able to provide enough torque to compress ETS and KS during knee extension. Also, this torque requirement should be kept as low as possible to reduce muscle fatigue. The ETS should be capable of compressing the HS during hip extension. The HS should be strong enough to flex the hip against gravity and the KS should be strong enough to flex the knee.

MATLAB scripts (Appendix B. Gas Spring Selection) were created to help pick the right combination of springs that would satisfy the conditions listed above. These scripts calculated the torques at the hip and the knee joint at each instant during the gait cycle. The hip joint experiences torque due to gravity, ETS, and HS; whereas the knee joint experiences torque due to gravity, ETS, and KS. The analysis was done for a user who is 180 cms tall and weighs 67 kgs. The anthropometrical measurements were taken from [46].

A gas spring rated 15 lbs could flex the knee to the desired position during return to equilibrium phase (Figure 2.12).

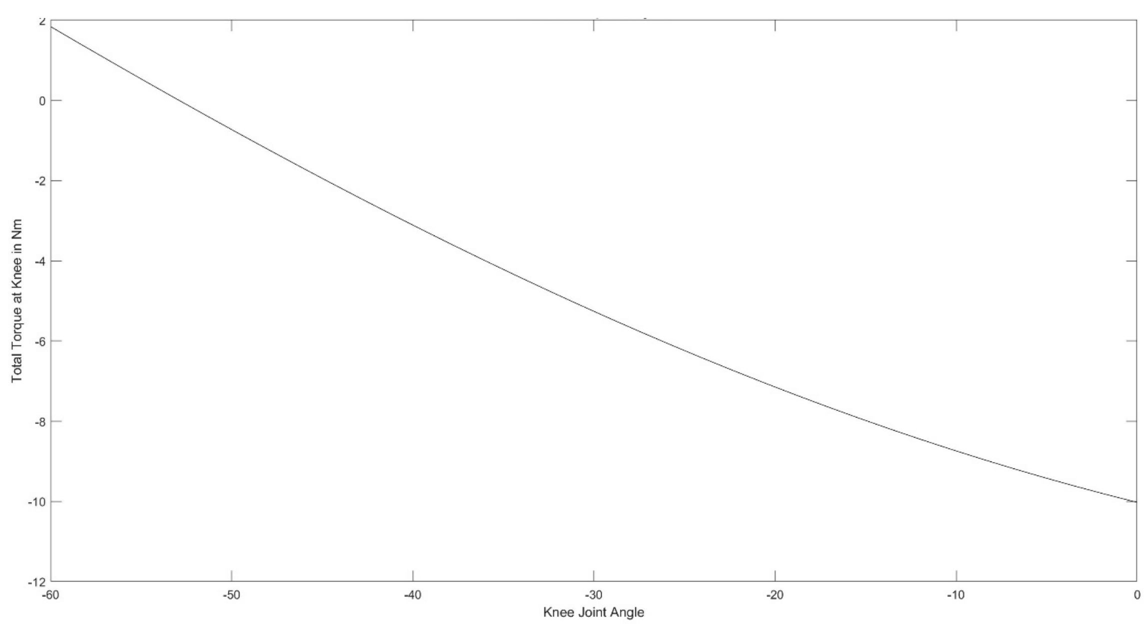

Figure 2.12: A knee torque analysis during the back to equilibrium phase shows that torque changes direction at 53 deg knee flexion which means that 15 lbs KS can flex the knee to a joint angle of 53 deg

For a KS rating of 15 lbs, the script specified several combinations of ETS and HS that would work (Table 2.3). There is a tradeoff between the hip flexion angle that can be achieved and the FES torque. As the hip flexion angle increases, the FES torque requirement also increases. The step length is also dependent on hip flexion angle, greater the flexion angle larger is the resulting step length. With a goal to have just enough hip flexion for the foot to clear the ground during knee extension even if this results in small step lengths, a HS and ETS of 30 lbs was chosen among the ones listed in Table 3.2. This was based on bench tests and testing done with a non-impaired individual. The resulting FES torque requirement is 23 Nm.

HS Force (lbs)	ETS Force (lbs)	Hip Flexion (deg)	FES torque (Nm)
15	15	7.5	17
20	20	8	19
30	30	10	23
40	40	12	27
50	50	13.5	30
60	60	15	34

Table 2.3: Gas spring combinations, their resulting flexion angles, and FES torque for a KS of 15 lbs

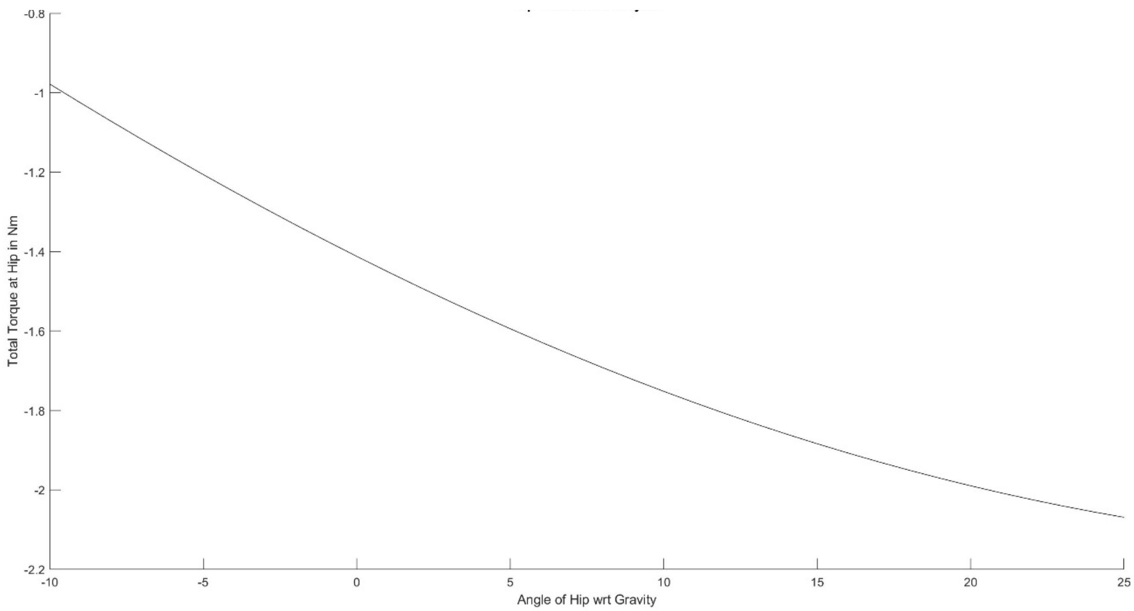

Figure 2.13: An analysis of the total torque at the hip during hip extension phase shows that a 30 lbs ETS can compress the 30 lbs HS.

The FES-ESO requires the user to use upper body strength to assist hip extension. During hip extension, ETS can only compress and store energy in the HS. If the ETS had to be strong enough to lift the user's torso against gravity, the FES torque requirement would be much greater than 50 Nm thereby making it non-feasible. Figure 2.13 shows that total torque at the hip remains negative (extension at the hip is taken as the negative direction). This means that the chosen ETS (30 lbs) can compress the HS (rated 30 lbs but compression force is 40 lbs).

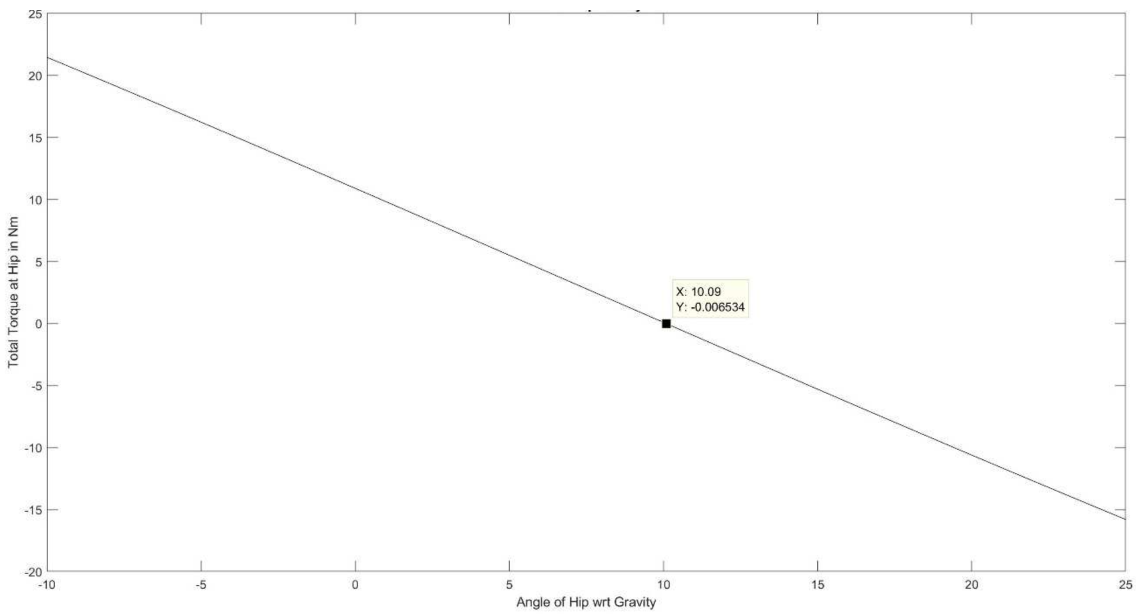

Figure 2.14: Total torque at hip during back to equilibrium phase changes direction at a flexion angle of 10 deg wrt gravity, which means that a HS of 30 lbs can flex the hip to 10 deg wrt gravity

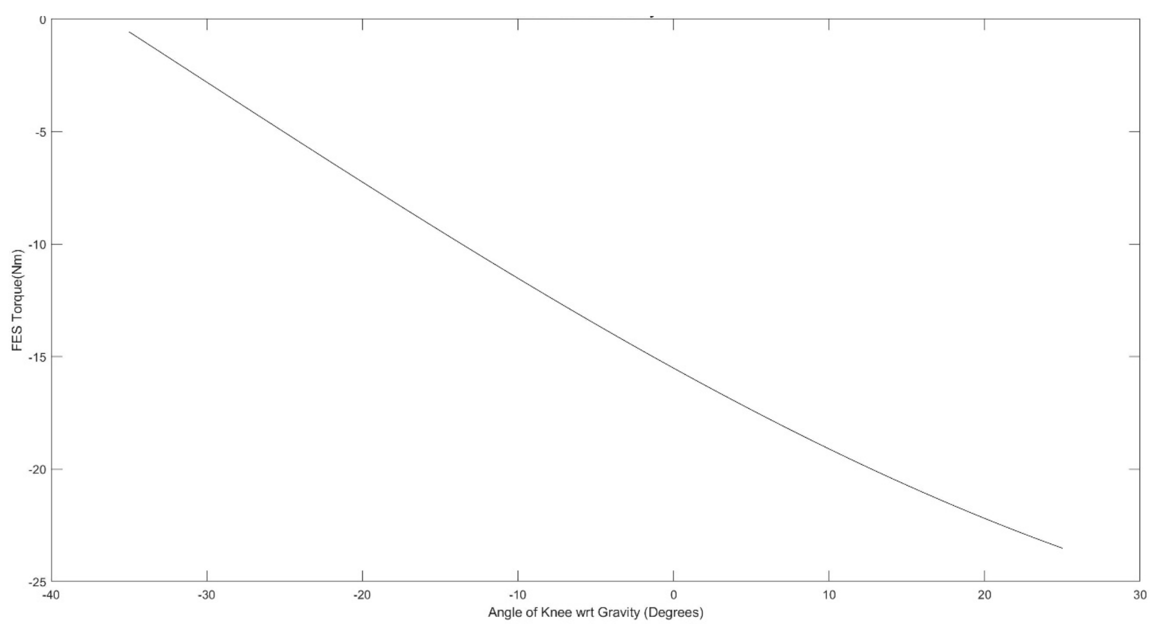

Figure 2.15: The analysis of knee extension phase for KS of 15 lbs, HS and ETS = 30 lbs show that 23 Nm of FES torque is required to store energy in the gas springs

Figure 2.14 shows that the hip can flex to about 10 deg with respect to gravity before the total torque reverses direction. Figure 2.15 is the result of analysis at the knee joint during knee extension. For KS of 15 lbs, HS and ETS of 30 lbs, the peak FES torque required to compress the ETS and KS resulting in stored energy is 23 Nm.

2.3.3 FES-ESO Structure

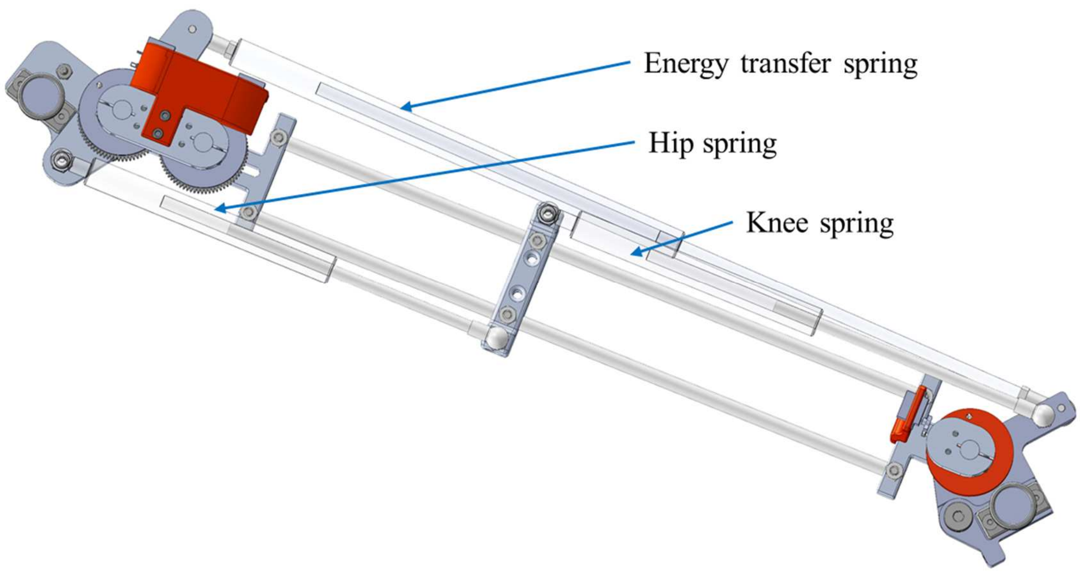

Figure 2.16: Rendering of the thigh component of the ESO

The thigh component (Figure 2.16) consists of the hip joint, the knee joint, and the gas springs attached on a structural aluminum frame. The gas spring attached between the hip and the knee is the energy transfer spring, the one between the middle of the thigh and the knee is the knee spring, and the one between the middle of the thigh and the hip is the hip spring.

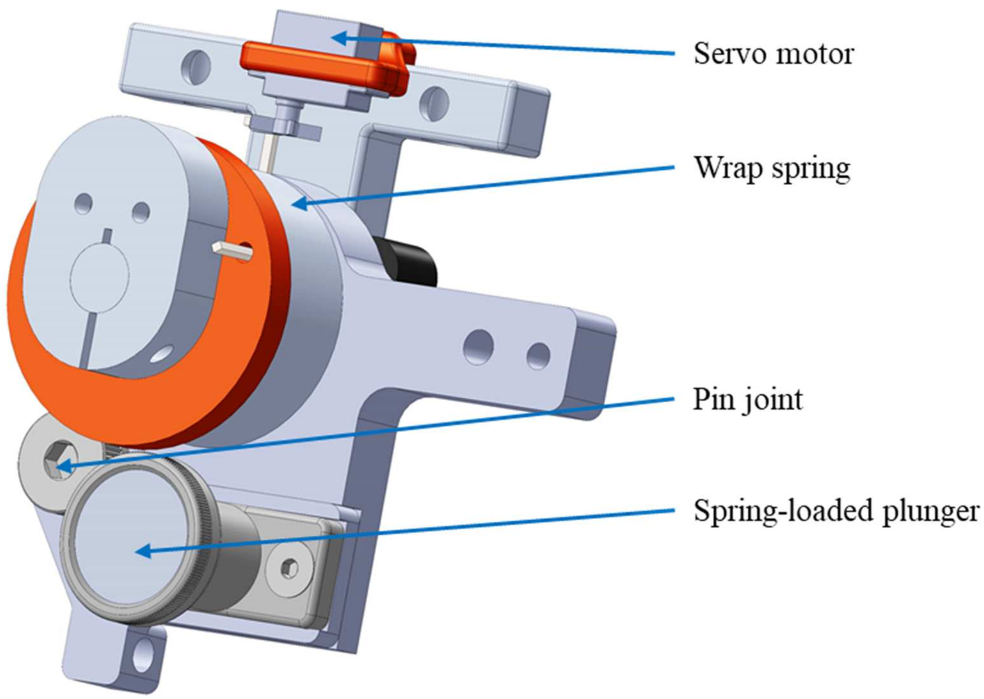

Figure 2.17: Rendering of the knee joint of the ESO

The knee joint is shown in Figure 2.17. A custom wrap spring (Reell Precision Manufacturing Corporation) is used for the braking mechanism with the dimensions of the spring optimized for this application [41]. The braking device has two ends: a fixed end and a control tang. The control tang is actuated with a servo motor that unlocks the brake to allow rotation. The wrap spring acts as a brake only in one direction so the joint is free to rotate in the other direction. Therefore, the knee joint has a unidirectional brake.

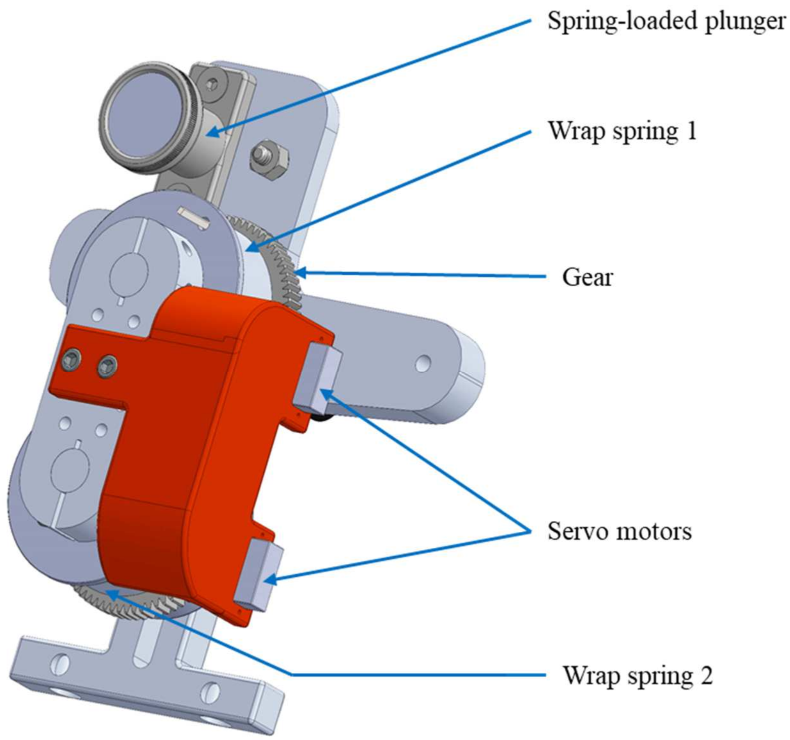

Figure 2.18: Rendering of the hip joint of the ESO

The rendering of the hip joint is shown in Figure 2.18. It uses two wrap springs coupled together with gears for a bi-directional braking mechanism. When the servos are off, the hip joint remains locked at a specific angle. Actuating one servo allows rotation in one direction and actuating the other servo allows rotation in the other direction.

The knee and the hip have a pin joint that can be unlocked by pulling a spring-loaded plunger away from the joint along its axis. This provides an additional degree of freedom, allowing the hip and the knee to decouple from the gas, which allows the user to sit while wearing the device.

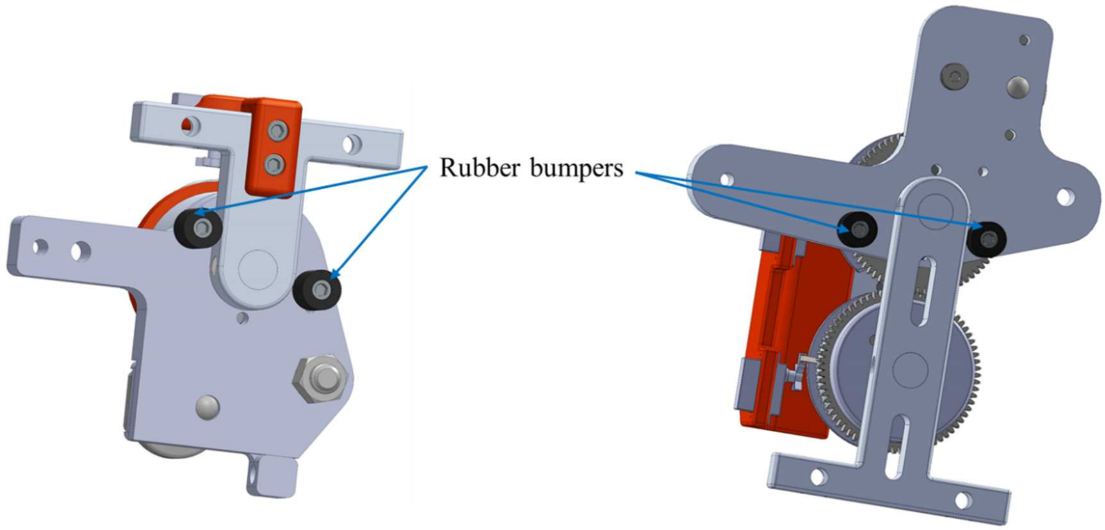

Figure 2.19: Rubber bumpers at the knee (left) and hip joints (right) act as rotational hard stops to constraint joint rotations

Rubber bumpers at the knee and hip joint constrain the joint rotations to a specific range thereby defining the equilibrium angles. They also act as an additional safety measure to prevent the collapse of the user in case the wrap spring brakes fail.

Body Attachment Parts

Torso Attachment

Figure 2.20: MOLLE military backpack

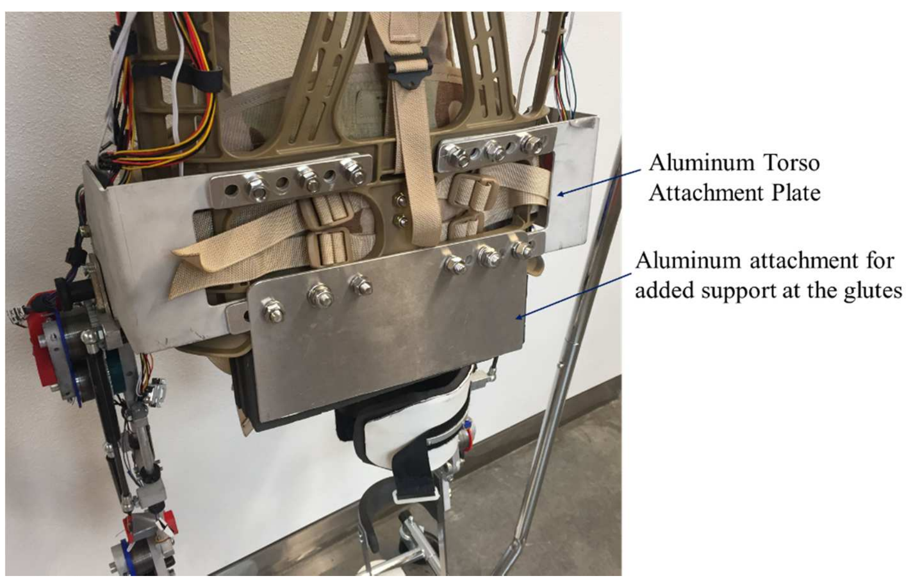

Figure 2.21: Aluminum plate to attach the backpack and the orthosis

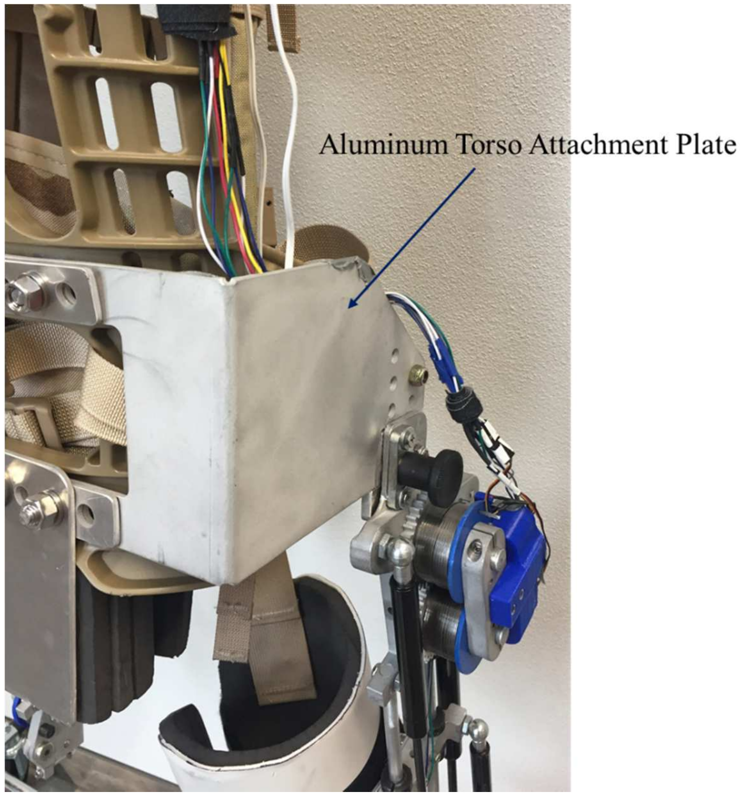

Figure 2.22: Closeup view of the aluminum torso attachment plate

A MOLLE military backpack frame (Figure 2.20) with shoulder straps and a hip strap was used to attach the orthosis to the user's torso. The purpose of the backpack is to provide support and stability to the user's torso while using the ESO. An aluminum attachment was added to the backpack to provide support to the user's glutes (Figure 2.21). This would prevent the user from slipping through the backpack at their hip. Figure 2.22 shows the aluminum plate that was used to connect the backpack to the orthosis.

Thigh Attachment

Figure 2.23: Thigh attachment part

The thigh attachment part (Figure 2.23) had two aluminum cuffs; each of these cuffs was 1.5” wide to provide rigidity. A long orthoplast sheet which covered the knee cap was used. The orthoplast sheet was lined on the inside by 0.5” thick padding. The orthoplast was essentially a thigh pad which was donned before donning the exoskeleton. After donning the exoskeleton, the thigh pad was secured to the structure of the ESO by using a Velcro strap.

Calf and Foot Attachments

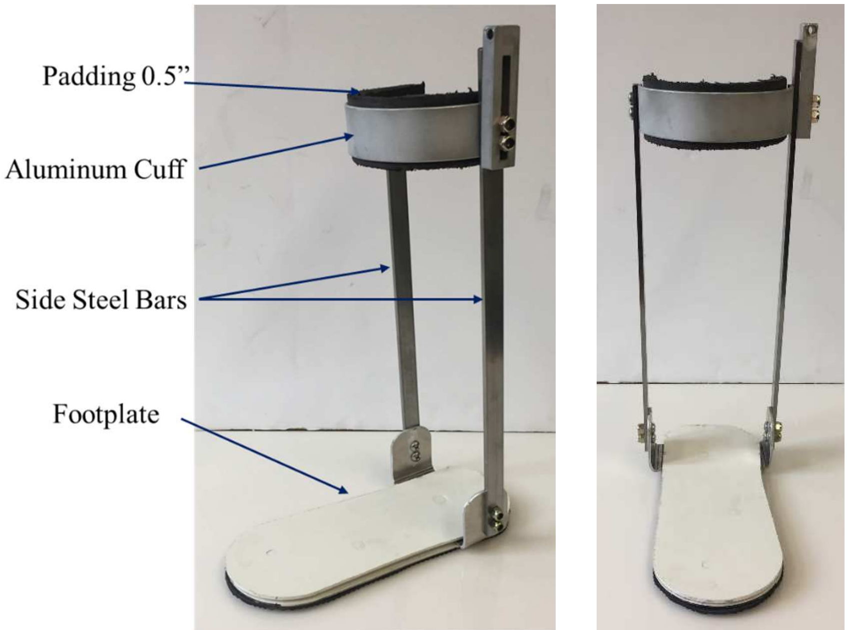

Figure 2.24: Calf attachment part version 2

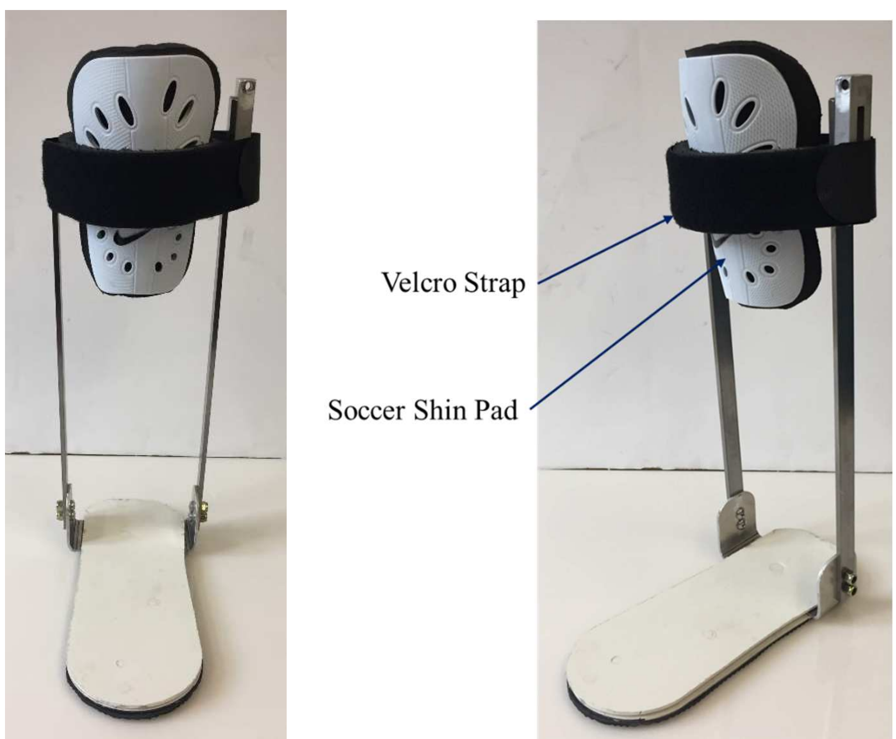

Figure 2.25: Calf attachment part with the shin pad

The design of calf attachment part was like an off the shelf metal AFO. An Aluminum cuff was connected to two side steel bars, of which the one on the lateral side was connected to the knee joint. The steel bars extended down the calf and connected to a footplate (Figure 2.24). The inside of the Aluminum cuff was lined with padding. A soccer shin pad goes on the inside of the cuff and an external Velcro strap is used to securely attach to the user's calf (Figure 2.25).

The footplate was made by sandwiching a bent Aluminum plate between two orthoplast pieces (Figure 2.26 and Figure 2.27). An adhesive was used to attach these pieces to one another. Finally, an adhesive was used to fasten a rubber shoe grip material to the bottom of the orthoplast material. This provides grip while walking.

Figure 2.26: Aluminum foot plate

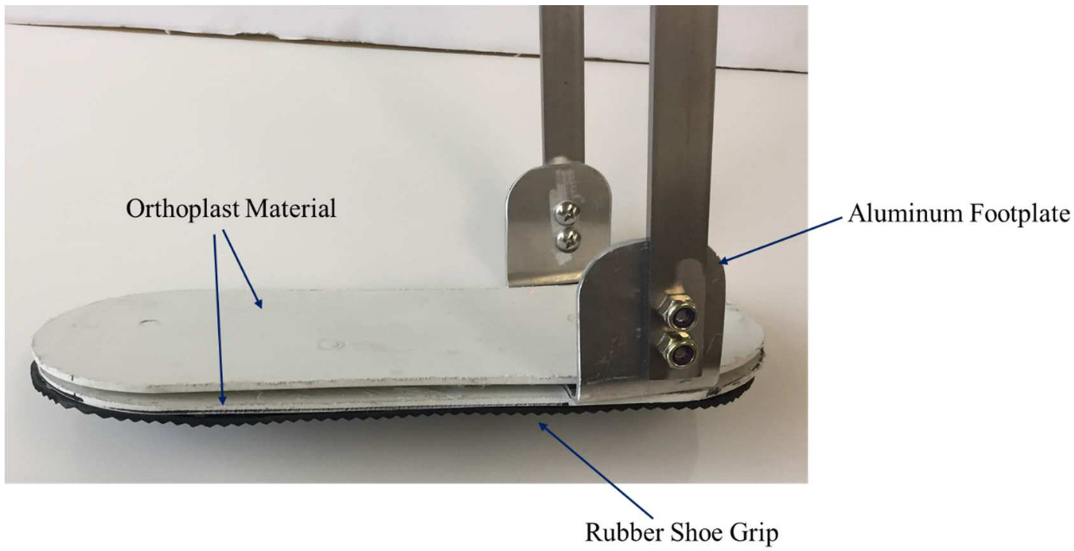

Figure 2.27: Footplate made by sandwiching an Aluminum plate between two orthoplast plates

2.4 Electronics

The electronics for the ESO need to actuate the servos that control the locking and unlocking of the joints, create and control the stimulation pulses to contract the quadriceps. An open loop control has been implemented with a specific timing of actuation of the servos and stimulation pulses.

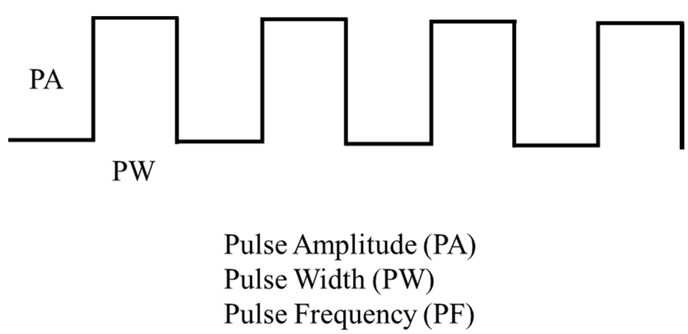

Figure 2.28: Theoretical stimulation pulse

The stimulation pulse needed to contract the quadriceps is shown in Figure 2.28. Pulse amplitude, width, and frequency are parameters that need to be controlled. A DAC (MCP4821) is used to control the amplitude of the pulse (shown in Figure 2.29). The output of the DAC is connected to a switch (MAX323) which allows the creation of a pulse with a specific width and frequency (Figure 2.29). The pulse width was 300 μs and the frequency was 40 Hz. The parameter of interest is the current passing through the electrodes placed on the quadriceps and not the voltage. To convert the constant voltage pulse into a constant current pulse, the output of the switch is sent to an operational amplifier (Figure 2.29). This constant current pulse flows from the secondary winding of a transformer to ground via a sense resistor. The transformer then converts low voltage, high current signal to a high voltage low current signal which is then sent to the electrodes placed on the quadriceps. We need signals with a current of around 100 mA which is why a transformer is essential. The output of the circuit was connected to an oscilloscope to verify that the pulses produced were accurate (Figure 2.30). During actual stimulation a string of pulses needs to be sent. However, the beginning can't be abrupt so there needs to be a ramp up and a ramp down. The slopes of these ramps can be set and the ramps are implemented by varying the pulse widths from a threshold minimum to the maximum (300 μs).

The servos are connected to an Adafruit servo driver which is conned to an Arduino metro mini. XBEE wireless is used to communicate wirelessly with the Arduino (Figure 2.29).

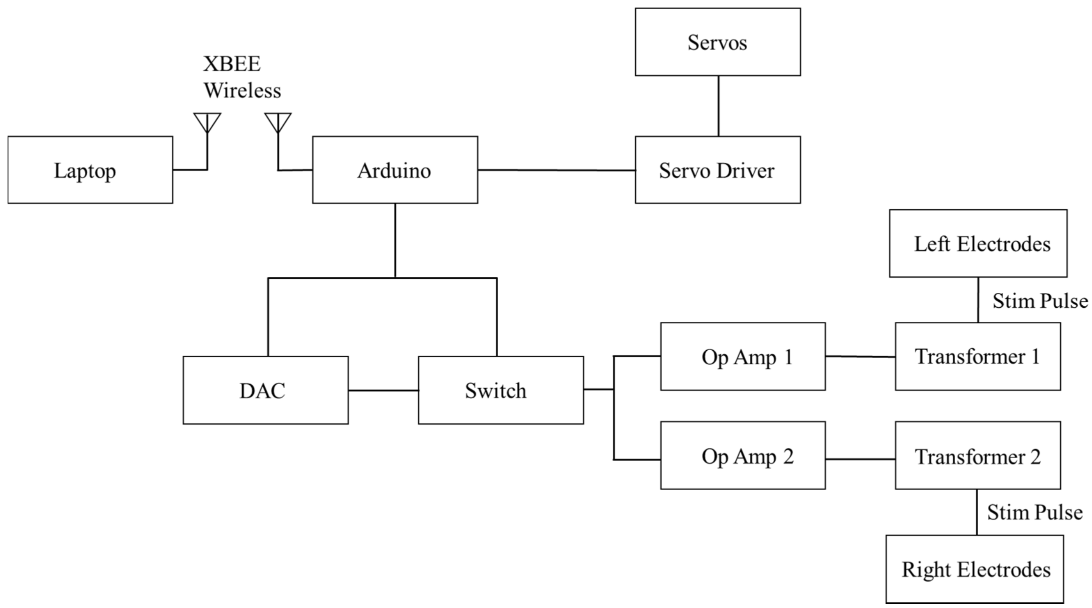

Figure 2.29: ESO electronics architecture

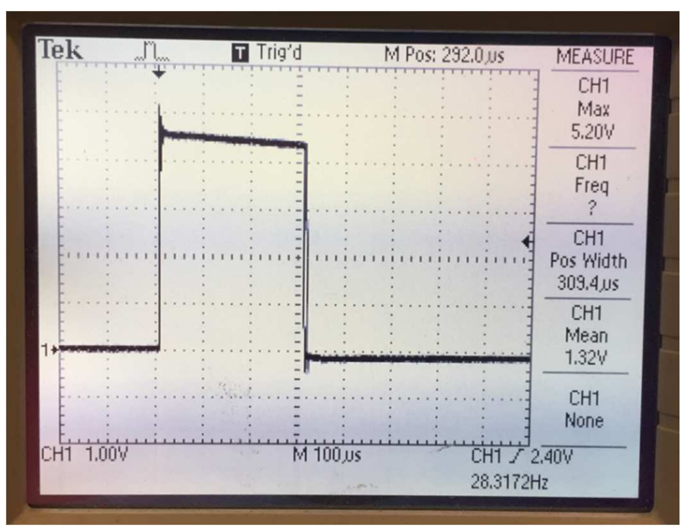

Figure 2.30: Single stimulation pulse on an oscilloscope

The Arduino side had functions to start and stop the actuation of the servos. It also had a function to send a single stimulation pulse and another that sent out a string of pulses with parameters such as pulse width, frequency, stimulation time etc. being passed to it. The Arduino constantly checked its serial port for any input bytes.

The GUI was implemented using Python (TkInter). The Python code sent a two-byte frame to the Arduino via the serial port. This frame included command and data. Command would tell the Arduino which function to call, and data would pass any information needed to that function (sendBYTES (cmd, data)). The software definition is described in the Appendix D. Electronics

The GUI (Figures 2.31 – 2.33) has multiple tabs. The first tab is a stimulation test window and is used to modify the parameters of stimulation to learn the right set to produce the required FES torque. On this tab, the user can set the pulse amplitude, frequency, width, stimulation time for the left and the right side by entering the values in the dialog boxes and clicking SetLP (for left). After setting the parameters, GoLStim button is pressed to output the desired stimulation pulse. The second tab has additional stimulation parameters that are used less frequently such as ramp up, ramp down slopes, and pulse width threshold. In the system test tab, individual servos can be actuated or stopped with a press of a button and the stimulation can be activated using a different button. The stim button on this tab uses the default parameters or the ones that were set in the first tab. The main tab is meant for two buttons, one for left step and the other for a right step. These buttons have inbuilt timings in the background, such as timings of actuation of various servos relative to stimulation (Figure 2.34).

The timeline of events occurring during a right step is shown in Figure 2.34. After the right step button is pressed, the right hip flexion is unlocked and after a delay of T2 the right knee is unlocked. This causes the hip and the knee to flex to attain the equilibrium state. Following this the right leg is stimulated, the hip and the knee are locked. This causes the knee to extend, thereby storing energy in the gas springs. The stimulation is then turned off. Then the right hip extension is unlocked for an interval of T7 following which it is locked. The time intervals between each event (T1- T7) and the stimulation strength are categorized into primary, secondary, and tertiary parameters depending on how often they need to be modified (Figure 2.34).

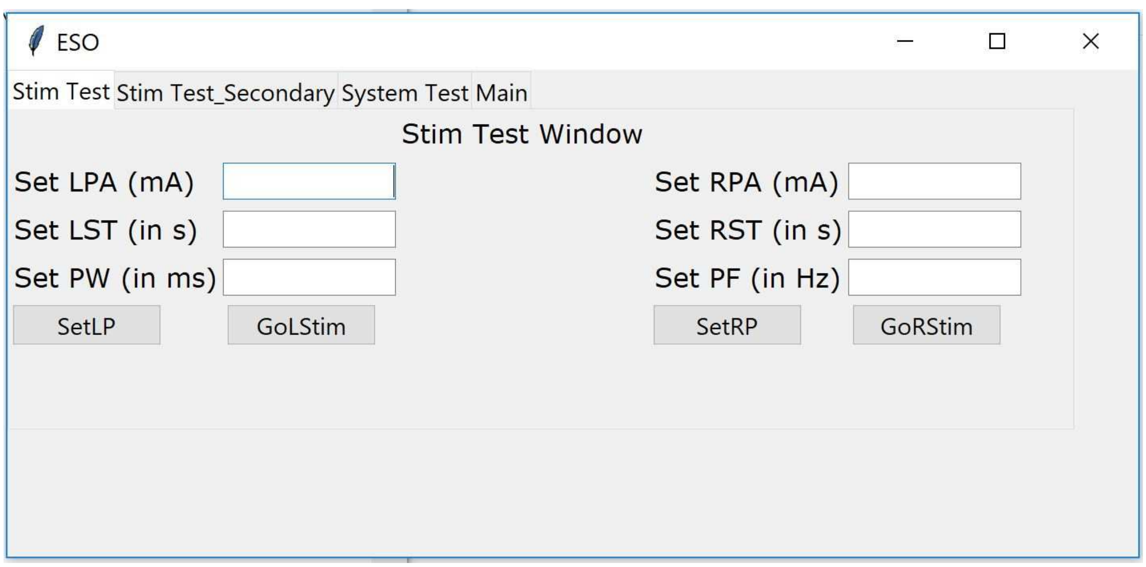

Figure 2.31: ESO GUI stim test tab

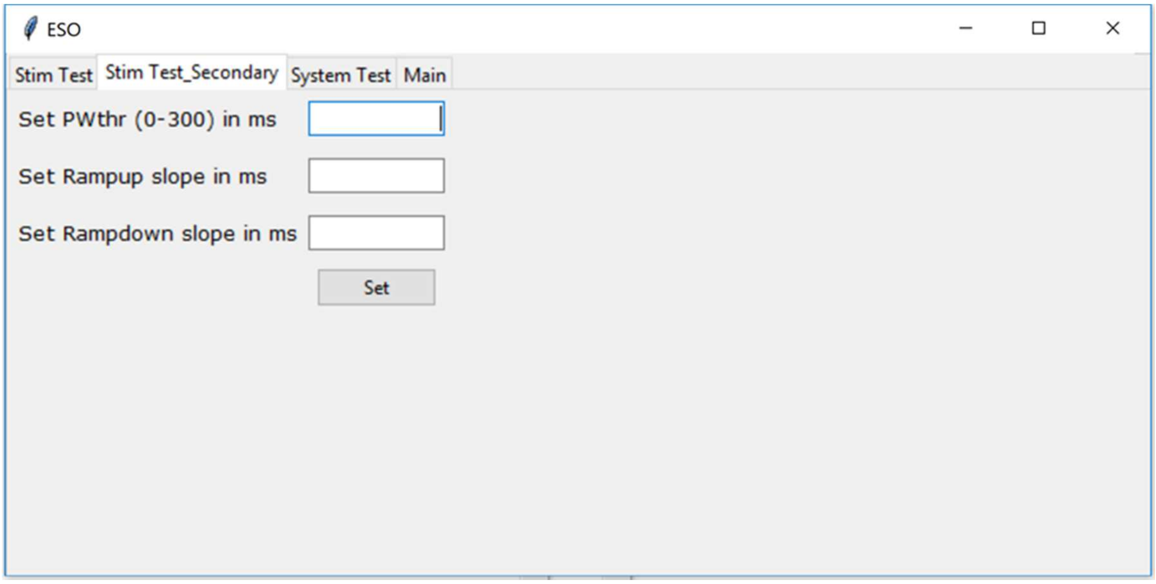

Figure 2.32: ESO GUI stim test_secondary tab

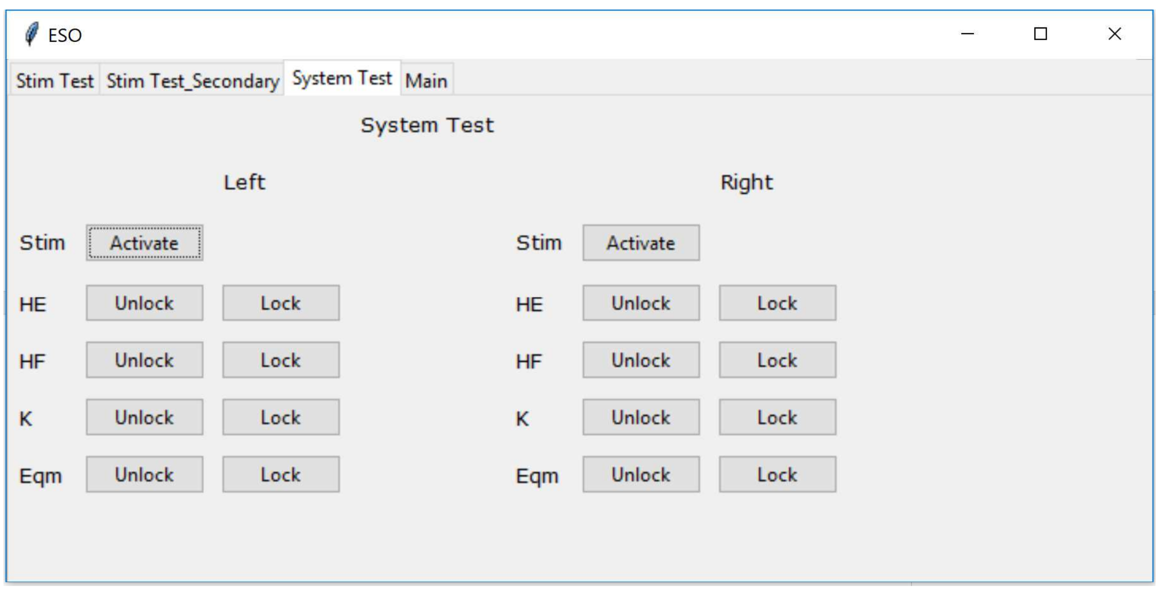

Figure 2.33: ESO GUI system test tab

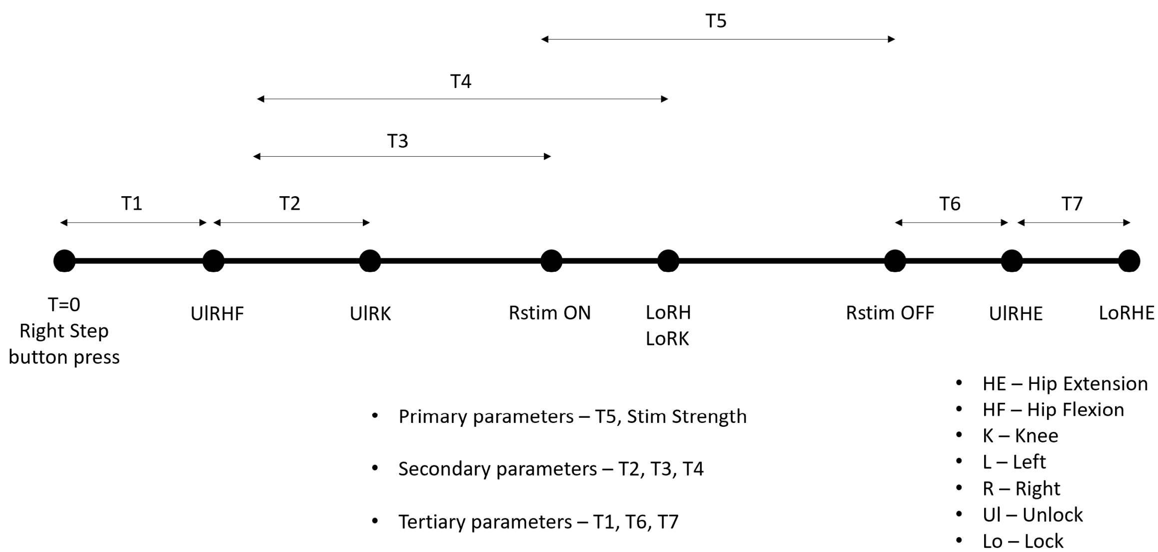

Figure 2.34: Timeline of events for a right step

3 Evaluation

3.1 Engineering Bench Tests

3.1.1 Wrap Spring Torque Tests

Objective

The hip and knee joints were tested to ensure they could hold at least 31 Nm of torque.

Methods

The test involved hanging weights off a moment arm connected to the joint and increasing the weights to see if the joints failed at a torque of 31 Nm. The moment arm was 6.8 cm at the knee and 7.4 cm at the hip. While loading the joint, the change in angle at the joint was recorded using a goniometer to measure the wrap spring slip. For example, first the joint was loaded with 1 kg and the angle was measured, then the load was increased to 5.7 kg and the angle was measured again. Each plot marker in Figure 3.2 indicates the torque due to load that at instant and the change in angle relative to the starting state (angle of slip).

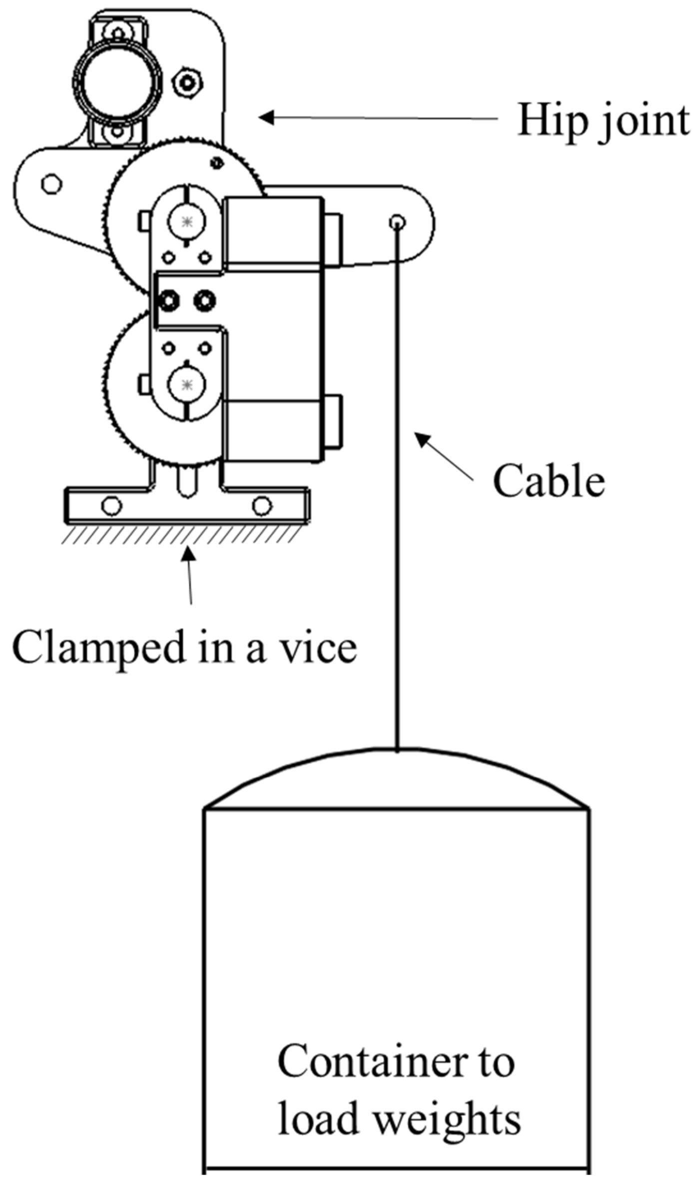

Figure 3.1: Hip joint wrap spring brake torque test

<u>Results</u>

The knee and the hip joints held a load of 50 Kg. This meant that the knee joint could hold at least 33.26 Nm and the hip joint could hold at least 36.17 Nm of torque. When the hip joint was loaded incrementally to around 53 kgs, the joint slipped by 8 deg (Figure 3.2).

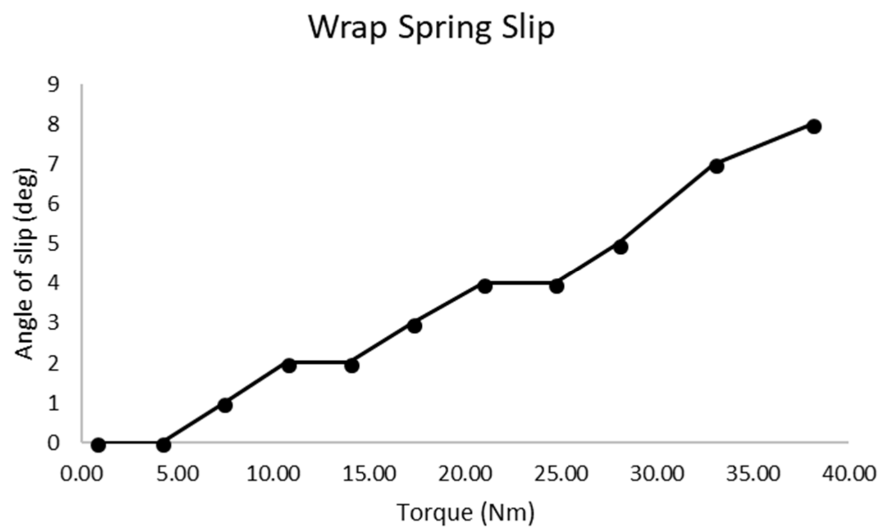

Figure 3.2: Wrap spring slip as the load varies

<u>Discussion</u>

The wrap spring brakes could hold the amount of torque required to prevent collapse of the user while standing. The slip of the wrap spring brake is its inherent characteristic. Each time the joint is loaded, the spring allows some movement before it coils down on the joint and stops the motion. This means that locking the joint at an exact angle is not possible with wrap spring brakes. However, to accomplish a gait cycle it is not necessary to accurately achieve the prescribed angles mentioned in Figure 2.2 which is why wrap spring brakes would work for this application.

3.1.2 Bench Top Gait Cycle Tests

<u>Objective</u>

Bench top tests were conducted to ensure that the ESO could achieve the three states during gait mentioned in Figure 2.2 (Equilibrium, Knee extension, Hip extension).

Methods

The prototype was hung on a wooden board. Weights were attached to the thigh and shank segments to simulate the weight of the user's leg. The knee was extended manually to go from equilibrium state to knee extended state. Then, the control tangs of the wrap springs were manually actuated to unlock the joints and transition from one state to another. The angles of the joints were measured using a goniometer.

Results

The prototype was able to achieve the states mentioned in Figure 2.2. The hip flexion angle in equilibrium state was 14 deg which matched the theoretically expected angle of 15 deg for a gas spring combination of 60 lbs (ETS), 60 lbs (HS), and 15 lbs (KS) (Table 2.3).

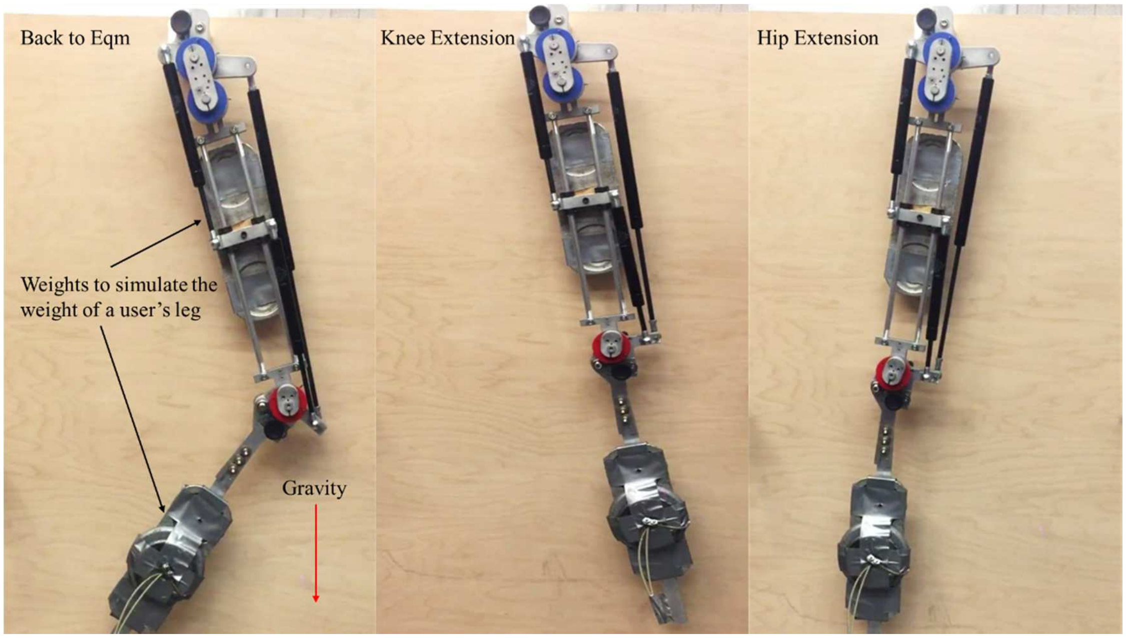

Figure 3.3: Bench top gait cycle run through

Component	Weight (kgs)
Left shank (including ankle)	1.5
Right shank (including ankle)	1.5
Left shin pad (padded)	0.05
Right shin pad (padded)	0.05
Left thigh cuff (orthoplast and padding)	1.45
Right thigh cuff (orthoplast and padding)	1.45
Left thigh segment	3.15
Right thigh segment	3.2
Backpack	3.95
Electronics box	0.75
Total	**17.05**

Table 3.1: Weight measurements of components of FES-ESO

Discussion

Bench top evaluation showed that the ESO was working as expected.

3.1.3 Non-impaired User Tests

Objective

Preliminary tests were conducted on a non-impaired individual using a one-sided early prototype to evaluate the fit and to go through a stepping sequence.

<u>Methods</u>

A one-sided early prototype was fabricated and evaluated on a non-impaired person (1.8 m, 66 kg) as shown in Figure 3.4. Video of the person walking was analyzed using the software package MAXTRAQ to determine the knee and hip range of motion and the time for each phase of gait. Distance covered during one right step was measured as the distance between the back of right heel at toe off and the back of right heel at toe off of next cycle.

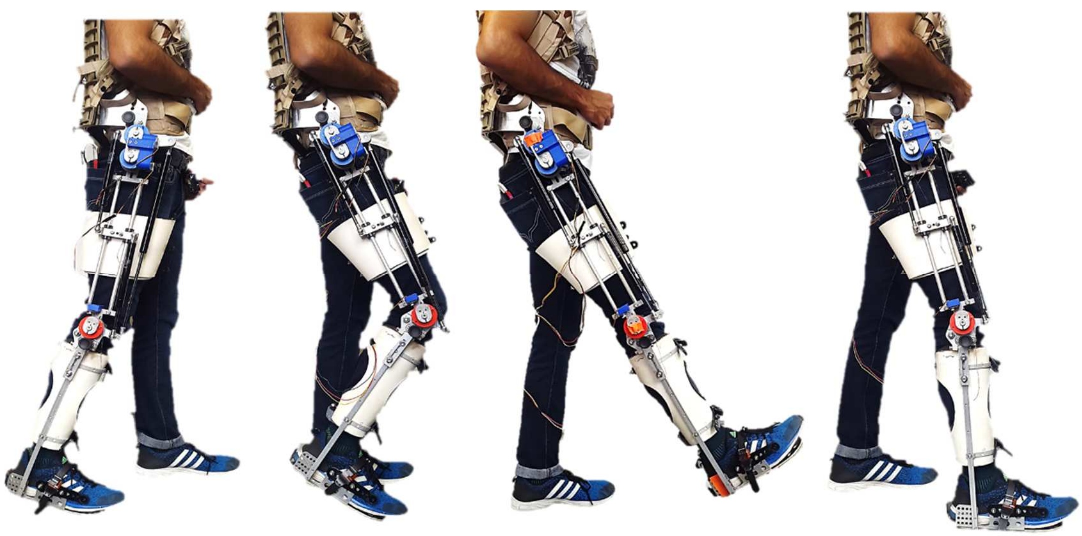

Figure 3.4: Stepping sequence captured while testing the prototype on a nonimpaired individual

<u>Results</u>

The person's foot cleared the ground and a complete gait cycle was possible. The data for each phase of gait cycle is shown in Figure 3.5Figure 3.6Figure 3.7. Hip and knee angles are the interior angles at the respective joints, where the hip and the knee angle are 0 deg when the person is standing upright. Hard stops should prevent hip angle from going below -10 deg, but during hip extension phase an angle of -26 deg was measured which was due to the relative movement between the orthosis and the body.

Figure 3.5: Knee and hip angles for the knee extension phase of gait. This phase took 0.5 sec, knee angle changed from 55 deg to 0 deg, and hip angle remained constant around 20 deg.

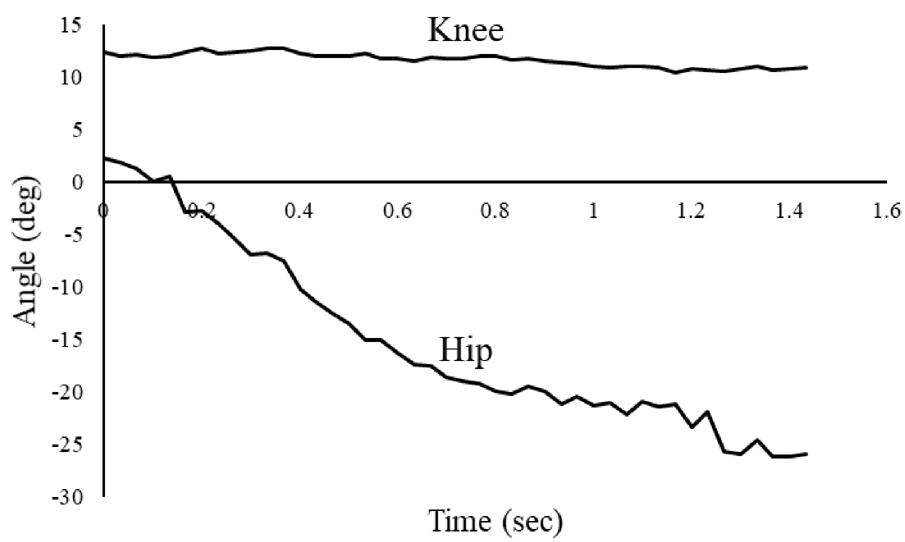

Figure 3.6: Knee and hip angles for the hip extension phase of gait cycle. This phase took 1.43 sec, knee angle remained constant around 11 deg, and hip angle changed from 2.3 deg to -26 deg

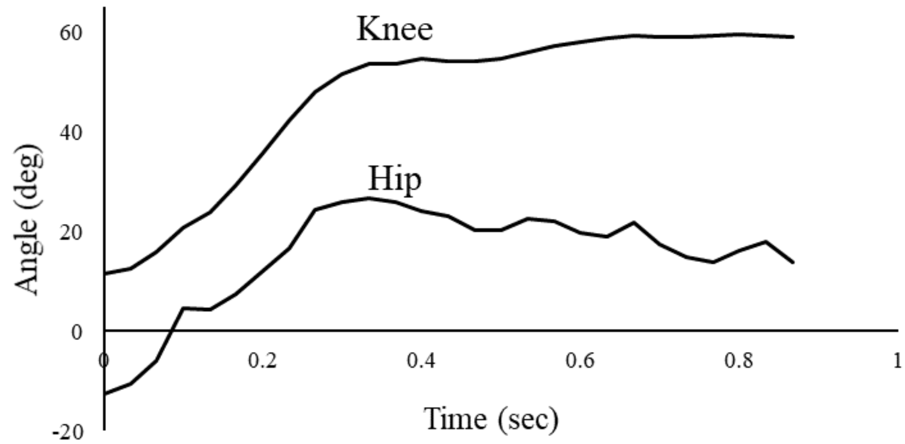

Figure 3.7: Knee and hip angles for the return to equilibrium phase of gait cycle. This phase took 0.87 sec, knee angle changed from 11.5 deg to 59 deg, and hip angle changed from -12.5 deg to 13.8 deg

The summed time for the three phases of gait was 2.8 sec. The distance covered during one right step was 0.76 m.

Discussion

Preliminary testing shows that the concept of FES-ESO for gait is feasible. The estimated maximum walking speed, assuming that the summed time of 2.8 sec remains consistent for each leg, was 0.27 m/s. Users may or may not get to a point where they are moving one leg while hip extension is happening in the other which is why 0.27 m/s is estimated as the maximum walking speed.

3.2 Preliminary User Tests

Objectives

The FES-ESO was tested on one volunteer with SCI to determine the feasibility of the concept. Fit of the exoskeleton, standing posture, and stepping were to be evaluated. All tests were conducted at the Minneapolis VA Medical Center.

Methods

The volunteer (1001) was a 57-year-old male with T5/T6 injury caused by a motorcycle accident. He used FES bikes on a regular basis. A measurement session was scheduled to get the volunteer's leg dimensions (form included in Appendix F. VA Study Forms). The exoskeleton's link lengths were then modified to fit the volunteer.

FES torque tests were conducted on the volunteer's leg to determine the peak quadriceps torque. These torque tests were conducted using our custom stimulator, a force gage (Mark 10 M5-200), and Axelgaard PALS 3"x4" rectangular electrodes. The electrodes were placed at appropriate locations on the volunteer's quadriceps and the stimulator leads were connected to the electrodes. A Velcro strap with a string was then strapped to the user's shank at 30 cm below the knee joint. The string was connected to the hook of the force gage which was held stationary by a person seated behind the volunteer's leg (Figure 3.8). When the quadricep was stimulated, the knee tends to extend but the force gage prevents it from doing so and records the maximum force. The maximum force multiplied by the moment arm, which in this case was 30 cm, gives the peak torque.

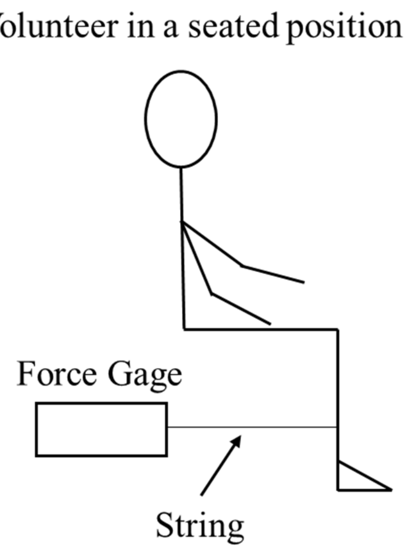

Figure 3.8: FES torque test set up

During gait testing sessions, first, the electrodes were placed on the volunteer's legs. Then the volunteer donned the exoskeleton, with only one leg attached, with the assistance of the physical therapist. The second leg of the exoskeleton was strapped to the volunteer's leg and then connected to the torso attachment of the exoskeleton. Various other ways of donning the exoskeleton were tried and this was the easiest and the quickest option. After donning the exoskeleton, the volunteer stood up with the assistance of the physical therapist and parallel bars. The locking of hip and the knee spring plungers was verified before allowing the volunteer to weight bear through the exoskeleton. Once a stable standing posture was achieved, the GUI would be used to test the stepping sequence.

<u>Results</u>

The exoskeleton fit the volunteer. The joints of the exoskeleton aligned with the joints of the user's body.

FES torque tests conducted on the volunteer showed that his legs could produce 34 Nm of peak torque. The gas spring combination that was initially selected required 34 Nm of torque. Pushing the quadriceps to their max capability for every stimulation would lead to quick fatigue. Hence, it was decided to use a gas spring combination that required 23 Nm of torque (Table 2.3). This would reduce the hip flexion angle from 15 deg to 10 deg which may impact toe clearance. This would become apparent during the gait cycle tests.

The volunteer was able to stand in the exoskeleton with the support of parallel bars. However, the body attachment parts on the exoskeleton permitted some movement of the volunteer with respect to the exoskeleton. This resulted in flexion at the knee and as a result, the volunteer couldn't maintain an upright posture. The wrap spring brake slipped by a few degrees with each load/unload cycle which lead to more flexion. Since people with SCI have atrophied muscles and weak bones, it was deemed unsafe for the volunteer to stand with a flexed knee. As a result, stepping tests couldn't be performed. The design of body attachment parts need to be modified to minimize the movement of volunteer's body relative to the exoskeleton.

Parameter	**Value**	**Measured value**
Device weight	< 10 kg	17.1 kgs
User height	1.6 to 1.9 m	1.8 m
User weight	< 114 kgs	70.5 kgs
Body coverage	Over legs, around hips and lower torso	Over legs, around hips and lower torso
Hip range of motion	110 flex to 30 deg ext	110 flex to 30 ext
Knee range of motion	95 flex to 2 deg ext	105 flex to 0 ext
Holding torque hip	31 Nm	>36.2 Nm
Holding torque knee	31 Nm	>33.3 Nm
Don/Doff time	< 15 min	Not met

Table 3.2: Design requirements compared to measured values

Figure 3.9: An image showing the volunteer's leg size

Figure 3.10: Photograph of the user after donning the FES-ESO

Discussion

Locking the hip and knee spring plunger after going from sit to stand to engage the gas springs was difficult. This was because the alignment between the plunger and the hole it needs to pop into needed to be exact for it to lock. To fix this, the hole was slightly oversized which would lead to slight rotation at the joint when it is not intended. Unlocking the plungers to go from standing posture to a seated position was also difficult. This was because while weight bearing the plunger tends to get jammed in the hole as there is significant non-axial force acting on it. In a future version, this could be addressed by either replacing the plunger with a different mechanical locking component or by creating a design that doesn't require an additional locking component.

4 Conclusion

The gas spring combination stores energy during knee extension and then transfers it to the hip and the knee joint as expected. Preliminary testing using a non-impaired individual demonstrated that the energy transfer component of the device worked. The exoskeleton fits the volunteer, the joints of the exoskeleton line up with the anatomical joints of the volunteer's body. There was relative movement between the volunteer's body and the structure of the exoskeleton which is undesirable and resulted in flexion at the knee while adopting a standing posture. This needs to be corrected to attain an upright posture while standing before conducting walking tests. A standing posture where the volunteer's knees are flexed results in significant forces acting on the volunteer's bones and is deemed unsafe. Walking tests can only be conducted once the relative movement issue is resolved and will help determine the feasibility of the approach.

5 References

[1] J. Verhaagen and J. W. McDonald, *Spinal Cord Injury*. Newnes, 2012.

[2] "Facts 2016.pdf." [Online]. Available: https://www.nscisc.uab.edu/Public/Facts%202016.pdf. [Accessed: 13-Jun-2018].

[3] "Spinal cord injury," *World Health Organization*. [Online]. Available: http://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury. [Accessed: 13-Jun-2018].

[4] J. Cao, S. Q. Xie, R. Das, and G. L. Zhu, "Control strategies for effective robot assisted gait rehabilitation: The state of art and future prospects," *Medical Engineering & Physics*, vol. 36, no. 12, pp. 1555–1566, Dec. 2014.

[5] J. W. McDonald and C. Sadowsky, "Spinal-cord injury," *The Lancet*, vol. 359, no. 9304, pp. 417–425, Feb. 2002.

[6] L. Phillips *et al.*, "Spinal Cord Injury: A Guide for Patient and Family," *Clinical Spine Surgery*, vol. 3, no. 3, p. 282, Sep. 1990.

[7] D. L. Brown-Triolo, M. J. Roach, K. Nelson, and R. J. Triolo, "Consumer perspectives on mobility: implications for neuroprosthesis design," *J Rehabil Res Dev*, vol. 39, no. 6, pp. 659–669, Dec. 2002.

[8] A. J. Kozlowski, T. N. Bryce, and M. P. Dijkers, "Time and Effort Required by Persons with Spinal Cord Injury to Learn to Use a Powered Exoskeleton for Assisted Walking," *Top Spinal Cord Inj Rehabil*, vol. 21, no. 2, pp. 110–121, 2015.

[9] C. Hartigan *et al.*, "Mobility Outcomes Following Five Training Sessions with a Powered Exoskeleton," *Top Spinal Cord Inj Rehabil*, vol. 21, no. 2, pp. 93–99, 2015.

[10] H. A. Quintero, R. J. Farris, and M. Goldfarb, "Control and implementation of a powered lower limb orthosis to aid walking in paraplegic individuals," in *2011 IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–6.

[11] R. J. Farris, H. A. Quintero, and M. Goldfarb, "Preliminary Evaluation of a Powered Lower Limb Orthosis to Aid Walking in Paraplegic Individuals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 6, pp. 652–659, Dec. 2011.

[12] I. Benson, K. Hart, D. Tussler, and J. J. van Middendorp, "Lower-limb exoskeletons for individuals with chronic spinal cord injury: findings from a feasibility study," *Clin Rehabil*, vol. 30, no. 1, pp. 73–84, Jan. 2016.

[13] J. Perry, M. Garrett, J. K. Gronley, and S. J. Mulroy, "Classification of Walking Handicap in the Stroke Population," *Stroke*, vol. 26, no. 6, pp. 982–989, Jun. 1995.

[14] S. R. Chang, R. Kobetic, M. L. Audu, R. D. Quinn, and R. J. Triolo, “Powered Lower-Limb Exoskeletons to Restore Gait for Individuals with Paraplegia – a Review,” *Case Orthop J*, vol. 12, no. 1, pp. 75–80, 2015.

[15] R. Lapointe, Y. Lajoie, O. Serresse, and H. Barbeau, “Functional community ambulation requirements in incomplete spinal cord injured subjects,” *Spinal Cord*, vol. 39, no. 6, pp. 327–335, Jun. 2001.

[16] X. Jin, Y. Cai, A. Prado, and S. K. Agrawal, “Effects of exoskeleton weight and inertia on human walking,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1772–1777.

[17] E. B. Marsolais and R. Kobetic, “Functional electrical stimulation for walking in paraplegia,” *Journal of Bone*, vol. 69, no. 5, pp. 728–733, Jun. 1987.

[18] J. O. Teeter, C. Kantor, D. L. Brown, and A. Stefanovska, *Functional electrical stimulation (FES) resource guide for persons with spinal cord injury or multiple sclerosis*. Cleveland (Ohio): FES Information Center, 1995.

[19] E. G. Claiborne and K. K. Baxter, “Introduction to the Parastep® System,” *Journal of Neurologic Physical Therapy*, vol. 18, no. 2, p. 11, 1994.

[20] V. K. Mushahwar, P. L. Jacobs, R. A. Normann, R. J. Triolo, and N. Kleitman, “New functional electrical stimulation approaches to standing and walking,” *J. Neural Eng.*, vol. 4, no. 3, p. S181, 2007.

[21] “Preliminary performance of a surgically implanted neuroprosthesis for standing and transfers--Where do we stand?,” *Journal of Rehabilitation Research & Development*, vol. 38, no. 6, pp. 609–617, Dec. 2001.

[22] E. Hardin *et al.*, “Walking after incomplete spinal cord injury using an implanted FES system: A case report,” *Journal of Rehabilitation Research & Development*, vol. 44, no. 3, pp. 333–346, May 2007.

[23] R. Kobetic, R. J. Triolo, and E. B. Marsolais, “Muscle selection and walking performance of multichannel FES systems for ambulation in paraplegia,” *IEEE Transactions on Rehabilitation Engineering*, vol. 5, no. 1, pp. 23–29, Mar. 1997.

[24] T. E. Johnston, R. R. Betz, B. T. Smith, and M. J. Mulcahey, “Implanted functional electrical stimulation: an alternative for standing and walking in pediatric spinal cord injury,” *Spinal Cord*, vol. 41, no. 3, pp. 144–152, Mar. 2003.

[25] J. Rose and J. G. Gamble, *Human Walking*. Williams & Wilkins, 1994.

[26] J. M. Hausdorff and W. K. Durfee, “Open-loop position control of the knee joint using electrical stimulation of the quadriceps and hamstrings,” *Med. Biol. Eng. Comput.*, vol. 29, no. 3, pp. 269–280, May 1991.

[27] C. L. E. Gerritsma-Bleeker, M. Heeg, and H. Vos-Niel, “Ambulation with the reciprocating-gait orthosis: Experience in 15 children with myelomeningocele or paraplegia,” *Acta Orthopaedica Scandinavica*, vol. 68, no. 5, pp. 470–473, Jan. 1997.

[28] J. Perez-Orive and R. E. Mayagoitia, “A closed-loop control system to be used a hybrid RGO system,” in *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1994, pp. 410–411 vol.1.

[29] M. Samadian, M. Arazpour, M. Ahmadi Bani, A. Pouyan, M. Bahramizadeh, and S. W. Hutchins, “The influence of orthotic gait training with an isocentric reciprocating gait orthosis on the walking ability of paraplegic patients: a pilot study,” *Spinal Cord*, vol. 53, no. 10, pp. 754–757, Oct. 2015.

[30] M. Arazpour *et al.*, “Effect of Orthotic Gait Training with Isocentric Reciprocating Gait Orthosis on Walking in Children with Myelomeningocele,” *Top Spinal Cord Inj Rehabil*, vol. 23, no. 2, pp. 147–154, 2017.

[31] W. B. Johnson, S. Fatone, and S. A. Gard, “Modeling effects of sagittal-plane hip joint stiffness on reciprocating gait orthosis-assisted gait,” *Journal of Rehabilitation Research & Development*, vol. 50, no. 10, pp. 1449–1456, Dec. 2013.

[32] M. Arazpour *et al.*, “Comparison of gait between healthy participants and persons with spinal cord injury when using the advanced reciprocating gait orthosis,” *Prosthet Orthot Int*, vol. 40, no. 2, pp. 287–293, Apr. 2016.

[33] D. Popovic, R. Tomovic, and L. Schwirtlich, “Hybrid assistive system-the motor neuroprosthesis,” *IEEE Transactions on Biomedical Engineering*, vol. 36, no. 7, pp. 729–737, Jul. 1989.

[34] T. Ohashi, G. Obinata, Y. Shimada, and K. Ebata, “Control of hybrid FES system for restoration of paraplegic locomotion,” in *Proceedings of 1993 2nd IEEE International Workshop on Robot and Human Communication*, 1993, pp. 96–101.

[35] R. Kobetic *et al.*, “Development of hybrid orthosis for standing, walking, and stair climbing after spinal cord injury,” *Journal of Rehabilitation Research & Development*, vol. 46, no. 3, pp. 447–462, Mar. 2009.

[36] M. Goldfarb and W. K. Durfee, “Design of a controlled-brake orthosis for FES-aided gait,” *IEEE Transactions on Rehabilitation Engineering*, vol. 4, no. 1, pp. 13–24, Mar. 1996.

[37] M. Goldfarb, K. Korkowski, B. Harrold, and W. Durfee, “Preliminary evaluation of a controlled-brake orthosis for FES-aided gait,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 3, pp. 241–248, Sep. 2003.

[38] S. R. Chang *et al.*, “A muscle-driven approach to restore stepping with an exoskeleton for individuals with paraplegia,” *J Neuroeng Rehabil*, vol. 14, May 2017.

[39] D. J. Nicol, M. H. Granat, S. J. M. Tuson, and R. H. Baxendale, “Variability of the dishabituation of flexion reflexes for FES assisted gait in spinal injured man1The work for this project was carried out at the Bioengineering Unit, University of Strathclyde, 106 Rottenrow, Glasgow G4 0NW, Scotland.1,” *Medical Engineering & Physics*, vol. 20, no. 3, pp. 182–187, Apr. 1998.

[40] W. K. Durfee and A. Rivard, “Design and Simulation of a Pneumatic, Stored-energy, Hybrid Orthosis for Gait Restoration,” *J Biomech Eng*, vol. 127, no. 6, pp. 1014–1019, Nov. 2005.

[41] A. Kangude, B. Burgstahler, J. Katsys, and W. Durfee, “Single Channel Hybrid FES Gait Assist System,” *J. Med. Devices*, vol. 3, no. 2, p. 027502, Jun. 2009.

[42] K. J. Boughner and W. K. Durfee, “Preliminary design of an energy storing orthosis for providing gait to people with spinal cord injury,” in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 2581–2584.

[43] A. Kangude, B. Burgstahler, and W. Durfee, “Engineering evaluation of the energy-storing orthosis FES gait system,” in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, 2010, pp. 5927–5930.

[44] A. Kangude, B. Burgstahler, J. Kakastys, and W. Durfee, “Single channel hybrid FES gait system using an energy storing orthosis: Preliminary design,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 6798–6801.

[45] C. F. Wiebusch, “Dial Clutch of the Spring Type*,” *Bell System Technical Journal*, vol. 18, no. 4, pp. 724–741.

[46] D. A. Winter, *Biomechanics and Motor Control of Human Movement*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2009.

6 Appendixes

A. Wrap Spring Torque Derivation

Torque of wrap spring in free direction

Notation:

l – length along the line of contact of spring on the arbor

μ - coefficient of friction between the spring and the arbor

r_2 – radius of the arbor

R_2 – radius to the neutral bending axis of the spring when on arbor

N – number of turns on arbor

P – compression in spring wire due to the applied torque; this is the resultant force acting across entire cross-section of the wire

f_0 – radial force of spring on arbor when no torque is applied, per length of contact line

Compression exists in the spring at any point when the rotation of the arbor unwinds the spring. Thus, there will be a radial force subtracted from f_0 at all points. This subtracting force is P/r_2. The increase in compression in the spring wire along the length of line of contact due to friction is given by Equation [1] which upon integration gives Equation [2] where C is the integration constant and is equal to $1/f_0$ as $P = 0$ at $l = 0$.

$$dP = \mu(f_0 - P/r_2)dl \quad (1)$$

$$l = -(r_2/\mu)\, ln[f_0 - (P/r_2)]\, C \quad (2)$$

$$P = r_2 f_0 (1 - \exp(-\mu l/r_2)) \quad (3)$$

Since $l = 2\pi r_2 N$

$$P = r_2 f_0 (1 - \exp(-2\pi N\mu)) \quad (4)$$

Since torque is equal to Pr_2

$$T = r_2^2 f_0 (1 - \exp(-2\pi N\mu)) \quad (5)$$

For $N\mu>1$ the exponential term becomes very small

$$T = r_2^2 f_0 \quad (6)$$

Torque of wrap spring in the gripping direction

If the torque is applied in the direction to wind up the spring, the force P'/r_2 adds to the inward force f_0 and the increase in compression is now given by Equation [7]. Following the same steps as adopted while deriving Equation [5], we get the torque in gripping direction which is given by Equation [9]. A small change in N or μ significantly affects the torque as these terms are in the exponent.

$$dP' = \mu(f_0 + P'/r_2)dl \tag{7}$$

$$P' = r_2 f_0(\exp(2\pi N\mu) - 1) \tag{8}$$

$$T' = r_2^2 f_0(\exp(2\pi N\mu) - 1) \tag{9}$$

<u>Radial force on the arbor (f_0)</u>

We can get the force f_0 by equating the potential energy of strain per unit length of the wire when on the arbor to the work done in expanding the spring from its free diameter to the diameter of the arbor.

E – Youngs modulus for the spring material

I – the are moment of inertia of the wire cross section

h – the radial thickness of the wire

R_1 – free radius to the neutral axis

r_1 – free inner radius of the spring

Consider a portion of the spring shown in Figure 6.1 expanded out from an initial radius of curvature R_1 to radius R_2. The region above the neutral axis is in compression whereas the one below the neutral axis is in tension. Consider y as the distance of any fiber from the neutral axis. The length of undistorted fiber is $L = (R_1 + y)\theta_1$; after bending this length changes to $L' = (R_2 + y)\theta_2$. The change in length per unit length or strain is given by Equation [10].

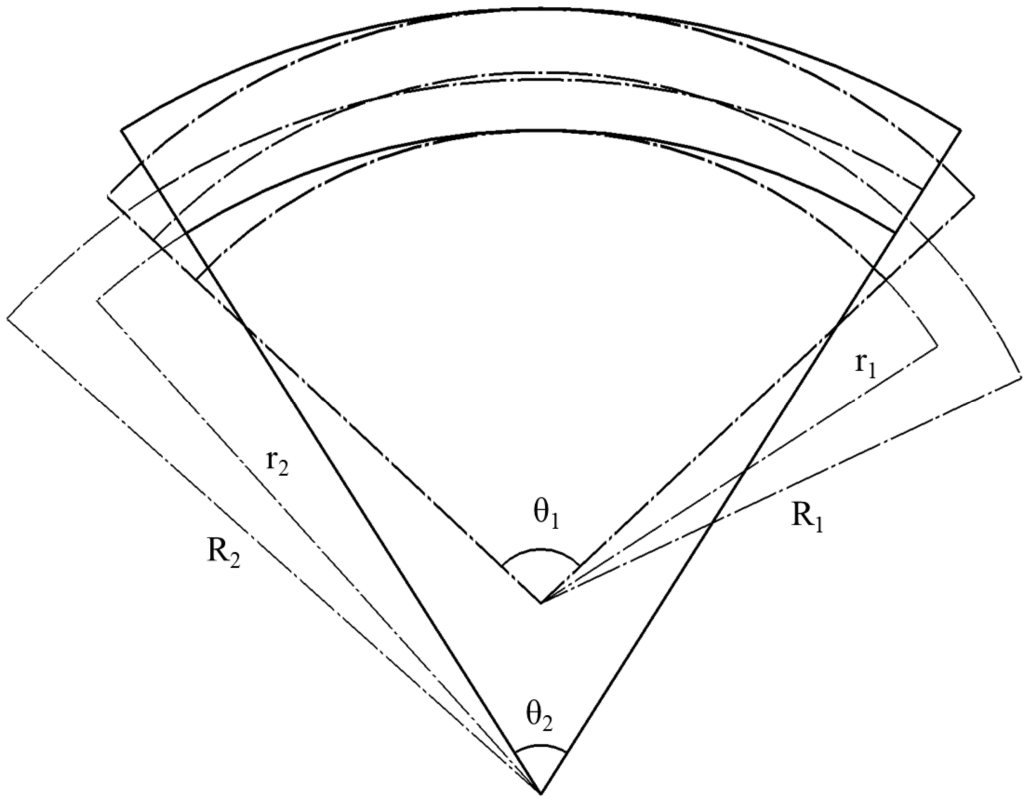

Figure 6.1: A part of spring in initial and expanded condition

$$\frac{L - L'}{L} = 1 - \frac{(R_2 + y)\theta_2}{(R_1 + y)\theta_1} \tag{10}$$

Since there is no change in length at the neutral axis, $\theta_2 = L_0/R_2$ and $\theta_1 = L_0/R_1$. Substituting these in Equation [10] we get Equation [11].

$$Strain = (y/R_2)(R_2 - R_1)/(R_1 + y) \tag{11}$$

The potential energy per unit volume in a material strained in tension or compression is

$$\frac{\mathrm{W}}{\mathrm{V}} = \frac{\mathrm{E}}{2}(\mathrm{Strain})^2 \tag{12}$$

Substituting strain from Equation [11] we get

$$\frac{\mathrm{W}}{\mathrm{V}} = \frac{\mathrm{E}}{2}\left[\frac{y(R_2 - R_1)}{R_2(R_1 + y)}\right]^2 \tag{13}$$

Let b be the width of the wire at point y. For a wire symmetrical about the neutral axis the strain energy per unit length is given by Equation [14] where $(R_1 + y)/R_1$ is the ratio of length of fiber at point y to the length along neutral axis. If all values of y are small compared to R_1, neglecting y leads to very little error. Hence, Equation [14] can be simplified into Equation [15].

$$\frac{W}{l} = \int_{-h/2}^{h/2} \frac{E}{2}\left[\frac{y(R_2 - R_1)}{R_2(R_1 + y)}\right]^2 \frac{R_1 + y}{R_1} b\,dy \tag{14}$$

$$\frac{\mathrm{W}}{\mathrm{l}} = \int_{-h/2}^{h/2} \frac{\mathrm{E}}{2}\left(\frac{\mathrm{R}_2 - \mathrm{R}_1}{\mathrm{R}_2\mathrm{R}_1}\right)^2 \mathrm{by}^2\mathrm{dy} \tag{15}$$

The integral by^{2}dy is the moment of inertia I of the section, hence

$$\frac{\mathrm{W}}{\mathrm{l}} = \frac{1}{2}\mathrm{EI}\left(\frac{\mathrm{R}_2 - \mathrm{R}_1}{\mathrm{R}_2\mathrm{R}_1}\right)^2 \tag{16}$$

This is equal to work done per unit length by the force per unit length F(ΔR) through distance ΔR

$$\int_0^{\Delta R} F(\Delta R)d\Delta R = \frac{1}{2}EI\left[\frac{\Delta R}{R_1(R_1 + \Delta R)}\right]^2 \tag{17}$$

Differentiating both sides with respect to ΔR and simplifying gives

$$\mathrm{F}(\Delta\mathrm{R}) = \frac{\mathrm{EI}}{\mathrm{R}_1}\frac{\Delta\mathrm{R}}{(\mathrm{R}_1 + \Delta\mathrm{R})^2} \tag{18}$$

Substituting $\Delta R = R_2 - R_1$

$$F(\Delta R) = EI\frac{R_2 - R_1}{R_1 R_2^2} \tag{19}$$

The equivalent force per unit length measured along the surface f_0 is greater then F (ΔR) in the ratio R_2/r_2 as the same total force is distributed over a shorter length.

$$f_0 = \frac{R_2}{r_2} F(\Delta R) = EI \frac{R_2 - R_1}{R_1 r_2 R_2^2} \tag{20}$$

The value of f_0 used in Equations [5] and [9] gives the torque of wrap spring in the free direction [21] and gripping direction [22] respectively.

$$\mathrm{T_f} = \mathrm{EI} \frac{\mathrm{r_2(R_2 - R_1)}}{\mathrm{R_1 R_2^2}} (1 - \exp(-2\pi N\mu)) \tag{21}$$

$$T_g = EI \frac{r_2(R_2 - R_1)}{R_1 R_2^2} (\exp(2\pi N\mu) - 1) \tag{22}$$

B. Gas Spring Selection

To select appropriate gas springs for the ESO, MATLAB scripts were written to calculate the torque at the knee and the hip throughout the ESO's gait cycle. To verify the results of MATLAB, a Simulink model was created for each phase of gait (Figure 6.3Figure 6.4Figure 6.5). The simulations showed if the motion was as expected for each phase of gait.

To model the ESO, a pivot, thigh link, shank link, and a foot link were created and interconnected using revolute joints. A view of what's under the mask for thigh link is shown in Figure 6.2. It consists of a structure defined by 3 Simscape multibody elements (Hole, Main, and Peg). The structure has some connection points for attaching gas springs, pivot, shank link. These position of these connections with respect to the structure of the thigh link is defined by using rigid transform blocks. The shank and foot links are modelled like the thigh link. The rotation of each joint is restricted by using rotational hard stop blocks. Simscape multibody force blocks are used to define the forces exerted by the gas springs, and a signal builder is used to impart FES torque to the knee.

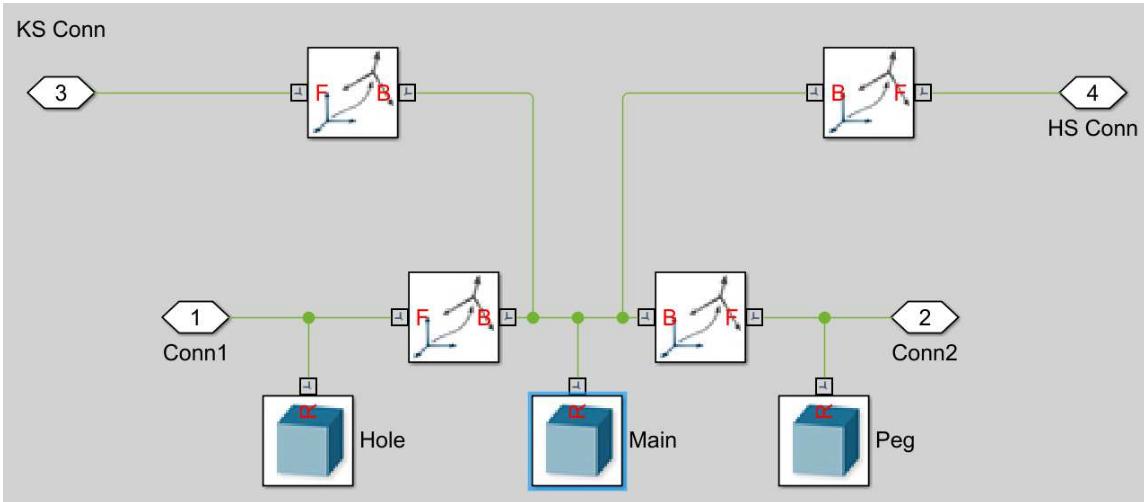

Figure 6.2: Simulink model of the thigh link of ESO

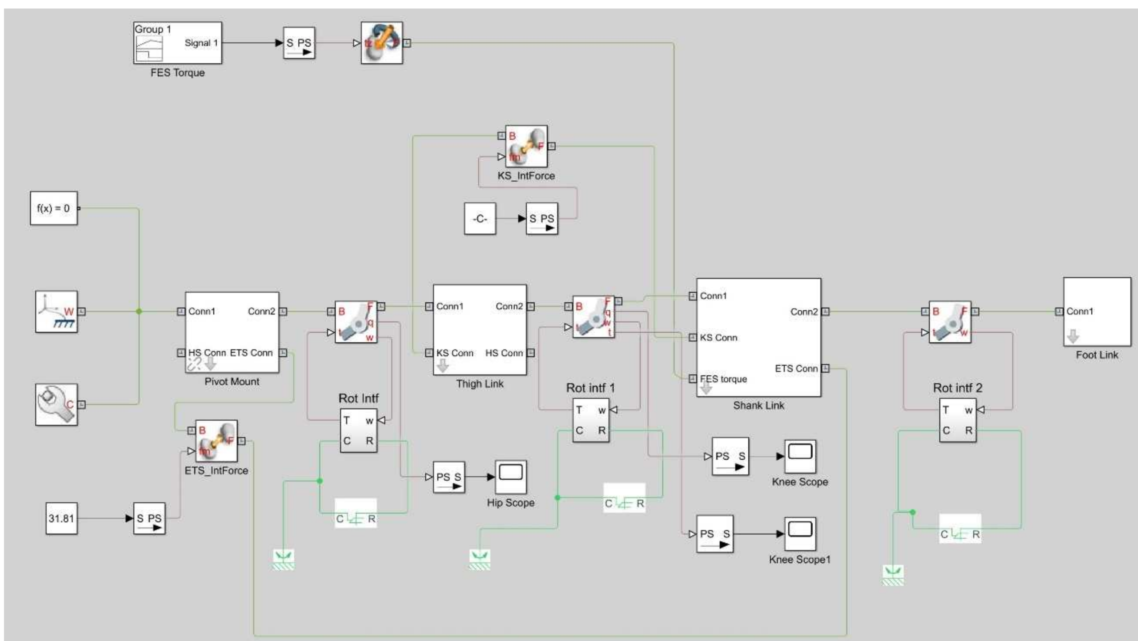

Figure 6.3: Knee Extension Simulation Model

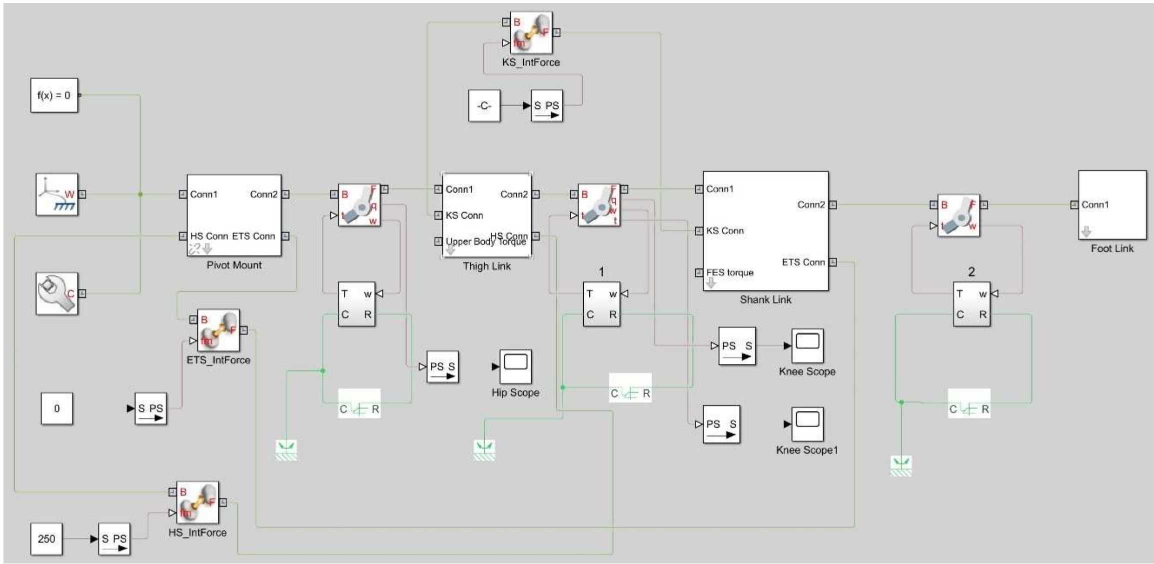

Figure 6.4: Hip Extension Simulation Model

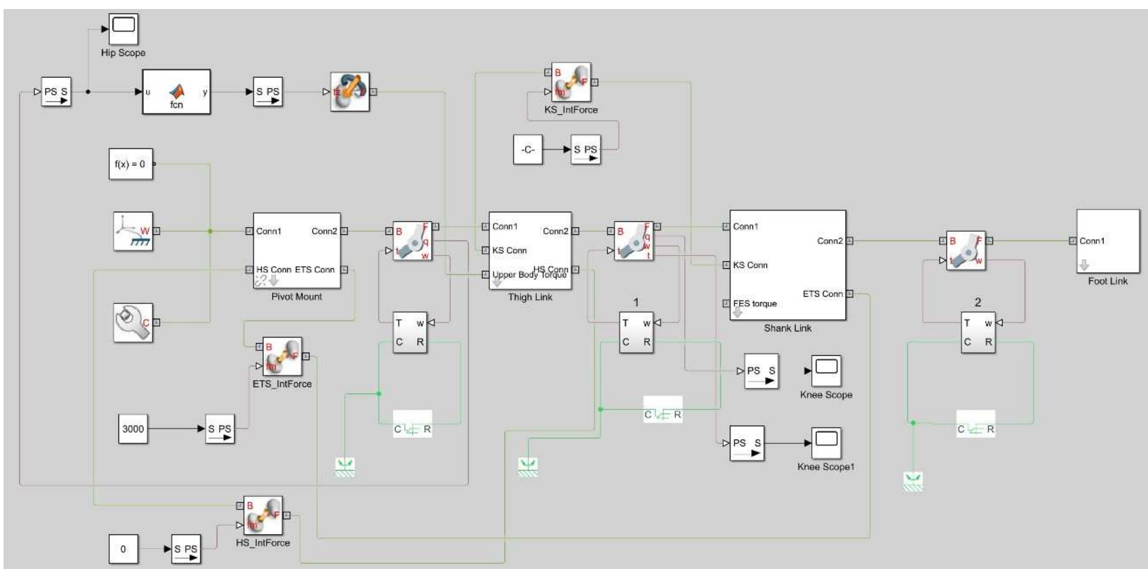

Figure 6.5: Return to Equilibrium Simulation Model

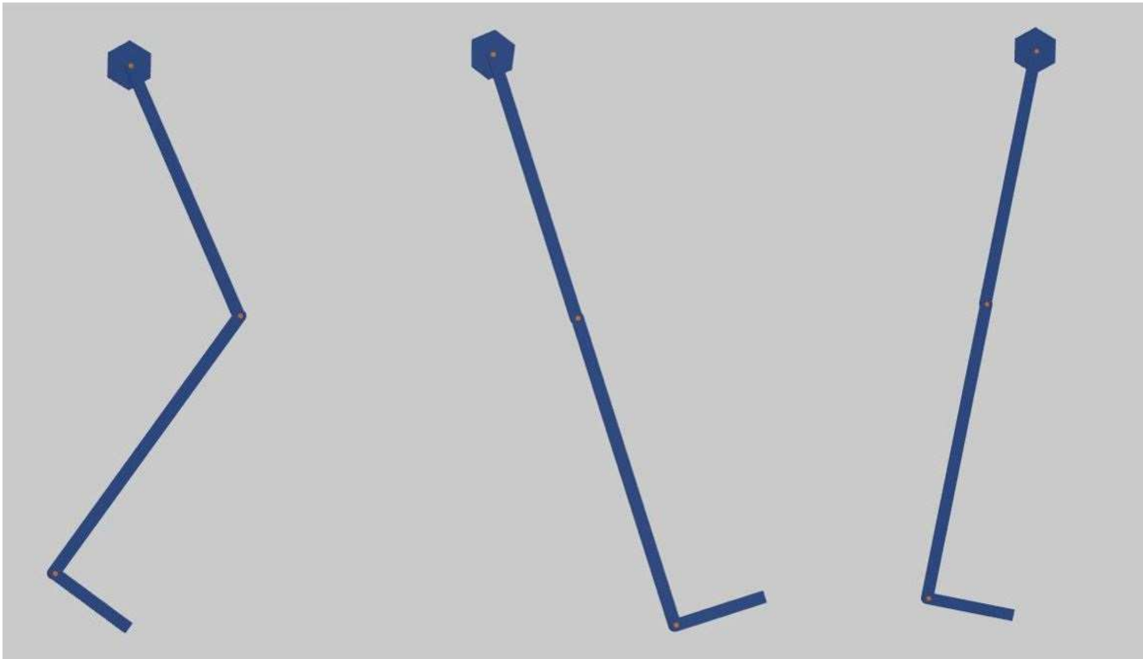

Figure 6.6: Results of simulation. From left to right, end of return to equilibrium, end of knee extension, and end of hip extension

```
% Knee Extension Analysis (Phase 1 of SCI Gait)
% Height: 180 cms
% Weight: 67 kgs
Wt = 6.7*9.81 + 4.5*9.81; %% weight of thigh in N, 4.5kg is
approximated weight of thigh part of device - NEED TO REPLACE WITH
ACTUAL VALUE
Ws = 4.087*9.81 + 1*9.81; %% weight of shank + foot (leg) in N, 1 kg is
approximated weight of shank part of device - NEED TO REPLACE WITH
ACTUAL VALUE
What = 45.4260*9.81; %% weight of HAT in N
Lt = 0.441; %% Length of thigh in m
Ls = 0.513; %% Length of shank + foot in m
Lhat = 0.846; %% Length of HAT in m
Dct = 0.190953; %% Distance of COG of thigh from proximal end of hip
Dcs = 0.310878; %% Distance of COG of leg from proximal end of knee
dtor = pi/180; %% degree to radian conversion
rtod = 180/pi; %% radian to degree conversion
theta = 25; %% Hip angle in degrees

%% Sign Convention CCW positive, CW negative
```

```
TkG = []; %% Torque at the Knee due to gravity
ThG = []; %% Torque at the Hip due to gravity
TkESS = []; %% Torque at the knee exerted by Energy storing spring
(ESS)
ThESS = []; %% Torque at the hip exerted by Energy storing spring
TkKS = []; %% Torque at the Knee exerted by Knee Spring (KS)
ThKS = []; %% Torque at the hip exerted by Knee Spring
TkHS = []; %% Torque at the knee exerted by Hip Spring (HS)
ThHS = []; %% Torque at the hip exerted by Hip Spring
TotH = []; %% Net Torque at Hip
TotK = []; %% Net Torque at Knee

alpha = -35:1:25; %% Angle of knee relative to gravity
thetahESS = 69.02:(-4.16/60):64.86; %% angle change for ESS torque at
hip
thetakESS = 34.02:(55.84/60):89.86; %% angle change for ESS torque at
knee
thetahKS = 15.07:(-6.3/60):8.77; %% angle change for KS torque at hip
thetakKS = 33.67:(53.71/60):87.38; %% angle change for KS torque at
knee
thetahHS = repmat(51.48,1,61); %% angle change for HS torque at hip
thetakHS = repmat(12.95,1,61); %% angle change for HS torque at knee


Fess = (40/2.2)*9.81; %% Force due to ESS in N, 90lbs during
compression due to friction
Fks = (20/2.2)*9.81; %% Force due to KS in N, 73 lbs during comp
Fhs = (120/2.2)*9.81; %% Force due to HS in N



RhESS = 0.073746; %% Moment arm for ESS at Hip
RkESS = 0.067818; %% Moment arm for ESS at Knee
RhKS = 0.2249; %% Moment arm for KS at Hip
RkKS = 0.054551; %% Moment arm for KS at Knee
RhHS = 0.04917; %% Moment arm for HS at Hip
RkHS = 0.2249; %% Moment arm for HS at Knee


%% Gravity Torques

TkG = -Ws*Dcs*sin(alpha*dtor);
ThG = -Wt*Dct*sin(theta*dtor)-
Ws*(Lt*sin(theta*dtor)+Dcs*sin(alpha*dtor));

%% ESS Torques

ThESS = -Fess*RhESS*sin(thetahESS*dtor);
TkESS = -Fess*RkESS*sin(thetakESS*dtor);

%% KS Torques

%ThKS = Fks*RhKS*sin(thetahKS*dtor);
TkKS = -Fks*RkKS*sin(thetakKS*dtor);

%% HS Torques
```

```
 ThHS = Fhs*RhHS*sin(thetahHS*dtor);
 %TkHS = Fhs*RkHS*sin(thetakHS*dtor);

%% Total Torques
TotH = ThG +ThESS +ThHS;
TotK = TkG +TkESS +TkKS;


%% plots


figure;
%%plot(alpha, TotH,'b');
%%hold on;
plot(alpha, TotK, 'k');
title('Knee Extension Analysis');
xlabel('Angle of Knee wrt Gravity (Degrees)');
ylabel('FES Torque(Nm)');
%%legend('TotH', 'TotK');
hold off;
```

```
% Hip Extension Analysis (Phase 2 of SCI Gait)
% Height: 180 cms
% Weight: 67 kgs
Wt = 6.7*9.81 + 4.5*9.81; %% weight of thigh in N, 4.5kg is
approximated weight of thigh part of device - NEED TO REPLACE WITH
ACTUAL VALUE
Ws = 4.087*9.81 + 1*9.81; %% weight of shank + foot (leg) in N, 1 kg is
approximated weight of shank part of device - NEED TO REPLACE WITH
ACTUAL VALUE
What = 45.426*9.81; %% weight of HAT segment in N
Lt = 0.441; %% Length of thigh in m
Ls = 0.513; %% Length of shank + foot in m
Lhat = 0.846; %% Length of HAT in m
Dct = 0.190953; %% Distance of COG of thigh from proximal end of hip
Dcs = 0.310878; %% Distance of COG of leg from proximal end of knee
Dchat = 0.5296; %% Distance of COG of HAT from proximal end of hip
dtor = pi/180; %% degree to radian conversion
rtod = 180/pi; %% radian to degree conversion
theta = 25; %% Hip angle in degrees
```

```
%% Sign Convention CCW postive, CW negative

TkG = []; %% Torque at the Knee due to gravity
ThG = []; %% Torque at the Hip due to gravity
TkESS = []; %% Torque at the knee exerted by Energy storing spring
(ESS)
ThESS = []; %% Torque at the hip exerted by Energy storing spring
TkKS = []; %% Torque at the Knee exerted by Knee Spring (KS)
ThKS = []; %% Torque at the hip exerted by Knee Spring
TkHS = []; %% Torque at the knee exerted by Hip Spring (HS)
ThHS = []; %% Torque at the hip exerted by Hip Spring
TotH = []; %% Net Torque at Hip
TotK = []; %% Net Torque at Knee


alpha = 25:-1:-10;
alpha1 = 25:-1:16; %% Angle of hip relative to gravity
alpha2 = 15:-1:-10;
thetahESS = 64.86:(35.75/35):100.61; %% angle change for ESS torque at
hip
thetakESS = 89.86:(0.75/35):90.61; %% angle change for ESS torque at
knee
% thetahKS = repmat(8.77,1,36); %% angle change for KS torque at hip
thetakKS = repmat(87.38,1,36); %% angle change for KS torque at knee
thetahHS = 51.48:(32.29/35):83.77; %% angle change for HS torque at hip
% thetakHS = 12.95:(-2.71/35):10.24; %% angle change for HS torque at
knee


Fess = (30/2.2)*9.81; %% Force due to ESS in N
Fks = (15/2.2)*9.81; %% Force due to KS in N
Fhs = (40/2.2)*9.81; %% Force due to HS in N 146in comp
F = -0.0556*What*alpha1+ 1.8889*What;
F = [F, repmat(What,1,26)];


RhESS = 0.073746; %% Moment arm for ESS at Hip
RkESS = 0.067818; %% Moment arm for ESS at Knee
RhKS = 0.2249; %% Moment arm for KS at Hip
RkKS = 0.054551; %% Moment arm for KS at Knee
RhHS = 0.04917; %% Moment arm for HS at Hip
RkHS = 0.2249; %% Moment arm for HS at Knee


%% Gravity Torques

TkG = -Ws*Dcs*sin(alpha*dtor);%+F.*sin(alpha*dtor)*(Lt+Ls);
ThG = What*sin(alpha*dtor)*(Dchat);%-Wt*Dct*sin(alpha*dtor)-
Ws*((Lt+Dcs)*sin(alpha*dtor))+

%% ESS Torques

ThESS = -Fess*RhESS*sin(thetahESS*dtor);
TkESS = -Fess*RkESS*sin(thetakESS*dtor);

%% KS Torques
```

```
% ThKS = Fks*RhKS*sin(thetahKS*dtor);
TkKS = -Fks*RkKS*sin(thetakKS*dtor);

%% HS Torques

ThHS = Fhs*RhHS*sin(thetahHS*dtor);
% TkHS = Fhs*RkHS*sin(thetakHS*dtor);

%% Total Torques
TotH = ThESS +ThHS;
TotK = TkG +TkESS +TkKS;


%% plots
hold on;
plot (alpha, TotH, 'k');
title('Hip Extension Analysis')
xlabel('Angle of Hip wrt Gravity');
ylabel('Total Torque at Hip in Nm');
hold off;
```

```
% Return to Eqmuilibrium Analysis (Phase 3 of SCI Gait)
% Height: 180 cms
% Weight: 67 kgs
Wt = 6.7*9.81 + 4.5*9.81; %% weight of thigh in N, 4.5kg is
approximated weight of thigh part of device - NEED TO REPLACE WITH
ACTUAL VALUE
Ws = 4.087*9.81 + 1*9.81; %% weight of shank + foot (leg) in N, 1 kg is
approximated weight of shank part of device - NEED TO REPLACE WITH
ACTUAL VALUE
What = 45.426*9.81; %% weight of HAT segment in N
Lt = 0.441; %% Length of thigh in m
Ls = 0.513; %% Length of shank + foot in m
Lhat = 0.846; %% Length of HAT in m
Dct = 0.190953; %% Distance of COG of thigh from proximal end of hip
Dcs = 0.310878; %% Distance of COG of leg from proximal end of knee
Dchat = 0.5296; %% Distance of COG of HAT from proximal end of hip
dtor = pi/180; %% degree to radian conversion
rtod = 180/pi; %% radian to degree conversion
```

```
%% Sign Convention CCW postive, CW negative

TkG = []; %% Torque at the Knee due to gravity
ThG = []; %% Torque at the Hip due to gravity
ThG_S = []; %% Torque at the Hip due to gravity on shank
TkESS = []; %% Torque at the knee exerted by Energy storing spring
(ESS)
ThESS = []; %% Torque at the hip exerted by Energy storing spring
TkKS = []; %% Torque at the Knee exerted by Knee Spring (KS)
ThHS = []; %% Torque at the hip exerted by Hip Spring
TotH = []; %% Net Torque at Hip
TotK = []; %% Net Torque at Knee


angHip = -10:1:25;
angKnee = 0:-60/35:-60;
thetahESS = 79.39:(31.59/35):110.98; %% angle change for ESS torque at
hip
thetakESS = 90.61:(-56.59/35):34.02; %% angle change for ESS torque at
knee
thetakKS = 87.30:(-54.31/35):32.99; %% angle change for KS torque at
knee
thetahHS = 83.64:(-33.06/35):50.58; %% angle change for HS torque at
hip


Fess = (30/2.2)*9.81; %% Force due to ESS in N
Fks = (15/2.2)*9.81; %% Force due to KS in N
Fhs = (30/2.2)*9.81; %% Force due to HS in N 146in comp

RhESS = 0.073746; %% Moment arm for ESS at Hip
RkESS = 0.067818; %% Moment arm for ESS at Knee
RhKS = 0.2249; %% Moment arm for KS at Hip
RkKS = 0.054551; %% Moment arm for KS at Knee
RhHS = 0.04917; %% Moment arm for HS at Hip
RkHS = 0.2249; %% Moment arm for HS at Knee


%% DH calculations

s=0;
alpha=0;
T = zeros(4,4,36);

for i=1:36

s=0;
alpha=0;
theta = (-90+angHip(i));
a=Lt;
T1 = TM(theta,alpha,a,s);

theta = angKnee(i);
a=Dcs;
T2 = TM(theta,alpha,a,s);

T(:,:,i) = T1*T2;
```

```matlab
end

%% Gravity Torques

TkG = -Ws*Dcs*sin((angHip+angKnee)*dtor);
ThG_S = zeros(1,36);
ThG_S(:) = -Ws* T(1,4,:);
ThG = -Wt*sin(angHip*dtor)*Lt+ ThG_S;

%% ESS Torques

ThESS = -Fess*RhESS*sin(thetahESS*dtor);
TkESS = -Fess*RkESS*sin(thetakESS*dtor);

%% KS Torques

TkKS = -Fks*RkKS*sin(thetakKS*dtor);

%% HS Torques

ThHS = Fhs*RhHS*sin(thetahHS*dtor);

%% Total Torques
TotH = ThG +ThHS;
TotK = TkG +TkESS +TkKS;


%% plots
 plot(angKnee, TotK, 'k');
 hold on;
 %plot (angHip, TotH, 'r');

 title('Return to Eqm Analysis')
ylabel('Total Torque at Knee in Nm');
xlabel('Knee Joint Angle');
hold off;
```

C. Preliminary body attachment parts

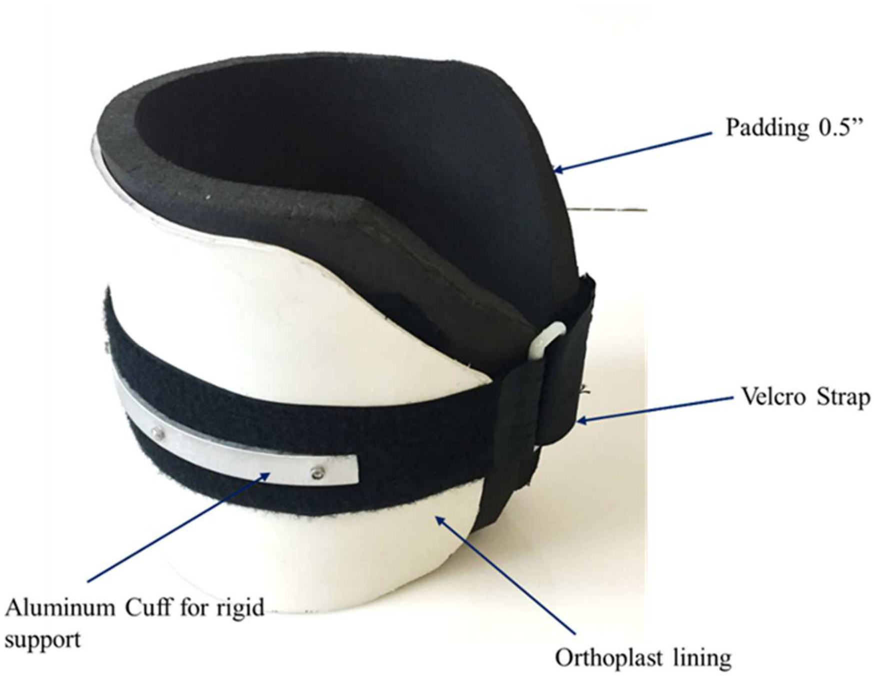

Figure 6.7: Thigh attachment part version 1

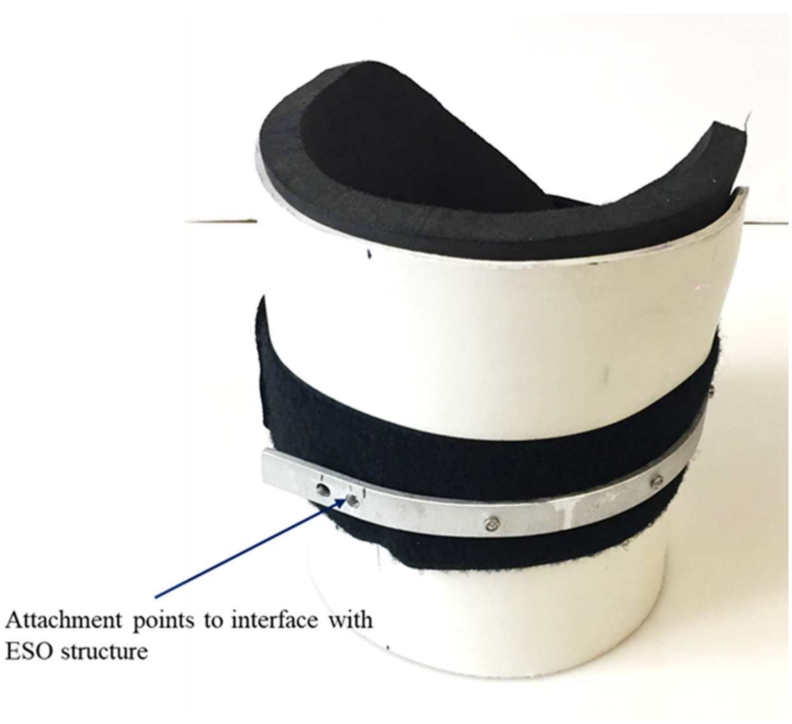

Figure 6.8: Attachment points of thigh cuff

As shown in Figure 6.7 and Figure 6.8, the initial design of thigh attachment part consisted of a rigid aluminum cuff which is lined on the inside with an orthoplast sheet. Inside the orthoplast cuff was a layer of padding which ensured that there is no chafing. The aluminum cuff, a Velcro strap, and the orthoplast were riveted together. This cuff was not rigid enough and allowed relative movement between the body and the exoskeleton. This was fixed by modifying the design to improve the cuff's rigidity.

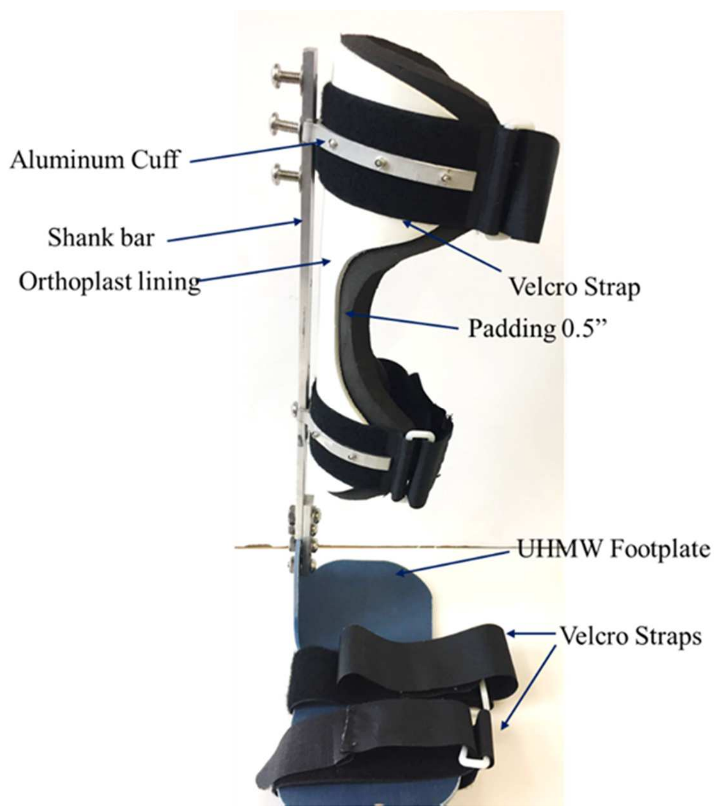

Figure 6.9: Calf attachment part version 1

The initial design of calf attachment part consisted of two aluminum cuffs connected to a shank bar which extended downward from the knee joint. The aluminum cuffs, Velcro straps, and orthoplast lining were riveted together. The inner side of the orthoplast material was lined with 0.5” thick padding. At the end of the shank bar was a footplate made of UHMW with two Velcro straps to securely attach the user’s foot and prevent foot drop. During clinical testing, it was found that this design of calf attachment allowed significant rotation at the ankle which is undesirable. This was mainly due to lack of stiffness at the ankle as a plastic part was being used. Moreover, the shank bar tended to rotate when the user was load bearing through his legs. To fix these issues a new calf attachment part was designed.

D. Electronics

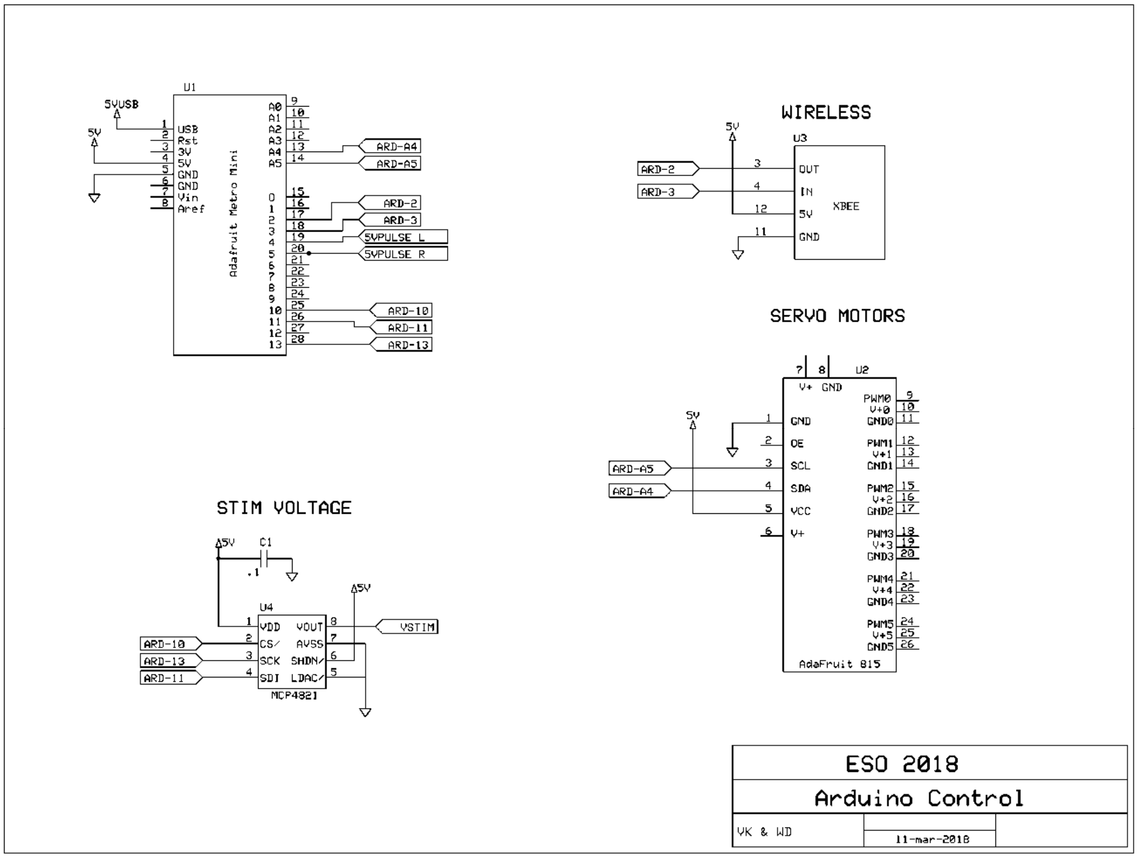

Figure 6.10: ESO schematic page 1

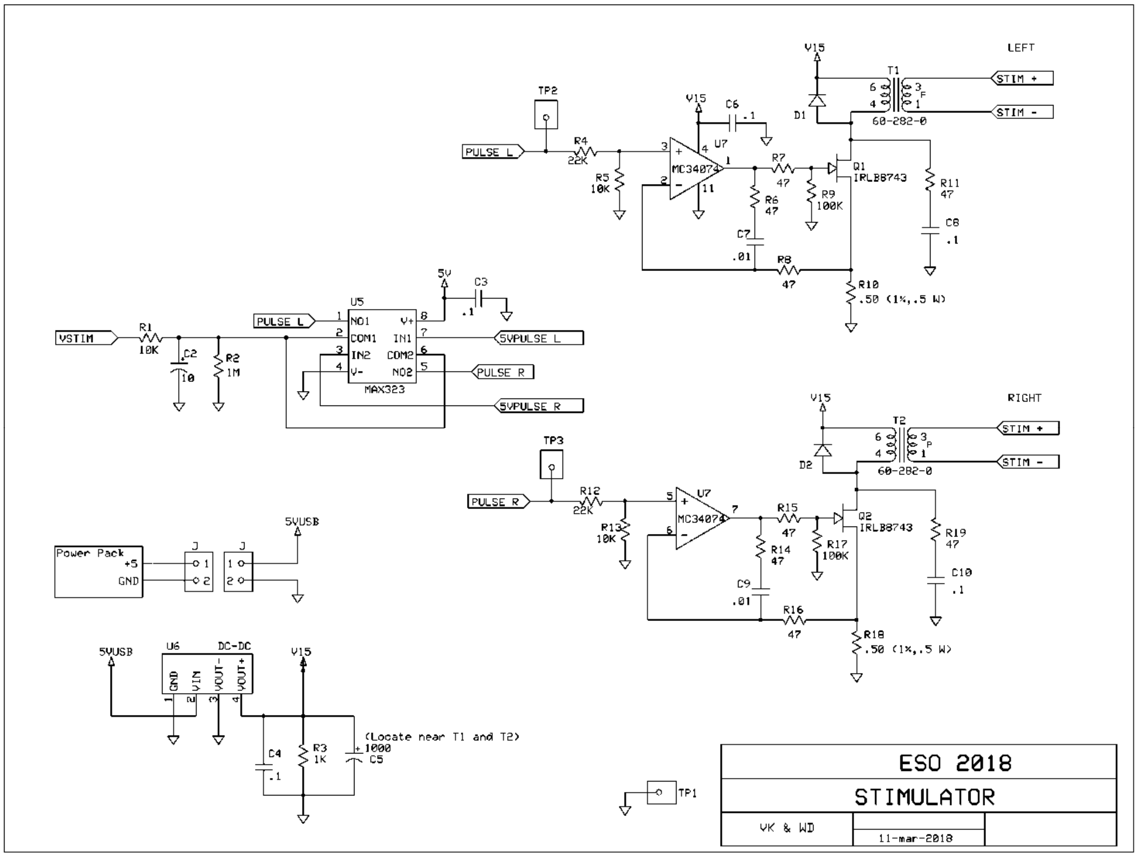

Figure 6.11: ESO schematic page 2

The schematics of the ESO electronics are shown in Figure 6.10 Figure 6.11. The electronics enclosure is shown in Figure 6.12Figure 6.13Figure 6.14. Each servo has 3 wires (PWM, Gnd, Pow) so two DB9 connectors have been used to connect the 6 servos (3 on each leg) to the Adafruit servo driver. The stimulation pulse is carried to the electrodes via Tens Lead Wires (3.5mm plug to Two 2mm Pin Connectors). These wires interface with the PCB via a 3.5 mm stereo jack. There are two jacks mounted on the wall of the enclosure (one for each leg) as shown in Figure 6.13. The enclosure has a DPDT switch mounted on its wall to act as a kill switch for stimulation in case something were to go wrong.

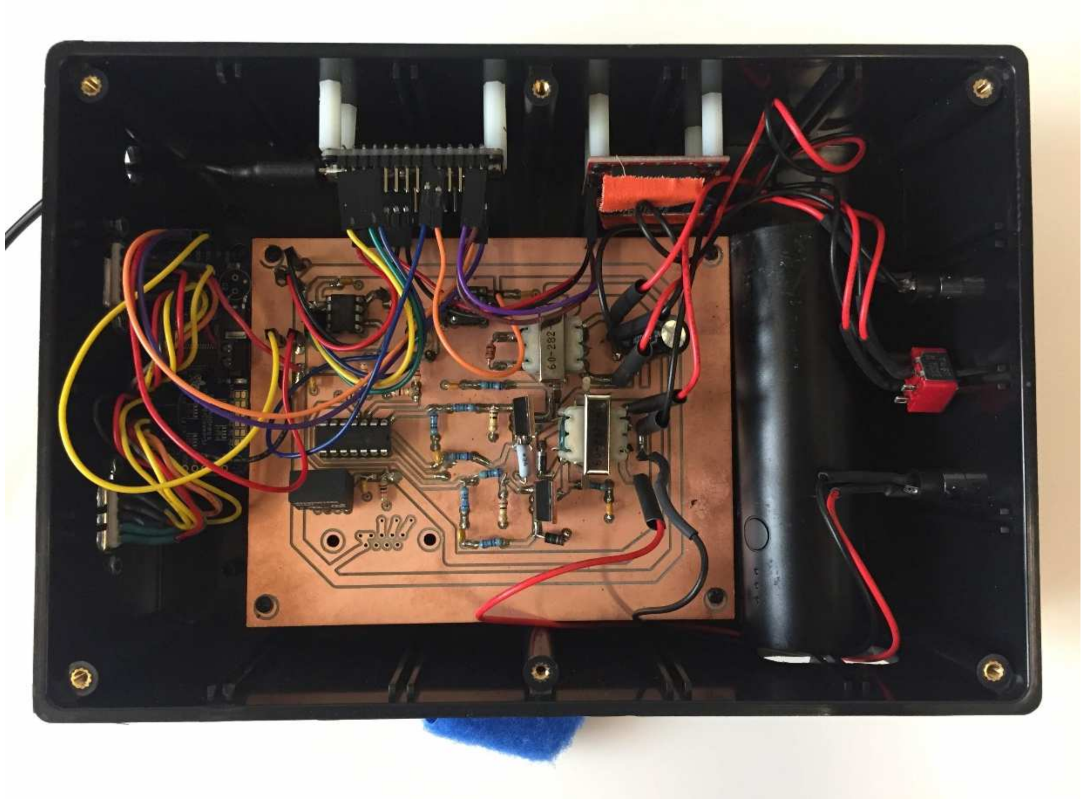

Figure 6.12: ESO electronics enclosure

Figure 6.13: Wall of the enclosure with two 3.5mm stereo jacks and a DPDT kill switch

Figure 6.14: 2 DB9 connectors are used to connect the servos with the servo driver

A 98 Ohm resistor was placed across the transformer and the stimulation pulse was monitored on an oscilloscope for various DAC values. The peak voltage recorded on the oscilloscope for a particular DAC value divided by the resistance (98) gives peak current. A plot of DAC value vs stimulation current was created to find the calibration factor (Figure 6.15).

For a DAC value of 4095 (max) the output current is not always the same even though a constant current circuit has been implemented. As the resistance across the transformer increases (or the resistance of the leg) the power demanded from the source increases. When this demand increases beyond what the power supply can provide the I_{max}, i.e. the current at max DAC value, decreases. This has been experimentally characterized by monitoring the current on an oscilloscope for various resistance values and the resulting plot is shown in Figure 6.16.

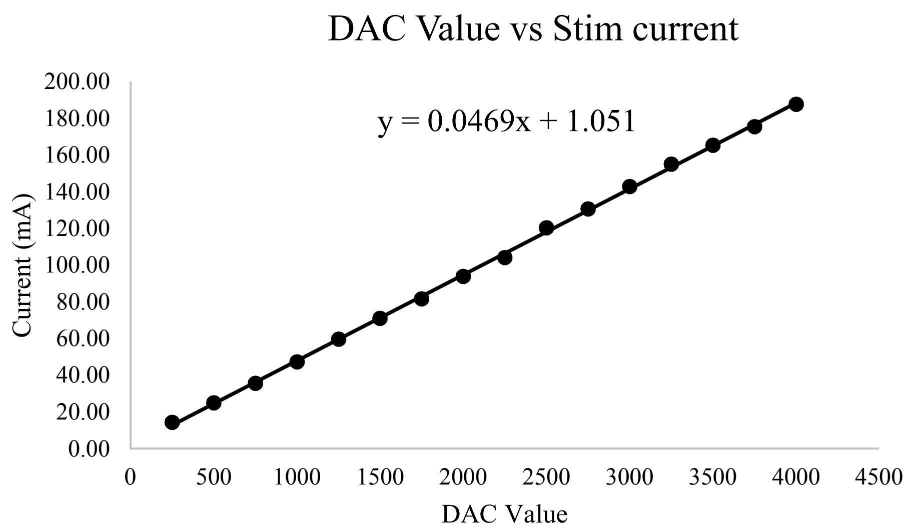

Figure 6.15: Calibration between DAC value and stimulation current

Resistance	Imax
46.2	210
98	188
216	187
465	169
980	143
1470	106
2190	75
3260	51
4580	34
6790	22
9720	20

Table 6.1: Maximum stimulation current for various resistance values

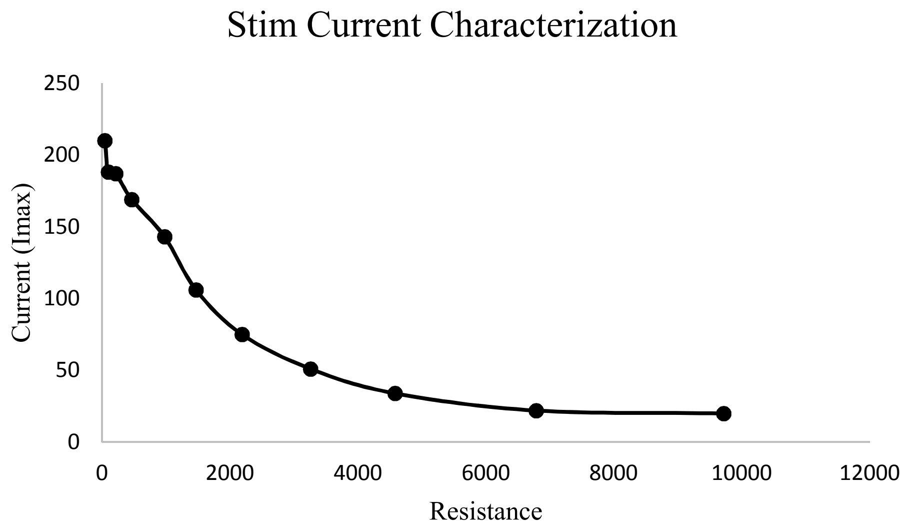

Figure 6.16: Maximum current that can be produced by the circuit decreases with increase in resistance

Arduino Code for ESO

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <SPI.h>



// called this way, it uses the default address 0x40
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
// you can also call it with a different address you want
//Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x41);
// you can also call it with a different address and I2C interface
//Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(&Wire, 0x40);

// Depending on your servo make, the pulse width min and max may vary,
you
// want these to be as small/large as possible without hitting the hard
stop
// for max range. You'll have to tweak them as necessary to match the
servos you
// have!
#define SERVOMIN  207 // this is the 'minimum' pulse length count (out
of 4096)
#define SERVOMAX  585 // this is the 'maximum' pulse length count (out
of 4096)
#define MAX_MILLIS_TO_WAIT 1000  //or whatever


/////INITALIZE GLOBAL PARAMETERS/////
```

```
unsigned long starttime;
uint8_t count = 1;
int prems = 0;
int myData = 0;
int cmd;
int data;
int LSPA, RSPA, LST, RST; //left stim pulse amplitude, right stim pulse
amplitude, left stim time, right stim time
int PW = 300; // pulse width
int PF = 40; // pulse frequency
int thr = 100;  //default pulse width threshold
int slpup = 20; //default ramp up slope
int slpdwn = 40; //default ramp down slope
int in_bytes[2];
int param_bytes[4];


// our servo # counter
uint8_t LHE = 11; // left hip extension servo pin
uint8_t LHF = 12; // left hip flexion servo pin
uint8_t LK = 13; // left knee servo pin
uint8_t RHE = 8; // right hip extension servo pin
uint8_t RHF = 9; // right hip flexion servo pin
uint8_t RK = 10; // right knee servo pin


const int PIN_CS = 10;

void setup() {                                               // setup
function for servo operation

  pinMode(PIN_CS, OUTPUT);
  pinMode(4, OUTPUT);  // Left stim
  pinMode(5, OUTPUT);  // Right stim
  SPI.begin();
  SPI.setClockDivider(SPI_CLOCK_DIV2);

  Serial.begin(9600);
  Serial.println("8 channel Servo test!");

  pwm.begin();

  pwm.setPWMFreq(60);  // Analog servos run at ~60 Hz updates

  pwm.setPWM(LHE, 0, 4096);

  delay(10);
}

// you can use this function if you'd like to set the pulse length in
seconds
// e.g. setServoPulse(0, 0.001) is a ~1 millisecond pulse width. its
not precise!
//  void setServoPulse(uint8_t n, double pulse) {
//  double pulselength;
//
```

```
//  pulselength = 1000000;   // 1,000,000 us per second
//  pulselength /= 60;   // 60 Hz
//  Serial.print(pulselength); Serial.println(" us per period");
//  pulselength /= 4096;  // 12 bits of resolution
//  Serial.print(pulselength); Serial.println(" us per bit");
//  pulse *= 1000000;  // convert to us
//  pulse /= pulselength;
//  Serial.println(pulse);
//  pwm.setPWM(n, 0, pulse);
//}


void setDAC(unsigned int val) {   // funtion that sets the DAC value
  //0.00V for val=0 (min), 4.095V for val = 4095 (max)
  byte lowByte = val & 0xff;
  byte highByte = ((val >> 8) & 0xff) | 0x10;
  PORTB &= 0xfb;
  SPI.transfer(highByte);
  SPI.transfer(lowByte);
  PORTB |= 0x4;
}

uint16_t pulselength = map(90, 0, 180, SERVOMIN, SERVOMAX);

//////////////////////////////////////////////////////////////////////
////////////////
//////////////////////////////////SERVO
FUNCTIONS/////////////////////////////////////
//////////////////////////////////////////////////////////////////////
///////////////

void fRHE(int inp){

  int var = inp;

  switch (var) {
    case 1:
      //Activate
      pwm.setPWM(RHE, 0, pulselength);
      break;
    case 2:
      //Deactivate
      pwm.setPWM(RHE, 0, 4096);
      break;
    default:
      // if nothing else matches, do the default

      break;
  }

}



void fRHF(int inp){

  int var = inp;
```

```
  switch (var) {
    case 1:
      //Activate
      pwm.setPWM(RHF, 0, pulselength);
      break;
    case 2:
      //Deactivate
      pwm.setPWM(RHF, 0, 4096);
      break;
    default:
      // if nothing else matches, do the default

      break;
  }

}


void fRK(int inp){

  int var = inp;

  switch (var) {
    case 1:
      //Activate
      pwm.setPWM(RK, 0, pulselength);
      break;
    case 2:
      //Deactivate
      pwm.setPWM(RK, 0, 4096);
      break;
    default:
      // if nothing else matches, do the default

      break;
  }

}


void fLHE(int inp){

  int var = inp;

  switch (var) {
    case 1:
      //Activate
      pwm.setPWM(LHE, 0, pulselength);
      break;
    case 2:
      //Deactivate
      pwm.setPWM(LHE, 0, 4096);
      break;
    default:
      // if nothing else matches, do the default
      break;
```

```
  }

}


void fLHF(int inp){

  int var = inp;

  switch (var) {
    case 1:
      //Activate
      pwm.setPWM(LHF, 0, pulselength);
      break;
    case 2:
      //Deactivate
      pwm.setPWM(LHF, 0, 4096);
      break;
    default:
      // if nothing else matches, do the default

      break;
  }

}



void fLK(int inp){

  int var = inp;

  switch (var) {
    case 1:
      //Activate
      pwm.setPWM(LK, 0, pulselength);
      break;
    case 2:
      //Deactivate
      pwm.setPWM(LK, 0, 4096);
      break;
    default:
      // if nothing else matches, do the default

      break;
  }

}


////////////////////////////////////////////////////////////////////////
////////////////
//////////////////////////STIMULATION
FUNCTION////////////////////////////////
////////////////////////////////////////////////////////////////////////
///////////////
```

```
void stim1(int PA, int MPW, int PF, int thr, int slpup, int slpdwn, int
pin, int t){
  int counter = 0;
  int number = 1;
  int PW = thr;
  int incr = slpup;
  int decr = slpdwn;
  int PInt = 1000/PF;
  int NRUP = (MPW-thr)/incr;
  int NRDP = (MPW)/decr;
  int NP = (t/PInt)-NRUP-NRDP;
  int firstpulse = 1;
  int curms = millis();

  while(PW!=0){
     curms = millis();
     if (((curms-prems) >= PInt) || (firstpulse == 1)){  /// if a
certain  number of milliseconds have gone by after previous pulse or if
it's the first pulse
      firstpulse = 0;
      switch (number) {
         case 1: //Ramp up
         onepulse(PA,PW,pin);
         PW = PW + incr;
         if(PW >= MPW ){ // ramp up till max pulse width with an
increment of incr then switch to next case
         number = 2;
         }
         break;
         case 2:
         onepulse(PA,MPW,pin); // send pulses at max pulse width
         counter++;
         if(counter>NP){
         number = 3;
         }
         break;
         case 3:
         onepulse(PA,PW,pin);
         PW = PW - decr; // ramp down with decr
         break;

         default:
         // if nothing else matches, do the default
         break;

     }

     prems = millis();

  }

    if (PW <= 0){ // while ramping down if pulse width becomes 0 or
less; stim is done so break out of loop
    prems = 0;
    break;
   }
```

```
 }

}


void onepulse(int PA, int PW, int pin){ // funtion that sends one pulse
of a certain amplitude and width

  setDAC(PA);
  digitalWrite(pin,HIGH);
  delayMicroseconds(PW);
  digitalWrite(pin,LOW);
}



void loop() {

  starttime = millis();

  while ( (Serial.available()<2) && ((millis() - starttime) <
MAX_MILLIS_TO_WAIT) )
   {
     // hang in this loop until we either get 2 bytes of data or 1
second
     // has gone by
   }

  if(Serial.available() < 2)
   {
     // the data didn't come in - handle that problem here
     //Serial.println("ERROR - Didn't get 2 bytes of data!");
   }
  else
   {
     for(int n=0; n<2; n++){
     in_bytes[n] = Serial.read();} // Then: Get them.
     cmd = in_bytes[0];
    data = in_bytes[1];

     switch (cmd) {
         case 1: //LHE
         fLHE(data);
         break;
         case 2: //LHF
         fLHF(data);
         break;
         case 3: //LK
         fLK(data);
         break;
         case 4: //RHE
         fRHE(data);
         break;
         case 5: //RHF
         fRHF(data);
```

```
        break;
        case 6: //RK
        fRK(data);
        break;
        case 7: //Set L stim PA
        LSPA = (data*20); // data was divided by 20 on the GUI side as
one byte has only 256 characters and we need to send values up t0 4095
        break;
        case 8: //Set R stim PA
        RSPA = (data*20);
        break;
        case 9: //Set L stim T
        LST = (data*1000);    // data is in sec LST is in millisec
        break;
        case 10: //Set R stim T
        RST = (data*1000);   // data is in sec LST is in millisec
        break;
        case 11: //Go L stim
        Serial.println("StimCalled");
        stim1(LSPA,PW,PF,thr,slpup,slpdwn,data,LST);
        break;
        case 12: //Go R stim
        stim1(RSPA,PW,PF,thr,slpup,slpdwn,data,RST);
        break;
        case 13: //L step

        break;
        case 14: //R step

        break;

        case 15: // Set Pulse Width
        PW = (data*10); // data divided by 10 on GUI side to be able
to send PW values greater than 256
        break;

        case 16: // Set Pulse Frequency
        PF = data;


        case 17: // Set Pulse Width Threshold
        thr = data;
        break;

        case 18: // Set Rampup Slope
        slpup = data;
        break;

        case 19: // Set Rampdown slope
        slpdwn = data;
        break;


        default:
        // if nothing else matches, do the default
        break;
```

```
      }
    }
}
```

Python code for ESO GUI

```
##---------------------------------------------------
##                 ESO GUI
##---------------------------------------------------

from tkinter import *  # import the gui library
from tkinter.ttk import * # use the pretty version
##from PIL import Image, ImageTk
from tkinter import ttk
import tkinter as tk
from tkinter.scrolledtext import ScrolledText
import serial          # import the serial library
import math
import time
from tkinter import Tk, Canvas, Frame, BOTH



#-----Send info to PICmicro
def sendBYTES(cmd,data):
    x=bytearray([cmd,data])  # form a two byte array
    ser.write(x)             # send to the arduino
    return

#-----Send 4 bytes to PICmicro
def send4BYTES(d1,d2,d3,d4):
    x=bytearray([d1,d2,d3,d4])  # form a two byte array
    ser.write(x)                # send to the arduino
    return

#-----Start handler
def dataStart():
    sendBYTES(1,0) # tell PIC to start sending data
    ser.flushInput() # clean out serial input
    updateData() # call update for first time
    return

#-----Stop handler
def dataStop():
    sendBYTES(2,0) # tell PIC to stop sending data
    return

#-----LED flash handler
```

```python
def flasher():
    global led
    if led:
        sendBYTES(3,0) # turn off
        led=False
    else:
        sendBYTES(3,1) # turn on
        led=True
    return

#-----Activate LHE
def ALHE():
    sendBYTES(1,1)
    return

#-----Deactivate LHE
def DLHE():
    sendBYTES(1,2)
    return

#-----Activate LHF
def ALHF():
    sendBYTES(2,1)
    return

#-----Deactivate LHF
def DLHF():
    sendBYTES(2,2)
    return

#-----Activate LK
def ALK():
    sendBYTES(3,1)
    return

#-----Deactivate LK
def DLK():
    sendBYTES(3,2)
    return

#-----Activate LEqm
def ALEqm():
    sendBYTES(2,1)
    sendBYTES(3,1)
    return

#-----Deactivate LEqm
def DLEqm():
    sendBYTES(2,2)
    sendBYTES(3,2)
    return


#-----Activate RHE
def ARHE():
    sendBYTES(4,1)
    return
```

```
#-----Deactivate RHE
def DRHE():
    sendBYTES(4,2)
    return

#-----Activate RHF
def ARHF():
    sendBYTES(5,1)
    return

#-----Deactivate RHF
def DRHF():
    sendBYTES(5,2)
    return

#-----Activate RK
def ARK():
    sendBYTES(6,1)
    return

#-----Deactivate RK
def DRK():
    sendBYTES(6,2)
    return

#-----Activate REqm
def AREqm():
    sendBYTES(5,1)
    sendBYTES(6,1)
    return

#-----Deactivate REqm
def DREqm():
    sendBYTES(5,2)
    sendBYTES(6,2)
    return

#-----Set Left Stim PA
def LStimPA():
##    x = int(text_label1.get())
##    if (x > 150):
##        x = 150
##    x = (x - 1.051)//(0.0469 * 20)  ###conversion factor to allow
transfer to arduino (1 byte has been allocated which means only 256
values)
##    LPA = math.ceil(x)
    LPA = int(text_label1.get())
    LPA = LPA//20
    sendBYTES(7,LPA)
    return

#-----Set Right Stim PA
def RStimPA():
    RPA = int(text_label2.get())
    RPA = RPA //20
    sendBYTES(8,RPA)
```

```python
    return

#-----Set Left Stim Time
def LStimT():
    LST = int(text_label3.get())
    sendBYTES(9,LST)
    return

#-----Set Right Stim Time
def RStimT():
    RST = int(text_label4.get())
    sendBYTES(10,RST)
    return

#-----Set Pulse Width
def PWidth():
    PW = int(text_label5.get())
    PW = PW//10
    sendBYTES(15,PW)
    return

#-----Set Pulse Frequency
def PFreq():
    PF = int(text_label6.get())
    sendBYTES(16,PF)
    return

#-----Set Pulse Width threshold
def PWthr():
    PWthr = int(text_label7.get())
    sendBYTES(17,PWthr)
    return

#-----Set Rampup slope
def SlopeUp():
    slpup = int(text_label8.get())
    sendBYTES(18,slpup)
    return

#-----Set Rampdown slope
def SlopeDown():
    slpdwn = int(text_label9.get())
    sendBYTES(19,slpdwn)
    return

#-----GO Left Stim
def GOLS():
    sendBYTES(11,4)
    return

#-----GO Right Stim
def GORS():
    sendBYTES(12,5)
    return

#-----Left Step
```

```
def LeftStp():
    return

#-----Right Step

def RightStp():
    return

#-----Set Left Param

def SetLP():
    LStimPA()
    LStimT()
    PWidth()
    PFreq()
    return

#-----Set Right Param

def SetRP():
    RStimPA()
    RStimT()
    PWidth()
    PFreq()
    return


#-----Set threshold, slope

def SetThrSlp():
    PWthr()
    SlopeUp()
    SlopeDown()
    return

#-----Close and quit
def alldone():
    ser.close()
    root.destroy()

##--------------------------
##  main
##--------------------------

#-----Set up the serial port
port="COM5"       #edit for your com port number
ser=serial.Serial(port, 9600, timeout=0, writeTimeout=0) #non-blocking
led=False
V = 5
counter=0
countdown1=0
countdown2=0
countdown3=0
countdown4=0
```

```python
#-----Create root-----
root = tk.Tk()    # main window
root.option_add("*Font", "verdana 10")
root.title("ESO")
root.geometry('{}x{}'.format(1920, 1280))

top_frame = Frame(root, width=750, height=200)
top_frame.grid(row=0, sticky="ew")
bottom_frame = Frame(root, width=750, height=250)
bottom_frame.grid(row=1, sticky="ew")

nb = ttk.Notebook(root)
nb.grid(row=0, column=0, columnspan=50, rowspan=49, sticky='NESW')

# Adds tab 1 of the notebook
page1 = ttk.Frame(nb)
nb.add(page1, text='Stim Test')

# Adds tab 2 of the notebook
page2 = ttk.Frame(nb)
nb.add(page2, text='Stim Test_Secondary')

# Adds tab 3 of the notebook
page3 = ttk.Frame(nb)
nb.add(page3, text='System Test')

# Adds tab 4 of the notebook
page4 = ttk.Frame(nb)
nb.add(page4, text='Main')

#-----Create sensor data frame-----

f1 = Frame(page4, width=250, height=200)
xf1 = Frame(f1, relief=GROOVE, borderwidth=2)
xf1.place(relx=0.01, rely=0.125, anchor=NW)
Label(f1, text='Sensor Data').place(relx=.06, rely=0.125,anchor=W)
f1.pack(side=LEFT)


####----Stim test window

lbl_label0=Label(page1,text="Stim Test Window",)
lbl_label0.grid(row=0,column=4,padx=5,pady=8,sticky=W)

lbl_label1=Label(page1,text="Set LPA (mA)",)
lbl_label1.grid(row=1,column=0,padx=5,pady=8,sticky=W)

text_label1=Entry(page1,width=10,justify=RIGHT)
text_label1.grid(row=1,column=1)

lbl_label2=Label(page1,text="Set RPA (mA)",)
lbl_label2.grid(row=1,column=5,padx=5,pady=8,sticky=W)

text_label2=Entry(page1,width=10,justify=RIGHT)
text_label2.grid(row=1,column=6)

lbl_label3=Label(page1,text="Set LST (in s)",)
```

```
lbl_label3.grid(row=2,column=0,padx=5,pady=8,sticky=W)

text_label3=Entry(page1,width=10,justify=RIGHT)
text_label3.grid(row=2,column=1)

lbl_label4=Label(page1,text="Set RST (in s)",)
lbl_label4.grid(row=2,column=5,padx=5,pady=8,sticky=W)

text_label4=Entry(page1,width=10,justify=RIGHT)
text_label4.grid(row=2,column=6)

lbl_label5=Label(page1,text="Set PW (in ms)",)
lbl_label5.grid(row=3,column=0,padx=5,pady=8,sticky=W)

text_label5=Entry(page1,width=10,justify=RIGHT)
text_label5.grid(row=3,column=1)

lbl_label6=Label(page1,text="Set PF (in Hz)",)
lbl_label6.grid(row=3,column=5,padx=5,pady=8,sticky=W)

text_label6=Entry(page1,width=10,justify=RIGHT)
text_label6.grid(row=3,column=6)

btn1 = Button(page1,text="SetLP",command=SetLP)  # Set Parameters
btn1.grid(row=4,column=0,padx=5,pady=4, sticky=W)

btn2 = Button(page1,text="GoLStim",command=GOLS)  # Set Parameters
btn2.grid(row=4,column=1,padx=5,pady=4, sticky=W)

btn3 = Button(page1,text="SetRP",command=SetRP)  # Set Parameters
btn3.grid(row=4,column=5,padx=5,pady=4, sticky=W)

btn3 = Button(page1,text="GoRStim",command=GORS)  # Set Parameters
btn3.grid(row=4,column=6,padx=5,pady=4, sticky=W)


#####-----Stim test secondary parameters

lbl_label7=Label(page2,text="Set PWthr (0-300)",)
lbl_label7.grid(row=1,column=0,padx=5,pady=8,sticky=W)

text_label7=Entry(page2,width=10,justify=RIGHT)
text_label7.grid(row=1,column=1)

lbl_label8=Label(page2,text="Set Rampup slope",)
lbl_label8.grid(row=2,column=0,padx=5,pady=8,sticky=W)

text_label8=Entry(page2,width=10,justify=RIGHT)
text_label8.grid(row=2,column=1)

lbl_label9=Label(page2,text="Set Rampdown slope",)
lbl_label9.grid(row=3,column=0,padx=5,pady=8,sticky=W)

text_label9=Entry(page2,width=10,justify=RIGHT)
text_label9.grid(row=3,column=1)
```

```
btn4 = Button(page2,text="Set Thr,Slope",command=SetThrSlp)  # Set
threshold, slope
btn4.grid(row=4,column=1,padx=5,pady=4, sticky=W)




####---------System Test Window

lbl_label10=Label(page3,text="System Test Window",)
lbl_label10.grid(row=0,column=4,padx=5,pady=8,sticky=W)

lbl_label11=Label(page3,text="Left",)
lbl_label11.grid(row=1,column=2,padx=5,pady=8,sticky=W)

lbl_label12=Label(page3,text="Stim",)
lbl_label12.grid(row=3,column=0,padx=5,pady=8,sticky=W)

btn5 = Button(page3,text="Activate",command=GOLS)  # Set Parameters
btn5.grid(row=3,column=1,padx=5,pady=8, sticky=W)

lbl_label13=Label(page3,text="HE",)
lbl_label13.grid(row=4,column=0,padx=5,pady=8,sticky=W)

btn6 = Button(page3,text="Unlock",command=ALHE)  # Set Parameters
btn6.grid(row=4,column=1,padx=5,pady=4, sticky=W)

btn7 = Button(page3,text="Lock",command=DLHE)  # Set Parameters
btn7.grid(row=4,column=2,padx=5,pady=4, sticky=W)

lbl_label14=Label(page3,text="HF",)
lbl_label14.grid(row=5,column=0,padx=5,pady=8,sticky=W)

btn8 = Button(page3,text="Unlock",command=ALHF)  # Set Parameters
btn8.grid(row=5,column=1,padx=5,pady=4, sticky=W)

btn9 = Button(page3,text="Lock",command=DLHF)  # Set Parameters
btn9.grid(row=5,column=2,padx=5,pady=4, sticky=W)

lbl_label15=Label(page3,text="K",)
lbl_label15.grid(row=6,column=0,padx=5,pady=8,sticky=W)

btn10 = Button(page3,text="Unlock",command=ALK)  # Set Parameters
btn10.grid(row=6,column=1,padx=5,pady=4, sticky=W)

btn11 = Button(page3,text="Lock",command=DLK)  # Set Parameters
btn11.grid(row=6,column=2,padx=5,pady=4, sticky=W)


lbl_label16=Label(page3,text="Eqm",)
lbl_label16.grid(row=7,column=0,padx=5,pady=8,sticky=W)

btn11 = Button(page3,text="Unlock",command=ALEqm)  # Set Parameters
btn11.grid(row=7,column=1,padx=5,pady=4, sticky=W)

btn12 = Button(page3,text="Lock",command=DLEqm)  # Set Parameters
```

```
btn12.grid(row=7,column=2,padx=5,pady=4, sticky=W)


lbl_label17=Label(page3,text="Right",)
lbl_label17.grid(row=1,column=7,padx=5,pady=8,sticky=W)

lbl_label18=Label(page3,text="Stim",)
lbl_label18.grid(row=3,column=5,padx=5,pady=8,sticky=W)

btn13 = Button(page3,text="Activate",command=GORS)  # Set Parameters
btn13.grid(row=3,column=6,padx=5,pady=4, sticky=W)

lbl_label19=Label(page3,text="HE",)
lbl_label19.grid(row=4,column=5,padx=5,pady=8,sticky=W)

btn14 = Button(page3,text="Unlock",command=ARHE)  # Set Parameters
btn14.grid(row=4,column=6,padx=5,pady=4, sticky=W)

btn15 = Button(page3,text="Lock",command=DRHE)  # Set Parameters
btn15.grid(row=4,column=7,padx=5,pady=4, sticky=W)

lbl_label120=Label(page3,text="HF",)
lbl_label120.grid(row=5,column=5,padx=5,pady=8,sticky=W)

btn16 = Button(page3,text="Unlock",command=ARHF)  # Set Parameters
btn16.grid(row=5,column=6,padx=5,pady=4, sticky=W)

btn17 = Button(page3,text="Lock",command=DRHF)  # Set Parameters
btn17.grid(row=5,column=7,padx=5,pady=4, sticky=W)

lbl_label21=Label(page3,text="K",)
lbl_label21.grid(row=6,column=5,padx=5,pady=8,sticky=W)

btn18 = Button(page3,text="Unlock",command=ARK)  # Set Parameters
btn18.grid(row=6,column=6,padx=5,pady=4, sticky=W)

btn19 = Button(page3,text="Lock",command=DRK)  # Set Parameters
btn19.grid(row=6,column=7,padx=5,pady=4, sticky=W)

lbl_label22=Label(page3,text="Eqm",)
lbl_label22.grid(row=7,column=5,padx=5,pady=8,sticky=W)

btn20 = Button(page3,text="Unlock",command=AREqm)  # Set Parameters
btn20.grid(row=7,column=6,padx=5,pady=4, sticky=W)

btn21 = Button(page3,text="Lock",command=DREqm)  # Set Parameters
btn21.grid(row=7,column=7,padx=5,pady=4, sticky=W)




###-----Create Manual Mode frame
##f4 = Frame(bottom_frame, width=500, height=250)
```

```
##xf4 = Frame(f4, relief=GROOVE, borderwidth=2)
##xf4.place(relx=0.01, rely=0.125, anchor=NW)
##Label(f4, text='Manual Mode').place(relx=.06, rely=0.125,anchor=W)
##f4.pack(side=RIGHT)


root.mainloop()
```

E. Bill of Materials

ESO Structure BOM

S.No	Description	Supplier	Part Number	Quantity	Unit Cost	Total Cost
1	Molle backpack frame, shoulder straps, waist belt, and padding set	Amazon	B00CHSR1C8	1	$65.00	$65.00
2	18-8 Stainless Steel Socket Head Screw 3/8"-16 Thread Size, 1-1/4" Long (Packs of 10)	McMaster Carr	92196A626	2	$7.62	$15.24
3	316 Stainless Steel Washer for 3/8" Screw Size, 0.406" ID, 0.75" OD	McMaster Carr	90107A127	1	$8.80	$8.80
4	High-Strength Steel Nylon-Insert Locknut Grade 8, Zinc Yellow-Chromate Plated, 3/8"-16 Thread Size	McMaster Carr	97135A230	1	$4.18	$4.18
5	Passivated 18-8 Stainless Steel Pan Head Phillips Screw 1/4"-20 Thread, 1-1/2" Long	McMaster Carr	91772A546	1	$9.39	$9.39
6	Super-Cushioning High-Strength EVA Foam Sheet 1/2" Thick, 29" x 47"	McMaster Carr	86095K24	2	$51.41	$102.82
7	High-Strength Steel Nylon-Insert Locknut Grade 8, Zinc Yellow-Chromate Plated, 1/4"-20 Thread Size	McMaster Carr	97135A210	3	$3.56	$10.68
8	Alloy Steel Shoulder Screw 1/4" Diameter 1/2" Long Shoulder, 10-24 Thread	McMaster Carr	91259A537	2	$1.09	$2.18
9	Plate-Mount Retractable Spring Plunger Steel, without Lock Nose, L 1.73", 3.5-6.3 lb. Nose Force	McMaster Carr	8478A2	4	$15.32	$61.28
10	316 Stainless Steel Hex Drive Flat Head Screw 82 Degree Countersink Angle, 10-24 Thread Size, 3/8" Long	McMaster Carr	90585A240	1	$3.04	$3.04
11	316 Stainless Steel Washer for Number 10 Screw Size, 0.203" ID, 0.438" OD	McMaster Carr	90107A011	1	$4.24	$4.24
12	18-8 Stainless Steel Nylon-Insert Locknut 10-24 Thread Size	McMaster Carr	91831A011	1	$6.27	$6.27
13	Aluminum 5052 12"x15"x0.125"	Discount Steel	1948	1	$25.37	$25.37
14	Al 6061 12"x24"x0.5"	Discount Steel	742	1	$99.77	$99.77
15	3D printing flament PLA	Amazon	B01FKESAMA	1	$23.00	$23.00
16	Wrap spring	Reel Precision Mfg	Custom	6	$0.00	$0.00
17	Al 6061 3"x12"x0.25"	Discount Steel	725	1	$19.00	$19.00
18	Al 6061 12"x32"x0.125"	Discount Steel	3422	1	$49.35	$49.35
19	ASTM A276-10 304 Stainless Steel Flat Bar 3/16" x 3/4"x 144"	Discount Steel	1959	1	$25.82	$25.82
20	Al 6061 rod 3/8" diameter 80" length	Discount Steel	12058	1	$8.71	$8.71
21	Al 6061 rod 0.5" diameter 18" long	Discount Steel	12059	1	$6.40	$6.40
22	Al 6061 3"x6"x0.75"	Discount Steel	788	1	$40.50	$40.50
23	Mild Steel 2"x 24"x 0.5"	Discount Steel	2623	1	$78.24	$78.24
24	Clamping Two-Piece Shaft Collar for 3/8" Diameter, Black-Oxide 1215 Carbon Steel	McMaster Carr	6436K13	4	$3.85	$15.40
25	18-8 Stainless Steel Socket Head Screw 8-32 Thread Size, 1/2" Long	McMaster Carr	92196A194	1	$6.88	$6.88
26	18-8 Stainless Steel Socket Head Screw 10-24 Thread Size, 1/2" Long	McMaster Carr	92196A242	1	$9.06	$9.06
27	18-8 Stainless Steel Socket Head Screw, 10-24 Thread Size, 3/4" Long	McMaster Carr	92196A245	1	$11.19	$11.19
28	Rubber Bumper with Unthreaded Hole, Heavy Duty, SBR with Steel Washer, 1/2" OD, 1/2" High, 5/32" Hole	McMaster Carr	9540K724	1	$11.72	$11.72
29	Dowel Pin, 316 Stainless Steel, 1/4" Diameter, 3/4" Long	McMaster Carr	97395A485	1	$11.99	$11.99
30	Plastic Round Shims, 0.005" Thick, 1/2" ID	McMaster Carr	90334A112	2	$13.31	$26.62
31	Ultra-Low-Friction Dry-Running Sleeve Bearing, Rulon J, for 1/2" Shaft Diameter & 3/4" Housing ID, 1/2" Long	McMaster Carr	6377K37	12	$7.30	$87.60
32	Passivated 18-8 Stainless Steel Pan Head Phillips Screw 1/4"-20 Thread, 1/2" Long	McMaster Carr	91772A537	1	$10.28	$10.28
33	Passivated 18-8 Stainless Steel Pan Head Phillips Screw 1/4"-20 Thread, 3/4" Long	McMaster Carr	91772A540	1	$13.09	$13.09
34	Passivated 18-8 Stainless Steel Pan Head Phillips Screw 1/4"-20 Thread, 1" Long	McMaster Carr	91772A542	1	$8.76	$8.76
35	18-8 Stainless Steel Socket Head Screw, 1/4"-20 Thread Size, 1/2" Long	McMaster Carr	92196A537	1	$9.61	$9.61
36	18-8 Stainless Steel Socket Head Screw, 1/4"-20 Thread Size, 3/4" Long	McMaster Carr	92196A540	1	$10.87	$10.87
37	18-8 Stainless Steel Socket Head Screw, 1/4"-20 Thread Size, 1" Long	McMaster Carr	92196A542	1	$14.93	$14.93
38	Alloy Steel Shoulder Screw, 1/2" Diameter 1/2" Long Shoulder, 3/8"-16 Thread	McMaster Carr	91259A707	2	$2.27	$4.54
39	316 Stainless Steel Shoulder Screw, 1/4" Diameter 1-3/8" Long Shoulder, 10-24 Thread	McMaster Carr	97345A168	2	$7.39	$14.78
40	Gas Spring, 9.65" Extended Length, 30 lbs. Extension Force	McMaster Carr	4138T533	2	$17.51	$35.02
41	Gas Spring, 9.65" Extended Length, 15 lbs. Extension Force	McMaster Carr	4138T531	2	$17.51	$35.02
42	Gas Spring, 15.63" Extended Length, 30 lbs. Force	McMaster Carr	9416K172	2	$15.78	$31.56
43	Ball Joint Linkage with Blended Rubber Seal, M6 x 1 mm Internal Thread, Right Hand	McMaster Carr	6275K53	4	$11.34	$45.36
44	Multiform Max, 36"W x 24"L x 1/8" thick, Solid Sheet, 2/cs	Alimed	4663	1	$226.75	$226.75
45	Hook and Loop Cable Tie with Buckle 36" Overall Length,black 2" wide	McMaster Carr	3955T289	2	$3.48	$6.96
46	Hook and Loop Cable Tie with Buckle 24" Overall Length, black 2" wide	McMaster Carr	3955T286	6	$2.85	$17.10
47	Hook and Loop Cable Tie with Buckle, 18" Overall Length 2" wide	McMaster Carr	3955T285	6	$2.52	$15.12
48	EZ ShoePAD 1/8" black shoe rubber soling sheet Shoe soles repairing rubber sheet. Shoes bottom repairing material 1 pack	Amazon	B075TTDBGT	2	$24.50	$49.00
49	4-Pack Polar Fleece Tie-Down Strap Sleeve Covers	Amazon	B00UHLNQ4O	2	$23.99	$47.98

ESO Electronics BOM

REF	DESCRIPTION	SUPPLIER	SUPPLIER PN	COST	Notes
MODULES					
U1	Arduino, Metro Mini 328, 5V, Adafruit 2590	Digi-Key	1528-1374-ND	$12.50	
U2	Servo Driver, 16-Channel, Adafruit 815	Digi-Key	1528-1008-ND	$14.95	
U3	Xbee Series 1, wire antenna, Digi XB24-AWI-001	Digi-Key	XB24-AWI-001-ND	$19.29	
	Xbee Series 1, trace antenna, Digi XB24-API-001	Digi-Key	602-1273-ND	$19.00	
	Board, Xbee Explorer, Sparkfun WRL-11373	Digi-Key	1568-1075-ND	$9.95	
	Board, Xbee Explorer, USB, Sparkfun WRL-11812	Digi-Key	1568-1305-ND	$24.95	
U6	DC-DC converter, 5Vin-15Vout, 3W, 4PSIP	Digi-Key	811-2357-5-ND	$11.84	
INTEGRATED CIRCUITS					
U7	MC34074 quad op amp, 14-PDIP	Digi-Key	MC34074PGOS-ND	$1.42	
U5	MAX323 analog switch, dual, 8-PDIP	Digi-Key	MAX323CPA+-ND	$3.58	
U4	MCP4821 12-bit DAC, 8-PDIP	Digi-Key	MCP4821-E/P-ND	$2.44	
SEMICONDUCTORS					
Q1	MOSFET, N-Chan, 30V, 78A, IRLB8743, TO-220	Digi-Key	IRLB8743PBF-ND	$1.57	
D1	Diode Standard 1000V 1A Through Hole DO-41	Digi-Key	1N4007DICT-ND	$0.16	
DISCRETES					
C7,C9	10000pF 50V Ceramic Capacitor X7R Radial	Digi-Key	490-8813-ND	$0.32	
C1,C3,C4,C6,C8,C10	0.1µF 50V Ceramic Capacitor X7R Radial	Digi-Key	490-8815-ND	$0.29	
C2	10µF 25V Ceramic Capacitor X7S Radial	Digi-Key	490-14506-ND	$0.80	
C5	1000µF 50V Aluminum Electrolytic Capacitors Radial	Digi-Key	4603PHBK-ND	$1.14	
R10,R18	500 mOhms ±1% 1W Through Hole Resistor	Digi-Key	PPC1D.50CT-ND	$0.25	
R6,R7,R8,R11,R14,R15,R16,R19	47 Ohms ±1% 0.25W, 1/4W Through Hole Resistor	Digi-Key	S47CATR-ND	$0.10	
R3	1 kOhms ±1% 0.25W, 1/4W Through Hole Resistor Axial Metal Film	Digi-Key	1.00KXTR-ND	$0.10	
R1,R5,R13	10 kOhms ±1% 0.25W, 1/4W Through Hole Resistor	Digi-Key	RNF14FTD10K0TR-ND	$0.10	
R4,R12	22 kOhms ±1% 0.25W, 1/4W Through Hole Resistor	Digi-Key	S22KCATR-ND	$0.10	
R2	10 MOhms ±1% 0.5W, 1/2W Through Hole Resistor	Digi-Key	PPCHF10MTR-ND	$0.54	
R9,R17	100 kOhms ±1% 0.25W, 1/4W Through Hole Resistor	Digi-Key	100KXTR-ND	$0.10	
T1,T2	Transformer, audio, Mode Electronics 60-282-0	See notes	60-282-0	$2.70	From onlinecomponents.com
	Servo Motors, Hitec HS-35 HD (Qty-6)	Digi-Key	1568-1572-ND	$35.00	
	Toggle Switch DPDT Panel Mount	Digi-Key	EG2400-ND	$2.79	
	PALS Axelgaard electrodes rectangle 3"x4"	Notes			Stocked at VA Minneapolis
CONNECTORS					
	9 Position D-Sub Plug, Male Pins Connector	Digi-Key	AE10972-ND	$0.51	
	9 Position D-Sub Receptacle, Female Sockets Connector	Digi-Key	AE11063-ND	$0.51	
	D connector cover	Digi-Key			
	3.5mm Stereo Panel Mount Jack	Amazon	B000ML4A2Q	$4.90	
	Tens Lead Wires - 3.5mm plug to Two 2mm Pin Connectors (2) - Discount Tens	Amazon	B01DZBDBZG	$7.95	
	USB 2.0 Cable A Male to Micro B Male 1.00'	Digi-Key	Q968-ND	$2.65	
	0.1" Crimp Connector Housing single row 1,2,3,4,5 pins	Notes			Sourced from ECE Depot UMN
	Wires with pre-crimped terminals m-f	Notes			Sourced from ECE Depot UMN
HARDWARE					
	Printed circuit board, ExpressPCB	Notes			Printed at Anderson Labs
	Enclosure	Digi-Key	HM1048-ND	$14.53	
	Nylon Pan Head Screws, Phillips, 2-56 Thread, 1/4" Longws	McMaster Carr	94735A707	$5.55	
	Nylon 6/6 Plastic Hex Standoff, 3/16" Hex, 5/8" Long, 2-56 Female Thread	McMaster Carr	92319A270	$1.78	
	Hook-Up Wire Kit 22G Stranded Wire, 25 ft. Spools	EE	27WK22STR25	$18.50	
	Anker PowerCore 5000, Ultra-Compact 5000mAh External Battery	Amazon	B01CU1EC6Y	$19.00	

F. VA Study Forms

Volunteer Information Form

Date:

Volunteer Number: 1001

Age:

Gender:

Height (cm):

Weight (Kg):

Anthropometric measurements:

	Left	Right
Length from center of hip to center of knee (cm)		
Length from center of knee to ankle (cm)		
Length from ankle to base of the foot (cm)		
Length of the foot (cm)		
Maximum width of the foot (cm)		
Width from hip to hip (cm)		
Circumference of largest part of calf (cm)		
Circumference of largest part of thigh (cm)		

Injury Information:

Date of injury:

Level:

Reason of Injury:

Description of Injury:

History of Stimulation:

Range of motion (hip, knee, ankle):

*Center of hip – Greater Trochanter; Center of knee - The midpoint between the femoral condyle and tibial ridge

Figure 6.17: Form to record the volunteers leg measurements

Session #:

Date:

Volunteer ID:

Purpose of session:

Start time:

End time:

Summary:

Notebook pages:

Additional comments:

Figure 6.18: Session log form

G. Recommendation

Testing at the VA brought to light some issues with the current design. Firstly, there were difficulties in getting the spring plungers to lock into place after going from sit to stand. This was because the joint angles of the volunteer's body needed to match up to the joint angles on the exoskeleton for the plungers to lock and getting them to match up is cumbersome for a person with SCI as he/she is bearing body weight through arms while the physical therapist is trying to get the joints to line up. Secondly, there was relative movement between the volunteer's body and the exoskeleton because the body attachment parts weren't rigid enough and the wrap spring slipped by a few degrees every time it was loaded. Also, in the current design FES torque requirement and achievable hip flexion angle are negatively correlated. To have sufficient flexion at the hip so that the volunteer's foot clears the ground, the FES torque requirement could potentially be greater than the capability of the volunteer's quadriceps.

The recommendation is to use a non-back drivable motor at the hip and gas spring at the knee, replace the rotational hard stops at the knee with an actuated hard stop, and use a ratchet brake at the knee. Motorizing the hip unlinks hip flexion and torque requirement from the quadriceps. This means that flexion angle of the hip can be increased without increasing the energy extracted from the quadriceps. Since there is no gas spring at the hip, there is no need for a spring plunger to go from sit to stand. The non-back drivable motor also acts as the braking mechanism at the hip eliminating the need for wrap springs at the hip. The knee joint has a gas spring connected to it for knee flexion. This retains the FES aspect which is important to prevent muscle atrophy. To go from sit to stand, the knee gas spring need not be unlinked if the gas springs range of motion covers the sit to stand range.

This overcomes the issues associated with the locking of spring plunger. An actuated hard stop can be used in place of the rotational hard stops and they could allow switching between walking range of motion and sit to stand range of motion at the knee. The wrap spring brake at the knee can be replaced with a ratchet brake to overcome the issue of wrap spring slip. The ratchet brake would require a high resolution of teeth.