Approaches to Minimally Actuated Exoskeleton Gait Development

by

Bradley Perry

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering: Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Homayoon Kazerooni, Chair
Professor Kameshwar Poolla
Professor Pieter Abbeel

Summer 2016

Approaches to Minimally Actuated Exoskeleton Gait Development

Abstract

Approaches to Minimally Actuated Exoskeleton Gait Development

by

Bradley Perry

Doctor of Philosophy in Engineering: Mechanical Engineering

University of California, Berkeley

Professor Homayoon Kazerooni, Chair

There are 6.8 million people in the United States that have mobility disorders and must rely on the use of assistive devices to aid in locomotion [23]. The human body was designed for a particular bipedal locomotion pattern, and deviations from that method of locomotion can result in secondary injuries. Many of the current solutions for mobility disorders are primitive, such as crutches or canes if possible and wheelchairs when not. In recent years, more intelligent, robotic systems have been developed to aid this population that needs assistance walking. Among these new robotic solutions are robotic exoskeletons.

Robotic exoskeletons have become a more popular rehabilitation tool in recent years, particularly for those with spinal cord injury. There are currently three robotic exoskeletons with US Food and Drug Administration approval for clinical use. One barrier to wider adoption is the cost of these devices. To reduce cost, a more minimal design was developed at the Human Engineering and Robotics Laboratory at the University of California with only two actuators at the hips and a semi-passive locking mechanism at the knee. While this design can help reduce the weight and cost, it introduces additional complexity to the gait development due to the lack of actuation.

The gait development strategies discussed in this paper are inspired by clinical gait data collected from healthy subjects. After all, the ultimate goal of an exoskeleton system as a rehabilitation device is to rehabilitate the pilot to the point that the patient's gait is restored to a natural walking pattern and the device is no longer necessary. Using clinical gait data and biomechanical studies, several models are developed for the design considered. A kinematic model was developed to better understand the angular constraints during both single stance and double stance. A dynamic model was also developed that models the behaviour of the system for different phases of the gait. Finally all these models were linked together in a finite state machine to form a hybrid automaton. The finite state machine specifies the switching conditions for each state.

One method for gait development is to design gaits that are tunable such that the gait practitioner can tune the gait to the pilot's comfort and rehabilitation needs. In this endeavour, a kinematic model is used to define the constraints of the double stance phase.

The gaits are then generated using a quartic polynomial spline using the node parameters from the kinematic analysis. This method empowers the gait practitioner by using tuning parameters that they understand from gait rehabilitation literature.

Another method for gait development is to use optimal gaits based on a hybrid automaton model of the system. The automaton implements three states or phases of the gait cycle including their associated dynamic models. Using this model, the gaits can be optimized for the torques necessary for these gaits under minimal input assumptions from the pilot in each of the phases. This optimization allows for a more detailed understanding of the system dynamics and an optimal gait given constraints of both the system and the gait cycle. Furthermore, the optimization method can be utilized as a gait generator that is tailored to each individual pilot, effectively reducing the workload of the gait practitioner.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank Professor Kazerooni for giving me the opportunity to work on projects that have a real impact on people's lives. Under his guidance, I have learned how to get to the root of a problem and quickly implement solutions.

Professors Poolla and Abbeel sat on both my qualification exam and my thesis committees. I am indebted to their graciousness and selflessness in doing so. Without their guidance, my thesis wouldn't be what it is.

Professor Mistree was a great inspiration to push me to grad school. Professor Attar inspired me to do research. Professor Fagan taught me how to be the kind of researcher and leader that I am today.

The Human Engineering and Robotics Laboratory is an incredible work environment with incredible people. I feel like it is a rare opportunity to get to work so closely with a multidisciplinary team in grad school. It was truly a joy to come to work everyday. Michael, Wayne, Mini, Yoon, Christina, Nick, and Shang-Li were all a huge part of what made HEL bearable. I was truly a pleasure working with my controls partners in crime, Grace, Octavio, and Hayley. It was an honor to mentor Hayley in her years as a Masters student. I also had the great opportunity to work with both Raghid and Andre to make the exo what it is today.

It would not have been possible to handle the stresses of grad school life without Elena, Kyle, and Jason. Elena got me outside to enjoy the wonderful world outside of campus (including swing dancing). Kyle was always somebody that was easy to relate to. All three were smarter than me and escaped with a Masters.

My long academic career has seen many hardships. I have been very fortunate to have my family to support me when times were hard. My mom always lent an ear to listen and a shoulder to cry on. My brother is someone I can always relate to and rely on for advise, even when I don't want to hear it. My dad was the man who taught me how to be a man. Without them, I wouldn't be who I am today.

I would like to thank my roommates Duncan, Abby, Marc, and Catie for giving me something to come home to that was not a thesis or more work.

My best friend, Jacob Stinnett, is the person I could go to for advise or help at any time day or night (because he too is a grad student). I sometimes regret not going to the same grad school as you, but I never regret asking you to be my roommate. Without your friendship from 2,122 miles away, I would have often broken down from the stresses of grad school and life. Danae was there when the thesis stresses were too much to handle, and I needed to break down. I owe my sanity over the past few months to her.

Last but not least, I am extremely blessed that I have had the many opportunities that I have had in my life. My achievements would not be possible without a little help from the big man. Thanks God.

Chapter 1

Introduction and Motivations

1.1 Why We Need Exoskeletons

There are 6.8 million people in the United States alone that have mobility disorders and depend on assistive devices to help with mobility, 1.7 million of whom are confined to wheelchairs or scooters [23]. Humans are designed to locomote through bipedal walking. Without the ability to locomote naturally, people will develop chronic injuries including joint pains and muscle imbalances [10]. These issues need to be corrected through gait analysis and therapy [50]. For those with neurological disorders, the brain may have lost the ability to control the body (cerebral palsy), and the patients may need therapy to help retrain the brain to overpower the body's reflexes. For those with a loss of neurological connection to their extremities (incomplete spinal cord injury) some function may be recovered through the use of therapy. For those with a complete loss of neurological connection to the lower extremities (paraplegia), some abilities may be regained through therapy, but the primary hope of human exoskeletons is to return these individuals to a better quality of life through more natural locomotion and avoid secondary injuries.

Therefore, depending on the condition, human exoskeletons have the ability to serve two distinct, yet equally important, goals for those with mobility disorders: therapeutic and at-home use. While the two objectives are different in theory, the realizations need not be drastically different. The major differences for at-home use in comparison to therapeutic use is that there needs to be extra safety considerations for out-of-clinic operation, and the cost must be affordable for an individual. Therapeutic use of human exoskeletons in hospitals is supervised, providing human safety checks. Additionally, hospitals and medical facilities can afford a higher price point for human exoskeletons because the products would be used in therapy for multiple patients.

At-Home Exoskeletons

In considering the potential benefits of at-home exoskeletons, this paper will focus on spinal cord injury (SCI) applications, particularly those resulting in paralysis, a common population

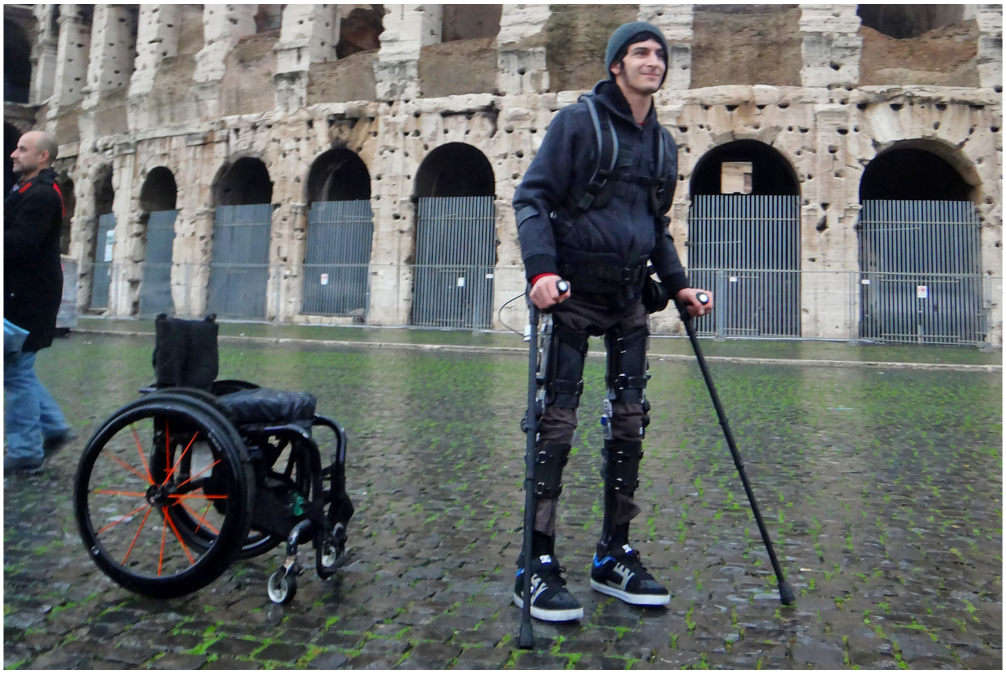


Figure 1.1: At-home exoskeletons are not confined to the pilot's house. Ideally, they can be used in any environment where the pilot would like to be upright.

considered for the use of human exoskeletons [42][14].

For those with complete spinal cord injury or paraplegia, the major risk to their health involves the secondary injuries that occur as a result of sitting or lying in a supine position for most of their lives. The human body was designed to stand and walk around. Without the bodyweight load on the legs, the lower extremity bones (including femur and tibia) can atrophy and result in bone mineral density levels, even lower than post-menopause osteoporosis [15]. The atrophy can be so extreme that something as simple as stubbing a toe could result in a broken foot. Keep in mind that these individuals typically cannot feel their lower extremities, so such an injury might not be noticed at the time of injury, potentially resulting in further complications. Studies have shown [25] that even with frequent exercise sessions, a paraplegic will not regain bone density once lost. There is hope that there would be less atrophy and bone loss with frequent daily use of an exoskeleton.

Similarly, without frequent motion and stimulation, human muscles and nerves begin to atrophy. These losses have implications for the cardiovascular health of the paraplegic individual. Paralysis can lead to a more sedentary lifestyle due to difficult mobility and lack of accessibility. The sedentary life leads to increased obesity and cardiovascular disorders in those with spinal cord injuries [29]. The human body will develop pressure sores if the body is loaded consistently in the same location. If a paraplegic does not take care to constantly shift his or her body weight, then their wheelchair-bound lifestyle could result in pressure sores. Furthermore, those with spinal cord injuries have a much higher likelihood of urinary tract infections and renal failures due to lack of bladder control and the need for catheters. Pressure sores and urinary tract infections are serious issues in that they can lead to septicaemia (bacterial infection of the blood) and potentially death [39]. The hope for at-home exoskeletons is to provide a more natural, less sedentary, and more productive lifestyle to those with mobility disorders like paraplegia in order to reduce these secondary injuries.

Last but not least, living life in a wheelchair has many social and psychological effects as well. In fact, suicide is one of the leading causes of death among those who have suffered a spinal cord injury [39]. Those with spinal cord injuries would benefit not only from the physiological effects of a more active and natural lifestyle but the psychological effects could be extremely beneficial as well.

Spinal cord injury patients are more likely than a healthy person to suffer from various maladies including those discussed above. This increased risk results in an increased death rate in some areas in particular. In Table 1.1, the cause of death for spinal cord injury patients is listed in terms of the standardized mortality ratio, that is the ratio of mortality relative to the expected mortality of the community as a whole given sex and age. Those with spinal cord injuries commit suicide at four times the normal rate and have blood infections at a staggering 172 times the normal rate. Better solutions for this population must be found.

Cause of Death	**SMR**
Septicaemia	172.3
Cancer	1.0
Diseases of the Nervous System	4.9
Diseases of the Pulmonary Circulation	26.2
Cerebrovascular Disease	2.3
Diseases of the Artery	2.2
Pneumonia and Influenza	32.5
Diseases of the Urinary System	22.8
Suicide	4.4

Table 1.1: Standardized Mortality Ratio for Spinal Cord Injury[39]

Therapeutic Exoskeletons

In considering the prospect of the benefits of therapeutic exoskeletons, this paper will focus on a case-study of cerebral palsy, a relatively new population considered for the use of human exoskeletons [7]. Many neurological disorders, such as stroke, osteoarthritis, Parkinson's disease, and multiple sclerosis, can be treated in part through gait therapy. It is the hope that exoskeleton's can be developed to the point that they can provide a better form of gait therapy in the future [12].

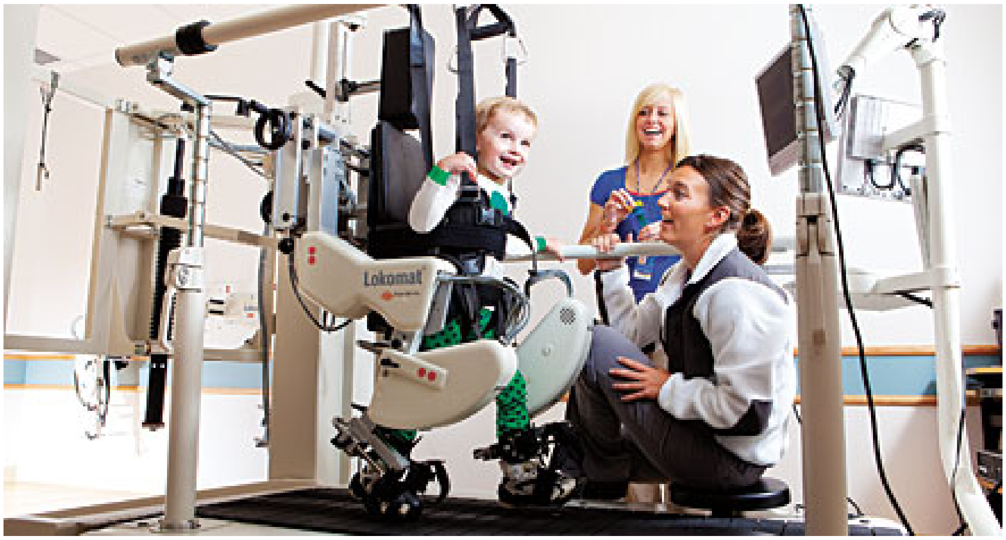


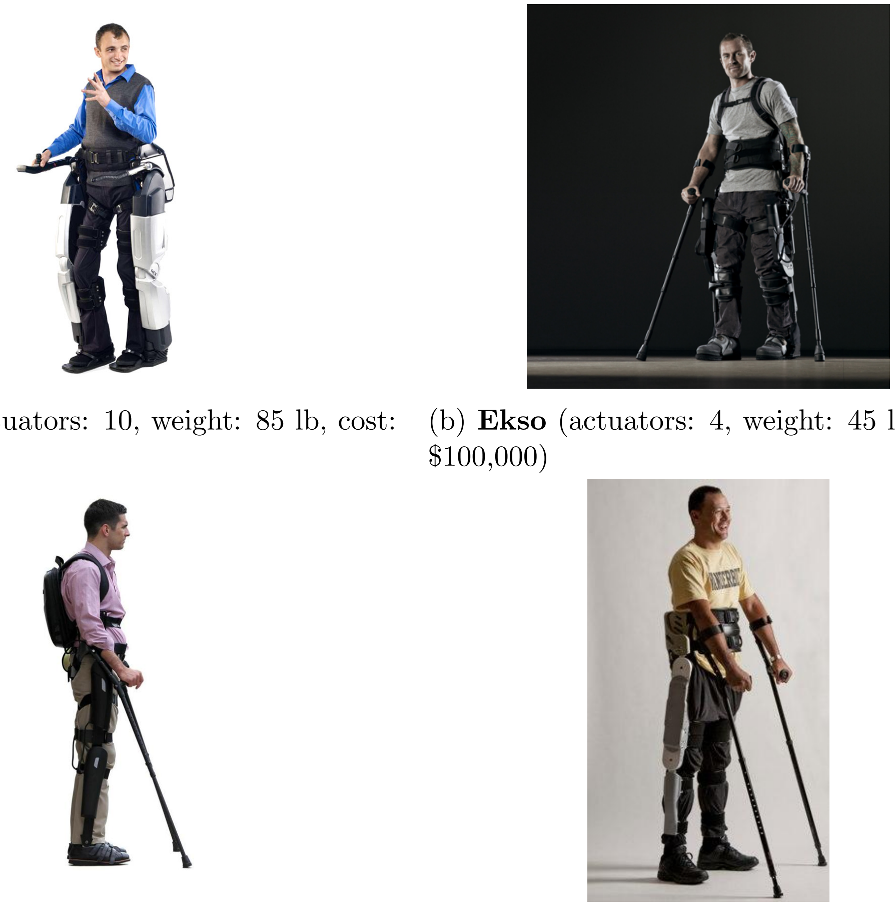
Figure 1.2: A child using the Lokomat, an in-clinic gait rehabilitation device [18].

Cerebral palsy is a neurological disorder that affects motor control, beginning in early childhood. Spastic cerebral palsy accounts for 70% of all cases of cerebral palsy[41], and is characterized by spasticity, or excessive muscle contractions due to a loss of motor neuron inhibition. The cause is a hypersensitive stretch reflex, most likely due to the inability of the central nervous system to overcome and inhibit these reflexes. These hypersensitive reflexes (hyperreflexia) cause hypertonia (abnormally increased muscle tone), and in a developing child, this can lead to further complications. In fact, in children with cerebral palsy, their gaits and range of motion tend to deteriorate as they develop due to the effects of hypertonia, muscle contractures (shortening of the muscles and tendons), and deformities on the still developing body [2].

Some of these effects can be alleviated through surgery, but there are risks. Often in the absence of continual intensive gait rehabilitation, the results of surgery are short-lived, potentially requiring additional surgery. There are some medicines that can be used to help suppress the stretch reflex including Botulinum toxin type A (commonly known as Botox) [24]. However, these affects can wear off and need reapplication.

The hope for therapeutic exoskeleton technology for cerebral palsy and similar neurological disorders is that exoskeletons will provide better gait rehabilitation so that the patients can learn to walk naturally without the device. For spastic cerebral palsy, exoskeletons could provide gait therapy much more quickly to try and prevent the deformities from deteriorating their gait before their condition worsens. These devices can't cure neurological disorders like cerebral palsy, but they can help diminish the effect on the pilot's gait and consequently improve their lifestyle.

1.2 Prior Art



(a) **Rex** (actuators: 10, weight: 85 lb, cost: $150,000)

(b) **Ekso** (actuators: 4, weight: 45 lb, cost: $100,000)

(c) **ReWalk** (actuators: 4, weight: 45 lb, cost: $70,000)

(d) **Indego** (actuators: 4, weight: 26 lb, cost: $80,000)

Figure 1.3: Exoskeletons on the market.

Exoskeleton research has exploded recently with many exciting new platforms hitting the market and acquiring Federal Drug Administration (FDA) and European Conformity (CE) approval for marketing and sale in the US and Europe, respectively. Some of these are displayed in Figure 1.3. Among the first on the market was Rex, the largest of the medical exoskeletons at 85 pounds and costing approximately $150,000. Rex has the most actuated degrees of freedom at ten and is able to be used without crutches.

Ekso was the first commercially available exoskeleton. Originally developed in the Human Engineering and Robotics Laboratory at the University of California, Berkeley, the Ekso GT is widely used in gait rehabilitation centers and sports four total actuators. Ekso's device is about the same price as Rex, but is much lighter at around 60 pounds.

ReWalk, out of Israel, was the first powered exoskeleton to receive FDA approval and is spreading to gait rehab centers across the United States and Europe. ReWalk is similar to Ekso in both actuation and weight, but at a cost of only $70,000 per unit. Indego is another four actuator exoskeleton out of Vanderbilt's Center for Intelligent Mechatronics which was licensed to Parker Hannifin for production and marketing. Indego received FDA approval in 2016 for sale to rehabilitation clinics. Indego is lighter than the competition at 26 pounds and at a competitive price of approximately $80,000 per unit.

The trend in the market seems to favor lighter, more mobile, and cheaper exoskeletons so that clinics and individual's can afford them. After all, a medical device is only as valuable as the number of people that can access it; and bulkier, slow-moving, and expensive devices are not as accessible.

Minimally Actuated Exoskeletons

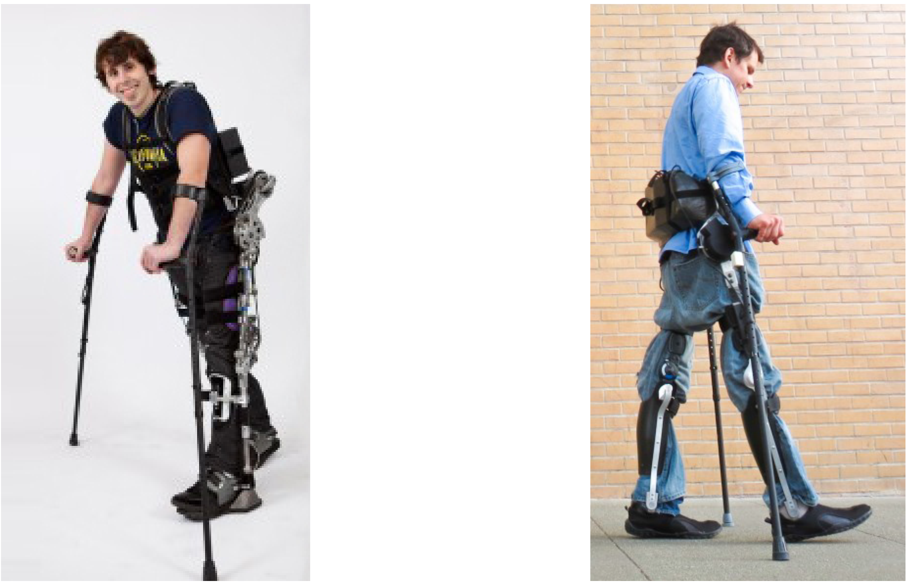
In the pursuit of a lightweight and affordable exoskeleton, the Human Engineering and Robotics laboratory at the University of California at Berkeley has been developing technologies that help reduce the cost by reducing the number of actuators of the system [32]. The research has focused on using exoskeletons with hip actuators only. These systems couple the dynamics of the knee to the hip in order to provide locomotion. The two methods used to provide locomotion to a hips-only actuated exoskeleton are a mechanical coupling of the knee joint to the hip and a dynamic coupling of the leg shank dynamics to the hip [32, 27]. These two methods are shown by the Austin and Ryan exoskeletons in Figure 1.4, respectively.

The hips-only actuation technology has been refined and a more modern version of these exoskeletons is described in detail in Chapter 2. The research in this thesis will focus on these exoskeleton platforms for developing and implementing gaits.

1.3 Biomechanics of Walking

There are a vast number of studies and terminology on the biomechanics of human bipedal locomotion. Before discussing the implementation of gaits on human exoskeletons, an overview of some of the major results of these biomechanical studies and definitions of some terminology is needed.

The body can be divided into three anatomical planes, illustrated in Figure 1.5. A transverse plane is parallel to the ground and separates the body into superior (upper) and inferior (lower) sections. The coronal (or frontal) plane is perpendicular to the ground plane and divides the human body into the anterior (front) and posterior (back) sections. The sagittal plane is also perpendicular to the ground plane and splits the body down the middle into left and right halves. Since most exoskeletons provide actuation primarily in the sagittal (or medial) plane (beside Rex) and the primary motion of human walking is also in the sagittal plane, this paper will focus on motion in this specific plane.



(a) The Austin exoskeleton used a mechanical coupling of the knee joint to the hip.

(b) The Ryan exoskeleton used a dynamic coupling of the knee joint to the hip through the pendular dynamics of the knee [27, 32].

Figure 1.4: Minimally actuated exoskeletons out of the Human Engineering and Robotics Laboratory at UC Berkeley.

Human Gait Cycle

Figure 1.6 describes the gait cycle and the terms associated with it. The gait can be broken down into single stance and double stance, as well as a stance phase and swing phase.

Firstly, a human gait is cyclical by definition; a gait is repeated in order to locomote. A stride is defined as the motion and subsequent distance accomplished during one full gait cycle period, i.e. the action that is continuously repeated. By this definition, it follows that a step is exactly half of a stride. Toe-off is the point at which the toe is lifted off the ground before the swing phase (or single stance) of the gait cycle [51].

Single Stance

The single stance phase (also known as single support) is when there is only one leg (the stance leg) in contact with the ground, also known as swing phase. The leg that is lifted off the ground during this phase is called the swing leg. Single stance is considered to be concluded on heel strike, when the swing heel hits the ground in front of the stance leg (as the other leg is lifted off the ground and “swinging” past the stance leg). Toe clearance is the distance from the ground to the toe during the critical time when the swing leg passes by the stance leg.

Toe clearance can be used as a metric of the robustness of a gait to variations, both

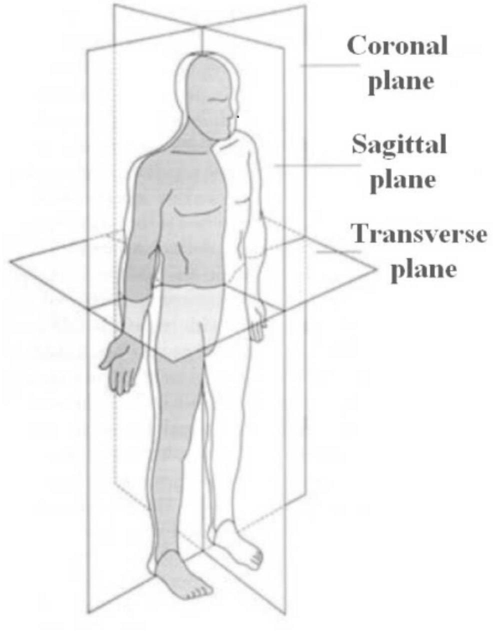


Figure 1.5: The anatomical planes of the human body [34].

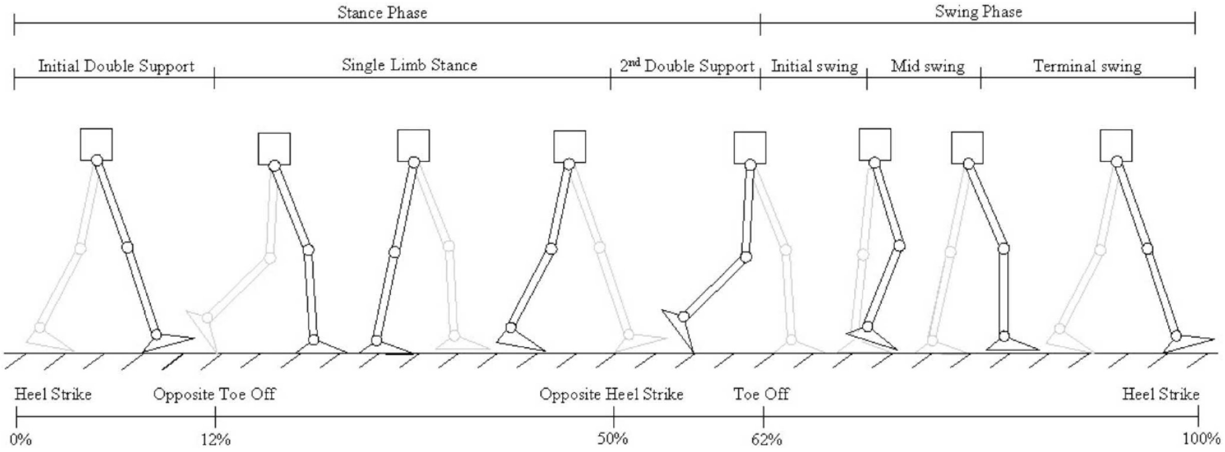


Figure 1.6: The human bipedal gait cycle [12].

in the ground conditions and to the gait itself. In fact, foot drop is a gait abnormality where the forefoot drops during the gait cycle due to weakness or lack of control. Foot drop decreases toe clearance and consequently increases the risk of falling. This gait abnormality is associated with many neurological disorders including stroke [20], hemiplegia [54], and Charcot-Marie-Tooth [13]. In fact there has been some research on assistive devices solely for the purpose of improving the robustness of the gait by decreasing foot drop and consequently increasing toe clearance [6]. If the ground is slightly higher in one spot or the gait trajectories

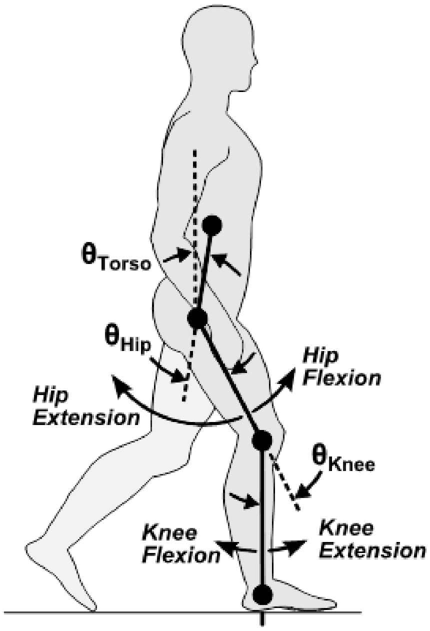


Figure 1.7: Definition of joint angles [32].

are not tracked perfectly, then the person should not fall with an assistive device. However, this robustness comes at the cost of added effort to lift the leg and consequently the swing toe higher.

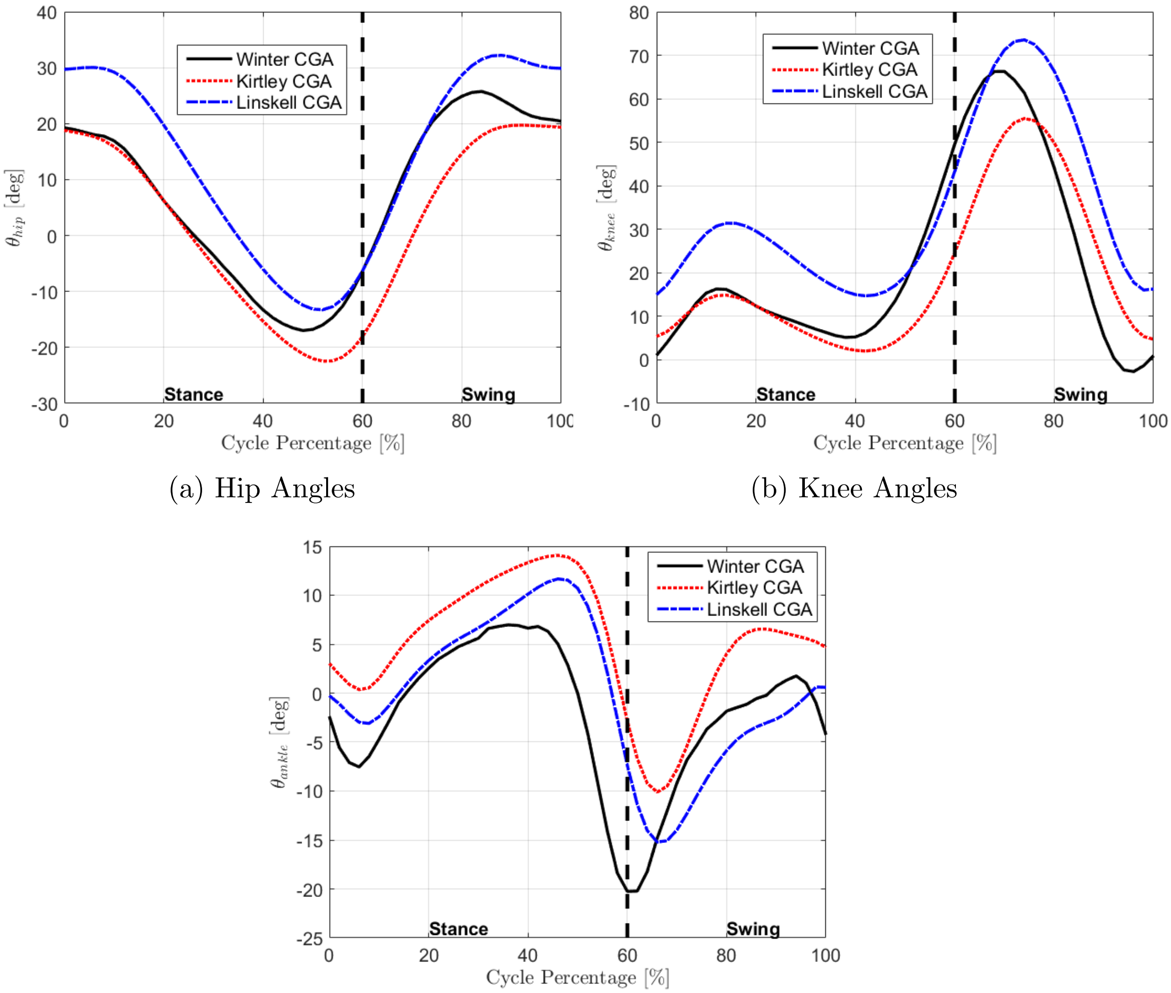
Double Stance

The double stance phase (also known as double support) is when both feet are on the ground. “Front” and “back” are used to describe the legs’ orientation in double stance. Double stance is responsible for transferring the weight from the back (previously stance) leg onto the front (next stance) leg. Once the weight is completely transferred from the back leg to the front, the back leg is free to be lifted for the swing phase to begin. Furthermore, double stance provides a mechanism to allow the momentum from the previous step to be continued forward into the next. With a proper double stance, the gait will be smooth and continuous, minimizing the energy needed and shock on the body [53]. Without it, the gait will appear like a rough march.

Joint Angles

Before discussing clinical gait analysis, some definitions need to be established. The definitions for the torso, hip, and knee angles are shown in Figure 1.7. To define joint angles, first linkages or lines are drawn connecting consecutive pivot points on the human body. The gait is then characterized by relative joint angles: angles that are relative to the next linkage in the chain instead of an absolute reference like a vertical line. The exception is the torso angle, which is relative to absolute vertical. These definitions are standard across clinical gait analysis research [51] [49].

Clinical Gait Data



(a) Hip Angles

(b) Knee Angles

(c) Ankle Angles

Figure 1.8: Clinical gait analysis angle data. Data from [52]

.

Clinical gait analysis is a key tool in both understanding the natural locomotion of humans and in the analysis and treatment of those with mobility disorders. By studying the gait of a patient in comparison to the normative, a gait specialist can identify weakness and suggest treatments [16]. A substantial effort was undertaken in past decades to document human date both for understanding and comparison [31] [51] [44] [43].

The joint angles of the lower extremities during the gait phase are shown in Figure 1.8. These angles and percentages roughly line up with the illustration in Figure 1.6. These plots are excellent for understanding what each joint is doing during each phase of the gait cycle. When developing rehabilitation gaits, these plots are the gold standard [16]. In clinical evaluation of a mobility assistance device, part of the performance is based on a comparison to some type of gait analysis similar to Figure 1.8.

1.4 Human Gait Modeling

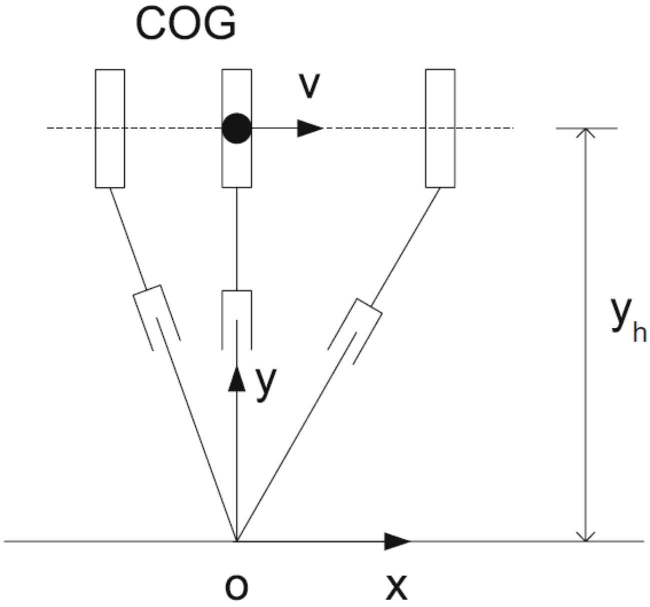


Figure 1.9: The simple two dimensional inverted pendulum model of the human gait. Adapted from [55].

There are many methods used to model and simulate human gaits. Some of the early research in this area focused on simplified modeling [22] [30]. Some of these drastically simplified models yielded surprisingly reasonable results such as the inverted pendulum model, which is governed by the one degree of freedom dynamics of the center of mass:

$$\ddot{x} - \frac{g}{y_h}x = 0 \tag{1.1}$$

where the definitions for these terms is shown in Figure 1.9. The x-direction is the direction of the walking motion, the y-direction is perpendicular to the ground plane, COG denotes the center of gravity of the biped model, v is the velocity of the mass, and y_h is the height of the COG. The constant gravitational acceleration near the Earth's surface is denoted by g. The premise of the inverted pendulum model is that the center of gravity stays relatively constant in the y-direction during a normal gait. Also, the stance leg is more or less what propels the body forward during walking. Therefore, one can model the entire system as simply a single inverted pendulum (stance leg) that slides up and down to keep the center of gravity constant in the y-direction. There has been some success employing control algorithms on robots based on this model [22].

While this model is surprisingly powerful for a simple biped model, a deeper understanding of the human gait cycle can be gained by studying passive dynamic walking. Passive dynamic walking is the idea that walking can require exactly zero input torque with a small amount of assistance from gravity, i.e. walking down a very slight slope. This was first proposed by McGreer [26] through the toy example of a compass biped, shown in Figure 1.10. The compass biped is composed of three masses: the hip mass m_h and two leg masses m. The leg masses are specified by their distance from the hip mass (b). The degree of slope that the biped walks down is specified by γ. The biped walks down the slope with each heel strike giving energy to the system and resetting to the initial condition. Passive dynamic walkers are difficult to find parameters for such that they will exhibit the passive dynamic behavior (motion without input), but McGreer showed this was possible [26]. The study of these walkers has helped shine light on some of the dynamic relationships of the human gait cycle that make bipedal locomotion such an energy efficient method of transport.

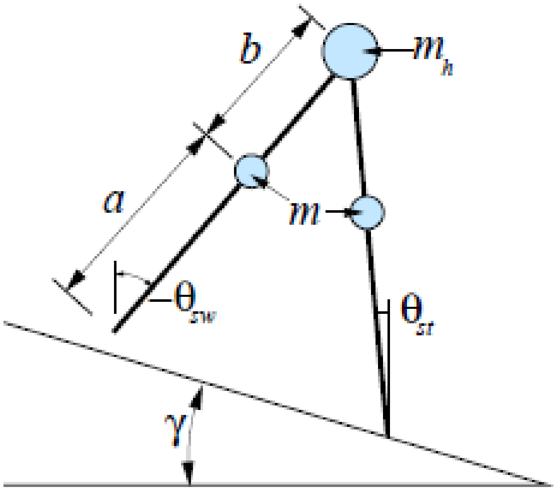


Figure 1.10: The first passive dynamic walking model: compass biped.

On the other end of the spectrum, there are much more complex models that provide more fidelity at the expense of increased difficulty in the analysis. A popular method to use for these more complex models is optimization or optimized control [55]. These models can be very complex and, consequently, very difficult to optimize. The state space is very large, most solvers can get stuck in local minima, and the right initial guess could mean the difference between an optimal solution and an infeasible one. There are two ways to optimize a physical model: forward dynamics and inverse dynamics. While forward dynamics are probably the more popular option, inverse dynamics have some advantages. Saidouni et. al used inverse dynamics along with a quartic spline constrained state space [38]. That is, the joint angles were assumed to track quartic splines with continuity constraints at gait phase transitions. This approach allows for optimization without drastically increasing the optimization variables through discretizing the state space, i.e. the cost function is an integral over a continuous function. The control inputs to track these splined trajectories is computed using inverse dynamics and the integral of the 2-norm of these control inputs over time is used as the cost function. Bessonnet applied the method of inverse dynamics using quartic splines on the seven link biped (Figure 1.11) for use on a large humanoid robot[4]. The results showed the potential benefit of polynomial-constrained gait generation.

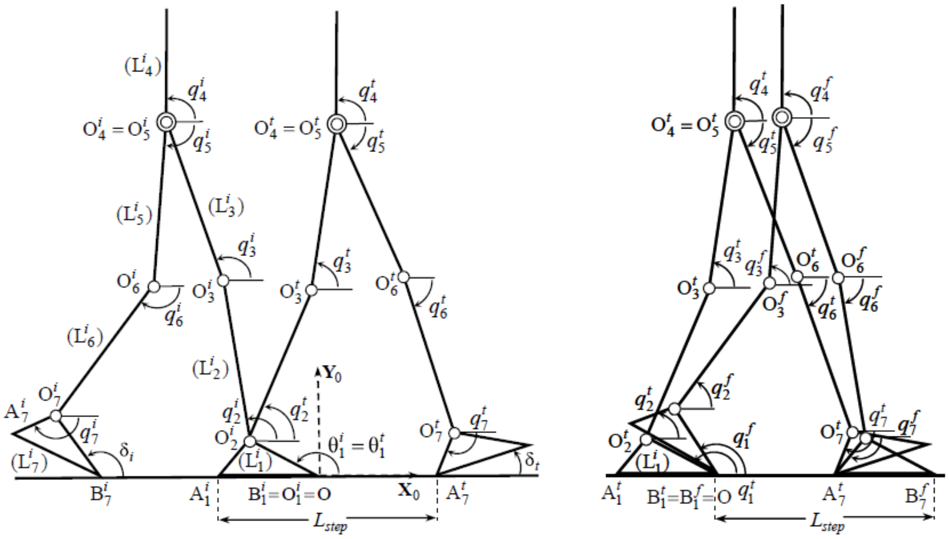


Figure 1.11: The biped model used for Bessonnet's optimization [4].

There are many ways to measure the performance and robustness of a gait. Performance metrics include: the total mechanical energy of the system, the total input energy to the system, and the sum of squares of the input torques. Robustness metrics include: toe clearance, friction margins on the feet, and zero moment point margins. From these metrics, it's clear that there is no one perfect gait for humans or even individuals. As such, it is difficult to correct the gait of a person to optimality. This difficulty also manifests itself in the selection of gaits for robotic systems, in particular human exoskeletons. The problem of finding the correct gait for each individual pilot introduces the possibility of two approaches: allowing the gait therapist to tune the exoskeleton to the pilots needs, or finding an optimal gait for the exoskeleton by some chosen metrics.

1.5 Thesis Contributions

The objective of this research is to develop gait trajectories that are smooth, continuous, and easily adapted to each pilot's anatomy and therapy needs. This thesis will focus on gait implementations on exoskeletons with only hip actuation, with a special emphasis on the Phoenix exoskeleton described in Chapter 2. To this end, there are two proposed solutions: easily-tunable gaits and gaits that are optimized per pilot.

Tunable Gaits

This thesis introduces the method of tunable exoskeleton gaits to exoskeletons with hip actuation only, or gaits that are adjusted by a gait therapist to suit an individual pilot's comfort and therapy needs. Tunable gaits have the possibility to work in both the clinical and at-home exoskeleton paradigms. If the pilot is only using the device in a clinic, then the device can be quickly adjusted to gait settings that are appropriate for their individual therapy. If the pilot is using an at-home exoskeleton, then the pilot would still need an initial fitting and tuning with a gait therapist. However, at this point, the pilot could use the exoskeleton until parameters need to be adjusted. A gait therapist would still be needed to either tune or advise the tuning of the device for the pilot.

Optimal Gaits

This thesis also introduces the method of optimal exoskeleton gaits to exoskeletons with hip actuation only, or a gait that is optimized to each pilot using his or her biological dimensions and properties. Through a tuning of a few constraints such as joint limits or toe clearance, optimal gaits can also be tuned to accommodate a pilot's individual needs. The optimal gait approach is useful, not only in providing gaits to the therapist, but also in providing analysis tools for the dynamics and control of the system. A development of the problem and barriers to implementation are presented.

Smooth and Continuous Gaits

All gaits for a human exoskeleton should be smooth and continuous to the point that the pilot will not encounter a jerkiness in its operation. Guaranteeing smoothness can be difficult in the case of walking because the pilot should always have the ability to stop walking the device if he or she wants to. This thesis introduces a quartic and cubic spline method that is novel to human exoskeletons. Using this method, acceleration can always be guaranteed to be continuous.

Chapter 2

The Phoenix Exoskeleton

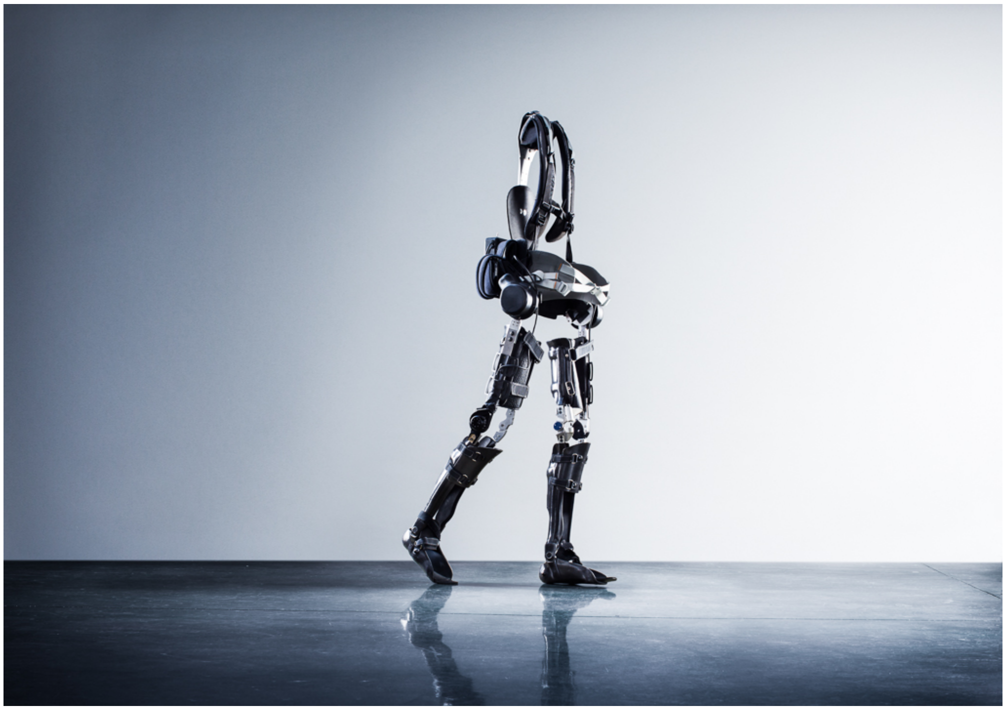


Figure 2.1: An alpha protoype of the Phoenix Exoskeleton.

The Phoenix is an exoskeleton that has its roots in the University of California Berkeley Human Engineering and Robotics Laboratory, much of the technology being initially researched and developed on campus. This exoskeleton platform will be the focus for system modeling and the implementation of algorithms developed in this paper. The Phoenix is minimally actuated, meaning that it contains the minimum number of actuators needed for its tasks (walking, sitting, and standing). The design objective of the Phoenix exoskeleton was to reduce the cost and weight as much as possible so that this device could be affordable and accessible to the average person for at-home use. In reducing the number of actuators to the minimum required, weight and cost are saved [46].

The Phoenix weighs only 26 pounds and can be easily worn while seated in a standard wheelchair. The device is commanded using a crutch-based user interface with a simple two-button pilot interface. The buttons are used in a simple "forward" and "backward" manner. The forward button will progress the pilot through states associated with taking steps, and the backward button will progress the pilot through states that ultimately end in a seated position.

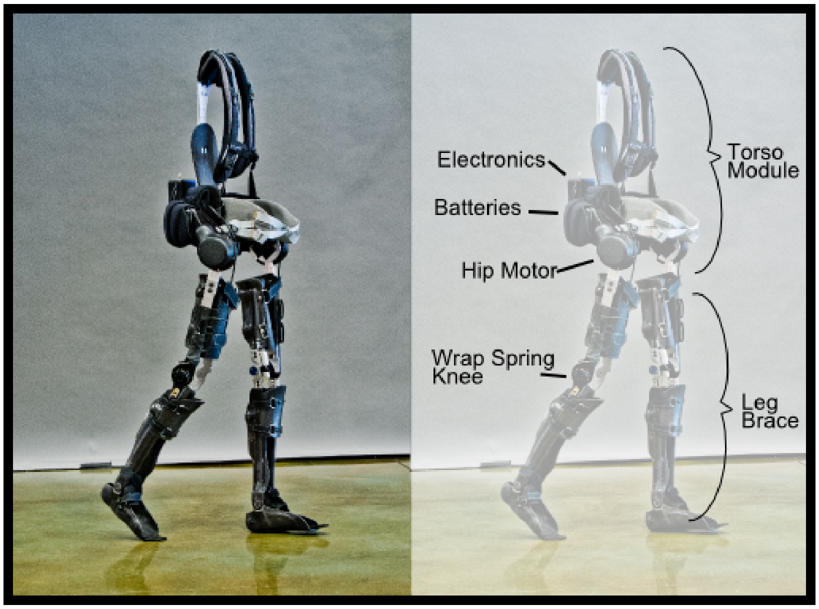
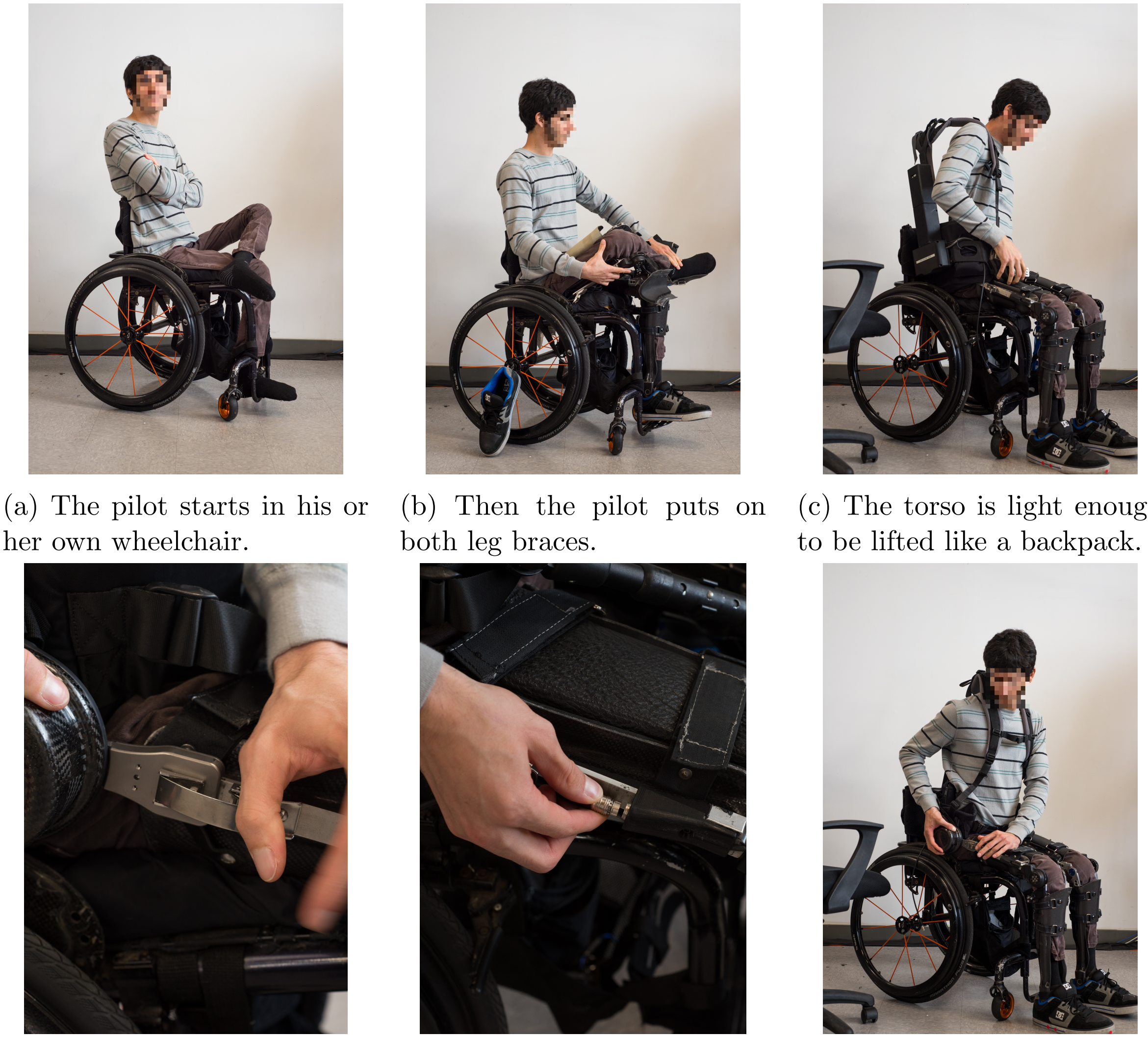


Figure 2.2: An overview of the Phoenix exoskeleton. From [27].

The Phoenix was designed to be modular and customizable to fit a variety of user needs. The user has the option of utilizing either custom-made orthotic leg braces or a set of general adjustable braces. The modular design allows for a hip-only solution for those user's who only need assistance at the hips, and a knee-only solution for those user's who only need knee assistance. The modular design also allows for simple donning and doffing. The pilot simply puts on the leg braces one at a time while seated in his or her wheelchair or chair. Then, the torso assembly can be simply transferred behind them and attached using the latch mechanisms.

2.1 Torso

The Phoenix was designed to fit the vast majority of the population of both men and women in the United States in both weight and height. In the design of an orthotic device, the actuators and joints need to align with the joint of the user as accurately as possible. To



(a) The pilot starts in his or her own wheelchair.

(b) Then the pilot puts on both leg braces.

(c) The torso is light enough to be lifted like a backpack.

(d) The hip and leg braces are latched into place.

(e) The knee cable is connected to the knee.

(f) The pilot can don the device without assistance.

Figure 2.3: Due to the modular design, donning and doffing is simple with the Phoenix. The pilot is completely independent.

accomplish this objective, the Phoenix has an adjustable torso piece. The torso back height can be adjusted from 20.75 inches to 31.25 inches. The hip width can be adjusted from 12.8 inches to 16.8 inches. The fore and aft position of the hip motors can be adjusted through an air pump on the back air cushion. It's important that the hip actuator's center of rotation is the same as the pilot's hip joint so that strange loading and nonlinear output torques are not observed. The back cushion is also used to distribute the physical interface forces, which can be particularly damaging to pilots who have severe muscle atrophy.

Finally, the torso has backpack-style shoulder straps that can be tightened and loosened

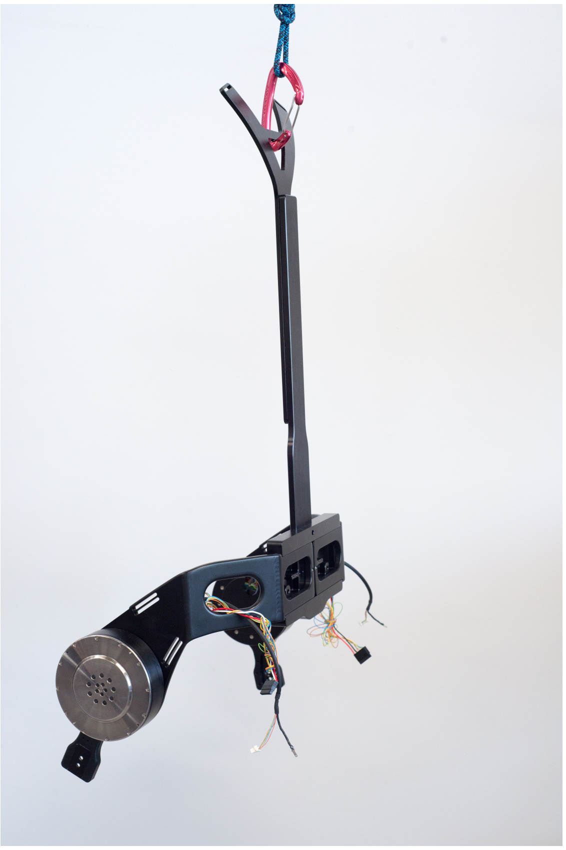


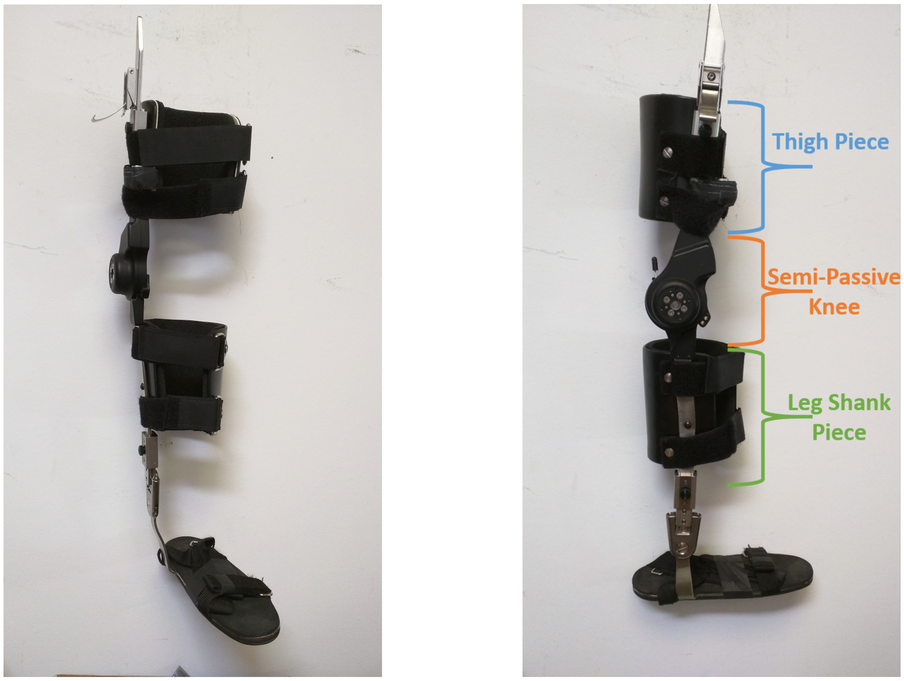
Figure 2.4: A stripped down version of the Phoenix torso piece showing the adjustability and minimal design.

to comfortably couple the exoskeleton back to the pilot's back. The hip piece also has a strap that is used to press the pilot into the back cushion and adjust the coupling of the pilot and device hips.

2.2 Leg Braces

The leg assembly can be either a custom-made knee-ankle-foot orthosis (KAFO) fitted with the semi-passive knee mechanism or a set of universal leg braces that similarly fit from the 5th percentile to the 95th percentile. The custom-made leg braces are more expensive but leverage the technology advancements in orthotics over the past 50 years. The custom-made braces are more comfortable and fit the pilot perfectly, but gait therapy clinics might not make the larger capital investment in custom braces. Therefore, the universal leg braces are the technology intended for most rehabilitation purposes while the custom-made braces are better suited for at-home use. Both options are designed to be easily donned and doffed.

The universal leg braces are designed in two sections: the leg shank (lower leg) and the



(a) Front View (b) Side View

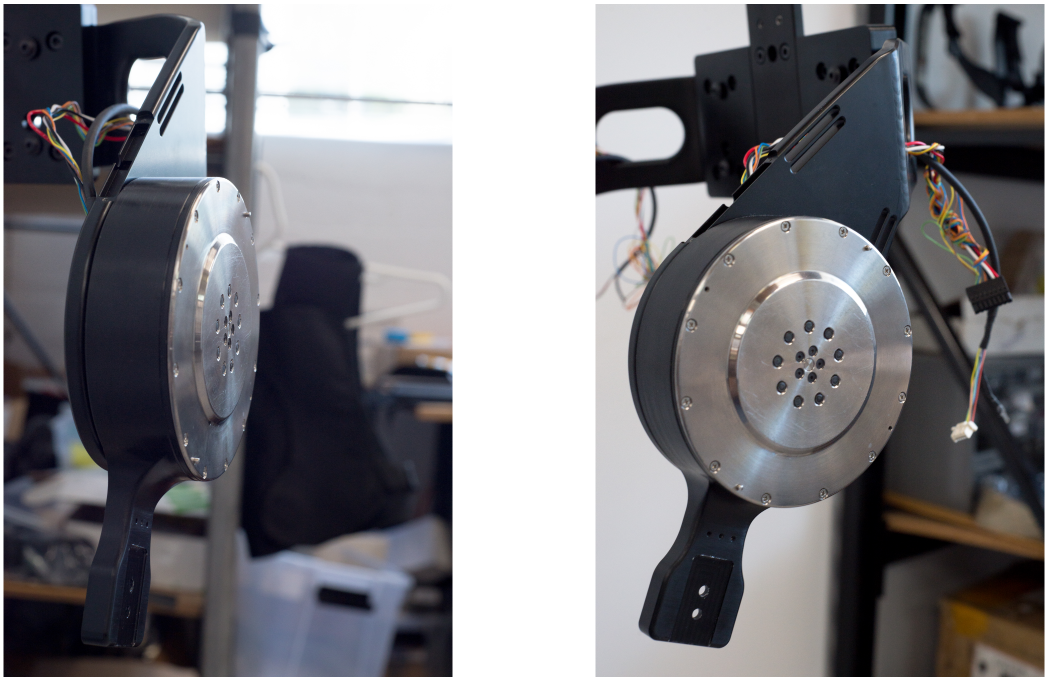
Figure 2.5: Phoenix exoskeleton adjustable leg braces.

thigh piece (upper leg). The two meet at the semi-passive knee mechanism, and the leg brace connects to the torso using the latch mechanism shown in Figure 2.3.

Similar to the torso, the universal leg braces are easily adjusted to align the device knee joint with the pilot's knee joint. To allow for such a wide range of adjustability, the leg braces come in three sizes: small, medium, and large. Having a range of sizes allows for the leg shank section to be adjusted for tibia lengths from 13.6 to 21.6 inches. The thigh piece is also adjustable to femur lengths from 12.7 to 20.0 inches to align the hip actuator with the pilot's hip joint.

2.3 Hip Actuators

The hip actuator for the Phoenix (shown in Figure 2.6)consists of a three-phase DC brushless motor coupled with a 100:1 harmonic drive transmission. The output link of the actuator is capable of producing 41.9 N·m of continuous torque, 152 N·m peak torque, and 90 watts of power. These numbers are adequate to support the torque requirements according to the clinical gait analysis data [21]. The actuators also have hard stops that prevent the legs from going out of the healthy range of motion. The hard stops are at roughly 40 degrees of extension during walking and 130 degrees of flexion while in the bent forward position before standing up.



(a) Motor front view.

(b) Motor side view.

Figure 2.6: The hip motor was designed to provide the necessary torque while also being thin enough to fit in the pilot's wheelchair.

The design of the hip actuator was focused on a very tight package with a high power density. The entire actuator thickness is 1.8 inches, which is thin enough to allow for the exoskeleton to be worn while seated in the pilot's wheelchair [21].

2.4 Semi-Passive Knees

In the pursuit of an affordable, lightweight exoskeleton, the knees of the Phoenix are not actuated. Instead, the knees have a semi-passive mechanism that allows the knee to "lock" in the sense that knee flexion is no longer allowed. When the knee is "unlocked", the knee joint is free to flex. Regardless of whether the knee is in a locked or unlocked state, the knee is always free to extend [47] to the extension hard stop limit. When the mechanism is against the hard stop limit and the mechanism is in the locked state, the joint cannot rotate in either direction.

The locking and unlocking action of the knees is allowed through the use of a wrap spring clutch. The clutch consists of a spring wrapped around a central hub connected to the output. When the clutch is unlocked, the spring is not in direct contact with the central hub allowing for a near frictionless rotation. However, when the clutch is locked, the spring is in contact with the hub that it's wrapped around. If the hub rotates in the direction of the

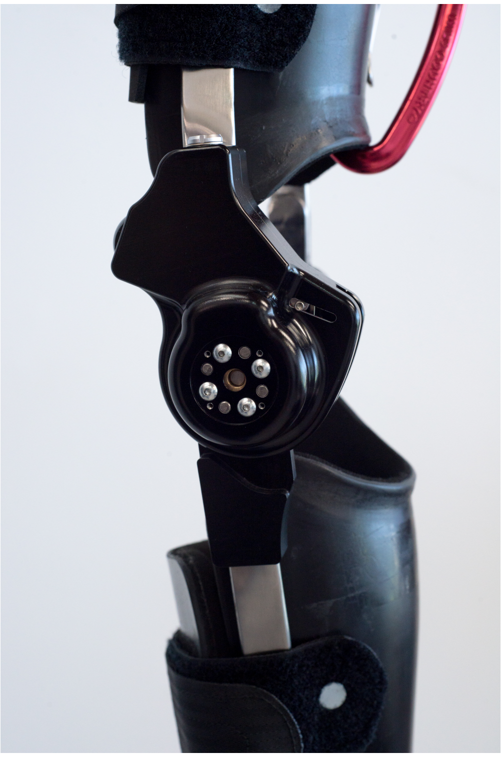


Figure 2.7: The Phoenix exoskeleton's lightweight, semi-passive knee mechanism.

spring's wrapping (flexion in this case), then the spring wraps tighter and prevents rotation. If the hub rotates in the opposite direction of the spring's wrapping (extension in this case), then the spring will slip and allow for rotation of the joint.

Whether the system is locked or unlocked is simply a function of the position of the spring tang. The spring tang extends from the wrap spring and through a small movement, will lock and unlock the clutch. This small movement is achieved through the use of a stepper motor coupled with a stepper motor chopper driver. The stepper motor allows for simple no-feedback control, similar to a solenoid but more power dense. To power a stepper motor, one simply needs to cycle current between the phases. The stepper motor chopper driver takes a simple motion command and manages the phase shifting to offload processing from the main micro-controllers. The chopper driver powers the stepper similar to a pulse width modulation (PWM) method by very quickly pulsing high and low voltages to vary the effective current. With the stepper motor and stepper motor chopper driver, lock and unlock commands can be developed through a simple driver abstraction code layer.

The chopper driver allows for go-to commands by programming a velocity profile. The starting velocity, maximum acceleration, maximum deceleration, maximum speed, and ending velocity are all programmed to the driver. The chopper driver then cycles the phases of the stepper to match the profile, coming to a stop at the commanded position. The current during the motion and holding are also set. Stepper motor operation is fundamentally dif-

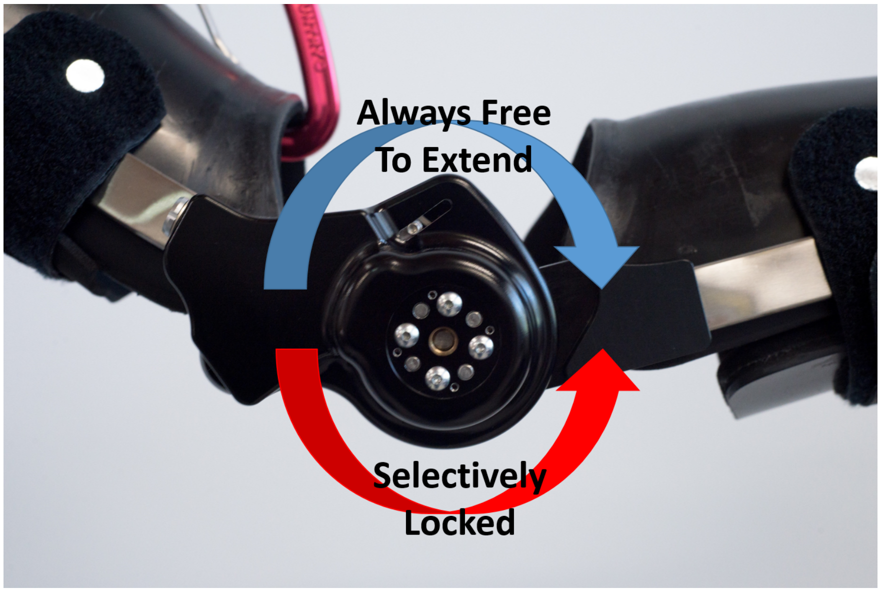


Figure 2.8: The knee mechanism is always free in extension. The flexion direction is capable of being locked and unlocked [47]

ferent than a regular DC motor operation in that for a stepper motor the supplied current controls the torque and the phase switching controls the velocity.

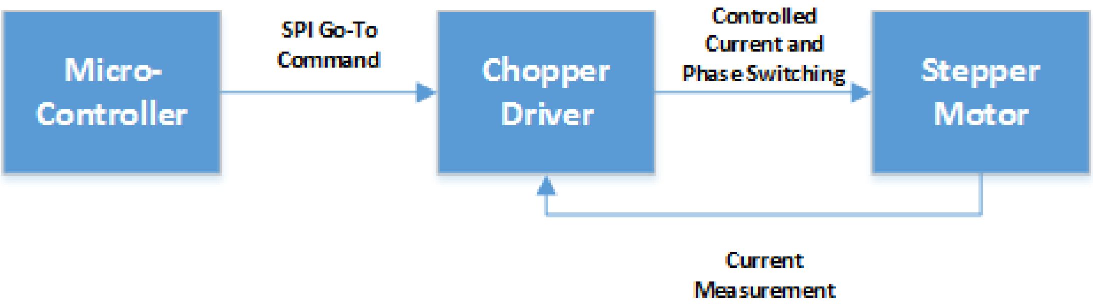


Figure 2.9: The stepper chopper driver controls the phase switching and current control.

2.5 Electronics

The main power board of the device was designed to have a low profile and efficient operation. The battery voltage of the device is constrained to 58 volts to provide for safe operation in a clinical setting [27]. To protect the device from its environment, it was tested under electrostatic discharge tests and emitted radiation tests. All internal communication buses are serial peripheral interface (SPI) due to its speed and reliability.

Micro-Controllers

The micro-controllers used were the mbed LPC1768, which is based on the ARM Cortex M3 microprocessor. The mbed platform was chosen due to its use of the ARM processor, widely regarded as the industry standard in embedded applications. Also, the mbed development platform allows for a very simple interface to the registers and Direct Memory Access (DMA). The mbed platform allows for very fast development without the need for directly accessing pins, writing drivers for the DMA, or writing to registers. Furthermore, the mbed platform is completely open source and provides support for complete integration of the design into a printed circuit board (PCB) design for production. The PCB for production has the NXP LPC1768 processors embedded in the PCB.

Amplifiers

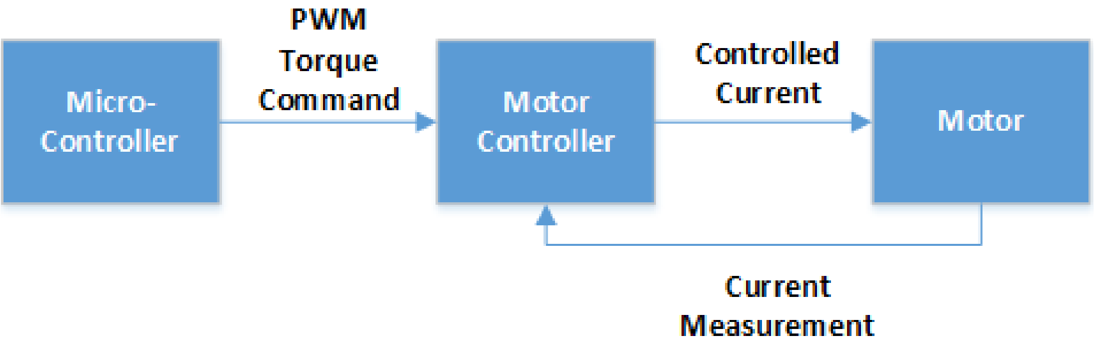


Figure 2.10: The motor controller removes the dynamic model dependence on the motor's electrical dynamics.

The Phoenix uses a motor controller for abstracting the system control design process dependence on the control of the motors (shown in Figure 2.10), meaning that the control algorithms don't have to consider the motor design at all in the models of the exoskeleton system. Essentially, the motor controller controls the current to the motor given a reference command from the micro-controller. The motor controller is in a direct feedback loop with the motor and uses current measurements to control the motor current for back electromagnetic force (back EMF), coil inductance, and other influences. This physical control architecture allows the micro-controllers to run simpler algorithms that only consider the motors as pure input torque devices without having to worry about the electrical characteristics of the motor.

Batteries

The battery pack for the exoskeleton had to be selected with the safety of the pilot in mind. With the invention of lithium ion battery technology, reusable batteries have become much more compact, lightweight, and quick to charge. However, these batteries run the risk of overheating, catching fire, and developing a cell imbalance. These problems are addressed through the use of a battery management system (BMS). A BMS and battery combination is referred to as a “smart” battery. The smart battery monitors the temperature, cell balance, state of charge, state of health, etc. of the battery and controls the charging and discharging in a safe manner. The BMS allows for the use of this new litium ion battery technology without introducing new safety hazards to the pilot and those around the device.

Sensors

For safe exoskeleton control, reliable and accurate sensors are required. For an exoskeleton to be financially accessible, the design must consider inexpensive sensors and using the fewest sensors possible. The design of the Phoenix leverages recent advancements in sensor technology which allow for inexpensive sensors with very high accuracy.

Encoders

There are four encoders on the device: one at each hip and knee. The encoders are a 14-bit absolute, magnetic encoder with parity and error checking (Austria Microsystems AS5048A). These encoders allow for extreme resolution (45 discrete measurements per degree of rotation) coupled with communication error handling. The absolute encoder does not have the computational downsides of a quadrature encoder in that an interrupt pin and counter are not needed for operation. Using an absolute encoder offloads computation from the micro-controller freeing up computation time for more important tasks.

Inertial Measurement Unit

An inertial measurement unit (IMU) is placed on the main control board for an absolute reference (Invensense MPU9250). The IMU has a built in digital motion processor (DMP) that combines the internal accelerometer, gyroscope, and compass measurements into absolute Euler angles. The DMP allows for offloading the computationally expensive sensor fusion calculation to the sensor itself. The IMU measures the torso angle and provides an absolute reference to the rest of the system since the encoders are all measuring relative angles.

2.6 Pilot Interface

The device is controlled through the use of an intuitive two button user interface. This interface establishes a “forward” and “backward” button that allows the pilot to navigate the

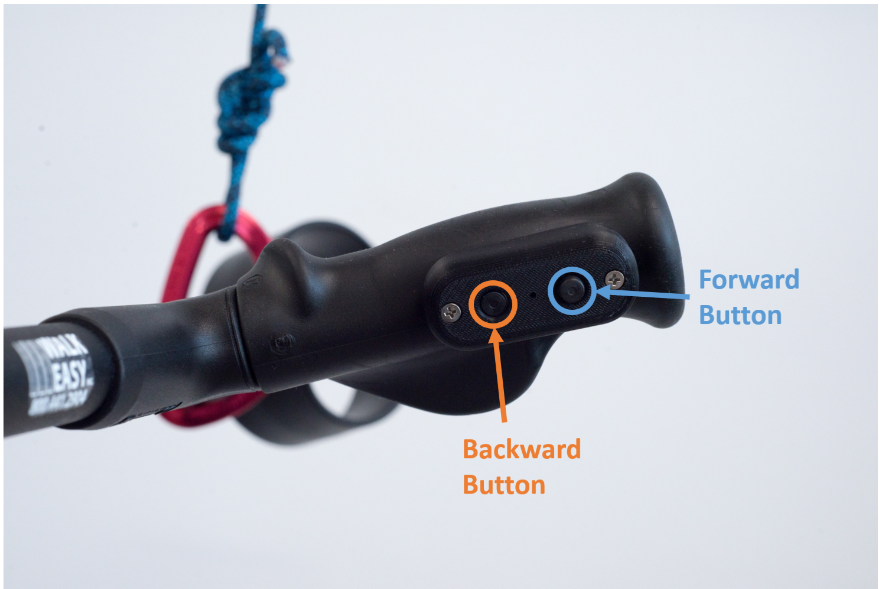


Figure 2.11: The user interface for the Phoenix exoskeleton is a simple forward and backward button configuration.

device's internal finite state machine (FSM) that controls the action states of the exoskeleton. The simple interface allows for very quick training of the pilots and high reliability. The interface is wirelessly coupled to the main control board through the use of two Bluetooth modules, one on the crutch and one on the main control board. The modules are both Bluetooth low-energy (BLE or Bluetooth Smart) and the crutch module under normal use can last for over a year on the energy of a standard coin cell battery.

The pilot interface was designed to be modular such that it can be adapted to whatever stability aid the pilot needs for therapy. The most common application is a crutch as pictured in Figure 2.11, but it could also be mounted on a walker.

2.7 Clinical Technician Interface

The device is designed to interface not only to the pilot but to the therapist administering the treatment as well. The clinical technician is able to select treatment options for the pilot as well as collect data for further analysis. The technical details of this interface will be covered in Chapter 4.

The interface runs on an Android tablet which communicates with the exoskeleton through the built-in bluetooth on all new Android compatible tablets. The protocol allows for rapid and secure transmission of data to and from the app. In fact, the app is capable of processing data at the control loop frequency of the exoskeleton (1 kHz). The Android App is an excellent platform to deliver both therapy customization and gait analysis tools.

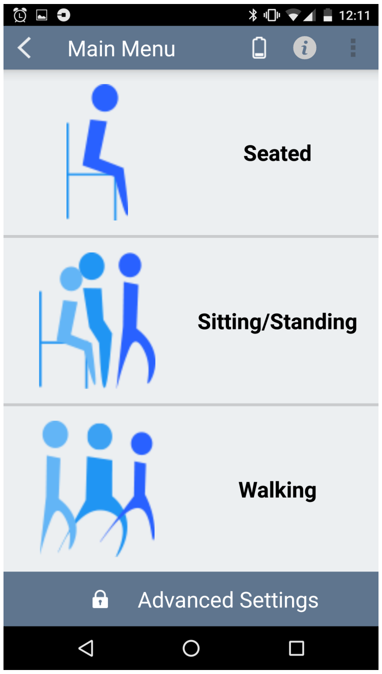


Figure 2.12: The Android app user interface for the gait practitioner.

2.8 Gait

The minimally actuated design of the Phoenix provides for a lightweight and cheap solution for both gait therapy and at-home applications, but it introduces some unique challenges in gait design. The Phoenix is designed to utilize the natural dynamics of the system to couple the motion of the hip to the knee. The underactuated design is a similar idea to the Austin Exoskeleton [33] where the knee was mechanically coupled to the hip motors, effectively mechanically programming the gait.

The Phoenix gait must use the dynamics of the system to swing the leg shank and provide enough toe clearance. This type of gait is biomechanically inspired in that in a normal human gait, the swing leg mostly takes advantage of the pendular dynamics of the legs to accomplish swing [32, 36]. The knee does not provide much power during the gait cycle, and it's mostly dissipative when it does. Figure 2.13 shows the power generated at the knee during a normal gait. Note that most of the power is negative and thus dissipative. Relying on these pendular dynamics to a greater degree than a human is the premise of the Phoenix gait. However, the underactuated design of the Phoenix does require a more careful gait design that will be covered in greater detail in later chapters.

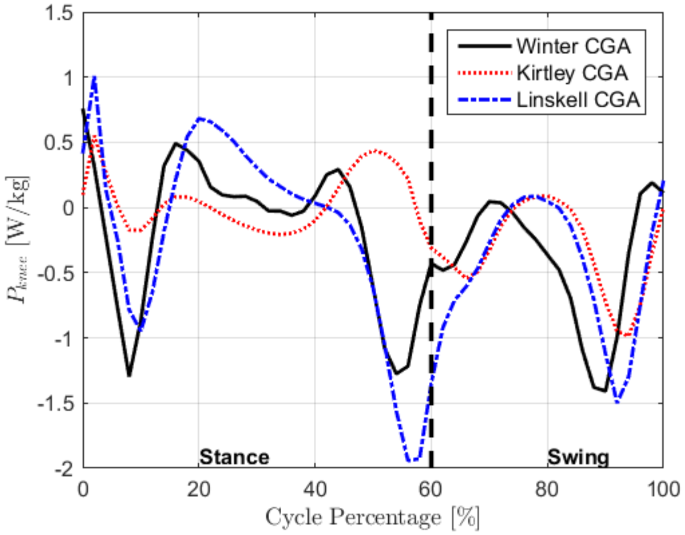


Figure 2.13: The CGA data for the knee joint power during normal walking. Data from [52].

Chapter 3

Exoskeleton Models

The gaits from the CGA data in Chapter 1 are a good place to start in the design of exoskeleton gaits. However, exoskeletons are far less actuated than the human body, so purely using CGA gaits as trajectories is not feasible for these systems. The gaits must be adapted to fit the constraints of the system while still providing a similar locomotion. The inspiration for exoskeleton gait development must come from both clinical data and modeling of the exoskeleton.

In order to study the gait of the exoskeleton, we need to develop a model for it. The model chosen to represent both the exoskeleton system and its pilot is a bipedal walker model. To this end, a kinematic, dynamic, and finite state machine model are generated. Each model will provide tools that can be utilized to both generate and analyze the gaits developed.

3.1 Kinematic Model

As discussed in the previous chapter, the Phoenix has an ankle locked at an approximate right angle and a knee mechanism that is always free in extension and selectively locked in flexion. Based on this design, the foot and leg shank are assumed to be locked rigidly together and can be modeled as one segment. Additionally, the knee locking and unlocking is assumed to be instantaneous, even though the stepper motor adds a delay to the knee mechanism.[1] The natural choice for these simplifications is a sagittal plane, 5 link biped, as depicted in Figure 3.2.

The five-link biped shown in Figure 3.1 is in the single stance phase of walking. In this case: link 1 is the stance leg shank and foot, link 2 is the stance thigh, link 3 is the torso, link 4 is the swing thigh, and link 5 is the swing leg shank and foot. The model angles are in absolute terms and are displayed in Figure 3.2. These angles will need to be translated

[1] Note: This is a good assumption, so long as the locking and unlocking time is short relative to the time in between locking and unlocking commands. The delay introduced by the stepper is around 100 ms. This delay is acceptable as long as the double stance period is greater than that.

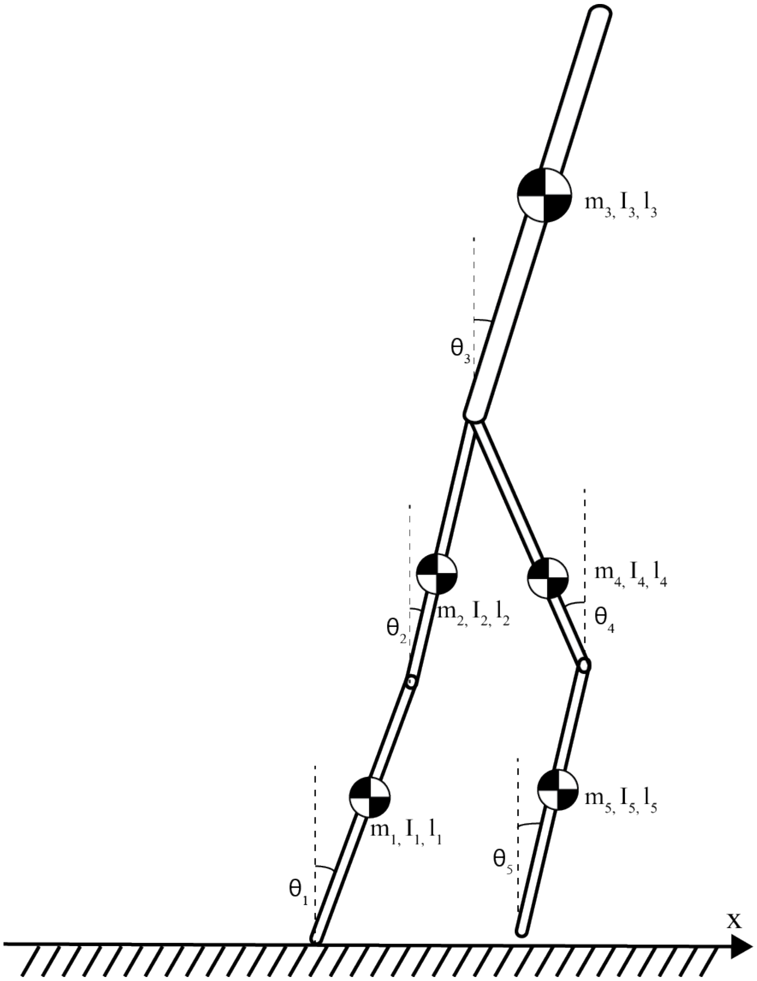


Figure 3.1: The 5-link biped model during single stance.

into relative terms according to medical definitions. This translation will be discussed later. Note: this system has five degrees of freedom and thus needs five actuators in order to fully control it. The absolute angles can be grouped into a single angle vector as follows:

$$\bar{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} \tag{3.1}$$

The five-link biped shown in Figure 3.2 is in double stance. The angle definitions are the same; again, they need translated into relative terms for analysis. The links are defined as: link 1 is the back leg shank and foot, link 2 is the back leg thigh, link 3 is the torso, link 4 is the front leg thigh, and link 5 is the front leg shank and foot. The distance from the back leg toe to the front leg heel is the step length.

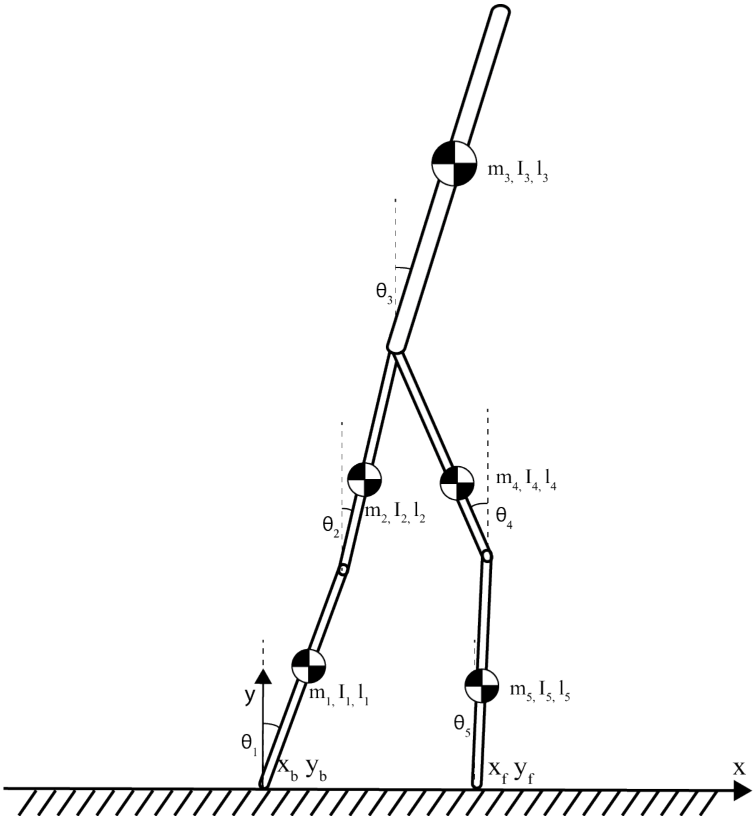


Figure 3.2: The 5-link biped model during double stance.

The kinematics of the five-link biped in double stance are such that there is a reduction in dimensionality. By definition during double stance, both feet must remain on the ground. This introduces a constraint that the front foot heel must always be on the ground. The constraints introduced are:

1. The front heel cannot slide along the ground. (X direction positional constraint)
2. The front heel cannot lift off the ground or push through the ground. (Y direction positional constraint)
3. The front heel cannot have any positional velocity. (X and Y direction velocity constraint)

Effectively, the legs of the exoskeleton form a four bar mechanism with a torso link attached at the top. Due to the constraints introduced by the front foot, the legs are now reduced to a two degree of freedom system. With the addition of the torso, the full system has three degrees of freedom. In order to reduce the dimensionality by one more degree, another constraint is needed. This constraint can be added by simply using the passive knee design described in Chapter 2 and locking the back knee. Now the legs are a four bar mechanism (one degree of freedom), and with the addition of the torso, the full system has two degrees of freedom. Now the system can be fully controllable in double stance with a two hip actuated system.

The lengths of the various limbs are given in Table A.1 according to Bartel [1].

Transformation to Medical Angles

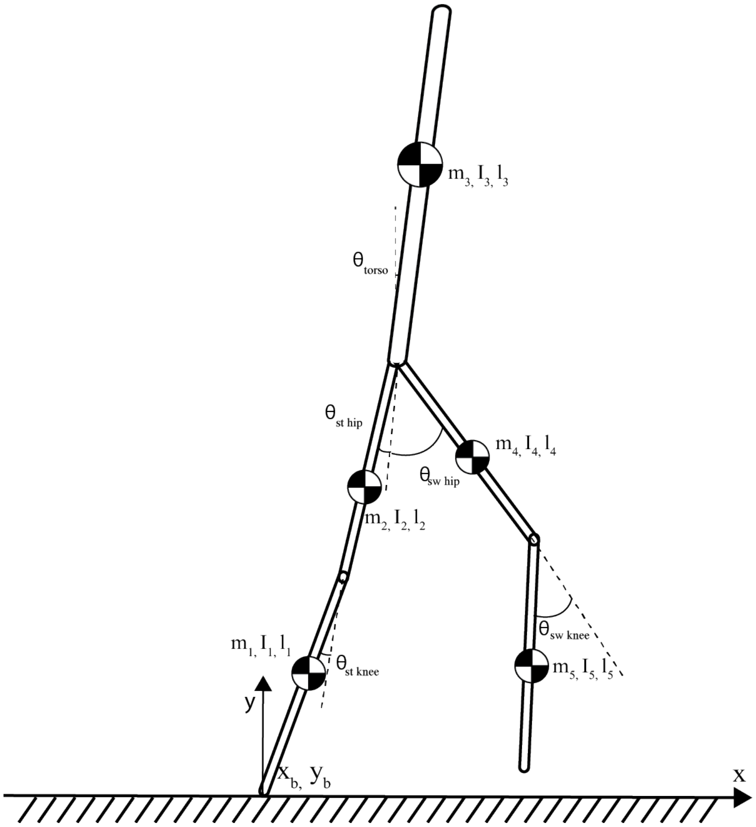


Figure 3.3: The 5-link biped model used for modeling the gait in medical terms.

In order to analyze the angles and angular velocities of the kinematic model, a transformation is needed from the absolute angles defined in Figure 3.2 to the medically defined angles in Figures 3.3 and 1.7. The medical joint angles are the torso angle (still absolute), stance hip angle, stance knee angle, swing hip angle, and swing knee angle. All the angles are relative to the torso angle, which itself is relative to the absolute vertical.

This can be simplified to a simple linear equation:

$$\underbrace{\begin{bmatrix}\theta_1\\\theta_2\\\theta_3\\\theta_4\\\theta_5\end{bmatrix}}_{\bar{\theta}} = \underbrace{\begin{bmatrix}0 & 0 & 1 & 0 & 0\\0 & -1 & 1 & 0 & 0\\1 & -1 & 0 & 0 & 0\\0 & 0 & 1 & 1 & 0\\0 & 0 & 0 & 1 & -1\end{bmatrix}}_{M} \underbrace{\begin{bmatrix}\theta_{\text{torso}}\\\theta_{\text{st hip}}\\\theta_{\text{st knee}}\\\theta_{\text{sw hip}}\\\theta_{\text{sw knee}}\end{bmatrix}}_{\theta_{\text{med}}} + \underbrace{\begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix}}_{b} \tag{3.2}$$

$$\bar{\theta} = M\theta_{\text{med}} + b$$

In double stance, the back leg will use the same definitions as the stance leg and the front leg will use the same definitions as the swing leg for single stance. This is simply to avoid defining a separate set of angles for double stance and translating between them for single stance.

3.2 Dynamic Model

Now that the kinematics of the system are fully defined above, a dynamic model can be generated to describe how torque inputs and constraints will affect the motions of the device. The dynamics of the system during single stance can be derived using Lagrangian dynamics. The derivation is not shown here for brevity. The full derivation is given by Tzafestas [48]. The equations that govern the motion are given by Equation 3.3.

$$D(\bar{\theta})\ddot{\bar{\theta}} + H(\bar{\theta})\dot{\bar{\theta}}^2 + G(\bar{\theta}) = \tau \tag{3.3}$$

where $D(\bar{\theta})$ is a 5x5 positive definite matrix known as the inertia matrix, $H(\bar{\theta})$ is a 5x5 matrix that represents the centrifugal and Coriolis effects, $G(\bar{\theta})$ is a 5x1 matrix that represents the gravitational effects, and τ is the set of joint torques acting on the system. The dot (e.g. $\dot{x}$) is used to denote derivatives with respect to time. The squared operator (e.g. x^2) on vectors in this case is defined as the elementwise square, that is each element of the vector is squared with itself. The values that make up the matrices D, H, and G are taken from the Tszafestas paper [48] and provided in Appendix B for completeness.

The dynamics given in Equation 3.3 are only valid for the single stance phase without any dynamic constraints on the system. In order for these dynamics to be applicable for the cases when constraints are applied, either the dynamics can be simplified by applying the constraint directly or a Lagrange multiplier can be utilized. In this paper, the Lagrange multiplier approach is utilized for simplicity and the possibility of backing out the constraint forces. Applying the Lagrange multipliers augments the system dynamics in Equation 3.3 to,

$$D(\bar{\theta})\ddot{\bar{\theta}} + H(\bar{\theta})\dot{\bar{\theta}}^2 + G(\bar{\theta}) = \tau + J^T(\bar{\theta})\lambda \tag{3.4}$$

where λ is the Lagrange multiplier that must be solved for at the same time as the dynamics, and $J(\bar{\theta})$ is the Jacobian matrix of the constraints in relation to θ. The Lagrange multiplier acts like a constraint force and the Jacobian matrix transforms the constraint forces into torques at each of the joints. The Jacobian matrix and number of Lagrange multipliers vary by the state of the system and the constraints imposed. Each phase of the gait will have a set of constraints and their associated Jacobians and Lagrange multipliers, e.g. locked knee constraint yields a knee constraint torque that keeps the knee from moving.

The masses, centers of mass, and moments of inertia of the various limbs are given in Table A.2 according to Winter [51].

Double Stance Phase 1

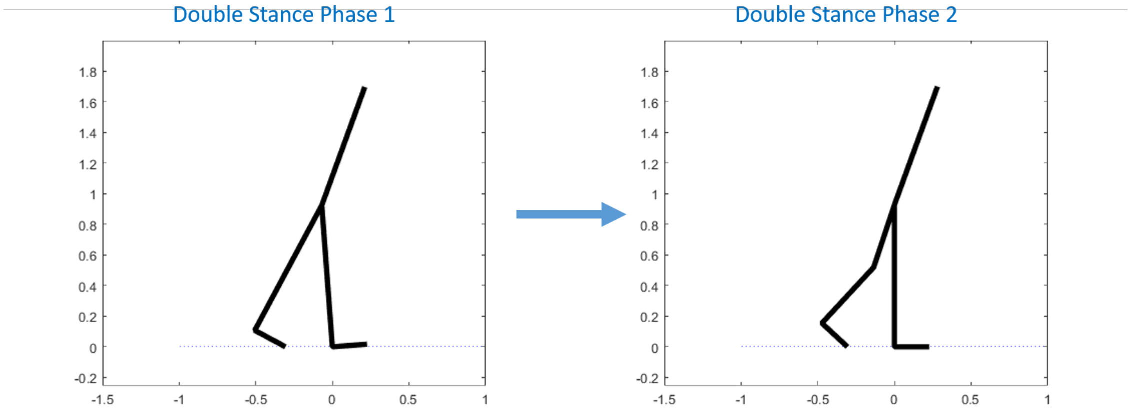


Figure 3.4: Double stance is split into two phases with different dynamic constraints.

This paper breaks double stance into two separate phases, double stance phase 1 and double stance phase 2, shown in Figure 3.4. In double stance phase 1, the bipedal walker starts at the end of single stance with both knees fully extended and the front heel and back toe in contact with the ground. The walker progresses through this phase by unlocking the front knee and shifting the weight forward with both feet still on the ground and the back knee completely locked in the fully extended state. The purpose of this phase is to smoothly and continuously transfer the weight towards the front foot after the end of single stance and prepare for another step.

As shown in Figure 3.5, the constraints for this system are that:

1. The front heel cannot translate in the x-direction.

2. The front heel cannot translate in the y-direction.

3. The back knee is locked.

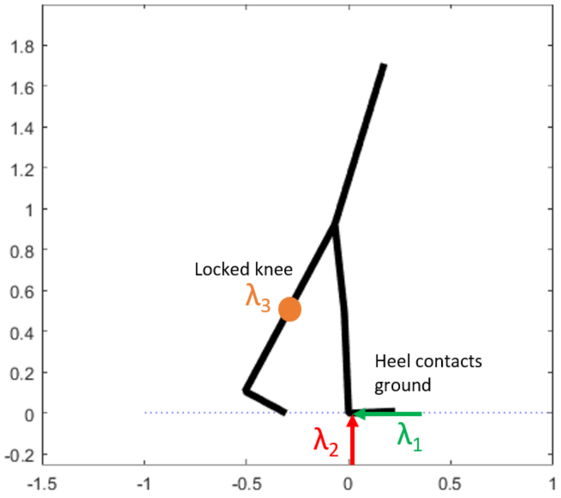


Figure 3.5: The 5-link bipedal walker model constraints during double stance phase 1.

These holonomic constraints can be expressed mathematically as:

$$\Phi(\bar{\theta}) = \begin{bmatrix} x_f - x_b - L_{\text{step}} \\ y_f - y_b \\ \theta_{\text{st knee}} \end{bmatrix} = 0 \tag{3.5}$$

where the x_b and x_f are the x-position of the back and front foot respectively, y_b and y_f are the y-position of the back and front foot respectively, and $\theta_{\text{st knee}}$ is the back leg's knee angle, as shown in Figures 3.5 and 3.2. The "stance" and "swing" definitions are recycled here so that new angle definitions are not required. Taking the derivative of $\Phi(\bar{\theta})$ with respect to $\bar{\theta}$ in Equation 3.5 yields the Jacobian matrix needed for the dynamics in Equation 3.4.

$$J(\bar{\theta}) = \begin{bmatrix} \ell_1 \cos(\bar{\theta}_1) & \ell_2 \cos(\bar{\theta}_2) & 0 & \ell_4 \cos(\bar{\theta}_4) & \ell_5 \cos(\bar{\theta}_5) \\ -\ell_1 \sin(\bar{\theta}_1) & -\ell_2 \sin(\bar{\theta}_2) & 0 & -\ell_4 \sin(\bar{\theta}_4) & -\ell_5 \sin(\bar{\theta}_5) \\ 1 & -1 & 0 & 0 & 0 \end{bmatrix} \tag{3.6}$$

λ for this phase of the gait will be a 3x1 matrix of the x and y ground reaction forces, respectively, and the torque needed to keep the back (stance) knee locked. The masses, centers of mass, and moments of inertia of the various limbs are given in Table A.2 according to Winter [51].

Double Stance Phase 2

The second stage of double stance is characterized by the front knee locking to extension and the back knee unlocking. This allows the center of gravity to continue moving forward while

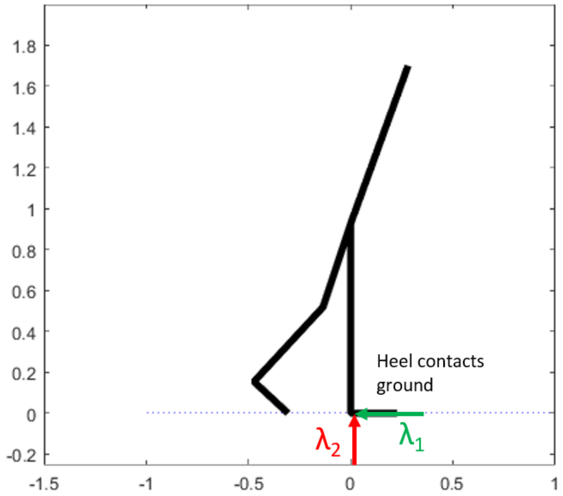


Figure 3.6: The 5-link bipedal walker model constraints during double stance phase 1.

preparing the back leg for liftoff. It is kinematically impossible to lift the back leg off the ground without bending the knee[2]. From Figure 3.6, the back knee needs to bend until the knee, toe, and hip are aligned. Without this back knee bend, it would only be possible to lift the toe off the ground by first moving the leg backwards. At the moment the knee aligns with the toe and hip, the forward velocity of the torso stops. If the back knee continued to bend, then the center of gravity of the body would lower and move backwards, which is the antithesis of forward locomotion. Therefore, it is clear that the exact moment that this alignment happens is the perfect moment to lift the toe off the ground to avoid a waste in energy.

The holonomic constraints for the bipedal walker during this phase are:

1. The front heel cannot translate in the x-direction.

2. The front heel cannot translate in the y-direction.

3. The front knee is locked to flexion.

As shown in Figure 3.6, these constraints can be expressed mathematically as:

$$\Phi(\bar{\theta}) = \begin{bmatrix} x_f - x_b - L_{\text{step}} \\ y_f - y_b \end{bmatrix} = 0 \tag{3.7}$$

There is another constraint, but it does not introduce a constraint force. The front knee can only extend during this phase. It does not make sense for the knee to flex further during

[2] It is actually also possible to simply lift the back leg backwards but would require a tremendous waste of energy.

double stance phase 2 because the front ankle should be at its most flexed state at this point. The front knee would have to come to a stop (zero velocity) at end of double stance phase 1 (beginning of double stance phase 2) because otherwise the knee motion would introduce a sudden jerk due to the sudden introduction of a new constraint force. Jerks are uncomfortable to the pilot and cause excessive wear on mechanical components. Therefore, there is a constraint that the knee cannot flex during this phase, but this constraint is not active in the generating the gait and is only for safety in implementation.

The Jacobian matrix associated with the constraints in Equation 3.7 to be plugged into Equation 3.4 is

$$J(\bar{\theta}) = \begin{bmatrix} \ell_1 \cos(\bar{\theta_1}) & \ell_2 \cos(\bar{\theta_2}) & 0 & \ell_4 \cos(\bar{\theta_4}) & \ell_5 \cos(\bar{\theta_5}) \\ -\ell_1 \sin(\bar{\theta_1}) & -\ell_2 \sin(\bar{\theta_2}) & 0 & -\ell_4 \sin(\bar{\theta_4}) & -\ell_5 \sin(\bar{\theta_5}) \end{bmatrix} \tag{3.8}$$

and λ is a 2x1 matrix of the x and y ground reaction forces, respectively. The masses, centers of mass, and moments of inertia of the various limbs are given in Table A.2 according to Winter [51].

Single Stance Phase 0

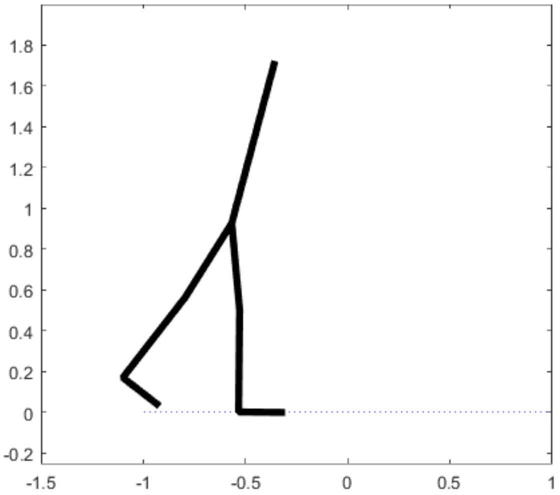


Figure 3.7: The 5-link bipedal walker model during single stance phase 0 has no constraints. This walker has the full dimensionality and is thus uncontrollable.

Similarly to double stance, this paper also breaks single stance into three separate stages. The first stage is single stance phase 0, where the back foot is lifted and starts swinging forward while the front knee moves towards full extension. In this phase, the foot constraints are no longer applied, the swing knee (formerly back knee) is now unlocked, and the stance knee (formerly front knee) is only locked to flexion. The stance knee will continue its extension from the previous phase, so the knee constraint actually isn't active. The flexion-locked knee is simply for safety in implementation. Therefore, this phase of the walking cycle has no active constraint forces. The phase will end once the stance knee is fully extended and locked instantaneously. The Lagrange multiplier λ in this case is simply 0.

Single stance phase 0 is a difficult phase to control due to the lack of dynamic constraints on the underactuated system. As mentioned above, dynamic constraints help reduce the dimensionality of a system and thus make it more controllable. With no active constraints on the system, the bipedal walker in single stance phase 0 becomes completely uncontrollable. This phase would depend on the momentum out of double stance to continue to the next phase. This would be very difficult to accomplish in both simulation and experiments; as such, this phase is not considered in the gait development in Chapters 4 and 5.

Single Stance Phase 1

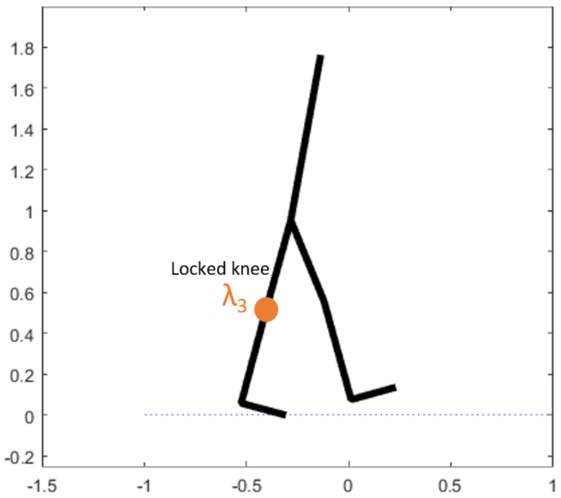


Figure 3.8: The 5-link bipedal walker model constraints during single stance phase 1.

The second phase of single stance has the stance knee fully locked while the swing leg swings through and locks at full extension. This stage comprises the bulk of the time of single stance phase of the walking cycle. The purpose of this portion of the gait is to provide propulsion by pivoting on the stance toe and moving the swing leg from behind the center of mass to in front of the center of mass. This will also move the center of mass towards the front of the stance foot and begin to tip the center of mass forward about the stance toe.

As shown in Figure 3.8, the constraints for this phase are relatively simple: the stance knee must be fully locked. This yields a very simple Jacobian matrix:

$$J(\bar{\theta}) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \end{bmatrix} \tag{3.9}$$

and a 1x1 λ of the torque at the stance knee required to keep it from moving. At the end of this phase, the swing knee will reach full extension and transition into single stance phase 2.

Single Stance Phase 2

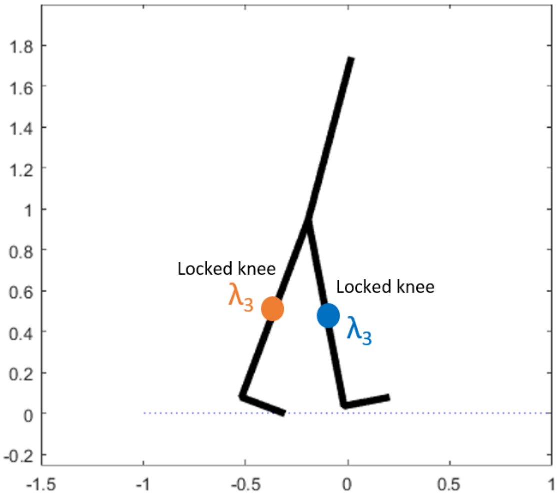


Figure 3.9: The 5-link bipedal walker model constraints during single stance phase 2.

Single stance phase 2 is the final stage of the walking cycle and serves to simply finish the swing phase and transition into double stance by placing the locked swing leg back onto the ground. There is a closure constraint that the end state of single stance phase 2 must be the same as the start state of double stance phase 1, thus completing the cycle. To apply this closure constraint, both knees must be locked and the swing foot must move to connect with the ground at the exact step length. Furthermore, to avoid an unnecessary jerk to the body, the heel strike must be impactless, meaning that the foot must hit the ground with zero velocity. This impactless heel strike will yield a gait that is both comfortable for the pilot and does not cause any unnecessary shock loads to the device.

As shown in Figure 3.9, the constraints for this phase are similar to the previous, but with the addition of the locked swing knee:

1. The stance knee is locked to flexion.

2. The swing knee is locked to flexion.

Translating these constraints into a holonomic constraint equation,

$$\Phi(\bar{\theta}) = \begin{bmatrix} \theta_{\text{st knee}} \\ \theta_{\text{sw knee}} \end{bmatrix} = 0 \tag{3.10}$$

and finally taking the derivative with respect to $\bar{\theta}$ yields a Jacobian

$$J(\bar{\theta}) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{3.11}$$

with a 2x1 λ matrix of the torque required to keep the stance and swing knees locked, respectively.

Single stance phase 2 can be difficult to consider in the development of gaits because the pendular motion of the leg shank is very time dependent. If the transition to phase 2 occurs too soon, then too much energy is used slowing down the natural motion. If the transition occurs too late, then the leg shank could oscillate slightly before locking to transition.

Conclusion

In gait development, single stance phase 0 would be very difficult to design for considering the lack of any dynamic constraints and the lack of actuators. This phase would depend on momentum to succeed. Single stance phase 2 would also be very difficult to design for, because it's difficult to determine how long the phase should be. While the single stance phase 0 and single stance phase 2 are likely to appear in the use of human exoskeletons due to the stochastic natural of the real world, they should be avoided during gait development if at all possible due to their complexity and relative actuator cost.

3.3 Finite State Machine

The physical model of the 5-link exoskeleton walker is a hybrid automaton, that is a model comprised of a finite state machine and a set of dynamics for each state. The highest level model for the bipedal walker is the finite state machine that shows how the system moves from one set of constraints, dynamics, and kinematics to the next. The finite state machine shows the possible states and the conditions on which it progresses. This finite state machine only deals with the walking states of the device. All other states related to sitting and standing should be considered separately. As discussed in the sections above, the bipedal walker model has two phases of walking: single stance and double stance, and five stages: two for double stance and three for single stance. A diagram of this finite state machine is shown in Figure 3.10.

The states, their purposes, and their conditions for triggering are detailed in Table 3.1.

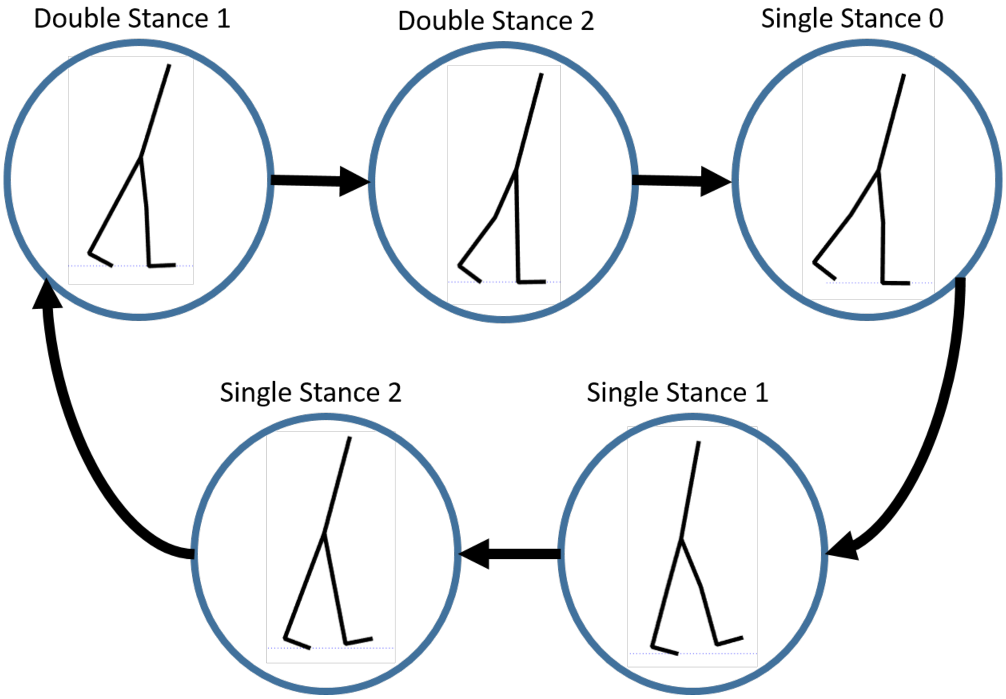


Figure 3.10: A hybrid automaton model of the exoskeleton.

State	Purpose	Trigger
Double Stance 1	Transition weight forward, continue forward momentum, connect front foot fully with the ground.	The front foot is fully in contact with the ground and the front ankle reaches its dorsiflexion limit.
Double Stance 2	Continue weight transition, continue forward momentum, bend back knee to allow toe-off	The back knee is aligned with the hip and back toe.
Single Stance 0	Continue forward momentum, begin swing, straighten stance knee, bring stance ankle to perpendicular.	The stance knee is fully extended.
Single Stance 1	Continue swing, tip weight forward, pivot about the stance toe, fully extend swing knee.	The swing knee is fully extended.
Single Stance 2	Conclude swing, bring foot to the ground at the step length without impact.	The foot touches the ground without impact in the same position as the start.

Table 3.1: Finite State Machine Transition Conditions

Chapter 4

Tunable Gaits

The gait for the Phoenix exoskeleton, like many other exoskeletons, is inspired by the natural human gait collected from clinical gait analysis (CGA). Because the Phoenix has actuation only at the hips and a semi-passive locking and unlocking knee mechanism, the gaits are implemented in the form of a hip trajectory that is tracked by the hip actuators and through the clever locking and unlocking of the knee mechanism. These hip trajectories are split into separate trajectories for the individual states of the gait for the stance and swing legs during both single stance and double stance. The states of this system are controlled by a finite state machine implemented on the Phoenix exoskeleton. The state transitions are triggered through the use of a crutch-mounted pilot interface. The hip trajectory is distilled into five important nodes inspired by the CGA data and from four-bar mechanism kinematics. These nodes are then used by a polynomial spline trajectory generator that outputs a trajectory for each control cycle of the Phoenix for the hip motor controller to track. This provides the pilot a smooth and continuous gait that he or she can use to walk continuously with the Phoenix.

4.1 Introduction

The objective of a gait rehabilitation device is to mimic a natural human gait in the hopes of restoring a natural gait to the pilot. The trajectory shown in Figure 4.2 is both the left and right hip angles for two cycles of a natural human gait from CGA data [52]. Similarly, Figure 4.3 shows the left and right knee angles for two cycles of a natural human gait from CGA data. These angles are graphically defined in Figure 4.1. The hip angle is the relative angle between the thigh and the line continuing downward from the torso. Hip extension is defined as the thigh rotating backwards about the hip joint. Hip flexion is defined as the thigh rotating forwards about the hip joint. The knee angle is defined as the relative angle between the leg shank and the line continuing downward from the thigh. Knee extension is a forward rotation about the knee joint, and knee flexion is a backward rotation about the knee joint.

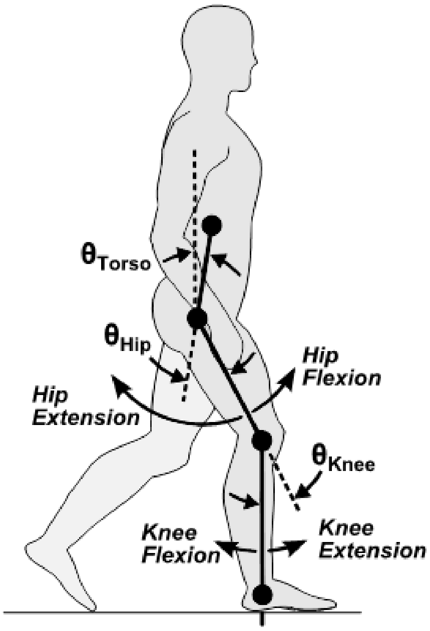


Figure 4.1: Definition of joint angles [32]. Repeated Figure 1.7 for clarity.

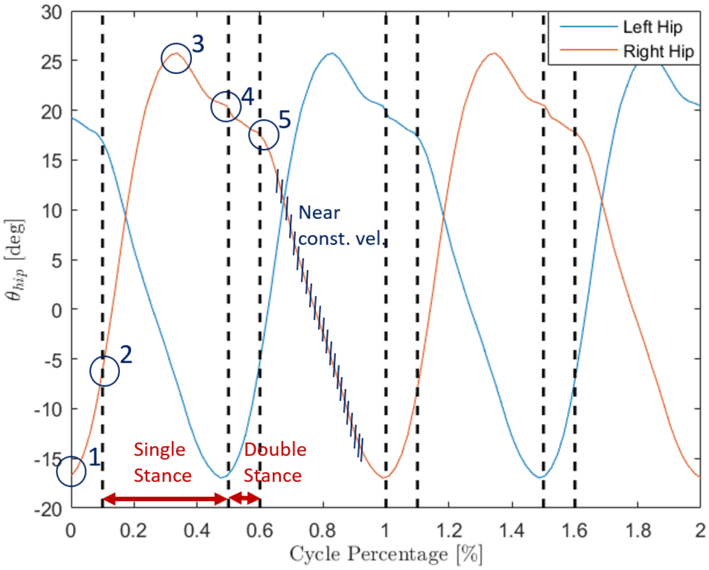


Figure 4.2: CGA hip angles for both single and double stance.

The single stance and double stance portions of the gait are separated by dotted lines where the short slices are double stance and the large slices are single stance. While any human exoskeleton should draw inspiration from CGA gaits, actual gaits vary from person to person. Additionally, during gait rehabilitation, therapists may want to train a pilot by

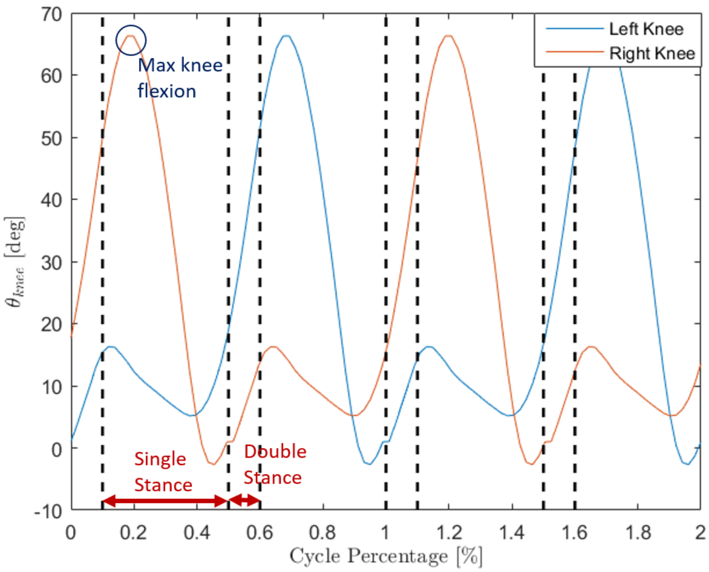


Figure 4.3: CGA knee angles for both single and double stance.

Plot Point	**Percent of Cycle Time**	**Angle Definition**
1	0%	Maximum Hip Extension
2	10%	Swing Start
3	35%	Maximum Hip Flexion
4	50%	Swing End
5	60%	Stance Start

Table 4.1: Points labeled on Figure 4.2

focusing on a particular part of the gait. Because there is no one ideal gait, tunable gaits allow for a more pilot-tailored rehabilitation experience. Remember, human exoskeletons are underactuated. In particular, the Phoenix has only hip actuation with a semi-passive locking and unlocking mechanism at the knees as shown in Figure 4.4. A tunable gait should draw tuning parameters from the clinical hip data shown in Figure 4.2 and 4.3.

Instead of defining the entire hip trajectory for a cycle, we define the hip trajectory by five major angles as shown in 4.2 and defined in the Table 4.1. In other words, these angles can be used as parameters to define the entire hip trajectory. However, the angles of double stance can be defined relative to each other due to the kinematic constraints of the system defined in Chapter 3. These constraints allow for a simplification where only three angles and three times are needed to describe the gait cycle.

Double stance is defined in Chapter 1 as the portion of the gait cycle where both feet are in contact with the ground. Since there are two legs in stance, the legs are defined as the “front” leg and the “back” leg depending on their position in relation to the torso during this gait phase. This phase of the gait is responsible for continuing the momentum from the single stance phase, propelling the center of mass forward, and transitioning the weight from the rear foot to the front foot such that the back foot can be lifted off the ground. To accomplish this under the kinematic constraints described in Chapter 3, the Phoenix must flex the front knee and hip while extending the stance hip, see Figures 4.2 and 4.3.

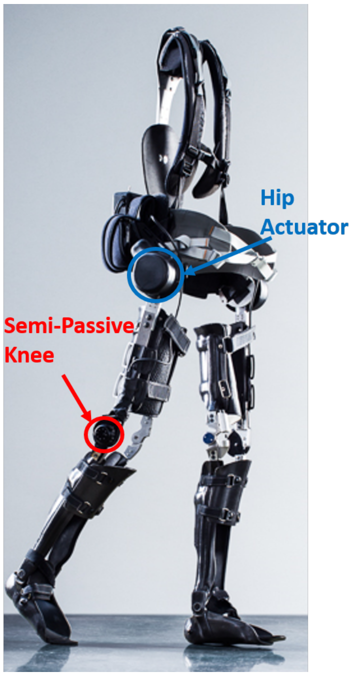


Figure 4.4: The Phoenix exoskeleton joint configuration.

The stance leg during single stance propels the center of mass forward while the swing leg moves forward. Since the center of gravity of the human body is approximately at the hips, the hips should move at roughly a constant velocity in the x-direction to prevent the extra energy expenditure associated with accelerating and decelerating the center of mass. To accomplish this, the stance hip angle should have a near constant angular velocity similar to the stance hip angle shown in Figure 4.2.

As discussed in the Chapter 1, single stance is the stage of the gait where only one foot is on the ground. During single stance, the swing hip swings forward from the *swing start angle* at the end of double stance reaching *maximum hip flexion* before extending back for heel contact at the *swing end angle*. Where the swing leg is defined as the leg that is not on the ground, and the stance leg is defined as the leg that is in contact with the ground. Furthermore, the *maximum hip extension angle* is a natural parameter choice because it describes the transition of the stance leg to the swing leg during double stance. The final angular parameter is used to define the hip flexion at the start of the stance trajectory is called the *stance start angle*.

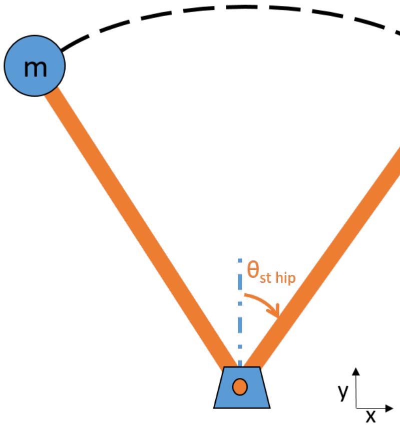


Figure 4.5: The inverted pendulum model for the stance leg.

The remaining parameters needed are simply the *step time*, the *percent time in double stance*, and the *percent time to maximum hip flexion*.

The Phoenix exoskeleton must take advantage of the kinematic constraints defined in Chapter 3 to accomplish a similar motion as the CGA data shows in Figure 4.2. The kinematic constraints allow for a simplification such that only points 1 or 2 and 4 or 5 in Figure 4.2 and Table 4.1 need to be defined to define the behaviour during double stance. Though the ideal gait would likely be similar to the CGA gait shown in Figure 1.8, the Phoenix gait must take into consideration it's underactuation and rely on the clever locking and unlocking of the knee to manipulate the kinematics. This yields a qualitatively different gait than the CGA data. The constraints from Chapter 3 manifest as point 5 flexing to allow for the front foot to come into contact with the ground and point 2 extending past point 1 due to the kinematic constraints. Points 2 and 4 are defined relative to points 1 and 5, respectively, in the Phoenix gait.

Another qualitative difference between the CGA data and the gait of the Phoenix is the exaggerated *maximum hip flexion*. The reason for the sinusoid-like motion from maximum hip flexion to extension is thought to be to take advantage of the pendular dynamics of the leg shank, reducing the overall energy cost of actuation [32]. The pendular dynamics of the knee are related to modeling the leg shank as a pendulum and using the hip input and dynamics to force the knee to flex and re-extend during swing. This causes the knee to reach peak flexion around the same time as the hip reaches peak velocity, as shown in Figures 4.2 and 4.3. The Phoenix tunable gait must take advantage of these pendular dynamics of the leg shank to a greater degree due to the lack of knee actuation. Using the clinical gait for inspiration, the *maximum hip flexion* is a natural parameter to tune.

4.2 Control Architecture

The control architecture for the Phoenix exoskeleton is given in Figure 4.6. The pilot gives commands via the user interface (UI) crutch which is fed into the finite state machine detailed above and implemented in the Phoenix control software. This gives the pilot full control over the state transitions of the exoskeleton. The gait therapist also has control over some parameters that he or she tunes through the use of the Android app interface, but the therapist cannot command state transitions from the app interface. The therapist-tunable parameters are listed in Table 4.2. These parameters allow the gait therapist to tune the therapy to the pilot's comfort and therapy needs. The sections below will detail the gait parameters and how the parameters map from the tablet to trajectories.

Parameter	**Minimum**	**Maximum**	**Unit**
Maximum Hip Extension	0	10	Degrees
Maximum Hip Flexion	20	40	Degrees
Stance Start	0	20	Degrees
Step Time	0.9	1.5	Seconds
Percent Time in Double Stance	5	20	Percent
Percent Time to Maximum Flexion	35	60	Percent
Walking Angle	0	10	Degrees
Standing Angle	-5	20	Degrees
Seated Angle	85	105	Degrees
Bent Forward Angle	110	135	Degrees

Table 4.2: Parameters that the therapist can tune.

The exoskeleton's implemented finite state machine takes in parameters from the gait therapist and adjusts state transitions accordingly. The parameters are also fed into the online trajectory generators. Each state's trajectories are generated at each control cycle due to the computational demands of generating each trajectory (1,000 to 3,000 points) and the limited computation resources of the embedded processor. The embedded processor also has memory limitations that make it difficult to store many large arrays of trajectories. The finite state machine selects whichever trajectory generator is associated with the current state and samples it every control loop, e.g. the step state will call both the swing leg and stance leg trajectory generators for the relevant legs and generate a point at each control loop.

The finite state machine will then take these trajectories and put them into a control algorithm for each leg. As discussed in Chapter 2, the control algorithms do not need to consider the dynamics of the motor because the motor controller controls the current by taking in the motor commands and controlling the torque directly. The control algorithms used are simple proporitonal (P) and proportional-derivative (PD) controllers that were tuned by hand using the Android app interface. The controllers output a motor signal

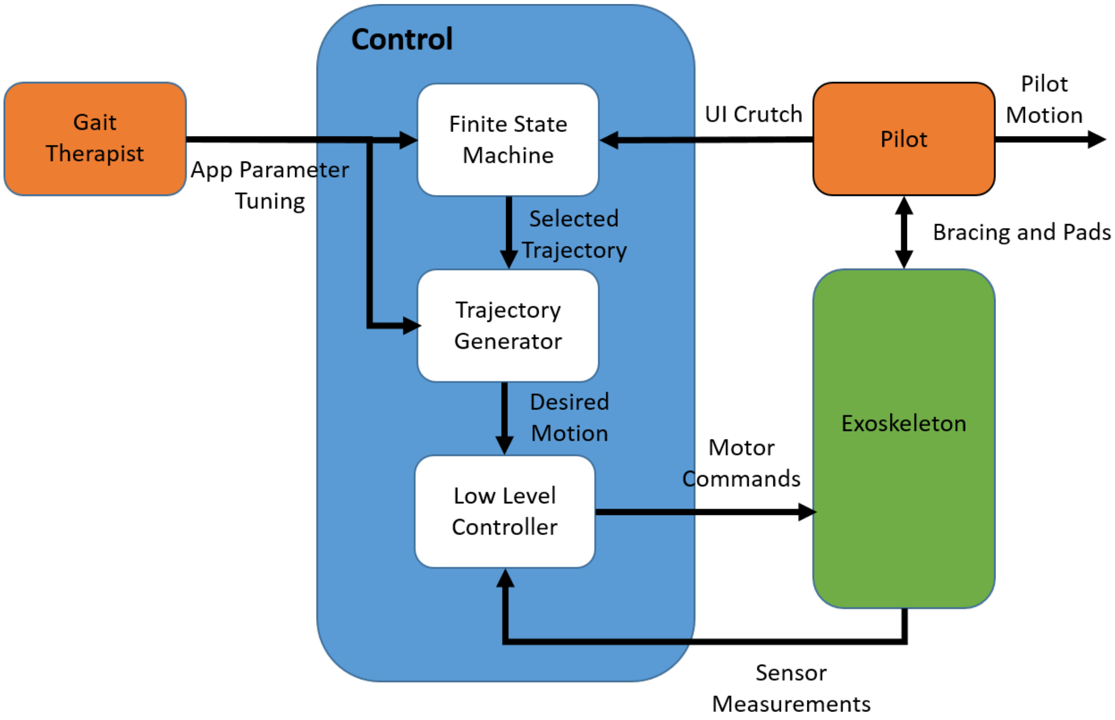


Figure 4.6: The Phoenix exoskeleton control architecture including both pilot and gait therapist input.

that the motor controllers use to power the motors on the exoskeleton. The exoskeleton provides absolute encoder angle feedback to the low level controllers which completes the close loop feedback. The pilot is rigidly attached to the exoskeleton using the bracing and pads discussed in Chapter 2. The coupling of the exoskeleton and pilot provides motion to the pilot as the motors are commanded and the exoskeleton moves.

Not pictured in Figure 4.6 is the visual and audio feedback that the exoskeleton gives to the pilot. This is to communicate safety and important state transition information to the pilot during operation. Also not pictured is the data that is uploaded to the Android app for the gait therapist's consumption. All the sensor readings are compiled into a data structure that is sent over Bluetooth to the Android app. The hope is that this data can be used to track the performance of each pilot and tailor the therapy to the pilot as performance improves.

4.3 Finite State Machine

The trajectories developed in the sections below are implemented in the easy-to-navigate finite state machine given in Figure 4.7. Unlike finite state machines used in the modeling, this finite state machine considers seated and standing states and the transitions between them as well as walking states. This figure shows how the forward commands transition "forward" and the backward commands transition "backward" through the finite state machine. Each of the button presses will trigger the associated state change. More detailed descriptions of the states are shown in Figure 4.8. The action states are defined as the states where the exoskeleton is tracking a trajectory. These states are characterized by red arrows in the state circles. The static states are simply regulating the system about a certain fixed position.

Figure 4.7 shows a transition from a group of seated states to the standing states using the forward command of the Phoenix pilot interface. From the standing states the pilot can then either transition into the walking states through a forward command or the seated states through a backward command. The walking states can transition to the standing states and subsequently the seated states through a series of backward commands.

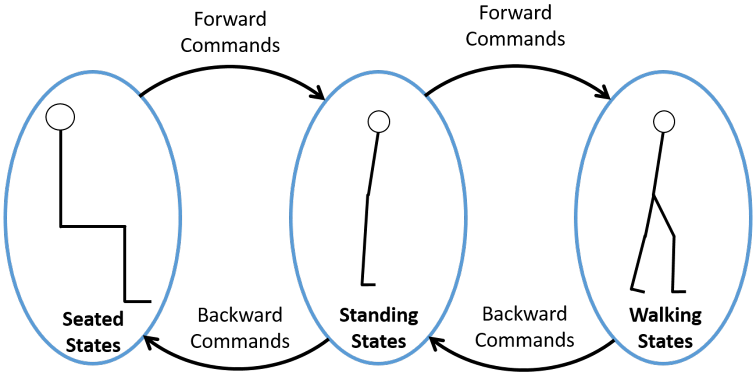


Figure 4.7: The overall finite state machine for the Phoenix Exoskeleton.

A closer look at the seated states is shown in Figure 4.8. Starting at the seated state, the pilot can transition to bent forward, in preparation of the standing up motion, by providing a forward command and moving through the seated to bent transitional state. If the pilot decides to reposition or abort the bent forward state, then a backward command can be given, and he or she will be returned to the seated state via the bent to seated state. From bent forward, the standing states can be entered through a forward command and passing through the standing up state. The seated states are entered from the standing states and involve the sitting down transitional state.

The standing states are a middle ground between the seated and walking states where the pilot can relax yet be upright or transition into one of the other sets of states. The standing state is entered by either the standing up state from bent forward (seated states) or the static split stance state (walking states) via the feet together state. The feet together state provides a very simple blending trajectory to bring the legs together in a standing posture. From standing, the pilot can transition back to seated by a backward command and passing through the sitting down state. A forward command will move towards walking by first commanding a first step state. The first step trajectory is different from the other walking trajectories in that the legs start together and the swing amplitude must be exaggerated for extra toe clearance. At the end of the first step, the exoskeleton will be in static split stance (a walking state). static split stance is where the exoskeleton is in double stance but is not actively walking. The device is in a static position in preparation for the next step.

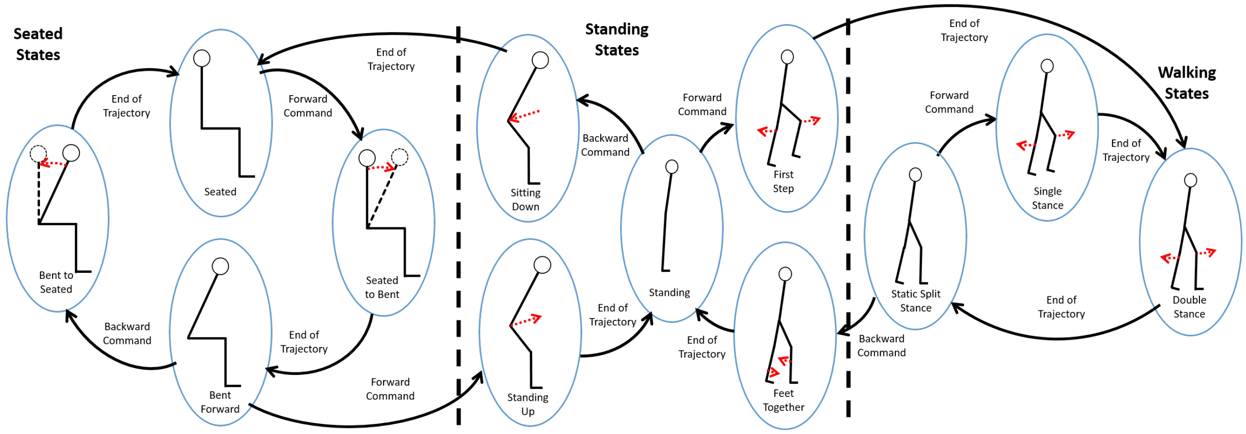


Figure 4.8: The full finite state machine for the Phoenix Exoskeleton.

The walking states are where the propulsion of the gait actually happens. The device will start in the static split stance state after having transitioned through the first step state. Now the device can be commanded to continually take steps through repeated forward commands. The steps are broken into the single stance and double stance states as described in the section above. These states are abstracted to be independent of which leg is in swing in stance through the use of an internal state which determines the swing side. However, the first step state always has the left leg be the swing leg to start. The feet together state will operate independent of which leg is in front with the same mechanism as single stance and double stance.

Static States

The static states in Figure 4.8 are depicted as the states without a red arrow (seated, bent forward, standing, and static split stance). Static states do not provide any motion to the pilot but instead hold the user in a fixed position. The states with a red arrow are considered active states. Each of these states are controlled by regulating about a particular set point. The states and their associated set points are described in Table 4.3.

The static states are controlled to regulate the set points through a proportional (P) controller. The controller in each of these states is not tuned to forcefully hold the pilot at the set point but more to guide the pilot to that position and allow freedom of movement. The pilot could be in these states for a long period of time, so comfort is very important.

Each of the set points given in Table 4.3 is associated with a parameter that the gait therapist can tune to provide a comfortable experience to the pilot.

Static State	Set Point
Seated	Seated Angle
Bent Forward	Bent Forward Angle
Standing	Standing Angle
Static Split Stance	Walking Angle

Table 4.3: Static State Set Points

Transition States

The transition states are the active states that transition from one static state to another without passing through a second active state. The transition states are all the active states that are not in the walking states set (bent to seated, seated to bent, standing up, sitting down, first step, and feet together). These transition states and their associated trajectories are given in Table 4.4.

Transition State	Hip 1 Trajectory	Hip 2 Trajectory
Bent to Seated	Bent to Seated Cubic	Bent to Seated Cubic
Seated to Bent	Seated to Bent Cubic	Seated to Bent Cubic
Standing Up	Standing Up Cubic	Standing Up Cubic
Sitting Down	Sitting Down Cubic	Sitting Down Cubic
First Step	Exaggerated Swing	Cubic Stance
Feet Together	Cubic Forward Blend	Cubic Backward Blend

Table 4.4: Transition State Trajectories

From Table 4.4, most of the trajectories are simply cubic spline transitions from one static state to the next. The cubic trajectory is built assuming zero velocity at the start and end points. The starting and ending positions are simply the set points associated with the previous and next static states respectively. While the cubic spline trajectories are based on the set points in Table 4.3, these trajectories are generated by a cubic spline algorithm, relieving the gait therapist from the need to these trajectories.

The unique transition state is the First Step state which is simply an exaggerated swing trajectory. This exaggerated swing is necessary to get proper toe clearance when starting with both feet together. The first step is more easily understood after the single stance and double stance trajectories are understood.

All the transition states are controlled using a Proportional-Derivative (PD) controller tuned for tracking the trajectory without causing discomfort to the user. The controller was tuned with the pilot in the system for direct user feedback.

These transition trajectories will be discussed further at the end of this chapter, after the double stance trajectories have been developed.

Locomotion States

The active states during walking are the same as the biological phases of walking: single stance and double stance. At the end of each step, the finite state machine will transition into static split stance until another step is commanded. Note: the finite state machine will immediately transition into the next step if the pilot continuously provides a forward command.

The trajectories for these states are given as swing and stance trajectories. While there is no swing leg during double stance, the same terms are used to describe the trajectories in double stance to avoid confusion. The "swing leg" during double stance describes the leg that was just in swing during the single stance state. This relationship holds true for the "stance leg" as well. The swing and stance trajectories flip upon entering static split stance. The locomotion states and their associated trajectories are given in Table 4.5.

Transition State	**Hip 1 Trajectory**	**Hip 2 Trajectory**
Single Stance	Swing	Stance
Double Stance	Swing Leg Double Stance	Stance Leg Double Stance

Table 4.5: Walking State Trajectories

Similar to the transition states, the locomotion state controllers are tracking-tuned PD controllers. The development of these trajectories and their tunable parameters is given in the sections below.

4.4 Static States

The static states in the finite state machine are used as states where the pilot can safely rest or prepare himself or herself for the next motion. The motors simply regulate about a set point with a loose controller that allows the pilot to move around without being impeded by the device.

In this section, each static state is expounded.

Seated

In the seated state, the pilot is sitting in a chair or wheelchair. The controller is regulating about the *seated angle* set point with a loose P controller. The knees are locked in this state. Usually, the pilot will don the device while seated in their personal wheelchair before turning on the device and transitioning to the standing and walking states.

This state is switched into from the bent to seated state at the end of the bent to seated trajectories. The seated state is also switched into from the sitting down state at the end of the sitting down trajectories. The seated state switches into the seated to bent state with a forward command.

Bent Forward

In the bent forward state, the pilot is bent forward in a seated position in preparation for the standing up active state. The controller is regulating about the *bent forward angle* with a loose P controller. The knees are locked in this state. Usually the pilot will only be in this state for a short period of time to position the crutches behind him or her. This crutch position is useful to provide a safe and stable stand-up.

The bent forward state is switched into from the seated to bent state at the end of the seated to bent trajectories. The bent forward state switches into the bent to seated state with a backward command. The bent forward state switches into the standing up state with a forward command.

Standing

In the standing state, the pilot is standing straight with his or her feet together. The controller is regulating about the *standing angle* with a loose P controller. The knees are locked in this state. The pilot is free to make small motions and work with his or her hands in this state. This state can be comfortable and therapeutic for pilots with spinal cord injury. Standing in and exoskeleton or standing frame can help prevent secondary injuries from prolonged sitting.

The standing state is entered into from the standing up state at the end of the standing up trajectories. The standing state is also entered into from the feet together state at the end of the feet together trajectories. The standing state switches into the sitting down state with a backward command and into the first step state with a forward command.

Static Split Stance

In the static split stance state, the pilot is standing with his or her feet apart in between steps. The controller is regulating the torso about the *walking angle* with a loose P controller. The knees are locked in this state. The pilot generally uses this state to take a quick break after a series of steps or to reposition himself or herself before taking another step.

The static split stance state is entered from the first step state at the end of the first step trajectories. The static split stance state is also entered from the double stance state at the end of the double stance trajectories. The static split stance state switches into the single stance state with a forward command.

4.5 Transition States

The transition states in the finite state machine are used as states where the pilot transitions between sets of static states. These states exclude the active walking states. The transition movements are mostly just cubic spline trajectories from one set point to the next assuming zero start and end velocity. The splines are all developed with the angle- and velocity-defined splines discussed in Appendix D.

In this section, each transition state is expounded.

Bent to Seated

In the bent to seated state, the pilot is transitioning from the bent forward state to the seated state. The controller is tracking a cubic spline trajectory from the *bent forward angle* set point to the *seated angle* set point assuming zero starting and ending velocities with a PD controller. The knees are locked in this state. This transition is mostly useful for aborting the crutch preparation for standing up.

This state is switched into from a backward command in the bent forward state. The state exits to the seated state.

Seated to Bent

In the seated to bent state, the pilot is transitioning from the bent forward state to the seated state. The controller is tracking a cubic spline trajectory from the *seated angle* set point to the *bent forward angle* set point with a assuming zero starting and ending velocities PD controller. The knees are locked in this state. This transition is necessary in the preparation for standing up.

This state is switched into from a forward command in the seated state. The state exits to the bent forward state.

Standing Up

In the standing up state, the pilot is transitioning from the seated state to the standing state. The controller is tracking a cubic spline trajectory from the *seated angle* set point to the *standing angle* set point assuming zero starting and ending velocities with a PD controller. The knees are locked to flexion in this state. This transition is necessary to get out of the wheelchair and enjoy the health benefits of a human exoskeleton for those with paraplegia.

This state is switched into from a forward command in the seated state. The state exits to the standing state.

The trajectory for standing up is shown in Figure 4.9. This data was collected in real-time on the device.

Sitting Down

In the sitting down state, the pilot is transitioning from the standing state to the seated state. The controller is tracking a cubic spline trajectory from the *standing angle* set point to the *seated angle* set point assuming zero starting and ending velocities with a PD controller.

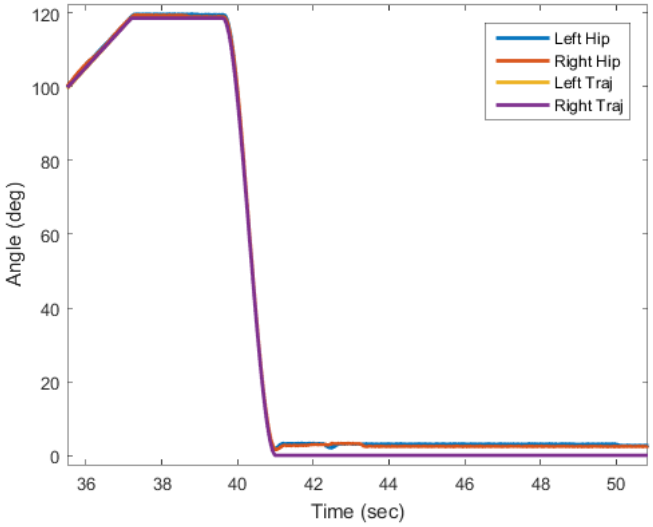


Figure 4.9: The standing up trajectory.

The knees are unlocked to allow the knees to flex in this state. This transition is necessary to get back in the wheelchair and doff the suit after a gait therapy session.

This state is switched into from a backward command in the standing state. The state exits to the seated state.

The trajectory for sitting down is shown in Figure 4.10. This data was collected in real-time on the device.

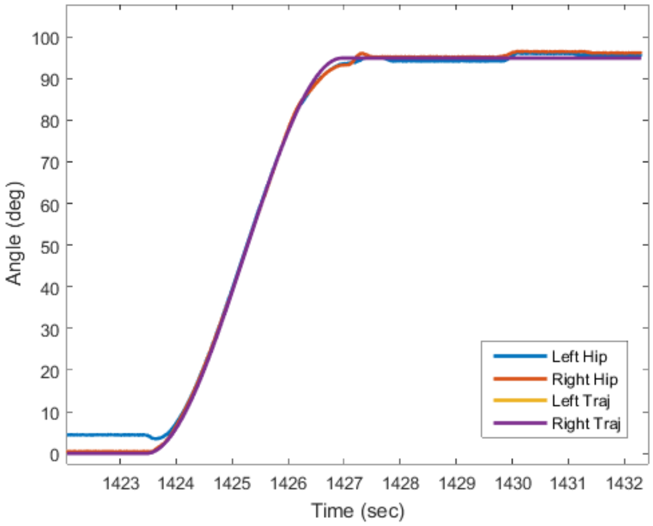


Figure 4.10: The sitting down trajectory.

First Step

In the first step state, the pilot is transitioning from the standing state to the static split stance state. The controller is tracking exaggerated single stance trajectories with a PD controller. The basis of the trajectory will be explained in the active walking states section. The difference is that this state starts at the *standing angle* and the hip swing is exaggerated in this state to achieve toe clearance. The swing knee is unlocked at the beginning to allow the swing knee to flex for toe clearance. At the peak hip flexion, the swing knee locks to flexion again, and the knee extends back to a straight leg. The stance knee is always locked. This transition is necessary to begin walking and to lift the foot off the ground when the pilot begins in the standing state. Similar to the swing trajectory, the first step trajectories are developed using the angle- and velocity-defined splines discussed in Appendix D.

This state is switched into from a forward command in the standing state. The state exits to the double stance state.

The trajectories for the first step are shown in Figure 4.9. This data was collected in real-time on the device.

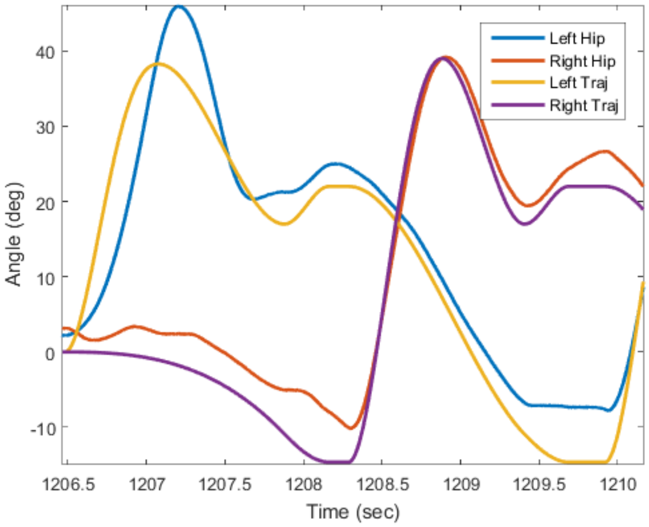


Figure 4.11: The first step trajectory before taking a regular step.

Feet Together

In the feet together state, the pilot is transitioning from the static split stance state to the standing state. The controller is tracking a cubic spline trajectory from the hip angles during static split stance to the *seated angle* set point assuming zero starting and ending velocities with a PD controller. The knees are locked in this state. This transition is necessary to get both feet back in line in order to stand in the more comfortable standing state. It is often easier to put the pilots weight on the back foot and bring the front foot backwards in this state.

This state is switched into from a backward command in the static split stance state. The state exits to the standing state.

The trajectories for feet together are shown in Figure 4.12. This data was collected in real-time on the device.

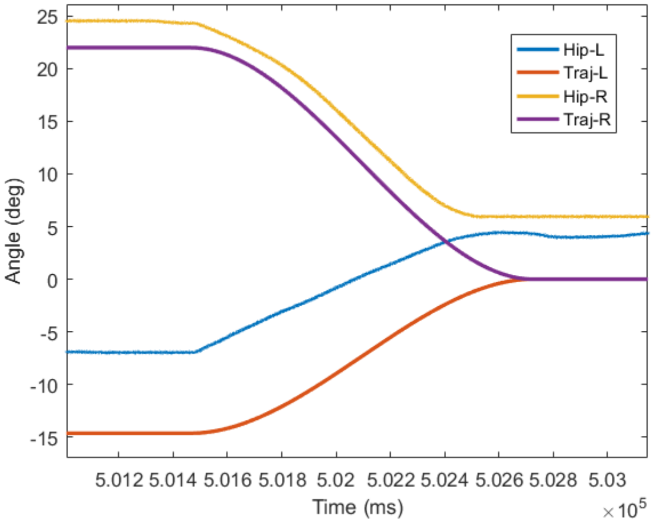


Figure 4.12: The feet together trajectories.

4.6 Walking States

In this section, the walking states are described. The static split stance state was already described in the static states section, so that state will not be discussed here. The double stance and single stance states are active states that provide the motion necessary for bipedal locomotion. These states are entered from the static split stance state by the pilot giving a forward command.

These states in the Phoenix finite state machine mimic the single and double stance phases of the CGA gait described in Chapter 1. The single and double stance states are the critical states where the pilot should spend most of his or her time during therapy to maximize the benefits of using the device.

Single Stance

In the single stance state, the exoskeleton is propelling the pilot's center of mass forward using the stance leg as support and swinging the swing leg forward. The controllers are tracking

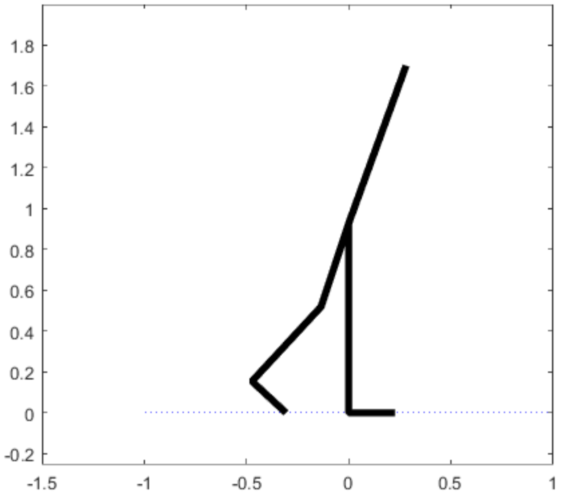


Figure 4.13: An illustration of the beginning of single stance phase 1.

quartic spline trajectories that mimic the CGA data in Figure 4.2 with a PD controller. The swing knee is unlocked at the beginning to allow the swing knee to flex for toe clearance. At the peak hip flexion, the swing knee locks to flexion again, and the knee extends back to a straight leg for heel strike. The stance knee is always locked. The stance hip and swing hip each have trajectory that uses the clinical gait as an inspiration.

This state is switched into from a forward command in the static split stance state. The state exits to the double stance state at the end of the single stance trajectories.

Stance Trajectory

The entire body can be modeled as an inverted pendulum during single stance with the stance leg being the pendulum and the torso being the mass, m. This stance leg model is shown in Figure 4.14. The center of gravity of the entire body is approximately at the hips, so if the hips move at a constant velocity, then so does the body. Over the range of motion of single stance, the small angle approximation can be made. Assuming a straight leg, which should be the case, the forward hip velocity is:

$$\begin{aligned} x &= \ell_{\text{leg}} \sin(\theta) \\ \dot{x} &= \ell_{\text{leg}} \dot{\theta} \cos(\theta) \\ &\approx \ell_{\text{leg}} \dot{\theta} \left(1 - \frac{\theta^2}{2}\right) \end{aligned} \tag{4.1}$$

And within about fifteen degrees of zero, the quadratic term can be ignored as well. These equations yield a close to proportional relationship with the stance hip's angular velocity and the hip position. The near-linear stance hip trajectory is also apparent in the CGA data shown in Figure 4.2. Of course, this linear region is just an approximation, but there is a great deal of freedom in single stance gait design because there are no closed-chain constraints.

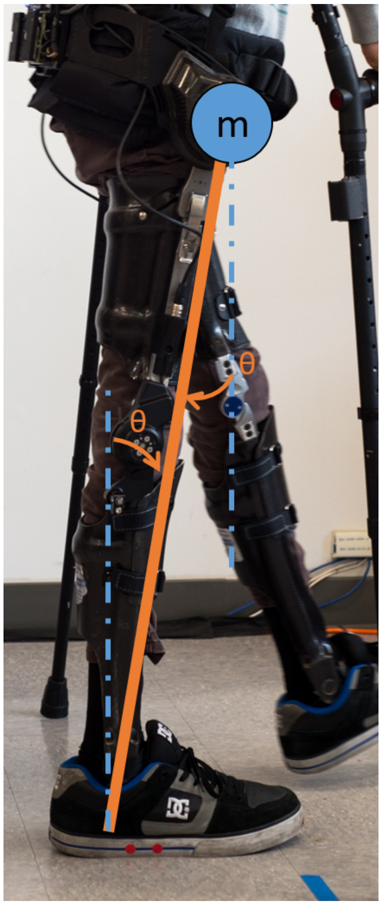


Figure 4.14: The simple model used for the stance leg.

Using the relationship above, it makes sense to try and have a somewhat linear gait through single stance and double stance phase 2. To accomplish this, the only two points from Figure 4.2 that are necessary are point 5 and point 1. The trajectory is implemented in the form of a angle- and velocity-defined spline discussed in Appendix D. Double stance phase 2 is included in the stance leg trajectory generation because the general motion is the same as single stance. Double stance phase 2 will be described in greater detail in the section below.

The stance leg's hip trajectory during double stance will be assumed to form a linear relationship between the points 1 and 2 in Figure 4.2. This linear relationship will keep the hips moving forward at a roughly constant velocity even during the beginning of double stance for the rear (previously stance) leg. The angle and velocity is derived by assuming a line, and based off the amount of time in double stance, the position can be calculated. The velocity is simply the slope of the line.

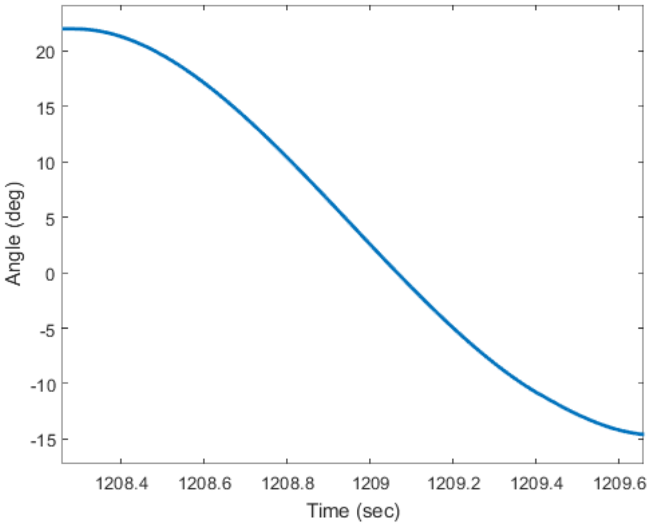


Figure 4.15: An example stance hip trajectory using this method.

An example of the gait generated using this method is given in Figure 4.15. A nearly linear trajectory is realized with larger variations at the start and end points. It starts at point 5 and ends at point 2.

Swing Trajectory

The other side of the gait cycle to consider during single stance is the swing leg. While the stance leg is propelling the body forward, the rear leg needs swing through and take a step forward. As mentioned above, the swing hip trajectory, while biologically-inspired, is exaggerated to increase the toe clearance for the less actuated system. The leg needs to hit a peak swing angle and then settle back on the ground for a heel strike with minimal impact.

The swing trajectory is developed in terms of a three-node angle- and velocity-defined spline discussed in Appendix D. The angle of the middle node of the single stance swing hip gait is denoted as *maximum hip flexion* (point 3 in Figure 4.2) because it is the point of *maximum hip flexion* throughout the gait cycle. The velocity at this point (point 3 in Figure 4.2) must be zero, because it is a maximum point.

The point 4 in Figure 4.2 for swing needs to be determined as the transition between swing leg single stance and front leg double stance. From the *maximum hip flexion* angle to the transition to double stance phase 1, the swing hip is extending until the heel touches the ground. Recall during the following stage of double stance phase 1, the hip will be flexing to allow the front knee to bend and the weight to transition forward. Therefore, at the point of heel touch, it is preferable to have the swing hip velocity equal zero. With this constraint in mind, the entire gait cycle is now defined in terms of the points in Figure 4.2 and Table 4.1.

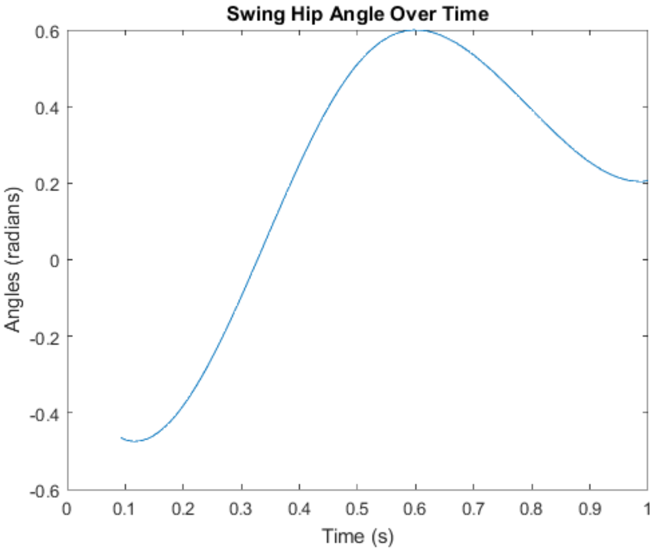


Figure 4.16: An example swing hip trajectory using this method.

An example swing trajectory is plotted in Figure 4.16. Notice the exaggerated *maximum hip flexion* angle in comparison to Figure 4.2.

Double Stance

In the double stance state, the exoskeleton is transitioning the pilot's center of mass forward over the front foot and continuing the forward momentum by cleverly locking and unlocking the knee mechanisms. The controller is tracking a quartic spline trajectory that mimic the CGA data in Figure 4.2 with a PD controller. The front leg knee is unlocked at the beginning of double stance to allow the front knee to flex and transition the weight forward in the first phase of double stance. At the end of double stance phase 1, the front knee locks to flexion again. The back knee is always locked. The front and back hip trajectories for double stance are related to each other by the closed chain kinematics. The trajectories in double stance are all developed with the angle- and velocity-defined splines discussed in Appendix D.

This state is switched into from at the end of the single stance and first step states. The state exits to the static split stance state.

Phase 1

The double stance phase 1 portion of the gait is controlled through a clever unlocking of the front knee just after heel strike. The moment before double stance begins for this system, the heel has just impacted the ground and both knees are locked. Now the device forms a triangle

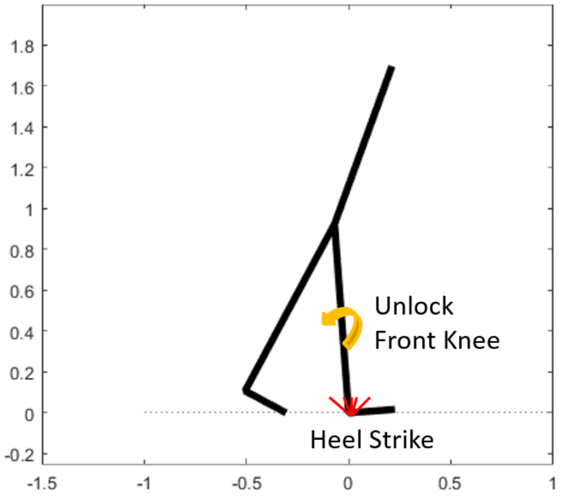


Figure 4.17: An illustration of heel strike and the beginning of double stance phase 1.

(see Figure 4.17) which cannot be moved without applying enough torque to rollover the heel. To resolve this issue, the front knee is unlocked. This relieves one kinematic constraint and allows the application of four bar mechanism kinematics to the legs of the system. In a four bar mechanism, only one degree of freedom exists. The exoskeleton system in double stance phase 1 has one degree of freedom for the legs and one for the torso with two inputs at the hips allowing for a completely controllable system. To generate a gait based on this kinematic chain, solving the kinematic equations of the lower body is the natural place to start; this analysis is given in Appendix C.

Unlocking the knee at heel strike is an unusual behaviour for a traditional locking knee orthosis. Most controllable locking knees will lock at heel strike and support the stance knee through both single and double stance. The control of this device is unique in that it cleverly unlocks at heel strike in order to take advantage of kinematic constraints to bring the system to full controllability.

The double stance for the Phoenix exoskeleton will differ from the natural human gait shown in Figure 4.2 in that during double stance phase 1 the stance hip will extend and the swing hip will flex. This behaviour is due to the double stance kinematic constraints and lack of knee actuation.

During this phase of walking the hip position is moving forward over the front foot. Because of the four bar mechanism constraints, the hip motion leads to the hips also moving down. While the forward motion is desirable, the downward displacement will have to be recovered during the next phase of the gait before Single Stance.

Using the four bar mechanism equations in Appendix C, the final angles of the exoskeleton can be immediately solved by simply specifying one angle in the chain. The forward (swing) heel angle (related to ϕ in Appendix C) was chosen as the angle that dictates the state of the lower body. Some front ankle flexion was allowed for this phase even though the rest of the model considers the ankle to be rigid. This constraint relaxation was to allow for considerations of the flexibility and slightly rounded nature of the shoes used with the exoskeleton. The overall allowance was a meager 6° of flexion.

By specifying a front ankle flexion at the end of double stance phase 1, the hip angles can be solved together. Fixing the front ankle's flexion at the end of double stance to 6° yields $\phi = \frac{\pi}{2} - 6\frac{\pi}{180}$ for the Freudenstein equations in Appendix C. The remaining angles are then back-solved and translated into the medical angles using Equation C.3. Using the system parameters defined in Appendix A and a step length $L_{\text{step}} = 30.68$ cm (see section below for details) which is a shorter step than the average man (78 cm) or woman (70 cm), resulted in an angle of $-22.65°$ for the *maximum hip extension angle* and an angle of 14.29° for the *stance start angle* from Figure 4.2 with a 6° torso angle. All these parameters can be tuned to fit the pilot's needs, but the angles defined from the kinematics is a good place to start.

Phase 2

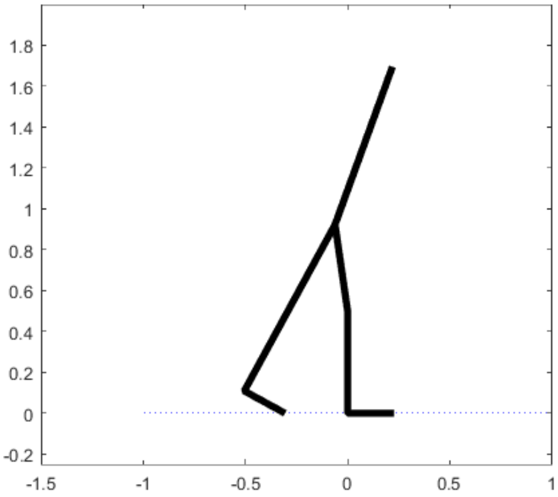


Figure 4.18: An illustration of the beginning of double stance phase 2.

Double stance phase 2 begins at the end of double stance phase 1 with the forward ankle at full flexion. To avoid the complicated single stance phase 0 dynamics, the front knee should be fully extended at the end of double stance ($\theta_{\text{st knee}} = 0$). As mentioned in Chapter 3, the goal of double stance phase 2 is to begin propelling the body forward by flexing the rear hip and extending the forward hip.

Throughout this phase, the position of the hip is moving upwards in the y-direction, regaining what was lost in the first phase. The hip is also moving forward in the x-direction, continuing the momentum that was established in phase 1. However, this effect only happens for some amount of bending in the rear knee. This is due to the line from the rear toe to the hips effectively increasing while the knee bends. Once this line is maximized, the hip position will start to move down and back again. This length is maximized when the hip, rear knee, and rear toe come in line. Therefore, double stance phase 2 and double stance in general ends when the rear knee comes in line with the toe and hip.

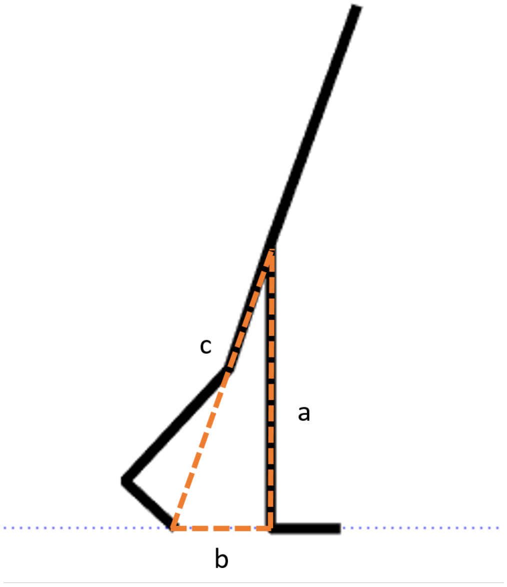


Figure 4.19: Illustrating the triangle used to help find the maximum step length.

In order for the rear knee to be in line with the rear toe and hip and the front knee to be fully extended, the step length must be within the correct range. These constraints cannot be met without also constraining the step length. Figure 4.19 illustrates the maximum of this range, where the front leg is vertical. The legs form a triangle with the back leg denoted as c and the front leg denoted as a; the step length can be solved for using the Pythagorean theorem.

$$\begin{aligned} c &= \sqrt{\ell_{\text{foot}}^2 + \ell_{\text{shank}}^2} + \ell_{\text{thigh}} \\ a &= \ell_{\text{thigh}} + \ell_{\text{shank}} \\ b &= \sqrt{c^2 - a^2} \end{aligned} \tag{4.2}$$

Using this equation to calculate the step length, the result is:

$$L_{\text{step}} = b + \ell_{\text{foot}} \tag{4.3}$$

Similar to double stance phase 1, the angles could be solved explicitly. The angles at the end of double stance phase 2 are easily calculated using the triangle in Figure 4.19. However, for simplicity, the double stance phase 2 trajectories will be included in the single stance trajectory generation, both swing and stance hips. The double stance phase 2 trajectories are defined by the geometry and the angles at the end of double stance phase 1. The two hips also move in the same direction in double stance phase 2 as the start of single stance, meaning this approximation is reasonable.

Walking Gait Generation

The points defined in the sections above were used to define nodes for quartic and cubic polynomial splines defined in Appendix D. These trajectories were applied to a five link bipedal model to verify the feasibility.

An example tunable gait is shown in Figure 4.20. This shows both the double stance and single stance phases. The knee trajectories were also designed using the methods above, but these joints are not actuated. Therefore, the knee trajectories can not actually be tracked and are designed as "desired" trajectories instead. Despite the fact that the knee trajectories cannot be tracked, it makes sense to specify a trajectory for them for validating the tunable hip trajectories.

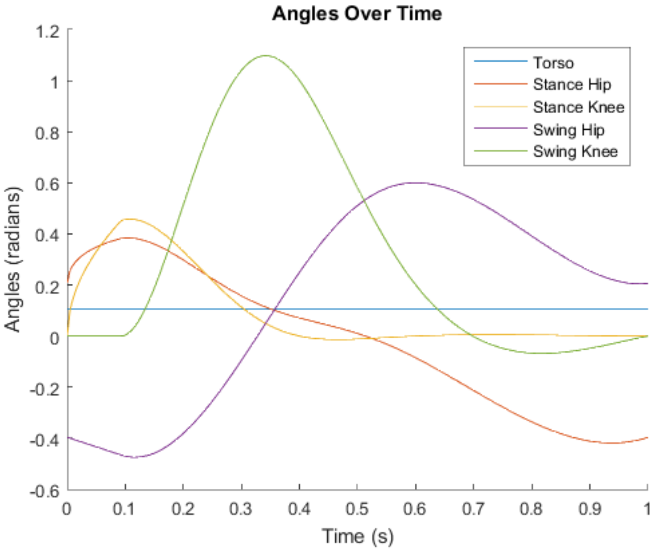
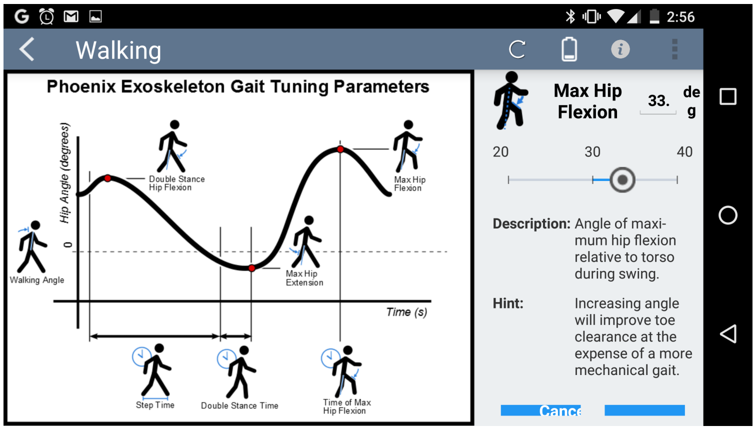


Figure 4.20: A example gait using the tunable gait method. The knee and torso angles are simply educated guesses based on kinematics.

4.7 User Interface

An Android-based interface was chosen for its familiarity with various users and ease of implementation. The Android app communicates with the device via Bluetooth and modifies the parameters that define the nodes described in the sections above. The gait practitioner



Max Hip Flexion 33. deg

Description: Angle of maximum hip flexion relative to torso during swing.

Hint: Increasing angle will improve toe clearance at the expense of a more mechanical gait.

Figure 4.21: An example of the Android app's user interface to tune the gaits.

additionally has control of the *step time*, the *percent time in double stance*, and the *percent time to maximum hip flexion*. The percent of the cycle time spent in double stance can perhaps be excluded in a released product because the standard time in literature is approximately 20% of the cycle [51].

The parameters were selected with the gait practitioner in mind. In a gait therapy session, the practitioner needs to be able to quickly tune the device's trajectories to account for the comfort and gait therapy needs of a new pilot. The difficulty comes in translating these terms into parameters on the exoskeleton that are useful for generating the trajectories and generating the spline coefficients that will ultimately generate the trajectory (Appendix D).

The design of the Phoenix exoskeleton itself requires input from the pilot in order to operate. The pilot commands steps using the crutch interface whenever he or she wants to command a step. Furthermore, the gait design tries to minimize the input from the pilot, but the pilot is still required to lean in the frontal plane in order to help shift the weight toward the front foot during double stance. During double stance, the pilots legs are split, and the system is stable in the sagittal plane. However, in the frontal plane, the center of mass of the body is more towards the side of the rear leg, so the exoskeleton will tend to fall forward and towards the side of the rear leg without crutch support. So some input from the pilot is expected in this device. In fact, this type of variation and concentration from the pilot can help improve the gait therapy.

4.8 Embedded Implementation

In the development of these gaits, the premise was that the state of the knee (locked or unlocked to flexion) could be changed at will. While this is true, there is a delay between the command to unlock and the time at which the knee actually unlocks. This delay can be up to 200 ms. The delay must be accounted for in the software design.

The gait trajectories must be switched quickly in the finite state machine. The control loop frequency is 1 kHz and runs on a NXP LPC1768 ARM Cortex-M3 processor. These resources are too limited to be able to compute a full trajectory on the fly, so a limited resource trajectory generation method must be used. This can be accomplished by taking the parameters and generating the next trajectory point when it's needed. So the parameters are stored on the exoskeleton, and the spline coefficients are recomputed when new parameters are set. This allows the micro-controllers to only compute one point along the trajectory at a time. However, this method does have the disadvantage of having to compute a trajectory point every control loop versus just accessing it from memory.

The Phoenix control flowchart is given in Figure 4.22. The flowchart starts with pilot interface commands and sensor measurements including the IMU and encoders described in Chapter 3. These measurements enter the finite state machine shown in Figure 4.7 which determines whether to trigger a state change or not. Regardless, the next trajectory point for both hips will be generated, and the motors will control the exoskeleton about those points. Then all the data from the sensors, inputs, and states of the device is stored to an SD Card. If a parameter is received, the system parameters will be updated and stored for loading on startup. If necessary, audio or visual feedback will be given to the pilot to communicate state and safety information.

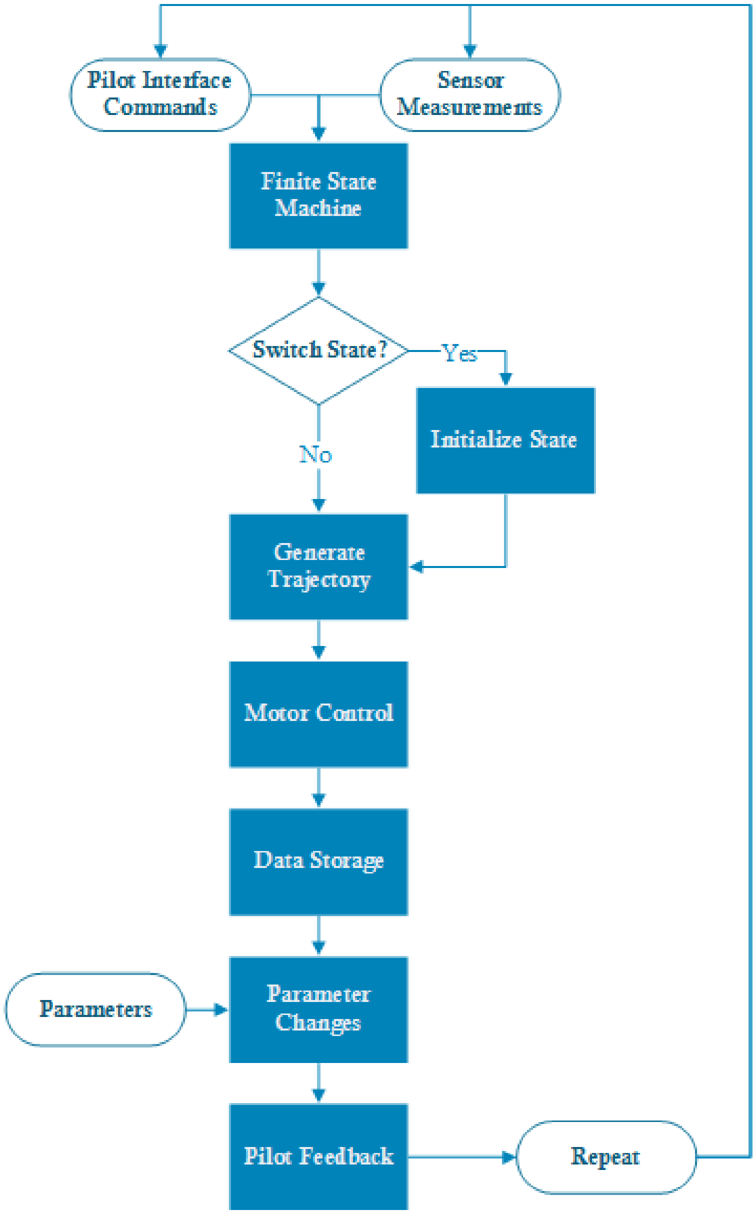


Figure 4.22: The logical flowchart for the control of the Phoenix.

Chapter 5

Optimal Gaits

An alternative to tuning each gait per the pilot's needs is to attempt to find the best possible gait for each individual pilot given the pilot's parameters (Appendix A) and set of meta-parameters such as step length and walking speed. There are multiple methods to search for optimal solutions. In this chapter, two methods are considered: parametric optimization and discretized optimization. This chapter provides methods and simulations that could be utilized in similar exoskeleton systems. The requirements for implementation are detailed in the final section.

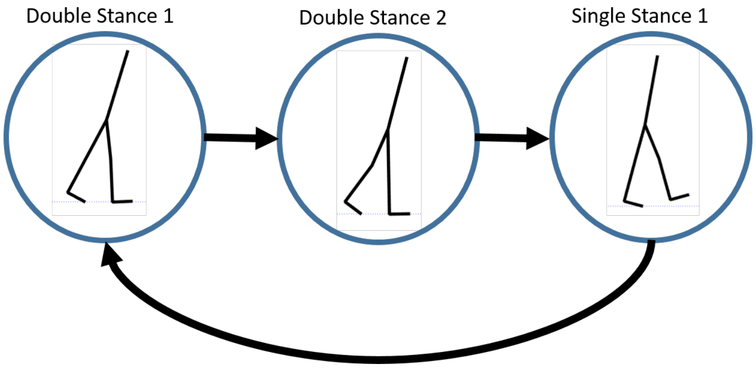


Figure 5.1: The finite state machine used for optimization.

To optimize the gait, a dynamic model and finite state machine (Figure 5.1) are needed. The exoskeleton model is composed of three of the states described in Chapter 3: double stance phase 1, double stance phase 2, and single stance phase 1. The transitions are listed in Chapter 3 but are quickly recounted here for completeness. The double stance phase 1 transitions to double stance phase 2 when the front ankle has bent to an allowable angle. The front ankle is considered rigid in the model but allowed to flex to account for some slight flexibility in the shoe and foot orthotic as well as some rolling of the shoe. The double stance phase 2 starts at the end of double stance phase 1 and transitions to single stance phase 1 when the rear knee comes in line with the hip joint and rear toe. This is the instant of maximum distance from the toe to the hip. Single stance phase 1 begins at this point and ends when the swing heel strikes impactless (with zero velocity in x and y) against the ground while simultaneously locking the swing knee with zero velocity. This is to avoid a knee slam and having to transition into single stance phase 2.

The step length was calculated by assuming that the front foot would be flat against the ground with the back knee straight at the end of double stance phase 2, as described in Chapter 2. The step length could be smaller than this and still satisfy these constraints if a slight ankle flexion is allowed similar to double stance phase 1. The step length could also be longer, but this would result in either the front foot not being in contact with the ground at the end of double stance (not a stable position) or the front knee not being completely straightened, requiring a transitional period of single stance phase 0. This case will not be considered.

The dynamics modeled in Chapter 3 are all accurate for continuous time. Unfortunately, continuous dynamic optimization is very difficult and extremely difficult to solve numerically with current methods [3]. The idea is to approximate these problems with similar systems either parametrically or by discretizing the dynamics into a finite number of discrete time steps.

In either case, the chosen cost function for the optimization problems is the 2-norm of the input torques squared over time, or

$$C = \int_{t_i}^{t_f} \tau(t) \cdot \tau(t) dt \tag{5.1}$$

where C is the cost, t_i is the initial time of integration, t_f is the final time of integration, and $\tau(t)$ is the input torques that are being minimized. This formulation should result in the minimal torque actuation necessary to locomote so long as the constraints and dynamics match those in Chapter 3. There are many possible cost functions to utilize in optimizing a gait [55], but Equation 5.1 was chosen to try to reduce the actuator size.

5.1 Parametric Optimization

Parametric optimization is a method that was first advanced by the space industry for trajectory simplified optimal trajectories [5]. The approach has since been generalized and applied to other fields [17, 19]. The basic idea is that the continuous optimization problem is very difficult to solve, and so the problem can be simplified by looking at a confined region of the solution space. For instance, the solution space could be constrained to be of a particular type, e.g. sum of sines, polynomial, exponential, etc. This allows for simulations that operate in continuous time but utilize solvers that operate on a finite number of variables [19]. Parametric optimization has found some recent uses in bipedal modeling [9].

Bessonnet et. al applied this method to bipedal walking models to find optimal gaits on a seven-link biped [37]. The biped operated under a two state finite state machine Lagrange multiplier constrained double stance and a free single stance with actuation at both ankles, knees, and hips. The walker is under a number of constraints including ground clearance and connectivity. Impacts can have a destabilizing effect on optimal solvers, so an approach using an impactless heel strike was developed [35]. Some preliminary results showed some promising simulations for how a bipedal model can take advantage of pendular dynamics similar to the Phoenix [36]. Bessonnet parameterized the gait of a fully-actuated 5-link bipedal walker through a quartic polynomial spline defined on angles as described in Chapter 1 [4]. This paper explores a similar method used on the five-link Phoenix exoskeleton model described in Chapter 3 with the only input being hip actuating torques.

Premise

The parametric optimization approach consists of finding the optimal, quartic polynomial spline trajectories (Appendix D) for each joint based on angle-defined nodes for each phase of the gait cycle, linking the phases together with closure constraints, and solving for the system as a whole. From these spline angle trajectories, the velocity and acceleration trajectories are trivial to solve. At this point, an inverse dynamics problem is formed, starting from trajectories and solving for input torques and constraint forces. Two approaches for solving the inverse dynamics were considered: a minimum norm solution [4] and using the pseudo Jacobian matrix [28]. These approaches provide a continuous function for the input torques and constraint forces based on the joint trajectories.

The pseudo-Jacobian inverse is defined by taking the general form of the system dynamics given in Equation 3.4 and the constraint equations such as Equation 3.5 or Equation 3.7 and explicitly solving for the constraint forces λ [28]. This is done by repeatedly taking the derivative of $\Phi(\bar{\theta})$.

$$\begin{aligned} \dot{\Phi} &= J(\bar{\theta})\dot{\bar{\theta}} = 0 \\ \ddot{\Phi} &= \dot{J}(\bar{\theta})\dot{\bar{\theta}} + J(\bar{\theta})\ddot{\bar{\theta}} = 0 \\ \Rightarrow & \dot{J}(\bar{\theta})\dot{\bar{\theta}} = -J(\bar{\theta})\ddot{\bar{\theta}} \end{aligned} \tag{5.2}$$

Now the dynamics in Equation 3.5 can be multiplied by JD^{-1} to get

$$-JD^{-1}\dot{J}\dot{\bar{\theta}} + JD^{-1}H(\bar{\theta})\dot{\bar{\theta}}^2 + JD^{-1}G(\bar{\theta}) = JD^{-1}\tau + \left(JD^{-1}J^T\right)\lambda \tag{5.3}$$

with substitution of Equation 5.2. The matrix $JD^{-1}J^T$ is called the pseudo-Jacobian [40]. The inertial matrix (D) is always invertible by definition, so the pseudo-Jacobian is also invertible so long as the Jacobian maintains full row rank. This equation can be solved simultaneously for λ and the dynamics of the system.

$$\begin{aligned} \left(JD^{-1}J^T\right)\lambda + JD^{-1}\tau &= JD^{-1}\left(-\dot{J}\dot{\bar{\theta}} + H(\bar{\theta})\dot{\bar{\theta}}^2 + G(\bar{\theta})\right) \\ \tau + J^T(\bar{\theta})\lambda &= D(\bar{\theta})\ddot{\bar{\theta}} + H(\bar{\theta})\dot{\bar{\theta}}^2 + G(\bar{\theta}) \end{aligned} \tag{5.4}$$

Bessonnett argues that a minimum norm solution is possible to find both the constraint forces (λ) and the input torques (τ) [4]. This leads to the simpler inverse dynamic formulation:

$$\begin{bmatrix} J^T & \mathbf{1} \end{bmatrix} \begin{bmatrix} \lambda \\ \tau \end{bmatrix} = D(\bar{\theta})\ddot{\bar{\theta}} + H(\bar{\theta})\dot{\bar{\theta}}^2 + G(\bar{\theta}) \tag{5.5}$$

Both these methods are nonlinear functions of the joint angle trajectories and their derivatives. Remember: the polynomial nodes are being optimized over. Therefore, it's entirely possible for the trajectories to violate the constraints, but by minimizing over both constraint forces and input torques, the constraints should be satisfied.

The integral defined in Equation 5.1 needs to be approximated to accommodate the nonlinear function of trajectories that is the inverse dynamics. This is possible by approximating the integral using MATLAB's *integral* function for each phase. The optimization problem can be easily solved by using MATLAB's *fmincon* function for nonlinear optimization, in particular the sequential quadratic programming (SQP) method [11]. The continuity constraints of the splines are implemented explicitly. The constraints of the exoskeleton system including step length, locked knees, closed chain, and between phase continuity constraints are all implemented as either linear or nonlinear constraints.

Optimization Results

Single Stance Phase 1

The parametric approach was applied to the single stance phase of the gait. The simulation assumed a step time of 0.6 seconds and three quartic polynomial spline nodes. Nonlinear optimization solvers are prone to get stuck in local optima, and the SQP solver is no exception. The solver was initialized by setting the initial guess to that of a linear blend between the nodes from the starting position to the ending position. The start and end positions can be solved explicitly based on the kinematic analysis in Chapter 3. The results from this simulation are shown in Figures 5.2 and 5.3.

The hip angles seem reasonable in comparison to the CGA data and the kinematic analysis. The swing hip quickly swings to a maximum angle allowing for maximum knee flexion and then slows the leg down for an impactless heel strike. The stance knee is constant, which is enforced through a Langrange multiplier while the stance hip moves nearly linearly towards the minimum, similar to the kinematic analysis. However, the torso angle seems to move over a large range, which is troubling. Many models consider many the torso angle to be constant during walking, and the CGA data largely agrees. It's reasonable to suspect that there would be a large torque cost associated with this.

The torque cost hypothesis turns out to be true. The torques are rather large by comparison to the CGA data. Winter's CGA data shows the torques relative to the mass of the subject and peaks at around 0.66 $\frac{\text{N}\cdot\text{m}}{\text{kg}}$ [52]. With the pilot weight of 80.7 kg which is roughly the average male weight in the United States, the optimization results in a peak torque of around 52.87 N·m. While the swing hip seems to be on a reasonable scale compared to CGA data, the stance hip is very far off. However, this could be due to the lack of other actuated joints. After all, the human body would be actuated at the knees, ankles, and hips also in

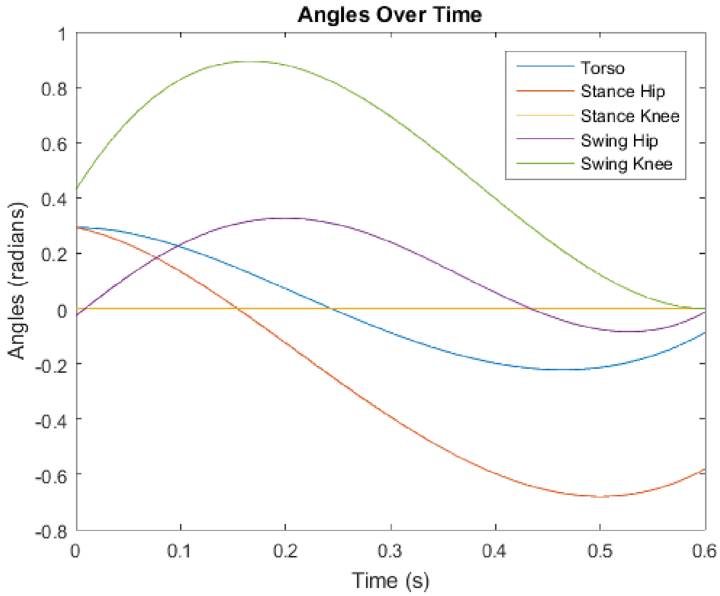


Figure 5.2: The joint angles of the parametric optimization of single stance phase 1.

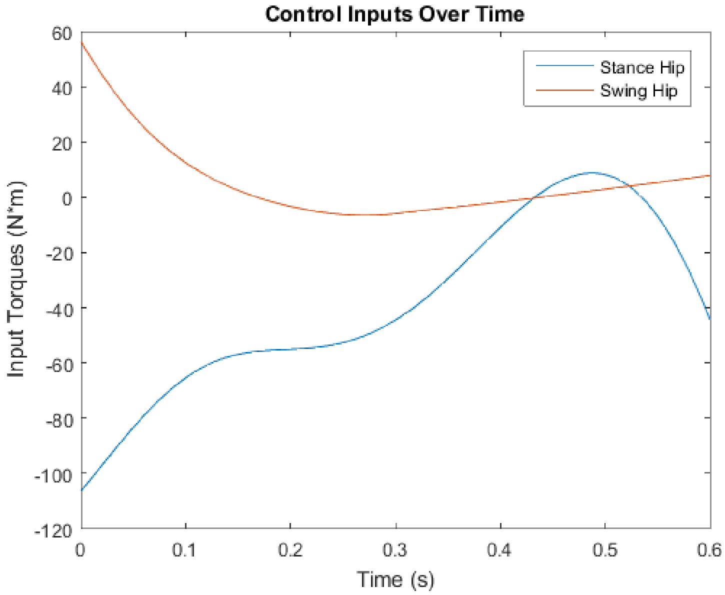


Figure 5.3: The hip torques of the parametric optimization of single stance phase 1.

the frontal plane. While only three nodes were used in this simulation, when more nodes were introduced, the trajectories and torques much wavier.

Double Stance Phase 2

Optimizing double stance phase 2 was attempted next, and the joint trajectories are shown in Figure 5.4. The simulation assumed a total time of 0.1 seconds spent in double stance phase 2 and three quartic spline nodes. Double stance phase 2 is more difficult to verify than single stance because it's very quick in a natural gait. The forward (swing) hip extends in order to straighten the knee at the end of the phase. This happens as expected and the forward knee straightens, ending with zero angular velocity. The stance hip barely flexes, which seems counter-intuitive, but there is a small sensitivity of the system to changes in the stance hip. This can also be seen in the double stance phase 1 later. The torso moves considerably, as in single stance phase 1. This is troublesome because one would expect large torques to result.

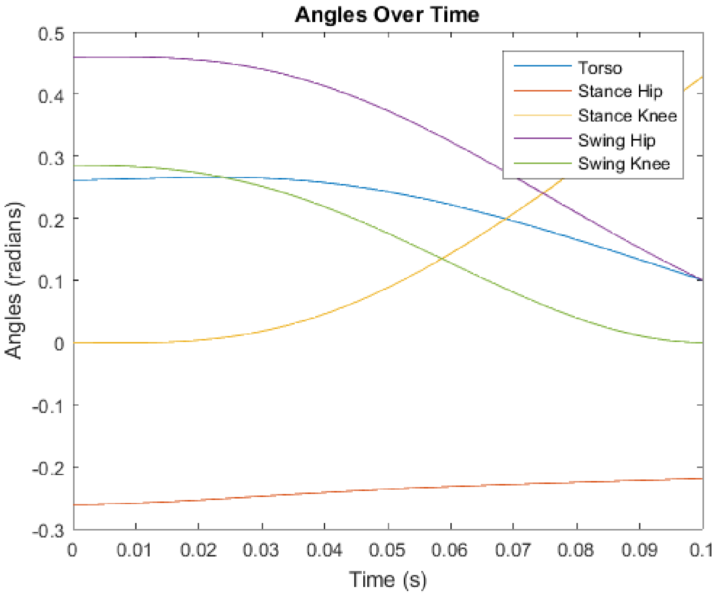


Figure 5.4: The joint angles of the parametric optimization of double stance phase 2.

In fact this is the case. The torques are even higher than in the single stance phase. These torques do not correlate at all with CGA data, and 350+ Nm is beyond the capability of any reasonably sized motor for an exoskeleton system. This result seems counter-intuitive since double stance has two fewer degrees of freedom. However, at least part of the problem is that a singularity occurs in the simulation that is not possible in reality. Notice in Figure 5.5 that there is a spike around 0.05 seconds.

This spike is also evident in the constraint forces shown in Figure 5.6. This singularity appeared in both inverse dynamics formulations and is due to both the pseudo-Jacobian and the minimum norm solution losing rank. The inertial matrix can't lose rank (and doesn't in the simulations as well) by definition.

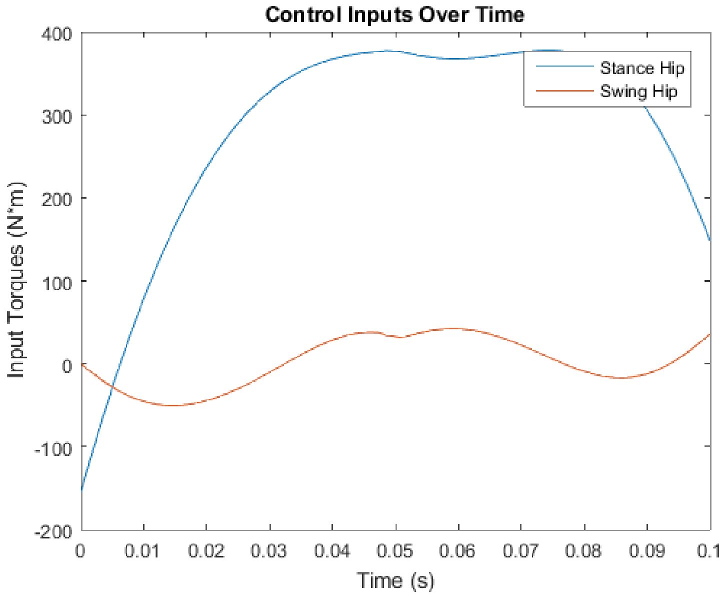


Figure 5.5: The hip torques of the parametric optimization of double stance phase 2.

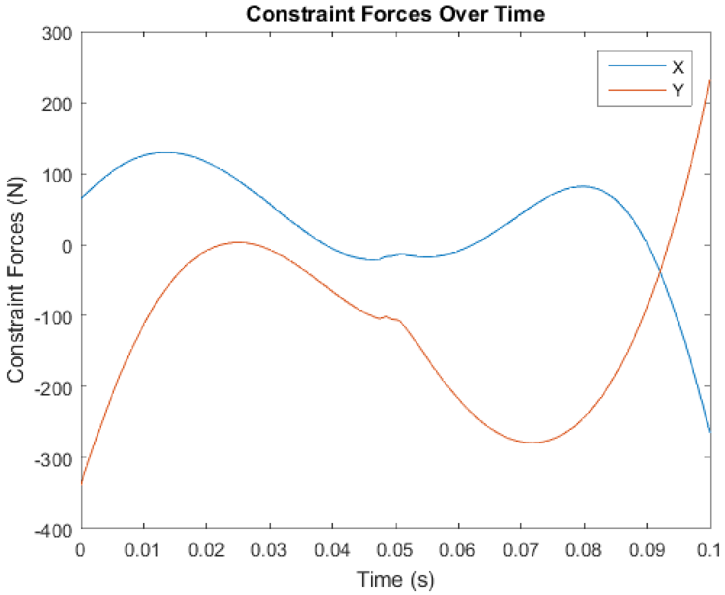


Figure 5.6: The ground reaction forces of the parametric optimization of double stance phase 2.

Conclusion

For very constrained dynamical systems like the Phoenix exoskeleton model, the quartic polynomial spline parametric trajectory optimization approach does not perform as expected.

This is partially due to the fact that the Lagrange multipliers do not work as well for constrained inverse dynamics. The trajectories generated by the polynomial splines are not kinematically constrained, and therefore can break the constraints. The trajectories could completely violate the constraints, which is not possible in a traditional Lagrange multiplier constrained dynamic system. Also, the singularities encountered during the inverse dynamics do not occur in the case of forward dynamic representations.

This formulation does not consider all the constraints as well as the dynamics. The forward dynamics consider both the constraints represented by both $\dot{\Phi}$ and $\ddot{\Phi}$ in Equation 5.2 and the constrained dynamics in Equation 3.4. In the minimum norm formulation, the constraints are not directly considered at all. In the pseudo-Jacobian formulation, the $\ddot{\Phi}$ constraint is considered but not the $\dot{\Phi}$. This results in torques that also do not produce the expected motion when plugged into the forward dynamics for simulating.

5.2 Discretized Optimization

Discrete optimization is a more typical approach for approximating continuous time optimal control problems. This is evident in much of the research on discretized finite time optimal controllers. The basic idea is to take a continuous time problem and use Euler discretization. That is:

$$\begin{aligned} \dot{x} &= f(x,u) \\ \dot{x} &\approx \frac{x(k+1)-x(k)}{\Delta t} \\ \Rightarrow x(k+1) &\approx f(x,u)\Delta t + x(k) \end{aligned} \tag{5.6}$$

where k is the current time step. The first equation describes the dynamics of the system. The second equation is the definition of a derivative if the limit as $\Delta t \to \infty$. Finally, the third equation is pure substitution. Applying this method to the dynamics in 3.4 yields:

$$\begin{aligned} &\bar{\theta}(k+1) = \dot{\bar{\theta}}(k)\Delta t + \bar{\theta}(k) \\ &\dot{\bar{\theta}}(k+1) = D(\bar{\theta}(k))^{-1}\left(\tau(k) + J^T(\bar{\theta}(k))\lambda(k) - H(\bar{\theta}(k))\dot{\bar{\theta}}^2(k) + G(\bar{\theta}(k))\right)\Delta t + \dot{\bar{\theta}}(k) \end{aligned} \tag{5.7}$$

Time discretization transforms a continuous differential equation into a simpler algebraic equation and transforms cost function integrals such as in Equation 5.1 into summations over a finite number of time steps. This approach allows for use of optimal solvers, as in the parametric approach.

Premise

Much of the mechanics for the optimization of the discrete time problem formulation are the same as the parametric problem formulation. However, the discrete approach uses the forward dynamics of the system, allowing the Lagrange multipliers to dynamically constrain the system properly. In the inverse dynamics case, the control inputs were solved at the same time as λ. Equation 5.2 still applies, but now λ and the next state state variables are solved together, meaning that the minimum norm and pseudo-Jacobian approaches are no longer valid. To solve the forward dynamics, the constraints in Equation 5.2 are considered directly. In matrix form this is:

$$\begin{bmatrix} I & 0 & 0 \\ 0 & D(\bar{\theta}(k)) & -J(\bar{\theta}(k)) \\ \dot{J}(\bar{\theta}(k)) & J(\bar{\theta}(k)) & 0 \\ J(\bar{\theta}(k)) & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\bar{\theta}}(k) \\ \ddot{\bar{\theta}}(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} \dot{\bar{\theta}}(k) \\ \tau(k) + J^T(\bar{\theta}(k))\lambda(k) - H(\bar{\theta}(k))\dot{\bar{\theta}}^2(k) + G(\bar{\theta}(k)) \\ 0 \\ 0 \end{bmatrix}$$
$$\begin{aligned} \bar{\theta}(k+1) &= \dot{\bar{\theta}}(k)\Delta t + \bar{\theta}(k) \\ \dot{\bar{\theta}}(k+1) &= \ddot{\bar{\theta}}(k)\Delta t + \dot{\bar{\theta}}(k) \end{aligned} \tag{5.8}$$

With these dynamic equations and no other constraints, the constraints laid out in Chapter 3 for each state will be enforced. For instance during double stance, the dynamic constraint that both feet must remain on the ground is not explicitly stated in a constraint equation. The Lagrange multiplier implicitly satisfies this constraint.

Again, MATLAB's *fmincon* SQP algorithm is used as the solver [11]. The torque cost function in Equation 5.1 is simplified for the discrete time case as

$$C = \sum_{i=1}^{N} \tau(i) \cdot \tau(i)\Delta t^2 \tag{5.9}$$

where N is the number of time steps in the simulation.

Results

The results for the discrete optimal gait generation strategies are discussed below.

Double Stance Phase 1

The double stance phase 1 was optimized assuming twenty time steps during this phase and a total phase time of 0.3 seconds. The gait was initialized assuming a linear blend from the start of the phase to the end with a constant velocity. The control inputs were all assumed to be 0. While the initial trajectory is not feasible either to the kinematics or dynamics, it should be close enough to get the optimization started.

The torque inputs for the optimization are given in Figure 5.7. These torques are substantially higher than the expected peak torque of around 52.87 N·m. This is due to the underactuation of the system. However, if the input of the pilot is considered, then perhaps the resulting torque could be decreased. Part of the objective of therapy is to encourage participation from the pilot. The pilot should not feel like he or she is simply riding the device.

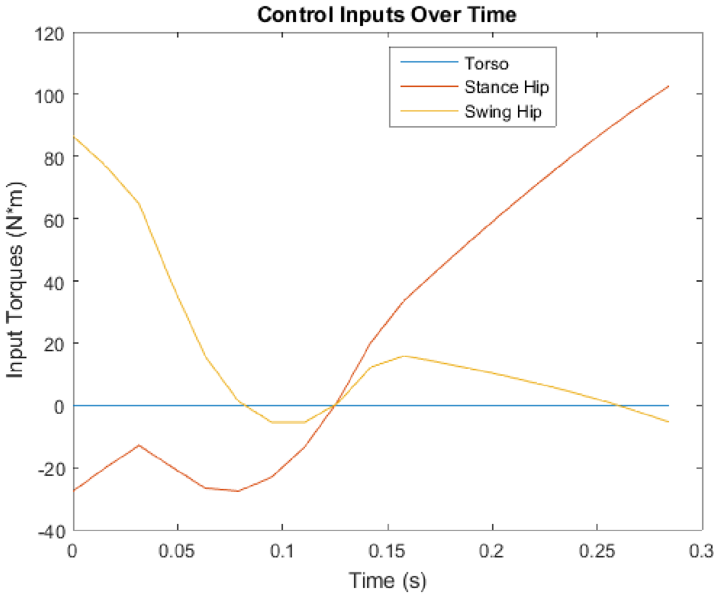


Figure 5.7: The joint torques of the discrete optimization of double stance phase 1 assuming only hip inputs.

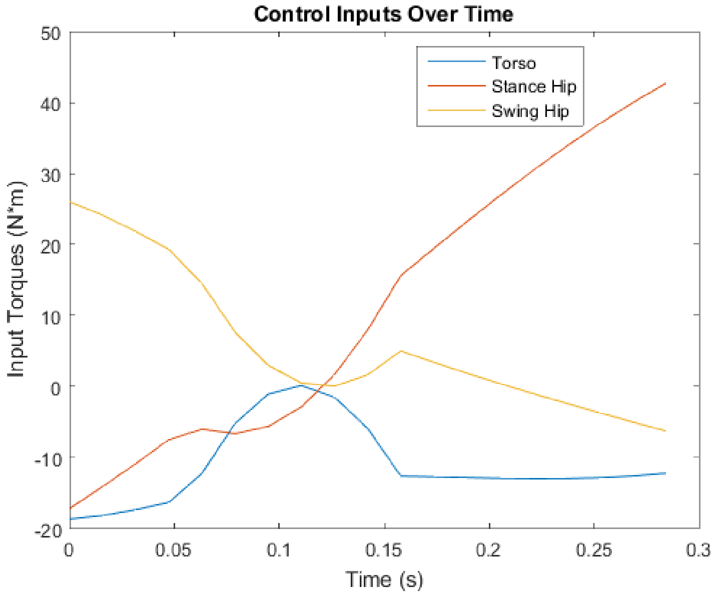


Figure 5.8: The joint torques of the discrete optimization of double stance phase 1 assuming hip and torso inputs.

For effective gait therapy, the device and pilot should collaborate to achieve motion. One way to model this is that the optimal trajectory minimizes both the input from the pilot and

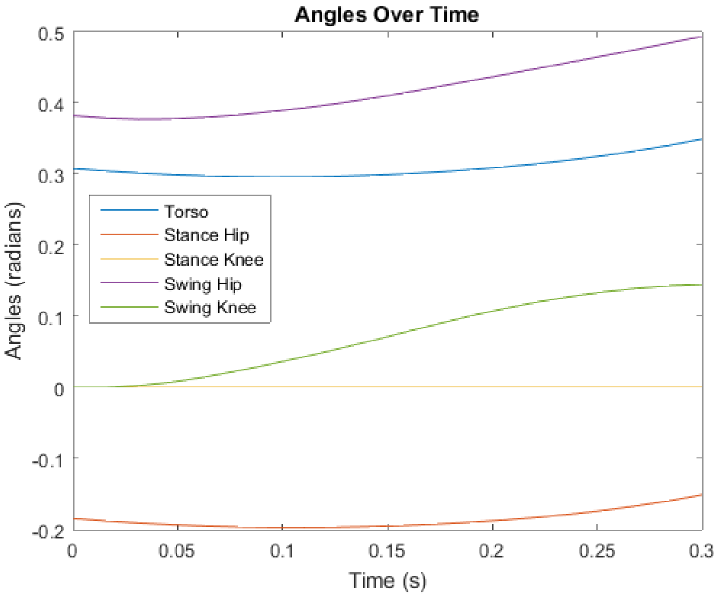


Figure 5.9: The joint angles of the discrete optimization of double stance phase 1 assuming hip and torso inputs.

the device. Using this paradigm with a paraplegic pilot, the pilot can provide input at the torso with abdominal muscles and weight-shifting with the use of crutches. This is modeled simply as a torque input at the torso, and the results are shown in Figures 5.8 and 5.9.

The angles from this simulation are shown in Figure 5.9. In this phase, the forward (swing) hip should flex allowing for the swing knee to flex and peak at the end of the phase. One would also expect that back (stance) hip to extend slightly, which is not observed in these results. This is due to the torso angle increasing during this phase. Remember: the hip angles are all relative to the torso angle. In this case, the thigh angle (absolute) would actually be extending, which is the expected behaviour. Large torso angle motions are not typical in a natural gait, but this could simply be an artifact of transitioning the weight forward in a limited actuation biped. The stance knee is held constant at zero, which is exactly as expected.

Double Stance Phase 2

The double stance phase 2 is optimized in a similar fashion to double stance phase 1. Twenty time steps with a total phase time of 0.3 seconds were used. The gait again assumed a linear blend from the start of the phase to the end with constant velocity. The start and end points are solved using the kinematic analysis in Chapter 3. The control inputs were all initialized to zero.

The angles generated by the optimization are shown in Figure 5.10. The graph shows the front (swing) hip extending back to being in line with the torso and knee with the center of

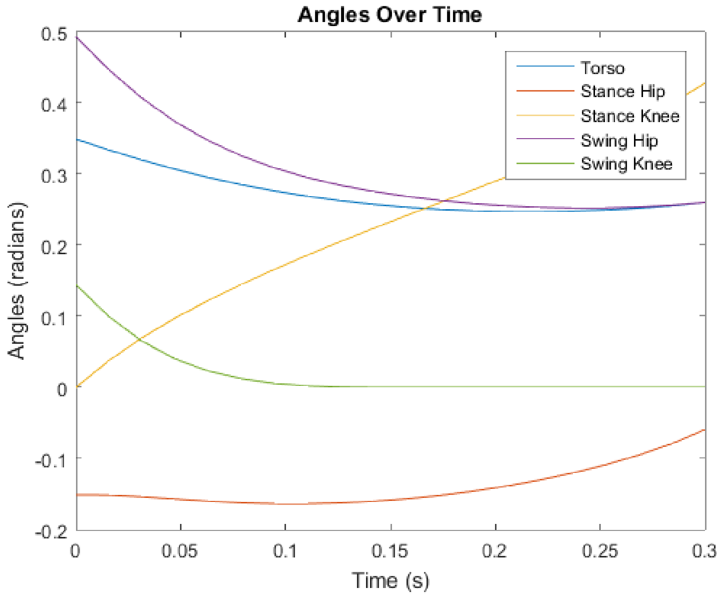


Figure 5.10: The joint angles of the discrete optimization of double stance phase 2 assuming hip and torso inputs.

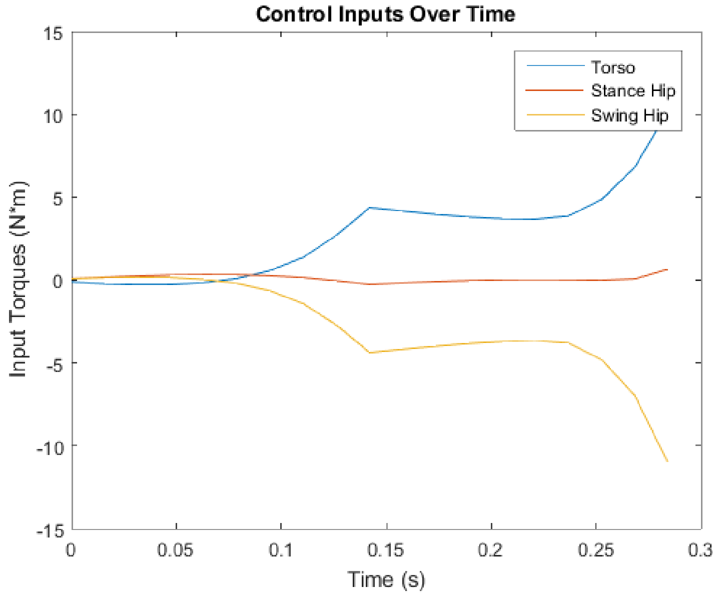


Figure 5.11: The joint torques of the discrete optimization of double stance phase 1 assuming hip and torso inputs.

mass over the front foot in preparation for the transition to single stance. The back (stance) hip extends and the back (stance) knee extends and accelerates to prepare for the pendular motion of single stance. The front (swing) knee comes to a stop when it reaches the zero position because it will lock and become the stance leg in the next phase.

Figure 5.11 show very small values for the hip torques. These torques are due to the carried momentum from double stance phase 1. Notice that the torques increase as some momentum is built up in anticipation for single stance. These low torques are also realistic in reality because the unlocked back knee is going to cause the back knee to bend due to gravity. This gravitational force will push the exoskeleton forward naturally without input. The rest is simply momentum.

These torques might be a bit low, but they would increase if the double stance phase 2 were shortened to spend less time in double stance. This is a balance that must be struck between speed, comfort, and the actuator limits. These actuator inputs are so low in comparison to the other phases that a speed up is probably necessary for this phase.

Single Stance Phase 1

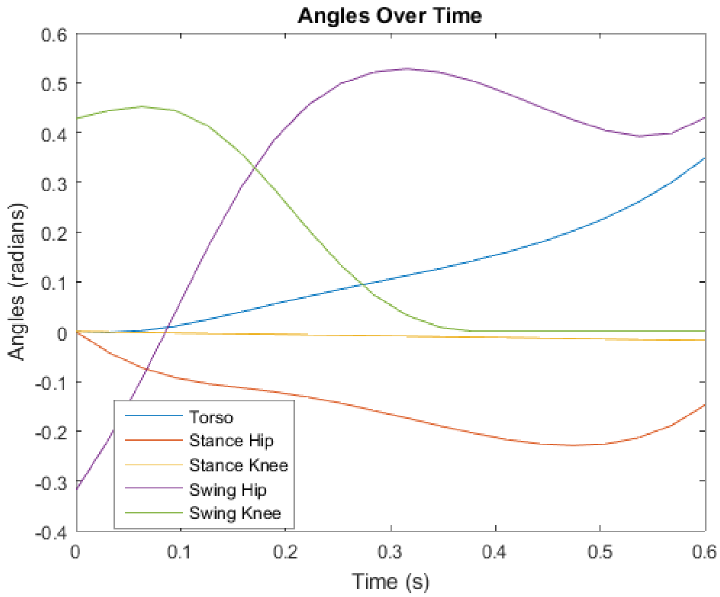


Figure 5.12: The joint angles of the discrete optimization of single stance phase 1 assuming hip and torso inputs.

Single stance phase 1 is optimized in much the same way as the previous two phases. The phase is broken up into twenty time steps and over a phase time of 0.7 seconds. The phase begins where double stance phase 2 ended and must place the swing heel on the ground with zero velocity in the x and y direction at the end of the phase to allow for an impactless heel strike. This ends where double stance phase 1 begins and the gait cycle repeats itself.

The angles for this gait phase are shown in Figure 5.12. The behaviour in single stance is somewhat similar to what was described in Chapter 3. The swing knee reaches a peak about

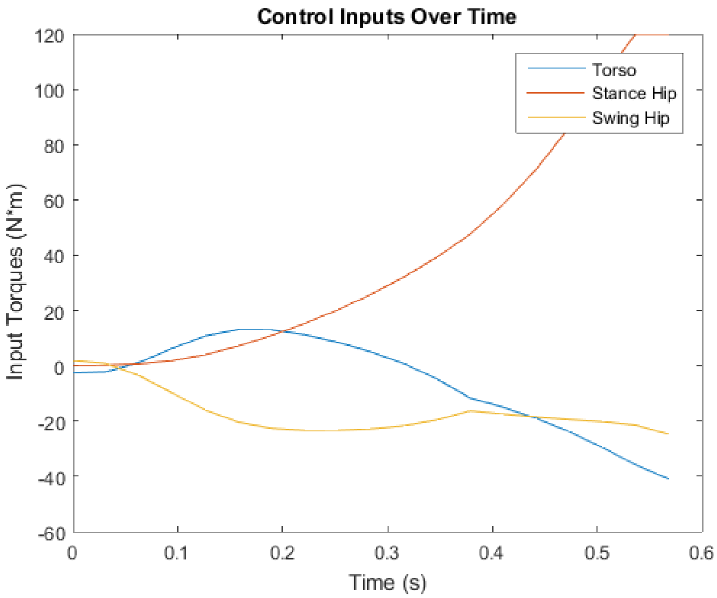


Figure 5.13: The joint torques of the discrete optimization of single stance phase 1 assuming hip and torso inputs.

where the angular velocity of the swing hip peaks and then comes down to approximately zero for the rest of the phase. The swing hip reaches a peak and then extends back to the start of double stance phase 1. The stance hip is similar to the linear hypothesis in Chapter 4], but starts to flex again towards the end. This is partially do to the extension of the torso angle at the end of single stance.

The same high torques are experienced in this phase as in the previous, even while considering the pilot's input as shown in Figure 5.13. The double stance phase 2 motion gets an assist from gravity when allowing the back (stance) knee to flex. The torques could be increased by decreasing the time spent in this state. With less time in the double stance phase 2 state, the motors would have to provide more input to compensate for less input from gravity.

5.3 Implementation

Developing optimal gaits serves a few purposes:

1. Understanding more about what is possible with system dynamics to inform design and control strategies.

2. Providing a baseline gait to be tuned from.

3. Providing gaits based on simple parameters.

While optimality is often not important, satisfying the constraints of an underactuated system is non-trivial. Studying the “best” gait possible is a good way to further understand the dynamics of the system. This enables comparisons of different designs based on what kinds of gaits can be produced. Furthermore, it enables analysis of state control decisions such as the locking and unlocking of the knee.

With gait development, it can be difficult to define where one should start. If the average pilot is placed in the device, what is the best gait to start with? A baseline gait can be developed for this scenario using an optimal approach, and then tuned by a gait practitioner for the pilot’s needs and comfort. Having a baseline gait that the practitioner knows *should* work with the pilot is a comforting place to start.

Finally, the final purpose to use optimal gaits is to provide a gait that is tailored to a particular pilot based on his or her parameters, such as weight and height. This method would allow for tuning of parameters such as step length and step time as well. Toe clearance tuning could also be introduced as an additional constraint. Introducing this method would greatly reduce the complexity for gait practitioners and increase the speed with which they could both provide therapy and tune a device for at-home use.

The computational power of the processors on the exoskeleton is extremely limited. Optimizing either in real-time or at initialization is not feasible. The Android device would also not have the resources necessary to optimize in a reasonable time frame. The most logical extension would be to have a cloud-based computing platform that optimizes for a given pilot and then communicates the result to the Android tablet. The gait would also have to be simplified before being streamed to the device for implementation in the gait. This is completely feasible using the splined methods described in Appendix D. The cloud server could simplify using spline before pushing the results to the Android and then to the exoskeleton. While the optimization would not be immediate, it could be greatly increased by having a bank of good initial guesses based on previous trials.

There are some obstacles in implementing optimal gait generation for exoskeletons. The first obstacle is that a solid infrastructure of cloud computing is needed. The strategies developed in this chapter would need to be ported to a platform like Amazon Web Services (AWS) that could handle the computation and communicate back to the tablet interface and back to the exoskeleton, as shown in Figure 5.14. Additionally, nonlinear optimization has a tendency to get stuck in local optima. A local optima would have the potential to be entirely different from the CGA gait and the gaits developed above. To avoid this problem, some safety checks would need to be implemented to verify that the gait generated by the optimal solver is actually a reasonable gait for the system. Some checks could be looking for a peak hip flexion motion, checking the actuator demands, and monitoring the accelerations and jerks of the device. While there are a few obstacles in the implementation of optimal gait generation for human exoskeletons, this is a promising area for further exploration and implementation in the future.

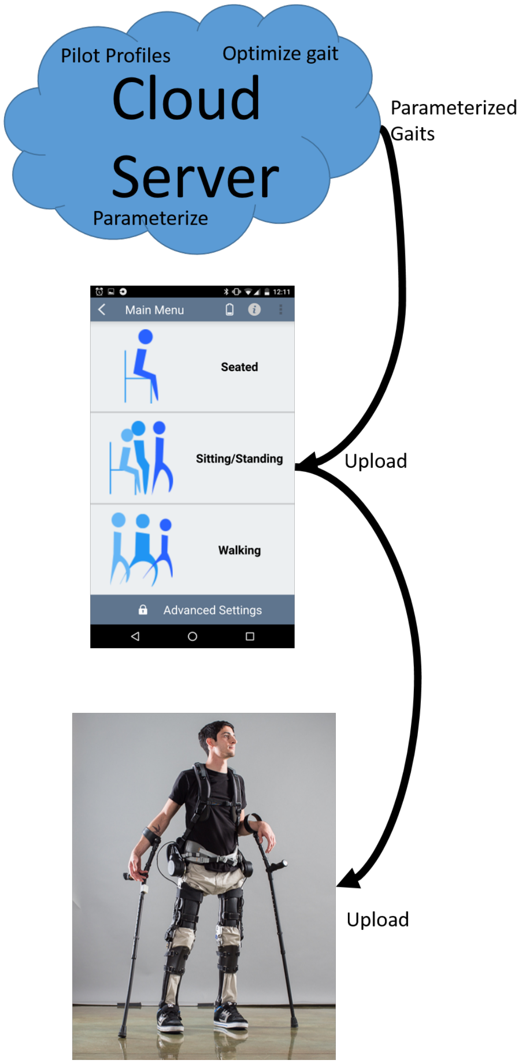


Figure 5.14: Optimal gait generation could be done in real-time through the use of a cloud server coupled with the Android app.

Chapter 6

Experimental Results

The tunable gait method discussed in Chapter 4 was applied to the Phoenix exoskeleton and evaluated in three ways:

1. Subjective feedback from the expert pilot
2. Data captured by the exoskeleton sensors
3. Point tracking data

The data indicates a much smoother gait than what was previously designed for similar exoskeletons [27, 32, 33].

The controllers used for the exoskeleton system are trajectory tracking Proportional-Derivative (PD) controllers for the active states and loosely regulating Proportional (P) controllers for the static states. The controller is selected depending on the state of the system in the finite state machine (Figure 4.8). Each leg has a separate controller, and the gains vary from state to state and depending on whether the controller is for the stance or swing leg. These controller gains were tuned experimentally by looking at tracking error and using the Android app interface to quickly adjust. The pilot's comfort was also taken into consideration during the tuning.

6.1 Pilot Feedback

The pilot noticed a significant improvement in the performance of the exoskeleton system when the tunable double stance gait was implemented. The pilot said that walking is much smoother with this gait, particularly in the transitional periods. The pilot was more balanced and stable in the device with the new gait and double stance. In fact, the pilot claimed that he needed less effort to both balance and propel himself. The device still requires side-to-side weight shifting input from the pilot, but the fore and aft weight shifting was drastically reduced. The device felt safer to the pilot in that the feet have better contact with the ground during double stance. The gait looks and feels more natural than the previous hip-only actuated exoskeleton gaits that largely ignored double stance. All in all, the pilot greatly appreciated the gait and felt safer, more stable, and more comfortable.

This gait enabled the pilot to walk continuously for the first time. In this case, continuous walking means that the pilot was able to continuously trigger steps without having to pause and reset his weight in between as in the previous gaits. This reduced the step time down to about 1 second for continuous walking.

6.2 Exoskeleton Data

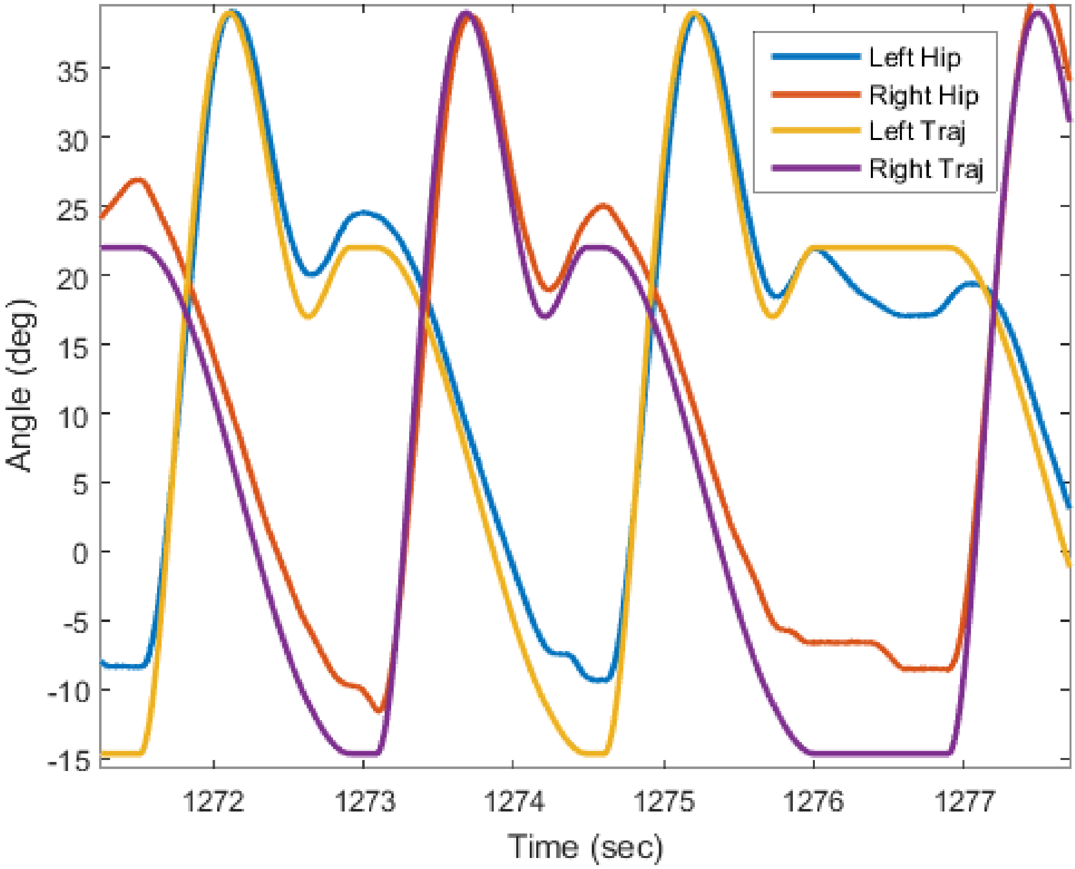


Figure 6.1: Hip data from the exoskeleton showing the controller's tracking performance.

For testing the device, debugging, and analysis, the Phoenix exoskeleton saves data at the control loop frequency from all the sensors, motors, and state information to a local SD card. The sensor data is all the torso-mounted IMU measurements and estimated Euler angles along with encoder angles for both hips. The state data shows the current state of the system as well the locked or unlocked state of the knees. The motor data is the torques that the hip motors are providing at the moment to control the hip angles. With this information, some performance of the exoskeleton for following the trajectories can be extracted.

Figure 6.1 shows a snapshot of the data collected from the exoskeleton during a trial in a lab setting. The green and yellow lines show the trajectories for the left and right legs respectively, while the purple and red lines show the actual hip angle. From the figure, it is clear that the trajectory tracking works quite well, at least on the swing leg. The stance leg tracks well but is not able to fully extend to what is commanded. This could be remedied by a stronger controller at the cost of a loss of comfort to the pilot or by tuning the gait further. The double stance sections of the gait show more variance than the single stance sections due to the uncertainty introduced by the knees.

Note the short flat periods in between steps. This flat spot is a brief pause in the static split stance state that the pilot takes before taking the next step. This pause allows for the pilot to reposition his or her crutch forward. Since continuous walking is possible with this gait, this repositioning is unnecessary. However, the pilot does not have to concentrate as much or transition his or her crutches so quickly if he or she allows himself or herself a brief pause. In order to increase the speed of this gait, the device must provide better stability to the pilot in the frontal plane while still allowing weight shifting.

Also, notice that the last period of static split stance in Figure 6.1 shows some significant deviations and motions away from the set point. This is due to the weaker controller provided during this static state. The pilot is comfortable in this state and is able to manipulate his or her body in the suit in order to prepare for the next step.

From the data collected from the exoskeleton, it appears that the device achieves what was desired:

1. Tight tracking of the locomotion state trajectories.

2. Looser regulating of the static states for pilot comfort.

3. A smooth and continuous gait that does not introduce jerks to the exoskeleton.

6.3 Point-Tracking Data

Figure 6.2 shows the point tracking software used to track stickers placed on the exoskeleton and pilot for external measurement of the joint angles. The point tracking software allows the operator to select an image mask that it then compares and tries to track as it moves through space [8]. The points can then be measured for angles using the software. These angles are exported to a tab delimited file which is then passed into MATLAB for post processing. The angles are adjusted to agree with the medical definitions defined in Chapter 3, converted to radians, and passed through a low pass filter with a relative frequency of 0.1. The filtering allows for a clean and offset-free dataset.

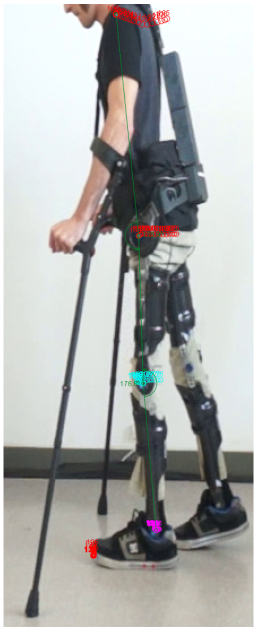


Figure 6.2: An image illustrating the use of the point tracking software.

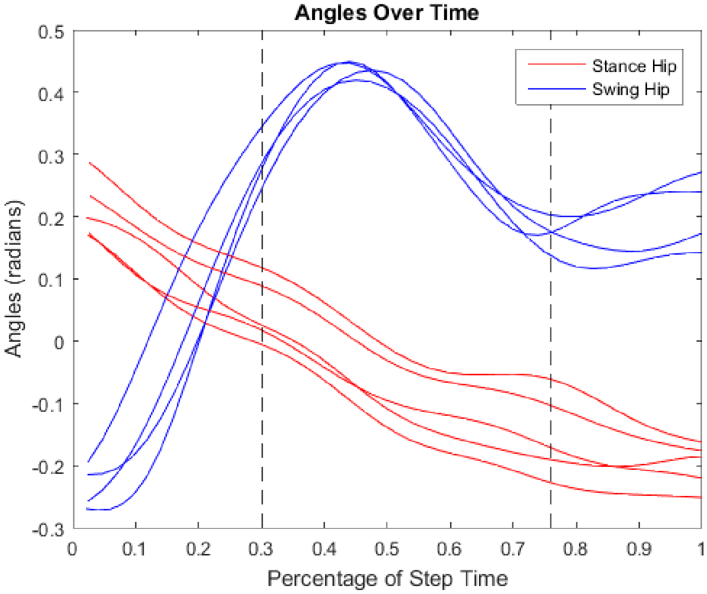


Figure 6.3: Hip Angles from several trials of the tuned trajectory for the expert pilot.

The hip angles from the experiments in the section above are also shown in Figure 6.3. The hip angles show a relatively small scattering. The line around 0.3 percent step time shows the approximate time of toe off; therefore, all the prior time is double stance phase 2. The line around 0.75 shows the approximate time of heel strike. While the trajectories look remarkably like those derived in the sections above, the proportion of time spent in double stance is rather high in comparison to the CGA data shown in Figure 4.2. In order for the walking speed to increase, the time spent in double stance must decrease. This means that the exoskeleton needs to provide additional stability for the pilot in order to go faster.

The stance hip angle shown in Figure 6.3 is roughly linear in single stance, as was prescribed Chapter 4. The linear stance hip motion is also in agreeance with the CGA data given in Figure 4.2. The swing hip shows a tight tracking of the maximum hip flexion both in angle and percent of the gait cycle in Figure 6.3. There is some variance in both hip angles in the transition to double stance phase 1 at roughly 75% of the step time. This is likely due to the role of uncertainty that the lack of knee control adds to the step length. If the heel strikes three inches back from the desired step length, then it could have large implications on the kinematics and dynamics of the system.

A similar plot for the knees is given in Figure 6.4. The knees exhibit some expected behaviour. The stance knee is locked throughout the entire single stance phase, as expected. The swing knee peaks around the point of maximum velocity for the swing hip. Double stance phase 2 shows the front knee locking while the back knee bends and accelerates to propel the leg through the pendular dynamics phase of single stance.

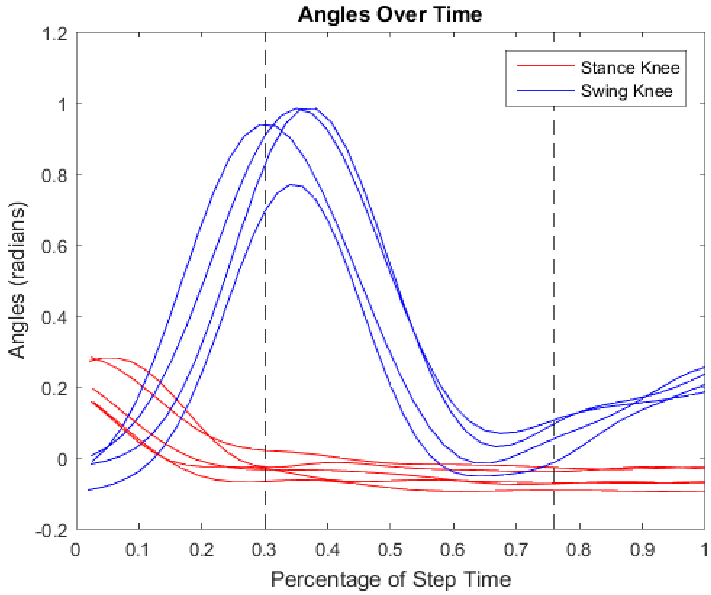


Figure 6.4: Knee Angles from several trials of the tuned trajectory for the expert pilot.

The knee data shown in Figure 6.4 shows a remarkable similarity to the “desired” knee trajectories that were designed in Figure 4.20. Because the desired knee trajectories were designed based on an understanding of the CGA data, the knee data also resembles the CGA data shown in Figure 4.3.

The point-tracking data solidifies that not only did the trajectories and controllers implement the desired motion on the hips, but the exoskeleton system matched the design of the gait.

6.4 Summary

This chapter presents the experimental results from testing the tunable gait method with a human test subject in the Phoenix exoskeleton. The results show that this gait is more comfortable and provides more stability for the user than the previous gaits that did not consider the smoothness or double stance in their development. The data collected by the exoskeleton demonstrates that the trajectories that were designed are actually being tracked with reasonable accuracy. The point-tracking results show that not only is exoskeleton tracking the trajectories, but the system is exhibiting the behaviour that was specified in Chapters 3 and 4. With these results, we can conclude that the tunable gaits are a viable solution and are ready to be implemented on exoskeletons such as the Phoenix immediately.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

This work illustrated how kinematic constraints can be utilized in a minimally actuated system to allow for smooth locomotion. The strategy of locking and unlocking the semi-passive knees proved to be very useful in manipulating the kinematic constraints. Two methods of gait development were shown: tunable gaits and optimal gaits. The tunable gait method gives absolute freedom to the gait practitioner to tune the device to the comfort and needs of the individual pilot. The optimal gait method attempts to take care of the majority of the complexities of gait generation.

The gaits developed were both smooth and continuous. This helps prevent sudden jerks from interfering with the operation of the device or causing discomfort. Pilot feedback showed that this style of gait generation is greatly preferred to more simplified approaches due to the increased stability, comfort, and step continuity.

The results from the tunable gait given in Chapter 6 are very promising in that they allowed a more natural gait that left the pilot feeling much safer and comfortable. Through the use of this gait, the pilot was able to use the exoskeleton in continuous walking for the first time. This was a breakthrough that allowed for much faster gaits than were achieved before.

7.2 Future Work

Further Simplified Gait Tuning

Future work should revolve around simplifying the tuning process for the gait practitioner. Currently, there are six parameters that tune the walking gait to each pilot. While the parameters are clear for gait practitioners, they are not clear to the average user. Simplifying these parameters to something that the pilot could understand would allow more freedom in at-home use. However, this adds the additional concern of guaranteeing safety for a tunable gait in the home.

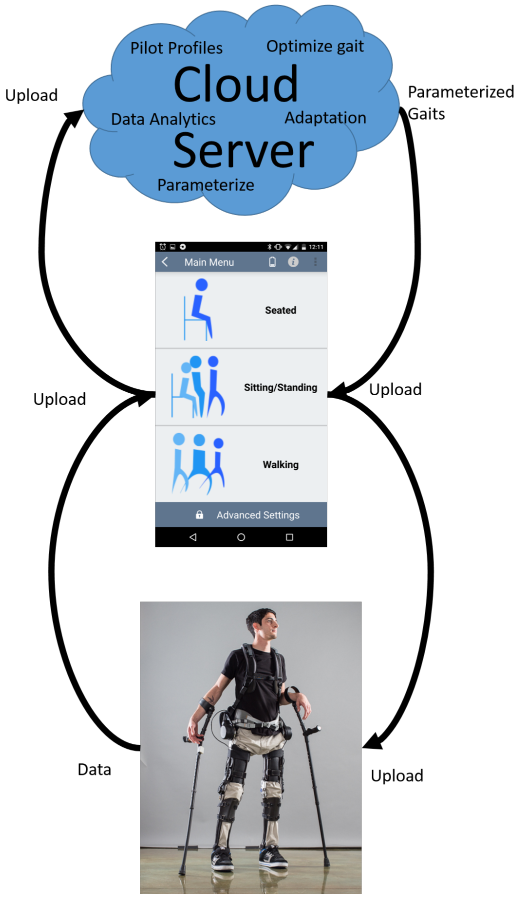


Figure 7.1: Optimal gait generation would better serve the pilot and gait therapist if data was fed back to the cloud server through the Android app. The data analytics and machine learning could help the exoskeleton adapt.

The optimal gait method could also be used in a cloud server configuration to design gait specifically for each pilot. This would avoid the problem of too many and too complex parameters for the average user to understand. In fact, with a few simple parameters like step time, step length, and toe clearance, a gait could be designed for each individual pilot. This would ideally reduce the workload of the gait practitioner, reduce the cost to the pilot, and empower the pilot to be more independent with their therapy. Gait practitioners are highly trained and very expensive specialists, often having PhD's in the area. The cost of gait therapy could be greatly reduced if the gait practitioner's role were reduced or eliminated.

In this endeavour, the exoskeleton must either make adaptations, should the gait need adjustments, or communicate with a cloud server which can then adjust the exoskeleton's gait after running some additional analysis. As discussed in Chapter 5, the cloud server would connect to the Android app and handle the higher computation cost tasks. Allowing the cloud server to adapt the gait or at least make recommendations to the gait practitioner would effectively close the loop on gait tuning. The cloud server could run optimization on an exoskeleton model and produce an optimal gait tailored to the pilots needs. Based on data from a few trials, the toe clearance could be increased through gait optimization parameters if the toe keeps scuffing the ground. The step length could be increased as the pilot's rehabilitation improves. The ultimate goal of exoskeletons should be a platform that is intelligent enough to recognize and adapt to the needs of its pilots.

Increased Stability

From the experimental testing, it is clear that the only way to increase the walking speed of the device is to decrease the time in double stance. There are two main portions of the double stance time: the designed double stance trajectory and the time spent in static split stance. The time spent in static split stance must decrease in order to increase the speed of the entire gait. This time is mostly spent by the pilot to help improve his or her stability and crutch placement before taking another step. The lack of stability is mostly in the frontal plane, rendering the hip actuators useless. In order to address this problem, a better crutch/stability mechanism must be developed or some control must be provided in the front plane. In the author's opinion, a better stability device would help improve the safety of the device and increase the walking speed while in it.

Appendix A

Exoskeleton Parameters Used in Simulation

The parameters chosen for the exoskeleton simulation were based on average human weight and height distributions per body segment [1]. This allows the simulations to be tunable to each pilot; all that is needed is the mass and height of the pilot. The average male human mass and height were used in all simulations in this paper (height= 1.75 meters, mass=80.7kg).

The proportions were taken from Orthopaedic Biomechanics: Mechanics and Design in Musculoskeletal Systems[1] pages 49-50. The parameters are illustrated in Figure A.1 (repeated Figure 3.1 for reference). They are summarized in Tables A.2 and A.1.

The center of mass proportions are all from the distal ends. The moments of inertia are all about the center of mass.

Body Part	Body	Foot	Leg Shank	Thigh	Torso
Proportion of Height	1	0.13	0.285	0.2450	0.470
Length (m)	1.75	0.2275	0.4987	0.4287	0.8225

Table A.1: Length Distribution Table

Body Part	Body	Leg Shank	Thigh	Torso
Proportion of Mass	1	0.061	0.1	0.678
Mass (kg)	80.7	4.9227	8.070	54.715
CoM Proportion of Length	N/a	0.394	0.567	0.626
CoM Distal Length (m)	N/a	0.197	0.243	0.515
Radius of Gyration (m)	N/a	0.416	0.323	0.496
Moment of Inertia (kg·m^2)	N/a	0.228	0.155	9.106

Table A.2: Mass, Center of Mass, and Moment of Inertia Distribution Table

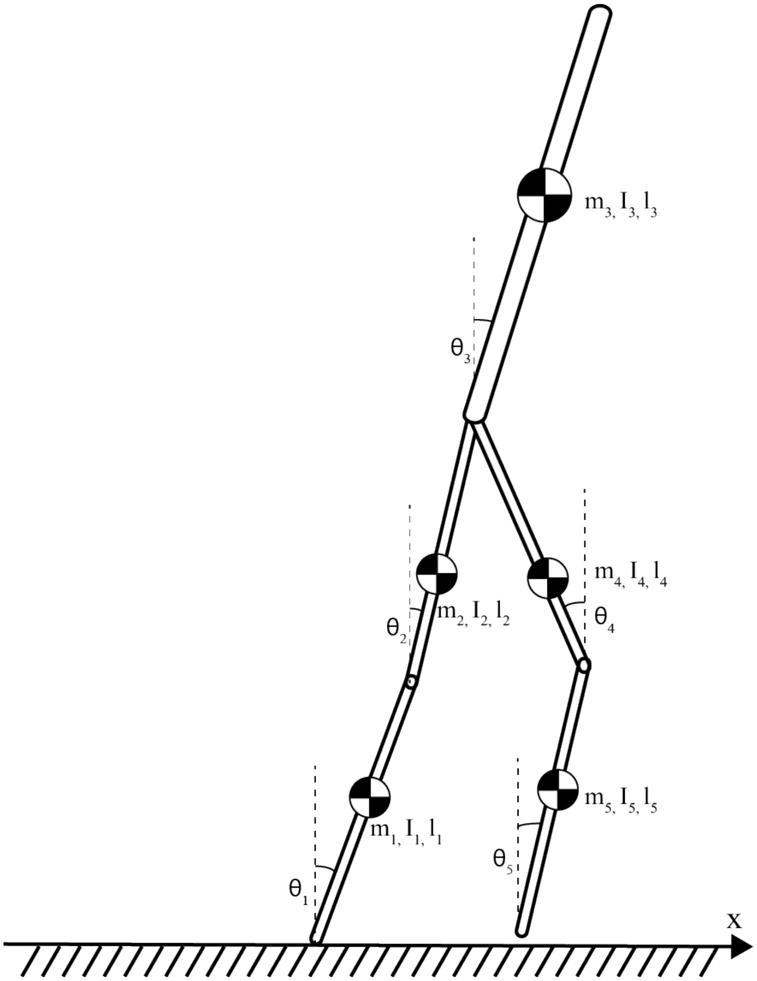


Figure A.1: The 5-link biped model during single stance with reference parameters.

Appendix B

Exoskeleton Dynamic Model Specifics

The matrices in the exoskeleton dynamics (Equation 3.3) are defined by Tzafestas [48]. The definitions are included here for completeness. Mu et al gives a more concise definition of these matrices [28]; however, an error was discovered. The lengthier Tzafestas definitions were used in all the dynamics in this paper.

The D matrix is defined as:

$$
\begin{aligned}
&D_{11} = I_1 + m_1 d_1^2 + (m_2 + m_3 + m_4 + m_5) l_1^2, \\
&D_{12} = p_{12} \cos(\theta_1 - \theta_2), \\
&D_{13} = p_{13} \cos(\theta_1 - \theta_3), \\
&D_{14} = p_{14} \cos(\theta_1 + \theta_4), \\
&D_{15} = p_{15} \cos(\theta_1 + \theta_5), \\
&D_{21} = D_{12}, \\
&D_{22} = I_2 + m_2 d_2^2 + (m_3 + m_4 + m_5) l_2^2, \\
&D_{23} = p_{23} \cos(\theta_2 - \theta_3), \\
&D_{24} = p_{24} \cos(\theta_2 + \theta_4), \\
&D_{25} = p_{25} \cos(\theta_2 + \theta_5), \\
&D_{31} = D_{13}, \;\; D_{32} = D_{23}, \\
&D_{33} = I_3 + m_3 d_3^2, \;\; D_{34} = D_{43} = D_{35} = D_{53} = 0, \\
&D_{41} = D_{14}, \;\; D_{42} = D_{24}, \\
&D_{44} = I_4 + m_4 (l_4 - d_4)^2 + m_5 l_4^2, \\
&D_{45} = p_{45} \cos(\theta_4 - \theta_5), \\
&D_{51} = D_{15}, \;\; D_{52} = D_{25}, \;\; D_{54} = D_{45}, \\
&D_{55} = I_5 + m_5 (l_5 - d_5)^2
\end{aligned}
\tag{B.1}
$$

where the p_{ij} are defined as:

$$\begin{aligned}
p_{12} &= m_2 d_2 l_1 + (m_3 + m_4 + m_5) l_1 l_2, \\
p_{13} &= m_3 l_1 d_3, \\
p_{14} &= m_4 l_1 (l_4 - d_4) + m_5 l_1 l_4, \\
p_{15} &= m_5 l_1 (l_5 - d_5), \\
p_{23} &= m_3 l_2 d_3, \\
p_{24} &= m_4 l_2 (l_4 - d_4) + m_5 l_2 l_4, \\
p_{25} &= m_5 l_2 (l_5 - d_5), \\
p_{45} &= m_5 l_4 (l_5 - d_5)
\end{aligned} \tag{B.2}$$

The p_{ij} values have the same units as the moments of inertia I_i. The H matrix is defined as:

$$\begin{aligned}
h_{12} &= p_{12} \sin(\theta_1 - \theta_2), \\
h_{13} &= p_{13} \sin(\theta_1 - \theta_3), \\
h_{14} &= -p_{14} \sin(\theta_1 + \theta_4), \\
h_{15} &= -p_{15} \sin(\theta_1 + \theta_5), \\
h_{21} &= -p_{12} \sin(\theta_1 - \theta_2), \\
h_{23} &= p_{23} \sin(\theta_2 - \theta_3), \\
h_{24} &= -p_{24} \sin(\theta_2 + \theta_4), \\
h_{25} &= -p_{25} \sin(\theta_2 + \theta_5), \\
h_{31} &= -p_{13} \sin(\theta_1 - \theta_3), \\
h_{32} &= -p_{23} \sin(\theta_2 - \theta_3), \\
h_{34} &= h_{43} = h_{35} = h_{53} = 0, \\
h_{41} &= -p_{14} \sin(\theta_1 + \theta_4), \\
h_{42} &= -p_{24} \sin(\theta_2 + \theta_4), \\
h_{45} &= p_{45} \sin(\theta_4 - \theta_5), \\
h_{51} &= -p_{15} \sin(\theta_1 + \theta_5), \\
h_{52} &= -p_{25} \sin(\theta_2 - \theta_5), \\
h_{54} &= -p_{45} \sin(\theta_4 - \theta_5),
\end{aligned} \tag{B.3}$$

The G matrix is defined as:

$$\begin{aligned}
G_1 &= -\left(m_1 d_1 + (m_2 + m_3 + m_4 + m_5) l_1\right) g \sin(\theta_1), \\
G_2 &= -\left(m_2 d_2 + (m_3 + m_4 + m_5) l_2\right) g \sin(\theta_2), \\
G_3 &= -\left(m_3 d_3\right) g \sin(\theta_3), \\
G_4 &= \left(m_4 (l_4 - d_4) + m_5 l_4\right) g \sin(\theta_4), \\
G_5 &= \left(m_5 (l_5 - d_5)\right) g \sin(\theta_5),
\end{aligned} \tag{B.4}$$

Appendix C

Four Bar Mechanism Kinematics

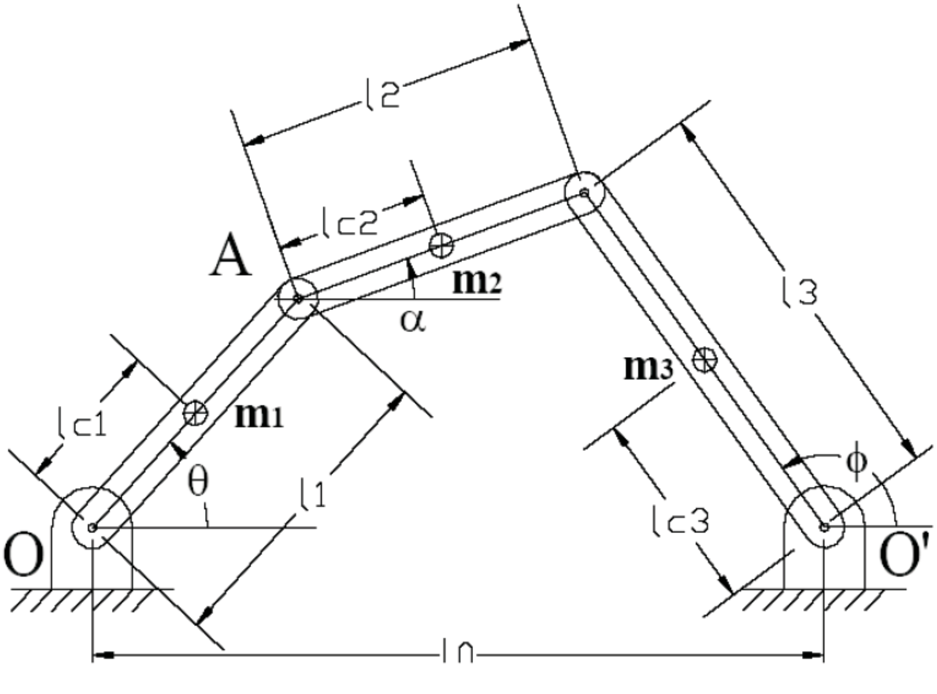


Figure C.1: Typical four-bar mechanism[45]

To solve for the leg joint angles for the exoskeleton based on the kinematic equations, Freudenstein's equation can be used [45]:

$$\begin{aligned} 0 &= k_1(\theta)\sin(\phi) + k_2(\theta)\cos(\phi) + k_3(\theta) \\ k_1(\theta) &= -2\ell_1\ell_3\sin(\theta) \\ k_2(\theta) &= 2\ell_3(\ell_0 - \ell_1\cos(\theta)) \\ k_3(\theta) &= \ell_0^2 + \ell_1^2 - \ell_2^2 + \ell_3^2 - 2\ell_0\ell_1\cos(\theta) \end{aligned} \tag{C.1}$$

where the ℓ's are defined in Figure C.1. This can be solved for ϕ and then α defined in Figure C.1.

$$\begin{aligned} \phi(\theta) &= 2\,\mathrm{atan2}\left(-k_1(\theta) \pm \sqrt{k_1(\theta)^2 + k_2(\theta)^2 - k_3(\theta)^2}, k_3(\theta) - k_2(\theta)\right) \\ \alpha(\theta, \phi) &= \mathrm{atan2}\left(-\ell_1 \sin(\theta) + \ell_3 \sin(\phi), \ell_0 - \ell_1 \cos(\theta) + \ell_3 \cos(\phi)\right) \end{aligned} \tag{C.2}$$

Now the challenge is to translate these equations into something that makes sense for the exoskeleton.

The transformation from the angles in Equation C.2 to the biomechanical angles defined in the sections above is as follows:

$$\begin{aligned} \theta_{\text{sw hip}} &= \theta_{\text{torso}} + \alpha + \frac{\pi}{2} \\ \theta_{\text{st hip}} &= \theta_{\text{torso}} + \theta - \frac{\pi}{2} - \mathrm{atan2}(l_{\text{foot}}, l_{\text{shank}} + l_{\text{thigh}}) \\ \theta_{\text{st knee}} &= 0 \\ \theta_{\text{sw knee}} &= \pi + \alpha - \phi \end{aligned} \tag{C.3}$$

As noted in the kinematics section, the rear (stance) knee is locked in double stance phase 1. The torso angle is free to constraints (besides the physical limitations of the pilot's body). Now the lengths need to be defined in terms of the exoskeleton gait:

$$\begin{aligned} \ell_0 &= L_{\text{step}} \\ \ell_1 &= \sqrt{l_{\text{foot}}^2 + (l_{\text{shank}} + l_{\text{thigh}})^2} \\ \ell_2 &= l_{\text{thigh}} \\ \ell_3 &= l_{\text{shank}} \end{aligned} \tag{C.4}$$

The ℓ_1 is the line from the back (stance) toe to the hip. This is solved using the Pythagorean theorem. All the parameters on the right side of Equation C.4 are defined in Appendix A.

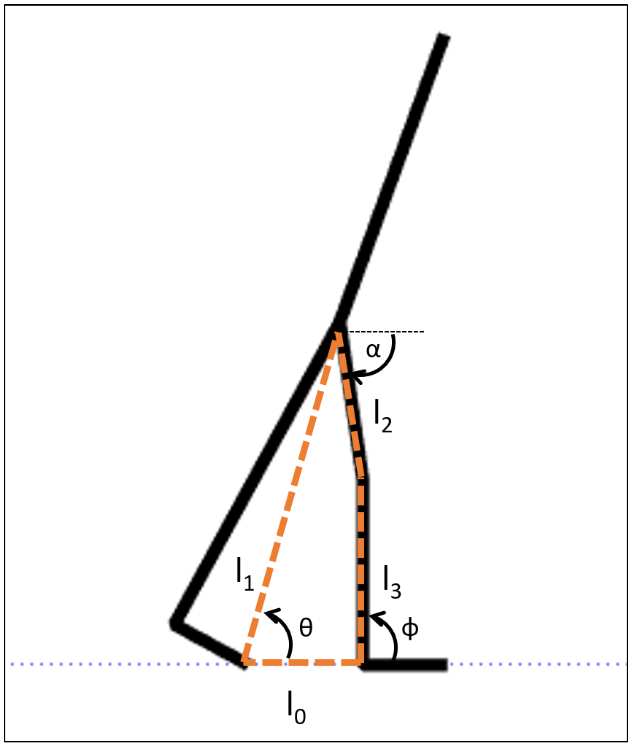


Figure C.2: Illustrating the connection between the four bar mechanism in Figure C.1 and the exoskeleton in Figure 4.17.

Appendix D

Spline Trajectories

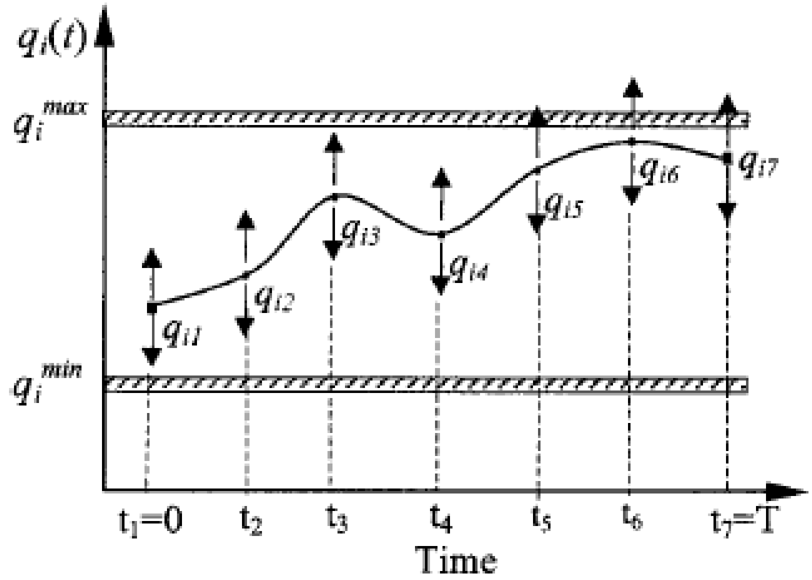


Figure D.1: An example of a spline trajectory.

The software on the exoskeleton needs to allow for the gaits to be quickly updated so that the practitioner can be agile in their approach to administering customized therapy. To allow for fast gait tuning, transferring as few parameters as possible from the clinician to the trajectory generator on the exoskeleton is necessary. The Phoenix exoskeleton operates at a control loop frequency of 1 kHz (0.001 seconds per cycle). With a step time of approximately one second, the trajectory generator needs to produce a trajectory with roughly 1,000 points. There are 8 states in the exoskeleton's finite state machine that require a trajectory; so then if each individual trajectory point were transfered from the clinical technician interface to the exoskeleton, it would need to transfer about 7000 data points or 28 kB of data. This type of data transfer is not feasible for this device to update in real-time given the bandwidth limitations of communication protocols such as Bluetooth and computational power of embedded processors.

To satisfy the limitations of the communication bandwidth, the trajectory has to be built by a trajectory generator based on parameters. These parameters are used to represent the gait as a whole. Now the issue is that the gait must be computationally inexpensive to generate. The microprocessors in the Phoenix operate at 96 MHz clock rate which is adequate for some simple calculations at a control frequency of 1 kHz but not fast enough to calculate all the trajectories within one control loop. Therefore, the gait must be generated one step at a time and not calculated and stored locally.

The gait must be smooth and continuous such that sudden jerks are not introduced in the trajectory tracking control of the device. To enforce these constraints, a quartic and cubic spline-based trajectory scheme was chosen. A spline is numeric function that is defined by piecewise polynomials. An n-order spline has the property that it is C^{n-1} continuous (one less than the degree of the spline). Meaning, cubic and quartic splines are continuous up to C2 and C3 respectively. We will be considering only quartic splines with the implied condition that cubic splines will simply have the highest order coefficient be 0.

Splines are controlled by nodes or handles. These handles are like waypoints for the trajectory to follow. They can be as simple as a position or as complex as a position, velocity, and acceleration at a particular time. Figure D.1 shows a spline trajectory with some angle handles. These nodes are the parameters that are used to both fully define the trajectory and are the tools that are used to tune it.

Angle-Defined Splines

An example quartic polynomial spline is given in Equation D.1.

$$\theta_{ji}(\tau_i(t)) = \phi_{ji0} + \phi_{ji1}\tau_i + \phi_{ji2}\tau_i^2 + \phi_{ji3}\tau_i^3 + \phi_{ji4}\tau_i^4 \tag{D.1}$$

where τ is the normalized time, j is the angle number (such as 0 for the swing hip angle), i is the polynomial number (such as 0 for the first polynomial of that spline trajectory), and the final numbered subscript of ϕ is the ϕ number for differentiating each ϕ. Each spline trajectory is made up of a number of piecewise polynomials; these polynomials are differentiated by the polynomial number. An example In Equation D.1, θ_{00} could refer to the swing hip angle (if 0 is associated with that angle) and the first spline in that trajectory

Each polynomial in the spline trajectory begins with $\tau = 0$ and ends with $\tau = 1$. The transformation from regular time to τ is given by:

$$\tau_i(t) = \frac{t_i - t_{i\text{start}}}{t_{i\text{end}} - t_{i\text{start}}} \tag{D.2}$$

where t is the time since the start of the spline trajectory, i is the polynomial number, "start" designates the starting time of the polynomial spline, and "end" designates the end time of the polynomial spline.

Quartic splines are a piecewise combination of quartic polynomials with a set of constraints that guarantee smoothness. Constraints must be introduced to enforce this continuity and smoothness with both the next and previous polynomials in each spline trajectory. Note: There are five coefficients that need to be solved for each polynomial; therefore, there must be five constraints per polynomial in the spline trajectory. Equation D.3 gives an example of these constraints for the a polynomial spline trajectory.

$$\begin{aligned}
\theta_{01}(0) &= \theta_{00}(1) \\
\dot{\theta}_{01}(0) &= \dot{\theta}_{00}(1) \\
\ddot{\theta}_{01}(0) &= \ddot{\theta}_{00}(1) \\
\dddot{\theta}_{01}(0) &= \dddot{\theta}_{00}(1) \\
\theta_{02}(0) &= \theta_{01}(1) \\
\dot{\theta}_{02}(0) &= \dot{\theta}_{01}(1) \\
\ddot{\theta}_{02}(0) &= \ddot{\theta}_{01}(1) \\
\dddot{\theta}_{02}(0) &= \dddot{\theta}_{01}(1)
\end{aligned} \tag{D.3}$$

These constraints are simply that the middle polynomial must have continuous angular position, velocity, acceleration, and jerk with the polynomials immediately before and after it. These constraints provide a no jerk solution, which yields an input torque without sudden shifts.

For efficiency, these constraints should be put in terms of the ϕ's by plugging in the appropriate τ, e.g. $\theta_{01}(0) = \phi_{010}$. To do this for the derivatives, the derivatives of τ are needed with respect to t for the chain rule, e.g. $\frac{d\theta}{dt} = \frac{d\theta}{d\tau}\frac{d\tau}{dt}$.

$$\frac{d\tau}{dt} = \frac{1}{t_{i\text{end}} - t_{i\text{start}}} = \frac{1}{\Delta t_i} \tag{D.4}$$

Using the product rule with Equations D.1 and D.4, yields the following for the first derivative of θ (angular velocity):

$$\begin{aligned}
\frac{d\theta}{dt} &= \frac{d\tau}{dt}\frac{\delta\theta}{\delta\tau} \\
&= \frac{1}{\Delta t_i}(\phi_{001} + 2\phi_{002}\tau + 3\phi_{003}\tau^2 + 4\phi_{004}\tau^3)
\end{aligned} \tag{D.5}$$

Now the second derivative of θ (angular acceleration)can be taken by using the product rule:

$$\begin{aligned}
\frac{d^2\theta}{dt^2} &= \frac{d}{dt}\left(\frac{d\tau}{dt}\frac{\delta\theta}{\delta\tau}\right) \\
&= \frac{d^2\tau}{dt^2}\frac{\delta\theta}{\delta\tau} + \frac{d\tau}{dt}\frac{\delta^2\theta}{\delta\tau^2} \\
&= 0 \cdot \frac{\delta\theta}{\delta\tau} + \frac{d\tau}{dt}\frac{\delta^2\theta}{\delta\tau^2} \\
&= \frac{1}{\Delta t_i}(2\phi_{002} + 6\phi_{003}\tau + 12\phi_{004}\tau^2)
\end{aligned} \tag{D.6}$$

Finally the third derivative (angular jerk) can be found using the product rule again:

$$\begin{aligned}\frac{d^3\theta}{dt^3} &= \frac{d}{dt}\left(\frac{d\tau}{dt}\frac{\delta^2\theta}{\delta\tau^2}\right)\\ &= \frac{d^2\tau}{dt^2}\frac{\delta^2\theta}{\delta\tau^2} + \frac{d\tau}{dt}\frac{\delta^3\theta}{\delta\tau^3}\\ &= 0\cdot\frac{\delta^2\theta}{\delta\tau^2} + \frac{d\tau}{dt}\frac{\delta^3\theta}{\delta\tau^3}\\ &= \frac{1}{\Delta t_i}(6\phi_{003} + 24\phi_{004}\tau)\end{aligned} \tag{D.7}$$

Now these equations can be plugged into the constraint equations (Equation D.3) and solved for in terms of the ϕ's. Using the chain rule,

$$\begin{aligned}\phi_{010} &= \phi_{000} + \phi_{001} + \phi_{002} + \phi_{003} + \phi_{004}\\ \phi_{011}\frac{1}{\Delta t_1} &= \frac{1}{\Delta t_0}(\phi_{001} + 2\phi_{002} + 3\phi_{003} + 4\phi_{004})\\ \phi_{012}\frac{1}{\Delta t_1} &= \frac{1}{\Delta t_0}(2\phi_{002} + 6\phi_{003} + 12\phi_{004})\\ \phi_{013}\frac{1}{\Delta t_1} &= \frac{1}{\Delta t_0}(6\phi_{003} + 24\phi_{004})\\ \phi_{020} &= \phi_{010} + \phi_{011} + \phi_{012} + \phi_{013} + \phi_{014}\\ \phi_{021}\frac{1}{\Delta t_2} &= \frac{1}{\Delta t_1}(\phi_{011} + 2\phi_{012} + 3\phi_{013} + 4\phi_{014})\\ \phi_{022}\frac{1}{\Delta t_2} &= \frac{1}{\Delta t_1}(2\phi_{012} + 6\phi_{013} + 12\phi_{014})\\ \phi_{023}\frac{1}{\Delta t_2} &= \frac{1}{\Delta t_1}(6\phi_{013} + 24\phi_{014})\end{aligned} \tag{D.8}$$

Therefore at each interior node of the spline, there are 4 connectivity constraints. Only one further constraint is needed to fully define the spline at the interior node: a node constraint. Each node of the spline requires an angle per the spline's design, in this case, these nodes define a critical point of the trajectory.

Additional constraints are still required for the exterior nodes (since the interior are already full defined). To this end, a velocity constraint can be applied to each start and end node, setting the velocity to a particular value at that point. One final constraint is required to fully define the set of splines. For example: using a two polynomial (three node) quartic spline, there will need to be 10 constraints (five for each polynomial). Four are taken care of by the connectivity constraints at the interior node. Three are taken care of by the node-angle constraints. Two are taken care of by the start and end point velocity constraints. The last spline is constrained to be cubic instead of quartic to eliminate this unnecessary free parameter.

The constraints in Equation D.8 can be combined with the node constraints into a linear system.

$$M_c\phi = M_x x \tag{D.9}$$

where x are the node definitions, Mx relates the node definitions to ϕ, and M_c relates ϕ's to themselves (as in Equation D.8) and to x. The spline coefficients, ϕ, can be solved simply by inverting M_c since it's always full rank [1]:

$$\phi = M_c^{-1} M_x x \tag{D.10}$$

Using this relationship, ϕ can be quickly solved, even when the number of nodes is large. The ϕ's must be solved for in order to ultimately generate the trajectories in real-time.

Angle- and Velocity-Defined Splines

Besides using a purely angle-based approach of defining splines, the velocity as well as the angle can be defined at each node, similar to the start and end points in the angle-defined spline. To free up parameters for velocity definitions at the nodes, the continuous jerk constraint must be dropped. This could lead to sudden jerks in the input torque but allows for added control of the spline.

Redefining the spline, the constraints are:

$$\begin{aligned}
\theta_{01}(0) &= \theta_{00}(1)\\
\dot{\theta}_{01}(0) &= \dot{\theta}_{00}(1)\\
\ddot{\theta}_{01}(0) &= \ddot{\theta}_{00}(1)
\end{aligned} \tag{D.11}$$

for each interior node. Note: the jerk constraint was removed. Now using the chain rule and the derivatives of τ defined in Equation D.4, Equation D.11 becomes:

$$\begin{aligned}
\phi_{010} &= \phi_{000} + \phi_{001} + \phi_{002} + \phi_{003} + \phi_{004}\\
\phi_{011}\frac{1}{\Delta t_1} &= \frac{1}{\Delta t_0}(\phi_{001} + 2\phi_{002} + 3\phi_{003} + 4\phi_{004})\\
\phi_{012}\frac{1}{\Delta t_1} &= \frac{1}{\Delta t_0}(2\phi_{002} + 6\phi_{003} + 12\phi_{004})
\end{aligned} \tag{D.12}$$

Again, some additional constraints are needed. In fact, if the same end point and node constraints (node position, velocity, and final cubic polynomial constraints) are used, only one new constraint is needed per interior node. This can be satisfied perfectly by defining a velocity for each interior node as well as the start and end nodes. This yields a fully defined spline.

A similar approach as the section above can be used to generate the ϕ's from the nodes.

[1] The full rank is by design. We purposefully chose linearly independent connectivity and node constraints such that this linear system would always be solvable.

Bibliography

[1] Donald L Bartel and Dwight T Davy. *Orthopaedic biomechanics: mechanics and design in musculoskeletal systems.* Prentice Hall, 2006.

[2] Katharine J Bell et al. “Natural progression of gait in children with cerebral palsy”. In: *Journal of Pediatric Orthopaedics* 22.5 (2002), pp. 677–682.

[3] Guy Bessonnet, Stephane Chesse, and Philippe Sardain. “Optimal gait synthesis of a seven-link planar biped”. In: *The International journal of robotics research* 23.10-11 (2004), pp. 1059–1073.

[4] Guy Bessonnet, Pascal Seguin, and Philippe Sardain. “A parametric optimization approach to walking pattern synthesis”. In: *The International Journal of Robotics Research* 24.7 (2005), pp. 523–536.

[5] John T Betts. “Survey of numerical methods for trajectory optimization”. In: *Journal of guidance, control, and dynamics* 21.2 (1998), pp. 193–207.

[6] Joaquin A Blaya and Hugh Herr. “Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait”. In: *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 12.1 (2004), pp. 24–31.

[7] Ingo Borggraefe et al. “Robotic-assisted treadmill therapy improves walking and standing performance in children and adolescents with cerebral palsy”. In: *european journal of paediatric neurology* 14.6 (2010), pp. 496–502.

[8] Douglas Brown. *Biomedical Data Resources.* 2015. URL: `http://physlets.org/tracker/`.

[9] Christine Chevallereau et al. *Bipedal robots: Modeling, design and walking synthesis.* John Wiley & Sons, 2013.

[10] DB Clement, JE Taunton, and GW Smart. “Achilles tendinitis and peritendinitis: etiology and treatment”. In: *The American Journal of Sports Medicine* 12.3 (1984), pp. 179–184.

[11] Thomas F. Coleman and Yin Zhang. *Optimization Toolbox: User’s Guide.* MATLAB. 2016.

[12] Aaron M Dollar and Hugh Herr. "Lower extremity exoskeletons and active orthoses: challenges and state-of-the-art". In: *Robotics, IEEE Transactions on* 24.1 (2008), pp. 144–158.

[13] Romildo Don et al. "Foot drop and plantar flexion failure determine different gait strategies in Charcot-Marie-Tooth patients". In: *Clinical biomechanics* 22.8 (2007), pp. 905–916.

[14] Alberto Esquenazi et al. "The ReWalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury". In: *American journal of physical medicine & rehabilitation* 91.11 (2012), pp. 911–921.

[15] Pea Frey-Rindova et al. "Bone mineral density in upper and lower extremities during 12 months after spinal cord injury measured by peripheral quantitative computed tomography". In: *Spinal cord* 38.1 (2000), pp. 26–32.

[16] JR Gage. "Gait analysis. An essential tool in the treatment of cerebral palsy." In: *Clinical orthopaedics and related research* 288 (1993), pp. 126–134.

[17] Abhijit Gosavi. *Simulation-based optimization: parametric optimization techniques and reinforcement learning*. Vol. 55. Springer, 2014.

[18] Cincinnati Children's Hospital. *Gift Provides First Steps*. URL: `http://www.cincinnatichildrens.org/giving/how-you-help/stories/ftc/archives/spring-2012/lokomat/`.

[19] David G Hull. "Conversion of optimal control problems into parameter optimization problems". In: *Journal of Guidance, Control, and Dynamics* 20.1 (1997), pp. 57–60.

[20] D Intiso et al. "Rehabilitation of walking with electromyographic biofeedback in foot-drop after stroke." In: *Stroke* 25.6 (1994), pp. 1189–1192.

[21] Yoon Jung Jeong and Homayoon Kazerooni. "Design of Low Profile Actuators for Medical Exoskeletons". In: *ASME 2015 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers. 2015, V003T03A094–V003T03A094.

[22] Shuuji Kajita, Tomio Yamaura, and Akira Kobayashi. "Dynamic walking control of a biped robot along a potential energy conserving orbit". In: *Robotics and Automation, IEEE Transactions on* 8.4 (1992), pp. 431–438.

[23] H Stephen Kaye, Taewoon Kang, and Mitchell P LaPlante. *Mobility device use in the United States*. Vol. 14. National Institute on Disability and Rehabilitation Research, US Department of Education, 2000.

[24] L Andrew Koman et al. "Botulinum toxin type A neuromuscular blockade in the treatment of lower extremity spasticity in cerebral palsy: a randomized, double-blind, placebo-controlled trial". In: *Journal of Pediatric Orthopaedics* 20.1 (2000), p. 108.

[25] MG Lazo et al. "Osteoporosis and risk of fracture in men with spinal cord injury". In: *Spinal Cord* 39.4 (2001), pp. 208–214.

[26] Tad McGeer. "Passive dynamic walking". In: *The international journal of robotics research* 9.2 (1990), pp. 62–82.

[27] Michael Geoffrey McKinley. “Design of Lightweight Assistive Exoskeletons for Individuals with Mobility Disorders”. PhD thesis. University of California, Berkeley, 2014.

[28] Xiuping Mu and Qiong Wu. “A complete dynamic model of five-link bipedal walking”. In: *American Control Conference, 2003. Proceedings of the 2003*. Vol. 6. IEEE. 2003, pp. 4926–4931.

[29] Jonathan Myers, Matthew Lee, and Jenny Kiratli. “Cardiovascular disease in spinal cord injury: an overview of prevalence, risk, evaluation, and management”. In: *American journal of physical medicine & rehabilitation* 86.2 (2007), pp. 142–152.

[30] Jong H Park and Kyoung D Kim. “Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control”. In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*. Vol. 4. IEEE. 1998, pp. 3528–3533.

[31] Jacquelin Perry, Judith M Burnfield, and Lydia M Cabico. *Gait analysis: normal and pathological function*. Slack Thorofare, NJ, 1992.

[32] Jason Ira Reid. “Control Strategies for a Minimally Actuated Medical Exoskeleton for Individuals with Paralysis”. PhD thesis. University of California, Berkeley, 2012.

[33] Jason I Reid et al. “A method of swing leg control for a minimally actuated medical exoskeleton for individuals with paralysis”. In: *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2013, V002T22A006–V002T22A006.

[34] Jessica Rose, James Gibson Gamble, and Janet M Adams. *Human walking*. Lippincott Williams & Wilkins Philadelphia, 2006.

[35] Mostafa Rostami and Guy Bessonnet. “Impactless sagittal gait of a biped robot during the single support phase”. In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*. Vol. 2. IEEE. 1998, pp. 1385–1391.

[36] Mostafa Rostami and Guy Bessonnet. “Sagittal gait of a biped robot during the single support phase. Part 1: passive motion”. In: *Robotica* 19.02 (2001), pp. 163–176.

[37] Mostafa Rostami and Guy Bessonnet. “Sagittal gait of a biped robot during the single support phase. Part 2: optimal motion”. In: *Robotica* 19.03 (2001), pp. 241–253.

[38] Tarik Saidouni and Guy Bessonnet. “Generating globally optimised sagittal gait cycles of a biped robot”. In: *Robotica* 21.02 (2003), pp. 199–210.

[39] RJ Soden et al. “Causes of death after spinal cord injury”. In: *Spinal Cord* 38.10 (2000), pp. 604–610.

[40] Naoki Sonoda, Toshiyuki Murakami, and Kouhei Ohnishi. “An approach to biped robot control utilized redundancy in double support phase”. In: *Industrial Electronics, Control and Instrumentation, 1997. IECON 97. 23rd International Conference on*. Vol. 3. IEEE. 1997, pp. 1332–1336.

[41] Fiona J Stanley, Eve Blair, and Eva Alberman. *Cerebral palsies: epidemiology and causal pathways*. 151. Cambridge University Press, 2000.

[42] Eliza Strickland. "Good-bye, wheelchair". In: *Spectrum, IEEE* 49.1 (2012), pp. 30–32.

[43] David H Sutherland. "The evolution of clinical gait analysis: Part II Kinematics". In: *Gait & posture* 16.2 (2002), pp. 159–179.

[44] David H Sutherland. "The evolution of clinical gait analysis part l: kinesiological EMG". In: *Gait & posture* 14.1 (2001), pp. 61–70.

[45] Chin Pei Tang. *Lagrangian dynamic formulation of a four-bar mechanism with minimal coordinates*. 2006.

[46] Wayne Yi-Wei Tung et al. "Design of a minimally actuated medical exoskeleton with mechanical swing-phase gait generation and sit-stand assistance". In: *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2013, V002T28A004–V002T28A004.

[47] Wayne Tung et al. "On the Design and Control of Exoskeleton Knee". In: *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2013, V001T09A005–V001T09A005.

[48] Spyros Tzafestas, Mark Raibert, and Costas Tzafestas. "Robust sliding-mode control applied to a 5-link biped robot". In: *Journal of Intelligent and Robotic Systems* 15.1 (1996), pp. 67–133.

[49] Michael W Whittle. "Clinical gait analysis: A review". In: *Human Movement Science* 15.3 (1996), pp. 369–387.

[50] Mary Elizabeth Wiley and Diane L Damiano. "Lower-Extremity strength profiles in spastic cerebral palsy". In: *Developmental Medicine & Child Neurology* 40.2 (1998), pp. 100–107.

[51] David A Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[52] David A Winter. *Biomedical Data Resources*. URL: `https://isbweb.org/resources/data-resources/140-movement-data/508-gait-data`.

[53] David A Winter. "Human balance and posture control during standing and walking". In: *Gait & posture* 3.4 (1995), pp. 193–214.

[54] TF Winters, JR Gage, and R Hicks. "Gait patterns in spastic hemiplegia in children and young adults". In: *J Bone Joint Surg Am* 69.3 (1987), pp. 437–441.

[55] Yujiang Xiang, Jasbir S Arora, and Karim Abdel-Malek. "Physics-based modeling and simulation of human walking: a review of optimization-based and other approaches". In: *Structural and Multidisciplinary Optimization* 42.1 (2010), pp. 1–23.