Design and Control of a Bipedal Robot

Derek Frei Lahr

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Mechanical Engineering

Dennis W. Hong, Chair
Kevin B. Kochersberger
Brian Y. Lattimer
Michael L. Madigan
Steve C. Southward

March 26, 2014
Blacksburg, Virginia

Keywords: humanoid robot, bipedal locomotion, biarticular actuation

Design and Control of a Bipedal Robot

Derek Frei Lahr

ABSTRACT

Emergency first responders are the great heroes of our day, having to routinely risk their lives for the safety of others. Developing robotic technologies to aid in such emergencies could greatly reduce the risk these individuals must take, even going so far as to eliminate the need to risk one life for another. In this role, humanoid robots are a strong candidate, being able to take advantage of both the human engineered environment in which it will likely operate, but also make use of human engineered tools and equipment as it deals with a disaster relief effort.

The work presented here aims to lessen the hurdles that stand in the way through the research and development of new humanoid robot technologies. To be successful in the role of an emergency first responder requires a fantastic array of skills. One of the most fundamental is the ability to just get to the scene. Unfortunately, it is at this level that humanoid robots currently struggle.

This research focuses on the complementary development of physical hardware, digital controllers, and trajectory planning necessary to achieve the research goals of improving the locomotion capabilities of a humanoid robot. To improve the physical performance capabilities of the robot, this research will first focus on the interaction between the hip and knee actuators. It is shown that much like the human body, a biped greatly benefits from the use of biarticular actuation. Improvements in efficiency as much as 30% are possible by simply interconnecting the hip roll and knee pitch joints.

Balancing and walking controllers are designed to take advantage of the new hardware capabilities and expand the terrain capabilities of bipedal walking robots to uneven and non-stationary ground. A hybrid position/force control based balancing controller stabilizes the robot’s COM regardless of the terrain underfoot. In particular two feedback mechanisms are shown to greatly improve the stability of bipedal systems in response to unmodelled dynamics. The hybrid position/force approach is shown through experiments to greatly extend humanoid capabilities to many types of terrain.

With robust balancing ensured, walking trajectories are defined using an improved linear inverted pendulum model that incorporates the swing leg dynamics. The proposed method is shown to significantly reduce the control authority (by 50%) required for satisfactory trajectory following. Three parameters are identified which provide for quick manual or numerical solutions to be found to the trajectory problem.

The walking and balance controller were operated on four different terrains successfully, strewn plywood, gravel, and high pile synthetic grass. Furthermore, SAFFiR is believed to be the first bipedal robot to ever walk on sand. The hardware enabled force control architecture was very effective at modulating ground reaction torques no matter the ground conditions. This in combination with highly accurate state estimation provided a very stable balance controller on top of which successful walking was demonstrated.

Acknowledgements

This dissertation could not have been accomplished without the ceaseless support of many individuals. The greatest thanks go to my wife, Becky, who has put up with the long hours and late nights without complaint. I love her very much.

I would also like to thank my family, Chris and Laurene, for their unending supply of support, constant praise, and advice. They have been following my work closely from the beginning and are my biggest fans. Without their encouragement, the path to this point would have been much more difficult.

It goes without saying that my advisor, Dennis Hong, deserves a great deal of gratitude. He has provided me and lab mates with the perfect environment for creative, fun, and interesting engineering projects. RoMeLa is the tinkerer's dream workshop and I feel privileged to have worked here for so long.

I thank the members of my graduate committee, Kevin Kochersberger, Steve Southward, Brian Lattimer, and Michael Madigan who have given me their time and support to better my education and research.

Thanks to everyone in RoMeLa. I look forward to coming in every day because of the great friends I have here and the amazing and fun work we do together. Special thanks go to Viktor Orekhov, Bryce Lee, Mike Hopkins, and Jack Newton for all their work in building and operating SAFFiR.

Finally, without the financial support of our funding agencies, none of this would have been possible. Thanks to the Office of Naval Research and DARPA for taking a chance on a growing lab.

Table of Contents

List of Figures

List of Tables

1 Introduction

Emergency first responders are the great heroes of our day, having to routinely risk their lives for the safety of others. Developing robotic technologies to aid in such emergencies could greatly reduce the risk these individuals must take, even going so far as to eliminate the need to risk one life for another. In this role, humanoid robots are a strong candidate, being able to take advantage of both the human engineered environment in which it will likely operate, but also make use of the human engineered tools and equipment as it deals with a disaster relief effort. In that vein, SAFFiR (Shipboard Autonomous Fire Fighting Robot) is being developed to further the technologies necessary to achieve that ultimate goal.

This project focuses on developing a robot to support sailors in surface ship firefighting and damage assessment/control activities. Fighting fires onboard ships is one of the most strenuous and dangerous activities that sailors need to perform. The use of robotics onboard ships is currently of interest to the U.S. Navy for “dirty, dull, or dangerous jobs”. Firefighting is both dirty and dangerous, making it an ideal application for robotics on ships.

A robot designed for firefighting activities will remove sailors from immediate exposure to conditions produced by a fire as well as allow for conditions to be monitored in the area close to the fire. Besides fighting fires, the robot can also serve as a tool used by sailors in helping to reduce the tasks that the sailor is required to perform. Robots will be capable of autonomously carrying out tasks; however, sailors will also be able to override robot behavior for a more manual use of the robot through teleoperation. With the recent reduced manning initiatives onboard ships, the additional support from a robot will be very beneficial.

In addition to the support to the sailors, the robot will also be capable of performing tasks that are not possible by humans. This includes allowing the robot to get more intimate with the fire for suppression activities, remain in the fire environments for longer durations, and use new suppression strategies that may be hazardous to humans. Beyond the U.S. Navy, the firefighting robot will transform how humans fight fires and has the potential to save thousands of lives every year in the U.S. alone.

A humanoid robot is uniquely suited to navigate around a shipboard environment because it is an environment designed for the human form. Some of the obstacles present include corridors, where width and height is limited, raised door sills, ladders, steps, stairways, and tight corners. For most conventional wheeled and tracked robots, these are significant obstacles. For example, to climb ladders and even stairs, these types of robots must have long wheelbases to prevent them from tipping over, however the length, if not segmented, limits their maneuverability in confined spaces. However, because this environment is designed for human travel, it follows that a humanoid robotic architecture could perform well. Additionally, the articulated legs allow the robot to balance itself well in rough sea states while wheeled or tracked vehicle cannot.

Legged robots, of which humanoid robots are a subset, are more easily shielded from heat sources than their wheeled or tracked counterparts. There are several approaches to environmentally hardening a robot, and one of the most effective and efficient methods is shielding, wherein the goal is to prevent as much radiant and convective heat as possible from reaching the bulk of the robot. Of the four most prolific robot architectures, legged robots, and in particular, humanoids are the easiest to shield from heat. This is because legged robots have no continuously rotating parts that must be outside of the shielding or any component that is in continual contact with the ground. So unlike wheeled or tracked robots, there are no axles, bearings, or wheels that have to pass through the shielding, all of which would produce vulnerable spots against heat. It follows that all the components of a legged robot can be covered with the shielding rather than having to develop high temperature capable wheels, tires, or tracks.

To be successful in the role of an emergency first responder requires a fantastic array of skills. One of the most fundamental of which is the ability to just get to the scene. And unfortunately it is at this level that humanoid robots struggle. The current crop of humanoids from around the world can at best travel at a walking pace across smooth relatively flat surfaces, at most traversing slight inclines and declines. So it is at this level that science must begin to understand the dynamic principles of bipedal locomotion in the hopes that one day humanoids will be able to crawl, shuffle, hop and run as effortlessly as a human. This research aims to do just that, beginning with a fundamental problem of locomotion in unstructured, irregular, or uneven terrain: the ability to take a single controlled stride from one step to the next assuming arbitrary initial and final conditions. This is dubbed, the stepping stone problem.

1-1 Background

Taking even one step, let alone stringing several together in the form of a walking gait, is a difficult task for a wide variety of reasons. The problems begin with the electro-mechanical hardware. For example, electric motors have relatively low power densities. Large powerful magnets and meters of copper windings come at great expense in terms of weight and inefficiencies through heat loss. Worse though, is the necessity for fragile, inaccurate, or inefficient geared transmissions to slow the rotation of electric motors to generate the large torques needed for legged locomotion. The handful of robots that use some form of hydraulic actuation, such as Boston Dynamics Petman or Raytheons SARCOS, require large on or off-board power supplies, pumps, and accumulators. Hydraulics also tend to be less efficient than other forms of actuation, however this may change as research in the field progresses.

The control of a humanoid presents an even wider variety of hurdles, only a few of which are presented here. The first of which is the high dimensionality of the system. On a typical biped alone, there are twelve degrees of freedom required to place the torso and feet in the correct position and orientation required in most tasks. Walking therefore requires the careful coordination and control of all these DOFs. The control laws meant to do this are not up to the task. Control theories developed in the early years of robotics were applied to robots of that time, which usually meant they had a fixed base. Legged robots are quite unlike these early manufacturing robots in that they use a floating base, and the controllers must take into account the hybrid dynamics of step to step transitions. Worst of all, a mobile robots interactions with the ground are through a unilateral constraint, affording only a fraction of the control authority of its fixed base counterpart.

Robots are meant to interact with their environment. Freed from the organized and structured environment of a manufacturing cell, humanoids unfortunately are destined to operate in the cluttered, unstructured, and downright confusing world of humans. If the obstacles were not hard enough to manage, the eventuality that any misstep could result in the injury of person or permanently damage the robot makes it that much more important that a humanoid can safely and robustly interact with its world. Modern actuators and controls are ill equipped to handle this eventuality. Stiff actuators under stiff control can impart a significant amount of damage while attempting to track a predefined trajectory, hence the need for robot safety cells.

The aim of this research is to address in some part these challenges of humanoid locomotion. A robot-wide approach is taken to address the individual but interconnected problems. Fundamental to the approach, is the codevelopment of hardware and software to enable more efficient actuation and control. Specifically, this work will research three avenues: biarticular and parallel actuation for more efficient bipeds, control approaches for robots utilizing both position and force controllable joints, and improved models for more accurate trajectory planning. Successful completion will make the long term goal of robotic emergency first responders one step closer to reality.

1-2 Current research field

There are currently two main approaches in the field of bipedal locomotion, those characterized by stiff position controlled algorithms that achieve forward progress, or alternatively those using the inherent dynamics to generate limit cycle motions that result in walking.

The origin of the former can be traced to the proliferation of industrial robots, and in industry, precision and speed are of utmost importance. The robots that were developed had rigid fixed bases and operated with powerful high gain motors and controllers that could quickly and accurately perform their task. When many early bipedal robots were first in development, they retained this style of actuation with designers working towards fully controllable motions. Legged robots do not have a fixed base however, and so their control authority is limited. To ensure the robots did not fall over, the Zero Moment Point was born.

The Zero Moment Point, ZMP, is defined as the point on the foot about which no moments act, only forces. It is alternatively called the center of pressure, COP. The ZMP can be either empirically calculated, or experimentally determined using a force torque sensor. It was proven that if the ZMP stays within the support polygon, then controllablity about that time period can be maintained. ZMP based walking algorithms therefore typically specify a series of joint trajectories that nominally generate a walking gait and ensure the ZMP remains within the foot. The trajectories are generated in a variety of ways[1]–[6]. Additionally, mechanisms to alter the trajectories are computed such that the stability is maintained through disturbances [7], [8].

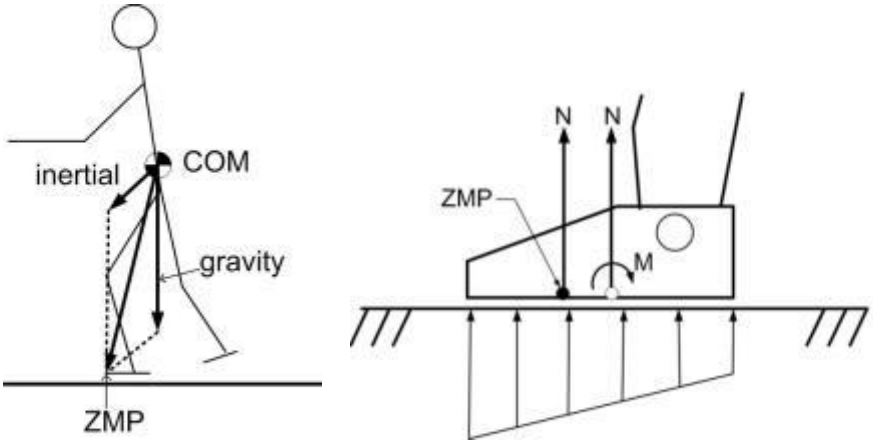

Figure 1-1. The ZMP is defined as the point between foot and ground about which no net moments are acting. It is sometimes called the center of pressure.

There are a wide variety of ZMP controlled robots. One of the earliest examples of which is the Honda P2, which was followed by the Honda Asimo [9]. Japan is a leader in this field, with other examples including Waseda Unversity's WABIAN[10], Sony's QRIO, HRP, and Tokyo University's H6 and H7. KAIST in Korea has develeped the HUBO series of robots, which are now capable of running[6]. The German Technical University of Munich has developed both Johnie and LOLA[11]–[15] which also display an impressive level of mechanical ingenuity.

ZMP controlled robots have made significant progress towards enabling the vision of humanoid robot helpers. As a whole, they display a high degree of versatility, being capable of omni-directional walking, running, stair climbing and some degree of rough terrain walking. Disturbance rejection is fair to good. However, with that versatility comes low energy efficiency. The joint trajectories are often generated with little regard to the natural dynamics or a cursory attempt at efficiency. Significant energy is expended then to drive the robot links against their natural tendencies. The Honda Asimo is said to use ten times more energy than a proportionally scaled human[16].

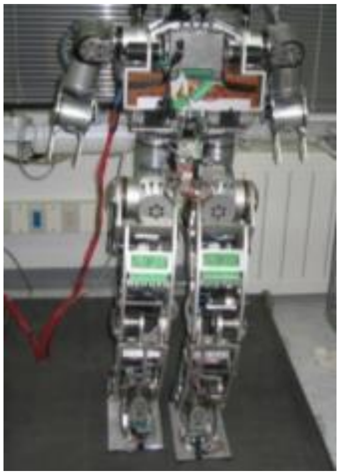

Figure 1-2. HUBO represent the current state of the art in humanoid robots.

Seeing the disadvantages of ZMP controlled robots, and faction of researchers breathed a second life into a concept dating back to the 1800's, that of passively stable limit cycle walking machines. The first instance of which was a wobbling toy patented by Fallis, that when properly initiated could walk down an incline. Just like this toy, early examples of passive dynamic walkers utilize the natural dynamics of a biped to in some part control the walking gait, with early examples having zero actuation. As their design has progressed, they have trended towards including more and more actuation, typically force controllable or low impedance actuators so as to maintain a large degree of passivity. The underlying principle though remains, exploit the dynamics for increased efficiency and stability.

Research into these passive dynamic robots was pioneered by that of McGeer who developed a now prevalent "Compass Biped" model in simulation whose dynamics could be easily computed and was shown to have a stable limit cycle gait[17]. Mechanical representations of this model have been made by many institutions. Cornell developed a biped with arms and legs that include three passive degrees of freedom and uses ankle push off to inject energy at each step [18]–[21]. MIT's example has two internal degrees of freedom and two DOF in both ankles for better control[22]. Delft University's work as gone as far as any to maximize the versatility of these passive dynamic robots[23]. Their latest work, Flame and Tulip, have no fully passive degrees of freedom, but employ force controllable actuators and a deep understanding of the natural dynamics of a walking stride to govern and control the legs. These last example exhibit the best disturbance rejection of all. In the United States, the work started at MIT continues at IHMC with the M2V2 robot[24][25]. Portions of their work though is very revelant to the research presented here.

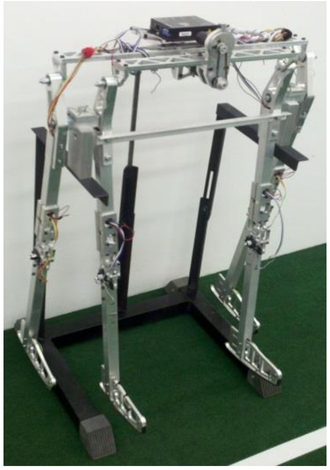

Figure 1-3. Passive Dynamic bipeds emphasize the natural dynamics of walking, in some cases by employing force control actuators.

The passive dynamic themed robots exhibit compelling energy efficiencies. A recent example from Cornell having walked 1[km] on a single battery charge[26]. The efficiency comes at the cost of verstility though. Passive joints make anything but straightline walking nearly impossible. Additionally the disturbance rejection has only been shown in a few cases, at best handling a step height discrepancy of around 20[mm][23]. Other anectodatal video evidence points to even worse performance. This approach though has shown the importance of the underlying dynamics and the flexibility of force controlled approaches.

In light of these two approaches, it is the goal of this research to investigate the space between. Literature shows there is value in both approaches. Both position controlled trajectories and force controlled joints have a place in bipedal walking. However, significant energy has not been spent understanding where an appropriate compromise should be made. This work aims to identify a suitable combination of the two approaches with the hope that it will produce a robust walking gait.

1-3 Our approach

1-3-1 Problem statement

Three specific problem areas are addressed in this research:

1) Electro-mechanical actuation for robotic applications lacks sufficient power, efficiency, and robustness to allow the widespread adoption of humanoids as helpers. The human body is very adept at producing and delivering high powers to its hand and feet. This is accomplished through the use of parallel and biarticular actuation of all the joints in a limb[27], [28]. A robotic analogy is currently missing.

2) Control schemes are in large part limited to position control of these actuators due to the impedance mismatch and friction that occurs as result of the large gear reduction units. Both mechanical and control solutions are needed to allow accurate and low impedance control of a robot limb. No algorithms exist that have demonstrated rough terrain walking on robots using both force and position controlled joints.

3) The complex dynamics and limited control authority of humanoids makes trajectory planning very difficult. Simple models and optimization based techniques have both been used with varying degrees of success. The simplified models are useful for step planning, but do not include all significant dynamic forces, such as the effect of leg swing on single support.

With these problems in mind, the research goals are:

1) Understand the role of the actuators in the hip and knee joint of a humanoid robot in various configurations including parallel, serial, and biarticular implementations

2) Find an optimal hip and knee joint configuration that minimizes the energy consumption during typical usage.

3) Extend the simple dynamic models of humanoid to include more of the significant dynamics affecting the COG in order to generate better trajectories.

4) Utilize a hybrid force/position control algorithm to enable walking in uneven and compliant terrain.

1-3-2 Approach

This research focuses on the complementary development of physical hardware, subsequent high and low level controllers, and trajectory planners necessary to achieve the research goals of improving the capabilities of a humanoid robot.

To improve the physical performance capabilities of the robot, this research will first focus on the co-development of a new lightweight linear actuator capable of force control and a humanoid robot platform that best exploits the new actuators by utilizing a parallel actuation architecture across each joint. Careful consideration is given to the placement of actuators to maximize the benefit of having multiple actuators span multiple joint (biarticular actuation).

Once the physical hardware is completed, the development of controllers will follow. Accurate state estimation of the COM in spite of sensor noise and unmodelled dynamics will first be solved. A balance controller will use ground reaction torques to stabilize the COM about a desired trajectory.

Once an understanding of the dynamics is attained, suitable states of the robot and their trajectories necessary to achieve a step can be determined. Furthermore, they can be parameterized to achieve a single stable stride. Control laws will be built on top of these trajectories to ensure they can be maintained even when disturbed. Simulation will be used primarily for the development of the controllers and filters, with final verification occurring on the actual robot.

1-4 Mapping

In Chapter 2, the design of humanoid robots is considered. In particular, the architecture of the hip and knee are investigated. The hip and knee are the most heavily loaded joints of the leg, and so offer the best chance to improve the efficiency of bipeds. To factors are considered, the orientation of the hip joint axes, and biarticular actuation across the hip and knee.

A new actuation architecture is used on the humanoid presented here, SAFFiR. It employs a new series elastic linear actuator in a parallel arrangement for optimal torque generation. The design of the actuator and the construction of the actuator are discussed in Chapter 3. For reference, Appendix 1 goes into the low level motor control of the actuators that enable force control.

Chapter 4 describes the literature review surrounding humanoid walking. It serves to frame the algorithm developed for SAFFiR. The walking algorithm is built on top of a robust balance controller, which is described in this chapter. Chapter 5 describes the walking state machine and trajectory generation that transforms the balance controller into a walking controller.

Chapter 6 presents an improved model of the center of gravity of a bipedal robot during the single support phase that takes into account the swing leg dynamics. The improved model more accurately represents the trajectory of the physical system and both improves trajectory following while reducing control torques.

2 Design for Efficiency

Like many other devices prevalent in our lives, humanoid robots are constrained by their energy usage. There are two breeds of humanoids at the moment, hydraulic and electrically powered. Neither have significant un-tethered run-times over an hour. In fact hydraulically powered humanoids are so energy hungry they required a large power and cooling tether when operated at the 2013 DARPA Robotics Challenge. RoMeLa has operated several battery powered electrically driven humanoid robots over the years. The DARwIn-OP [29] series robots operate up to 30 minutes on one battery charge. Similarly, the author has personally driven CHARLI [30] for close to 45 minutes before the battery was replaced. Finally, SAFFiR has walked for 30 minutes with little depreciable loss of battery charge, such that a 1.5 to 2 hour life is expected. Ultimately though, to be useful as a tool, humanoids need large scale improvement in energy storage and efficiency technology.

There are several avenues by which humanoids can increase their runtime. Batteries and other energy storage technology will likely improve, offering a larger energy reservoir. Similarly, improvements in transmissions and motors will reduce energy conversion losses. In this work, we instead investigate an avenue of energy savings not yet explored, specifically, the energy savings that can be attained through joint orientation and actuation optimization, referred to as architectural improvement.

We propose that considerable gains in efficiency are possible by the clever arrangement of either the joint orientations or the actuator placement about the joint. The following evidence supports this hypothesis. Like all robots, humanoids undertake very repetitive motions. Considering the legs only, the majority of their time is spent either locomoting or standing. While doing either, the torques and velocities of each motor therefore fall within a reduced operating range. Furthermore, a study of electric motors will show that they are most efficient in a small portion of their operating range. We show in this work that by carefully choosing the robot architecture, one can shift operating range required by the repetitive tasks, such as walking, to the more efficient operating range of the motor.

This chapter is divided as follows. Conventional humanoid designs will be introduced as background art. In particular, the serial nature of the designs will be highlighted. These designs will be compared to that of the human body, which serves as inspiration for parallelly actuated designs later. A good portion of the effort is in the comparison of parallel and serial actuated designs and the respective effect on the efficiency. An overview of both architectures is given. Next a simple 2DOF actuated u-joint is analyzed. It is shown that both parallel and serial actuation can achieve the same performance. This work is extended to 3DOF and 3DOF with biologically inspired biarticular actuation architectures. An optimization is run and compared to a control architecture for efficiency gains.

2-1 Conventional humanoid design

By their very nature, human beings and humanoid robots undertake very similar actions, including standing, walking, and jumping. However, the means by which their actuators and skeleton accomplish these actions is very different. This is in part due to the nature of the “technology” within. Because the type, characteristics, and controls of each actuator are different, the way in which they are implemented is also different.

A muscular-skeletal arrangement of a human leg that has effectively the same number of degrees of freedom as a humanoid, but it uses extensive parallel actuation while humanoids are almost strictly serially actuated. Among a variety of explanations for this difference, we will focus on inspiration and design principles from the human leg in order to build a better robot leg. In this section, the primary characteristics of both humanoid and human leg are discussed and areas of improvement for humanoids highlighted.

Many existing humanoid robots stand out as a good indication of the available technology. To emphasize the importance of the hip and knee design, this section will focus primarily on those portions of humanoid robots in existence. It will be seen that due to the demanding and often antithetical constraints of these robots (a lightweight, high strength, and powerful robot), similar designs have been reached by their builder. In general, all humanoid legs include either 6 or 7 DOFs with both DC motors with harmonic drive units and belt reductions, as seen in Figure 2-1. A short description of several of these robots will follow and is only meant to give a frame of reference with which to compare to SAFFIR.

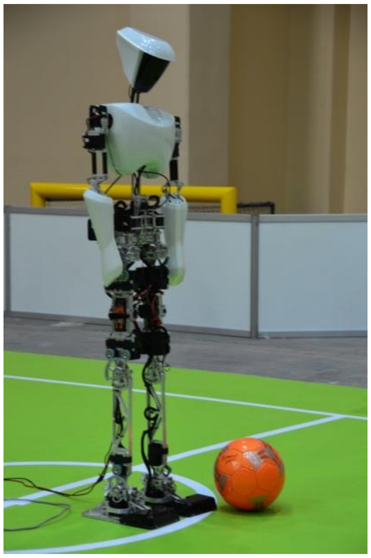

Figure 2-1 Cognitive Humanoid Autonomous Robot with Learning Intelligence (CHARLi)

LOLA

LOLA was built by the Technical University of Munich for a research program dating back to 1998, whose first creation was the humanoid Johnnie[12], [14]. Its current form stands 180[cm] tall and has 22 degrees of freedom (DOFs). LOLA is unique for two reasons, one it utilizes linear actuators in both the knee and ankle joint for their superior mass distribution and adjustable torque curve. Secondly, LOLA is the only robot presented here to utilize an actuated toe joint.

KAIST HUBO

The HUBO robot from the Korean Advanced Institute of Science and Technology (KAIST) began as the KHR-0 robot in 2001 [31], [32]. This platform has 41 DOFs, is 125[cm] tall, and weighs 55 [kg]. It is notable in that it has five independent fingers on each hand, stereo vision, and a specialized six axis force/torque sensor for balance.

WABIAN-2

WABIAN-2 was developed in 2003 at the Waseda University based off of research dating back to 1966 [33]. As seen in Figure 1, the robot is 1.54[m] tall with a weight of 64.5[kg]. Like HUBO, this platform also has 41 DOFs but they are distributed differently. WABIAN-2 is unique in that it has a spherical joint in the ankle as well as a 2-DOF trunk joint and waist, and a passive toe joint. These extra DOFs reduce the impact of the heel strike when walking, allow WABIAN-2 to better cope with rough terrain, and maintain a more human-like posture.

Honda ASIMO

ASIMO is the result of research by the Honda Motor Company dating back to 1986[9]. In its current form, ASIMO can run at 6km/h in a straight line, and up to 5 km/hr in a circle. Like HUBO, it also has stereo vision, can climb stairs, and interact with humans. It only has 33 DOFs but is considered one of the most state of the art humanoid robots at this time.

2-1-1 Existing Joint Configurations

As this paper focuses primarily on the leg design of Virginia Tech’s SAFFIR, this section will focus on the various leg configurations of existing humanoid robots. It will be seen that due to the demanding and often antithetical constraints of these robots (a lightweight, high strength, and powerful robot), similar designs have been reached by their builders. All legs have either 6 or 7 DOFs and primarily utilize DC motors with harmonic drive units and belt reductions.

Hip Joint

Like the human hip joint, all these humanoids have three DOFs at the hip. In these cases, such a joint is achieved using three revolute joints whose axes intersect at a point, as can be seen in Figures 2.

Both HUBO and Wabian-2 employ harmonic drives mounted on the joint which are driven through a toothed belt by DC motors. This belt allows a more compact configuration and more human like proportions to be maintained as the assembly does not extend outside the leg. It is also interesting to note that the hip yaw axis of LOLA is inclined towards the vertical axis of the robot for better power distribution unlike the vertical orientation of both HUBO and Wabian-2.

Knee Joint

While the human knee joint is kinematically one of the simpler joints within the body, it must also be one of the more powerful on a robot. While no high resolution photos exist of ASIMO’s mechanical design, it is known that like both HUBO and Wabian-2, it utilizes an electric motor driving a harmonic drive unit through a toothed belt. The ASIMO robot also uses a longer belt drive so as to place the motor as high on the thigh as possible, thereby reducing the moment of inertia of the leg. Because the knee joint sees both high speeds and large torques, it must be very powerful. To this end, the HUBO knee is driven by two smaller 150[W] motors for a higher power to weight ratio, as seen in Figure 3 left. The additional motor is only used for force control, eliminating the need for a mechanical differential.

The one noteworthy exception to this trend is LOLA. They have chosen to use a ball screw actuator on the knee, and while this design will have a non-linear torque curve, this curve can actually be tailored to the torque requirements of the knee.

Ankle Joint

For the most part, the ankle joint is very similar in appearance to the hip joint. All robots studied here with the exception of Wabian-2, use a 2-DOF ankle joint. Wabian-2 has an additional degree of freedom in the yaw direction. While this allows it considerable more flexibility, it comes at the cost of weight and the increased inertia of the leg. DC motors and harmonic drives with belt connections are implemented in HUBO, Wabian-2, and ASIMO, as can be seen in Figure 2-2. Although not pictured, ASIMO also places the DC motor as high on the shin as possible to again reduce the legs moment of inertia.

Again, the LOLA robot differentiates itself with the use of two linear actuators in the ankle. Although it has two actuators for two degrees of freedom, it can use them in parallel in both the ankle pitch motion as well as in roll. In pitch, the actuators move together, and for roll they move against one another. So for either motion, twice the power is available given the same motors as compared a revolute based joint. The previous iteration as it appeared on Johnnie is pictured in Figure 2-2 right. The current version can be seen in Figure 1.

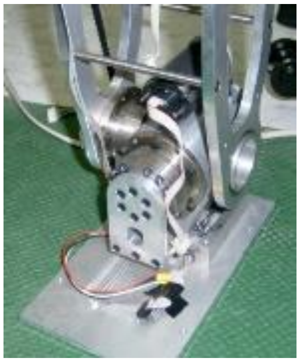

Figure 2-2. 2-DOF ankle joint of HUBO.

2-1-2 Electric motors

All the designs above are electrically driven. They convert electrical energy delivered from a battery to mechanical energy through some type of electromagnetic motor. Electromagnetic motors use energy in the form of moving electrons to generate a magnetic field. This magnetic field then interacts with a permanent magnet, exerting a force thereon. When this force is large enough to overcome both the internal friction and the external load, the motor will move, and mechanical power is produced. This internal friction can represent a significant amount of energy lost. In order to generate the aforementioned magnetic field, electric current is driven through a length of wire, and it is in here that additional energy is lost as heat.

There are two ways to operate the motor in a more efficient region. The first and most common is to carefully select the transmission ratio between the motor and the joint. In addition to that method, we will investigate altering the orientation and interconnection between the degrees of freedom of the leg as a way of better distributing the torque between actuators. The inspiration for which comes from the anatomy of the human body, as will be introduced next. The goal of this work is to operate the motor in a range such that these two losses are minimized.

2-2 Biarticular actuation

The limbs of all the robots describe above can be modeled as serial manipulators. The anatomy of human body is not built in the same way. Instead of serial actuation, the hip, knee, and ankle joints are driven in parallel by over 20 major muscles [34]. Of those, ten directly exert torques on the hip joint. The major muscle groups that control the pitch axes of the hip and knee can be seen in Figure 2-3**.** Those include the *m. gluteus maximus*, hamstring, *rectus femoris*, and *m. vastus.* Notable is the fact that two of these muscles, the hamstring and the *rectus femoris* are biarticular. Both muscles span more than one joint, the hip and the knee.

It has been shown that these biarticular muscles are responsible for transferring power between the joints especially during powerful motions such as jumping [28], [35]. Literature shows that approximately 300[W] of power flows through the *m. rectus femoris* to the knee joint in the preflight extension of a leg while jumping. In doing so, this muscle has constant length, and therefore produces no power of its own. The trend continues to the ankle via the m. gastrocnemius, which delivers 200[W] from the knee to the ankle.

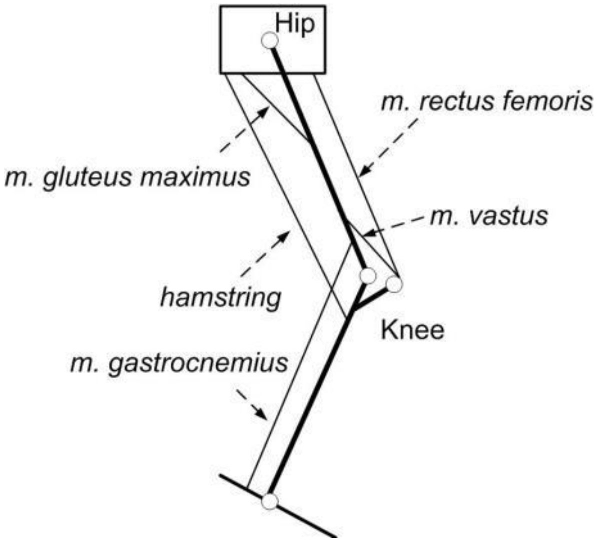

Figure 2-3. The major muscle groups of the hip and knee

The human body would appear to be optimized around power delivery or perhaps ideal dynamics. In this work we are interested instead in the maximum efficiency of each actuator, which will likely lead to a very different solution. The inspiration though is drawn from the means by which nature has achieved this optimization. Specifically, the human body utilizes over actuated, parallel, and biarticular features across the joints of a leg. This work will work to understand how parallel and biarticular actuation can improve the efficiency of humanoid robots.

2-2-1 Biarticular Joints

By their very nature, human beings and humanoid robots undertake very similar actions, including standing, walking, and jumping. However, the means by which their actuators and skeleton accomplish these actions is very different. This is in part due to the nature of the “technology” within. The ability to transfer power is a valuable one. This is particularly true in humanoid robots, where often one or two joints are much more heavily loaded than others, (hip roll and knee pitch) [10], [32]. Incorporating biologically inspired bi-articular actuators to humanoid legs can allow them to better distribute loads and power across several actuators. There are a handful of robot designs that make use of bi-articular actuators [28], [35], [36].

Our contribution to this area is a better understanding of the mechanics and constraints important to parallel and bi-articular actuators in humanoid robots especially with respect to the energy lost. We also adopt a simple method of representing such complex systems [37]. This representation lends itself towards optimization in that physical design constraints are easily defined. A conceptual design of the full implementation is proposed. Like the analysis above, this work will consider the inverse transpose of the Jacobian as the representative model of the system as it nicely captures all the necessary interactions between actuators.

Figure 2-4 below illustrates the two joints being studied, the hip and knee. Of the hip, only the pitch and roll joints are of interest. The yaw axis which is typical of most hips is considered independent because it is use less frequently and with less intensity. As a note, this work considers both serial and parallel actuation of the hip. As the work below will show, the two can be made to perform identically with respect to the actuator loads in the nominal configuration. There are some significant differences with regard to implementation though when considering biarticular actuation.

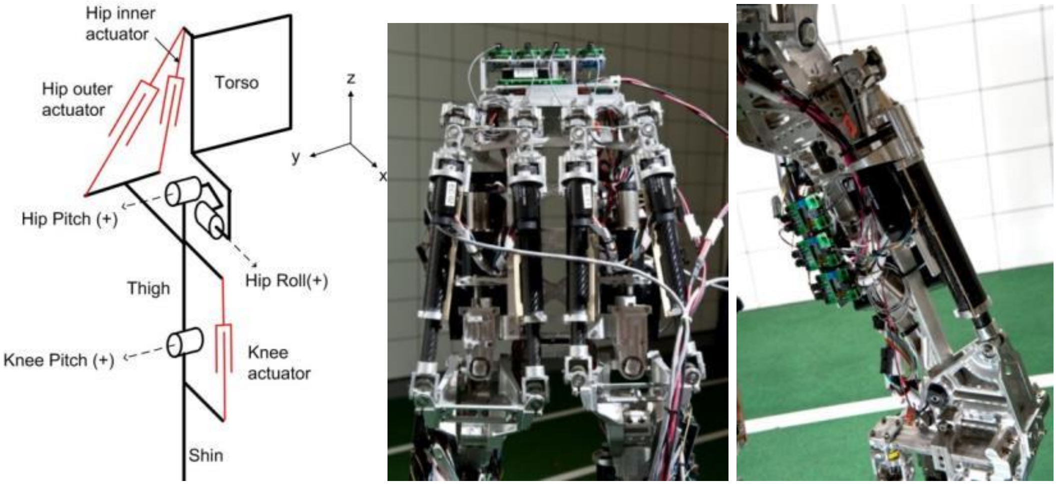

Figure 2-4. Parallel hip actuators and Knee actuator as seen on the SAFFiR

First the parallel actuated version will be discussed as it is more straightforward. The purpose of biarticular actuation is to interlink two distally located joints, in this case the hip and the knee. This is accomplished by allowing an actuator to exert torques about these two joints. In the human body this is accomplished by connecting the two ends of the muscle to the links on the far side of both joints. For example, the hamstring extends from the pelvis to the shin, thus biarticularly actuating both the hip and the knee. This in principle is easily accomplished with linear actuators on humanoid robots. In practice though, this can negatively affect the range of motion of the joint due to interference issues. This is particularly problematic when the designer wishes to exert opposite torques about two joints which would require the actuator to pass by or through the link that separates the two joints.

An alternative is depicted below in Figure 2-5. Instead of directly attaching to the shin for example, a biarticular hip actuator exerts a torque on a lever rotatably mounted to the thigh. This lever drives a pulley around which a cable is attached. The cable can then be used to exert either positive or negative torques on the shin depending on the configuration. Of the two actuators in the hip, one or both can be made biarticular. Furthermore, the knee actuator can be arranged such that it terminates on either the shin, the trunion or the pelvis, making it biarticular to none, one or both of the hip degrees of freedom.

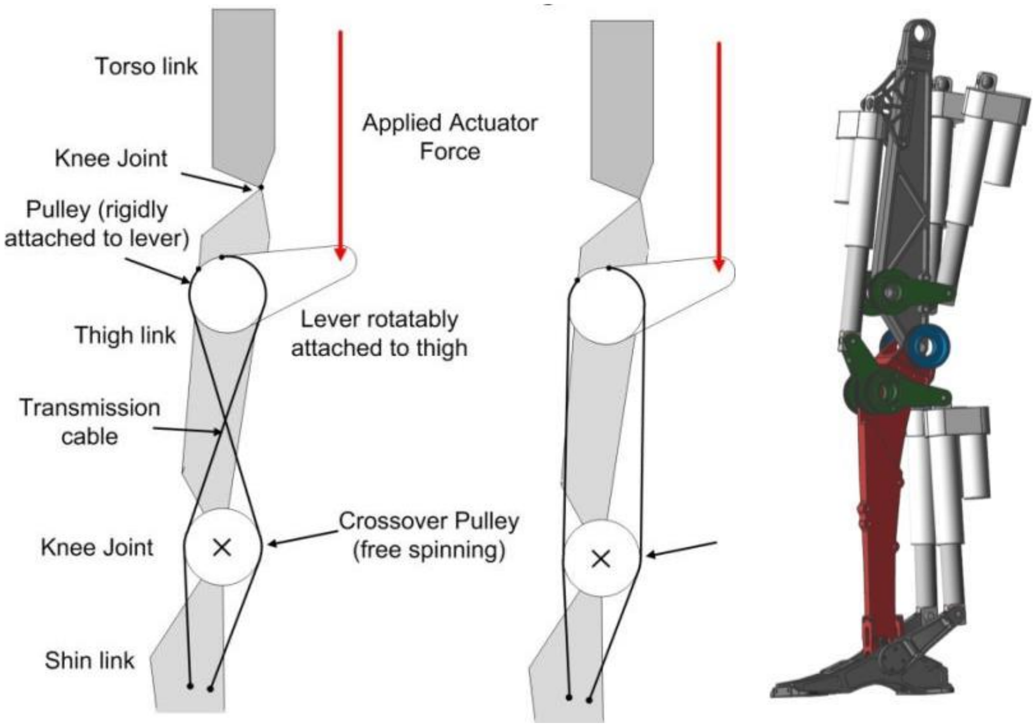
Torso link
Knee Joint
Pulley (rigidly attached to lever)
Thigh link
Transmission cable
Knee Joint
Shin link
Applied Actuator Force
Lever rotatably attached to thigh
Crossover Pulley (free spinning)

Figure 2-5. A cable pulley arrangement can be used to implement biarticular actuation across the hip and knee.

Serial actuation poses a slight challenge to implementing biarticular actuation. This is most clearly evidenced by understanding why it is so easy with parallel actuators. All parallel actuators that span the hip terminate on the thigh. They must only jump one more link to be made biarticular. However, at least one (for a 2dof hip such as the one studied here, only one) serial actuator of the hip is not directly connected to the thigh. To be made biarticular this particular actuator must jump two links and one joint. Crossing this second joint while not effecting the torque about that joint must be done carefully. In practice this can be accomplished through the use of a constant velocity joint, such as the one seen in Figure. A similar implementation is studied and found to be feasible in Appendix 0 for reference. Therefore to make both actuators of the hip biarticular is difficult. However, making one does not require this complexity, since one is always directly connected to the thigh.

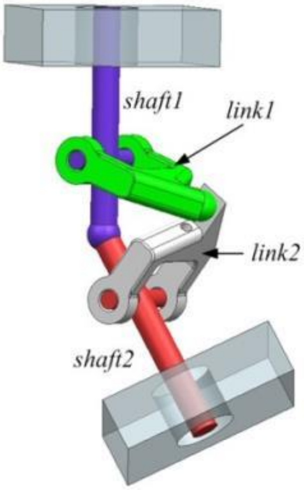

Figure 2-6. Parallel hip actuators and Knee actuator as seen on the SAFFiR

Allowing the knee to be biarticular requires similar care. The knee is two joints and one link removed from the torso. There are two biarticular options: 1) make the knee biarticular with regards to the nearer of the two hip joints, whatever it may be or 2) allow the knee to exert a torque about the hip via a constant velocity joint that passes through the hip center. In this way the knee can exert arbitrary toque vectors on the torso by reorienting the output shaft of the CV joint.

These various configurations will be addressed independently in the optimization below as compared to parallel implementations.

2-3 Optimization

Optimizations were run to evaluate the effectiveness of various configurations including: alternative joint axes orientation and most biarticular arrangements. The optimization serves to reduce the energy lost by the motors and is therefore dependent on the motor trajectory. These motor trajectories are derived from the joint trajectories, so several joint trajectories and motions were chosen and weighted to represent the expected behavior of the robot. For example, the lower body of the robot is primarily used for locomoting, standing, and rising from a squatted position, so these motions are modeled and applied to the cost function with some duty cycle. Other constraints are easily enforced to ensure that the joint maintains some specific capabilities in terms of force or velocity to ensure a well-rounded joint design.

This section will discuss the models and motions, the scoring function, the constraints, the optimizations, and finally the results.

Unlike the upper body, the lower body of a humanoid robot has much more limited purpose. It needs only to support the upper body and move it through space. It is therefore a good candidate to be optimized around the specific motions it will likely be performing. In general the upper body must perform a larger set of operations and requires a more general approach. In this work we look at two motions: walking, and standing from a squat (referred to simply as standing from hereafter). This section will cover the general parameters of the motions. A detailed derivation of the torques and velocities can be found in the appendix.

The standing motion will be covered first. The purpose of this motion is to replicate load on a leg due to the robot rising from a squatted position using both or just one leg. In the author's experience, it is a difficult motion for both humans and robots. It is fundamentally useful for picking objects off the ground, and standing after a fall. In particular, standing from a squat on one leg also places similar loads as would be experienced when stepping up or across a long distance. This motion was derived from a dynamic model of the robot. Joint trajectories were defined such that the center of gravity of the robot rose nearly linearly from a height of 0.4[m] to 0.8[m]. The velocity of the torso begins at 0[m/s] and finishes at 0.2[m/s]. A non-zero motion was selected to simulate jumping, and ensure that higher motor velocities could be achieved. This also has the effect of emphasizing this portion of the motion by ensuring mechanical work is being done even though average joint torques are much lower due to the leg configuration.

The stepping motion was pulled from actual physical experimentation. Bipedal locomotion experiments on the SAFFiR robot have yielded joint torques and velocities during a step. Both standing and stepping are presented in Figure 2-8.

This work only uses trajectories obtained during simulation or experimentation of the SAFFiR platform. Therefor one significant assumption should be addressed. Specifically, the author makes the assumption that the torque and velocity profiles used herein are representative of all humanoids. This is a safe assumption for a number of reasons. First, humanoids by their very nature are very similar. The joints and links are ordered and orientated in very similar fashion. Furthermore the actions they perform are similar because many of the trajectories are driven by the robots form. Standing on one leg for example must be done with the center of gravity directly above the stance foot, resulting in nearly identical loading conditions no matter the robot.

Secondly walking trajectories are by in large also very similar between humanoids. In large part his is due to the kinematic constraints imposed on a robot that walks with its feet flat on the floor. Therefore the one significant limitation of the conclusions drawn from this work is its applicability to robots that actively use a toe to reduce the loading on the knee. However the premise of actuator orientation effecting efficiency and the methods proposed to identify and apply realizable solutions still hold no matter how the robot walks. In some part then, the results are a very indicative example of how the efficiency of humanoids can be improved. Finally, this analysis does not compare or make claims about all legged robots (although the process and optimization developed are applicable).

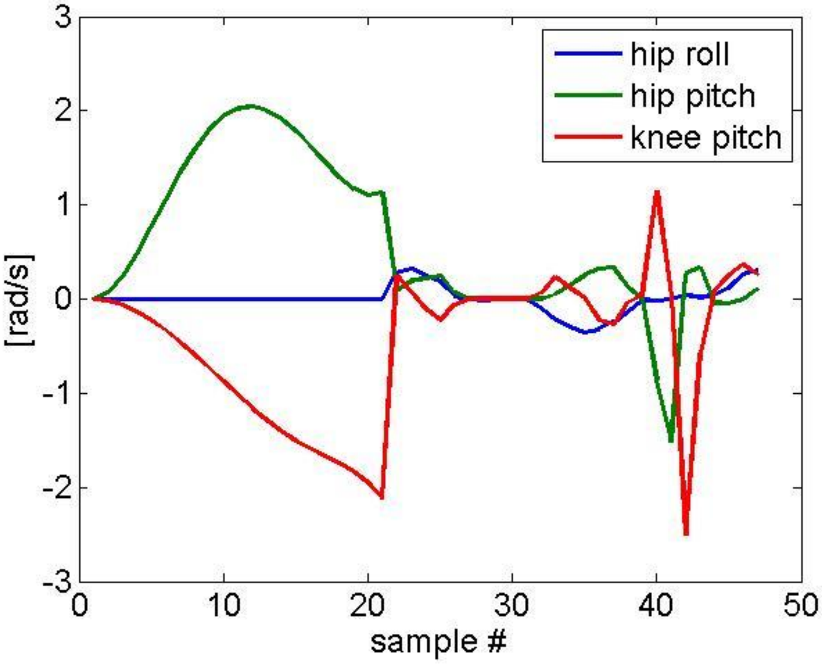

Figure 2-7. Velocity of the hip and knee joints during standing and stepping.

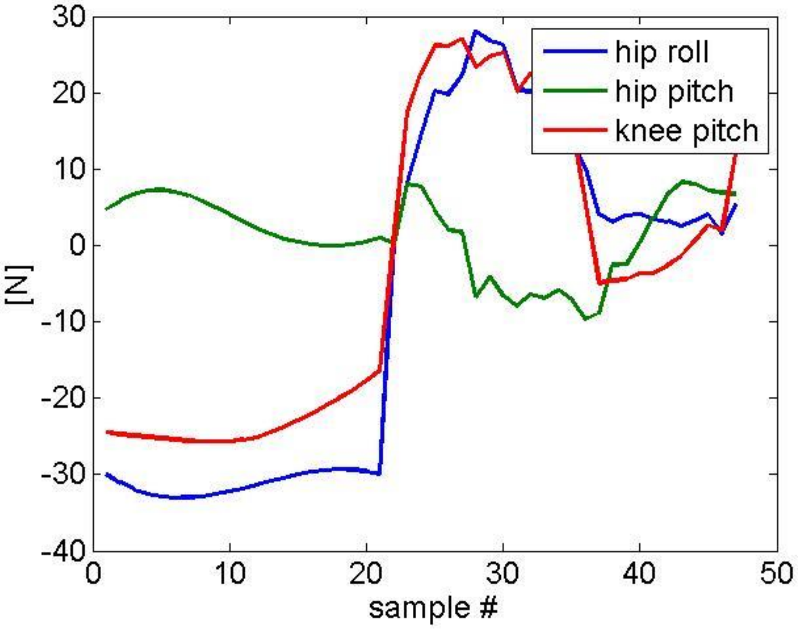

Figure 2-8. Joint velocity and torque trajectories compiled for standing up and stepping.

The optimization seeks to minimize the energy lost during these motions over the entire operation of the robot. To calculate those losses, it is first necessary to better understand the losses in the electric motors that drive all the joints.

2-3-1 Electric Motor Losses

As mentioned above, there are two primary losses within an electric motor, internal friction, and electrically resistive losses. The frictional losses are characterized by the motor manufacturer as the no-load current, which describes the amount of current required to drive the rotor when under no external load. Because current is assumed to be proportional to torque, the no-load torque is given as in Equation 2-1.

$$\tau_0 = k_t * I_0 \tag{2-1}$$

where τ_0 is the no load torque, k_t is the motor torque constant, and I_0 is the no load current. The power lost due to this torque is given as in Equation

$$\dot{E} = \tau_0 * |\omega| \tag{2-2}$$

where $\dot{E}$ is the time rate of change of energy, and ω is the angular velocity of the motor. There are small velocity dependent losses, essentially viscous friction, however these are even smaller and in fact most often not included in the manufacturers specifications and are therefore ignored. Electrical losses are conventionally accepted as the most predominant losses in an electric motor. The motor windings are long and thin wires with significant resistance. The losses within the wire are given as in Equation 2-3:

$$\dot{E} = I^2 R_0 \tag{2-3}$$

where R_0 is the resistance of the wire as given by the manufacturer. The overall power lost to a motor is given by Equation 2-4 [38]:

$$E^{\cdot} = I^2 R_0 + \tau_0 * \omega \tag{2-4}$$

Since current is proportional to torque, this can be rewritten as in Equation 2-5

$$E^{\cdot} = \left(\frac{\tau}{k_t}\right)^2 R_0 + \tau_0 * |\omega| \tag{2-5}$$

where τ is the motor torque. Because of the squared term, it is beneficial to reduce the motor torque; however, at some point the losses due to an increasing angular velocity will increase and present a limit further increasing the gear ratio.

Given the relationships above, the losses associated with electromechanical motors will now be expanded to include the various architectures above. First, Equations 2-5 and 2-18 and 2-19 are combined to solve for actuator losses as a function of joint torques and joint velocities. This yields Equation 2-6:

$$E^{\cdot} = \left(\frac{G \cdot F}{k_t}\right)^2 R_0 + \tau_0 * |H^t \cdot V| \tag{2-6}$$

where G represents the transformation between joint torques and actuator forces. Equation 2-6 simplifies to

$$E^{\cdot} = F^t G^t G F \frac{R_0}{{k_t}^2} + \tau_0 * |H^t \cdot V| \tag{2-7}$$

This result is interesting because it shows that rotating the two axes of the manipulator together has no effect on the current losses. That is, any orientation of axes that at least have the same angle between axes, will incur the same losses due to the internal motor resistance. This is akin to applying a rotation matrix, T, to G. Because rotation matrices are orthogonal, it will drop out of the term: $F^tG^tT^tTGF$. It does have an effect though on the losses due to the motor no load torque or internal friction.

By applying weights to the standing and stepping motion representative of the time spent in each motion, an estimate of the total energy lost can be calculated as a percentage of the runtime of the robot. For this analysis, a weighting of 0.5 is applied to the stepping motion, and a weight of 0.5 is applied to the standing motion. The energy lost equation then becomes as in Equation 2-8:

$$E = \sum_{i=1}^{n} w_i \left({F_i}^t G^t G F_i \frac{R_0}{{k_t}^2} + \tau_0 * |G^t \cdot V_i| \right) dt_i \qquad (2\text{-}8)$$

where *w* is the weight given to the motion, *dt* is the timestep between data points of the trajectory, and *i* is the number of timesteps.

Constraints on motor velocity and motor torque are limited to the manufacturer's specifications. Torque is limited by the motor controller to 0.27[Nm]. Motor velocity is limited to 15000[rpm] or 1570[rad/s] [39]. The enforcement of these constraints is discussed in the particular algorithms' section.

Transmission comparison

The above analysis does not take into account the transmission attached to the electric motor. It can be shown though that the different types of transmissions used in electromechanical robots are comparable and thus can be compared without loss of generality. The three primary types of transmissions employed are harmonic drives, cycloidal drives, and ballscrews.

The efficiency of these three drives is very similar. In ballscrew transmissions, the ball bearings eliminate friction, both static and dynamic, resulting in frictional losses comparable to harmonic drives, or around 90%. The newest iterations of cycloidal drives (as seen by the author but not published yet) similarly employ rolling elements to reduce friction. It is safe to assume that these types of drives will also achieve similar efficiencies simply by the fact that if they were much worse than either ballscrew or harmonic drives, they would either not be used or improved until they did.

Nominally, the weight of the system affects the loads going through the joints, which in turn could affect the amount of energy lost. However, the weight of the linear actuators appropriate for a humanoid robot falls entirely in the range of harmonic drives. The harmonic drive models typically used weigh between 0.9 and 5 [kg] whereas an appropriate linear actuator weighs between 0.9 and 3.2 [kg][48].

While slight gains in weight are possible with linear actuators due to improved structural efficiency, the resulting actuator positions studied would still yield a similar effect, that is reducing the energy lost in the electric motor. It is true however that the optimum actuator position could be different. This is best illustrated by looking at Equation 2-7, which shows that if you multiplied the first term of the right side by a constant and compared the results between two matrices *G*, the same matrix would provide reduced losses under either loading condition (multiplied or not).

For thoroughness, it is also worth making a note regarding the roll of the transmission in the losses that are studied here. In this work, only those losses within the electric motor are considered. There will always be a fixed amount of energy lost due to the mechanical work that must be done to accelerate and decelerate the masses of the robot. Said differently, the mechanical work will always travel through some sort of transmission that loses an amount of energy proportional the total energy passing through the transmission. Since these loses cannot be changed, it is not necessary to consider them in the analysis.

It is theoretically possible for the gear ratio to affect the transmission efficiency. However, this can be represented in the frictional losses in the motor if it is of a great concern. Instead of using the no-load torque listed in the motor specifications, this value could be increased to represent additional losses incurred as a result of having to increase the transmission ratio. The net effect of which is to force the optimizer to favor designs with lower motor velocities, which is the same result if an additional term were added to Equation 2-7 that represented transmission losses.

2-3-2 Additional Constraints

The efficiency of the robot should not be the only determining factor in the specification of the actuator orientation simply because it is not feasible to predict with any certainty all the loads and velocities a humanoid leg will undergo. It is therefore necessary to also specify some baseline performance in terms of joint speed and joint torque that may be desirable for unexpected situations, for example: fall recovery. On the positive side, a wide range of literature exists that demonstrates acceptable joint parameters that have been successfully used on hardware.

A benefit of the optimization approach taken here is the simplicity with which these desired parameters can be incorporated. Specifically, desired joint torque and joint velocities are appended to the list of torque and velocities graphically represented in Figure 2-7 and Figure 2-8. An associated *dt* for these appended values is zero, thus they do not affect the efficiency. Instead, should a configuration cause the motor torque or velocity to be exceeded by these desired joint torques or velocities, an associated penalty will be assessed to that design. In this way, general desired joint parameters are neatly encoded into the optimization

Size constraints are added as the final restriction on the optimization. These are encoded into the limits of the parameters that are optimized. Specifically, the optimization is restricted to search in between the specified limits of each parameter. This is carried by the piece of genetic algorithm code that decodes the chromosome into parameter values. In this optimization, the elements of the matrix H are optimized. Fortunately each element represents a physical parameter and so assigning a limit is easy. In this case each element has units [meters], and represents the effective lever arm a linear actuator would have about the joint axes. This is significant in that designs which require large lever arms will be bulky, take up a large volume, and require very long linear actuators. In this study, the lever arms are limited to 0.1[m] length. If larger “effective” lever arms are desirable, it is possible to increase the leverage in other ways, such as the ballscrew pitch or other transmission mechanisms in the design.

2-3-3 Parameters to adjust

Finally the representation used to describe the actuator implementation is considered. It is again easiest to study the transformation relating actuator forces to joint torques, the matrix H. To reiterate, each element of H is the torque exerted by one actuator about one joint for one unit force output from the actuator. Therefore, each column of H represents one actuator, and each row represents on joint. For example, the element in the 3rd row and 2nd column is the torque exerted by the 2nd actuator about the knee joint (the third joint). The joints are numbered as follows: 1 = Hip roll, 2 = Hip Pitch, and 3 = Knee pitch. Actuators 1 and 2 are nominally the hip actuators, and actuator 3 is nominally the knee actuator.

From above, the matrix *H* of a 2DOF hip is extended to represent both the hip and knee joints as in Equation 2-9:

$$H = \begin{bmatrix} a & b & p_3 \\ c & d & p_4 \\ p_1 & p_2 & e \end{bmatrix} \qquad (2\text{-}9)$$

where the extra column and row are added to represent the knee pitch actuator (column) and knee torque (row). The elements labeled with a *p* indicate that those respective actuators are biarticular across a certain axes. To represent biarticular actuators, the elements of H must indicate that the actuator exerts torque about the joint axes of both the hip and the knee. Assigning a value of zero to any of the *p* elements indicates that the particular actuator associated with that column is not biarticular.

Elements a through d represent the orientation of the hip. Each element of matrix *H* has units [m], and so represents the lever arm about which a linear actuator exerts a torque about the respective joint axes. To impose length and orientation constraints these elements are computed as in Equation 2-10 and 2-11:

$$a = r_1 * sin(\theta_1)\,, b = r_1 * cos(\theta_1) \qquad (2\text{-}10)$$

$$c = r_2 * sin(\theta_2)\,, d = r_2 * cos(\theta_2) \qquad (2\text{-}11)$$

where *r1* and *r2* are the lengths of each leverarm, limited to 0.1[m] as mentioned above. *Θ1* and *θ2* are the orientation of the two hip actuator axes and can take values of $-\pi$ to π . Only half of this spectrum is actually needed to represent the entire set of configuration if no biarticular actuators are present. When one hip actuator is biarticular, allowing for the full range of orientations simplifies the search. In the searches below, element e is limited to 0.08[m]. The knee must rotate the most out of all joints, and so in practice, its lever arm must be shorter than the rest so that the actuator length does not get too long. This would be of a lesser concern if rotary actuators were used instead. The optimization assigned 6 bits to each parameter, discretizing the angle *θ* to 3degrees and the radius to 3[mm], suitable discretization for a full size robot.

Table 2-1. Parameter bounds for the numerical search and genetic algorithm.

Parameter	Symbol	Lower bound	Upper bound	Units
Lever length	r_1, r_2	0	0.1	Meters
Lever angle	θ_1, θ_2	0	π	Radians

Biarticular lever	p_1- p_4	0	0.05	Meters

In any configuration, the matrix G from Equation is computed from matrix H as in Equation 2-12:

$$G = k_g * H^{-1} \tag{2-12}$$

where kg is an extra gear ratio representing the reduction between the force generated by the linear actuator to the motor torque. For rotary actuation, kg would be unity. The losses are then computed as in Equation 2-8.

2-3-4 Genetic algorithm coding

Genetic algorithms have in recent years been used for a wide variety of type and dimensional synthesis and optimization problems. They are unique among search algorithms in that they combine elements of both random and directed search routines. The genetic algorithm described here is applied to the joint orieintation optimization problem. Genetic algorithms use the breeding and evolutionary mechanisms of natural systems to root out an optimal solution, where the optimal is determined by a user defined fitness function. The strength of these algorithms stems from the combination of elements from both random walk type searches and broader and more computationally intensive enumerative type algorithms. The result is a rigorous search which can be used for many types of optimizations of multi-criteria problems.

The algorithm used here is applied to a dimensional type synthesis problem. In this approach, the various parameters of significance, (elements of the matrix G) area coded into a series of bits which make up a chromosome. A large population is then built up of many different chromosomes. Each individual in the population is then scored for a fitness level using the fitness function. Members of the population are then combined and crossed using a mating function which gives priority to the individuals with higher fitness. Mutations are also allowed to occur in every generation. With each new generation, the population is rescored by the fitness function, and the cycle continues. The result is a powerful search algorithm built upon very basic and easy to program functions. A flowchart representing the structure of the genetic algorithm can be seen in Figure 2-9.

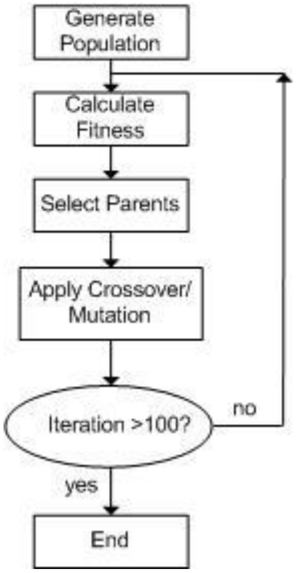

Figure 2-9. Flowchart of the genetic algorithm process.

The specific details of genetic algorithms are described in Appendix 0. Assuming some basic knowledge of genetic algorithms, the specifics of this implementation are laid out here, including the chromosome coding and fitness functions. The GA here seeks to minimize the energy lost during standing and stepping by adjusting the elements of the inverse transpose of the jacobian, *G*. For the simple case here of a 2dof joint, there are four elements. Each element can take a value between -12 and 12, chosen by experience because they would represent a mechanical system with an overly large mechanical advantage. Each parameter is encoded by a six bit number, for a chromosome length of 24 bits.

The fitness value and the energy lost are related by Equation 2-13:

$$f(c) = \frac{-E}{350} + 1 \; when \; E \leq 350 \; and \; f(c) = 0.01 \; when \; E > 350 \qquad \textit{(2-13)}$$

where *E* is the energy lost by the motors. This function was chosen such that no energy lost represented a unity fitness value. In practice it was seen that energy losses averaged around 200[J], and therefore 350[J] was the highest energy In application, the lowest fitness value was 0.01 in order to allow even the poorest performing individuals to be allowed to reproduce, in case there was some valuable genetic information within the chromosome. Additionally, a penalty function was instituted to discount the fitness value if motor torque or velocity constraints were violated. Specifically, if the motor torque or motor velocity during any portion of the trial exceeded their limits, the fitness was docked 0.2.

As implemented here, this mutation occurs with probability of 0.033. The population was seeded with 100 individuals. It was allowed to run for 300 generations. The results showed convergence at this time as seen in Figure 2-10.

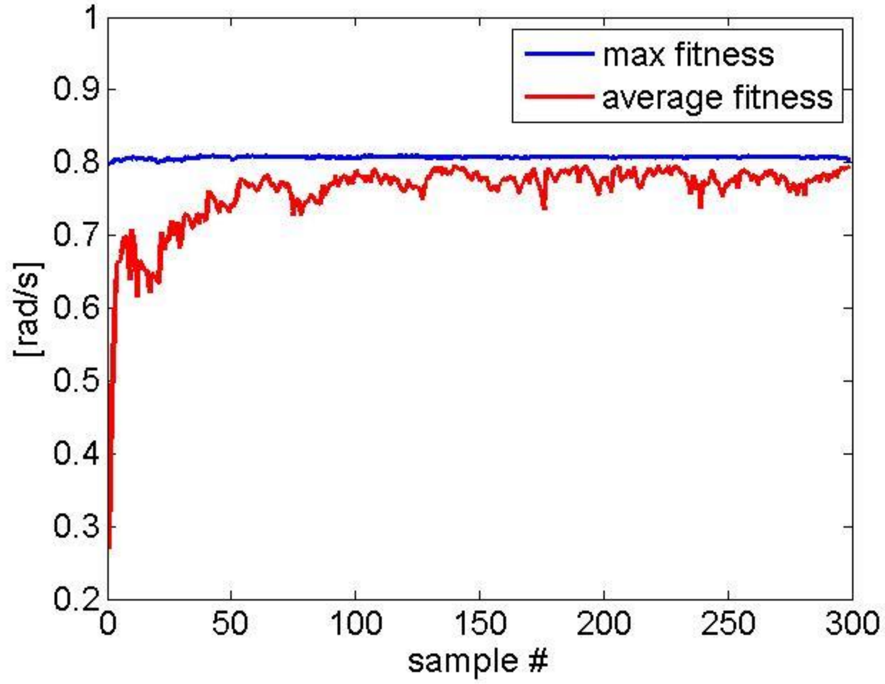

Figure 2-10. Maximum and average fitness values across 300 generations showed convergence.

2-4 Results

An extensive study was undertaken to understand three concepts, the effect of hip axes orientation on robot efficiency, the effect of biarticular actuation on efficiency, and the interplay if any between the hip axes and biarticular actuation. A total of 14 different configurations were tried. The results can be seen in Table ?. The configurations are labeled by their hip and biarticular configurations. The two hip configurations are free and fixed, where free indicates the joint axes can be in any orientation, while the fixed indicates that only the gear ratios can be varied and that the axes are oriented in the x and y directions. The biarticular label indicates which actuator of the three in question is attached in a biarticular fashion and across which joints it spans. For example the knee actuator can actuate the knee, the knee and hip pitch, the knee and hip roll, or the knee and hip pitch and roll. The hip actuators can only be arranged such that they either span the hip, or the hip and the knee.

The results are indicated by the energy consumed, and the resulting actuator axes. The energy consumed is given as Joules, and is that calculated by Equation 2-8. The results also show the portion of that energy that was consumed by resistive losses versus that consumed by frictional losses. The actuator axes are indicated by the columns of matrix H. Matrix H also shows the degree to which a biarticular actuator affects adjacent joints. Each element of matrix H has units [m], and so represents the lever arm about which a linear actuator exerts a torque about the respective joint axes.

Some trends can be drawn from the overall results. The first is that a majority of the losses are due to resistive losses. This is also indicated by the size of the elements of H, many of which are maxed out against their limits. Clearly then, higher efficiencies could still be attained by increasing the motor's leverage within the linear actuator, effectively increasing its “leverarm” over the joint. The other generalization that can be drawn from the results is that with more design freedom comes higher efficiency, which should be expected. The greater number of variables that can be manipulated, the greater possibility there is for improvement. Within these greater trends though, there are micro-trends that are of particular interest which will be discussed next.

The discussion of the results is divided into four sections. The first looks solely at the effect the hip orientation has on the efficiency. A discussion of the configuration with the maximum efficiency follows. With higher degrees of biarticularity, the mechanical complexity increases. Therefore the best improvement in efficiency with the least mechanical complexity is addressed finally, being divided into two sections, one for a biarticular knee and the other a biarticular hip.

Table 2-2: Results from the genetic algorithm.

Name	Hip orientation	Biarticular actuator	Energy lost	Resistance losses	Friction losses	H
A	free	none	79.3	57.6	21.6	0.0812 0.0876 0 -0.0584 0.0449 0 0 0 0.0800
B	fixed	none	82.9	62.7	20.2	0.1000 0 0 0 0.0873 0 0 0 0.0800
C	free	1hip, knee roll	62.4	36.4	26	0.0975 0.0043 0.0373 -0.0223 0.0856 0 0.0484 0 0.0800
D	free	1hip, knee pitch	60.9	37.8	22.9	0.0866 0.0917 0 -0.0500 0.0258 0.0437 0.0500 0 0.0787
E	free	knee roll	66.25	44	22.2	0.1000 0.0098 0.0500 0.0025 -0.0979 0 0 0 0.0800
F	free	knee pitch	78.7	59.2	19.5	0.0985 0.0714 0 0.0174 -0.0630 0.0167 0 0 0.0800
G	free	knee both	70.5	50.4	20.1	0.1000 0.0382 0.0309 0.0025 -0.0907 0.0309 0 0 0.0800
H	fixed	knee roll	66.0432	45	21	0.1000 0 0.0500 0 0.0873 0 0 0 0.0800
I	fixed	knee pitch	82.9	63	19.8	0.1000 0 0 0 0.0857 0.0024 0 0 0.0800
J	fixed	knee both	66	45	20.9	0.1000 0 0.0500 0 0.0873 0.0012 0 0 0.0800
K	fixed	1 hip	62	37.8	24.5	0.1000 0 0 0 0.0905 0 0.0500 0 0.0787
L	fixed	2 hip	61.7	37.5	24.2	0.1000 0 0 0 0.0968 0 0.0500 0.0214 0.0787

M	free	1 hip	61.1	36.7	24.4	0.0981 0.0376 0 0.0074 -0.0892 0 0.0500 0 0.0800
N	free	2 hip	59.1	33.5	25.7	0.0862 0.0612 0 0.0442 -0.0730 0 0.0500 0.0421 0.0749

2-4-1 Hip orientation effect

The control system for comparison of these two optimizations was a traditional 2dof serial hip joint and no biarticular actuators. The two axes were perpendicular and configured in the x and y direction. Two configurations were compared, both which allowed the full optimization of the gear ratios. The optimization was handled in the same way as described above; however, the appropriate elements of H were set to zero to ensure perpendicularity of the joint axes. The resulting optimum matrix, H, for the control system was found to be:

$$H = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.087 & 0 \\ 0 & 0 & 0.8 \end{bmatrix} \tag{2-14}$$

The energy lost by the system was 83[J]. Of that, 63[J] was lost to the electrical resistance of the wire, while 20[J] was lost to the no-load torque losses of the motor. The high values of the elements of H indicate that the optimization was seeking to increase the gear ratio between the motor and the manipulator such that the windage losses were increased and the resistive losses minimized. In addition, two of the elements (in this case the hip roll and knee pitch) are maxed out against their respective limits. This agrees with the split between the two types of losses. The optimization was driving the leverarms of those actuators to larger lengths in order to reduce the higher resistive losses.

The optimization was run several more times but allowed to optimize both the orientation of the joint axes in the plane, the gear ratio of each actuator, and certain biarticular configurations. As a direct comparison to the control system, configuration 2 is discussed next. Configuration B had a freely oriented hip and no biarticular actuators. The matrix H from this system can be seen in Equation 2-15:

$$H = \begin{bmatrix} 0.08 & 0.087 & 0 \\ -0.058 & 0.045 & 0 \\ 0 & 0 & 0.08 \end{bmatrix} \tag{2-15}$$

The losses of the unconstrained system were close but reduced from that of the control with 79[J] of losses accounted for by 58[J] to the resistive and 21[J] to the frictional losses. Optimizing the orientation of the hip axes appears to results in a small improvement, 4% improvement in efficiency.

By in large, this trend continues when looking at other configurations. For example, M (free, 1hip) is slightly more efficient than K (fixed, 1hip); F (free knee pitch) is slightly more efficient than I (fixed, knee pitch). The trend is not absolute though. E (Free, Knee roll) is the same efficiency as H (fixed, knee roll).

The results indicate that the system efficiency can be slightly improved by altering the orientations of joint axes. The improvement is small though, less than 5%. These results are to be expected though. From the loss equation, it was shown that applying a rotation matrix to the joint axes has no effect on the resistive losses. The effect of this rotation will only be seen in the frictional losses, which represent smaller losses. With fewer parameters to optimize, it should expected that lower gains would be possible.

The following section will investigate the implementation of interlinked joints by incorporating biologically inspired biarticular actuators.

2-4-2 Optimization of Biarticular Actuation

The discussion above described several biarticular configurations that are possible with parallel and serially driven hip and knees. To reiterate, one or more hip actuators can also exert torques on the knee, and the knee can also exert torques on the torso about an arbitrary axis. In this section, the various permutations of actuation will be investigated with regard to their effect on the energy lost.

Knee as biarticular

Several biarticular arrangements were tried. Of those, six utilized only the knee as a biarticular element. They consisted of the knee acting across the hip pitch, hip roll, or both the hip pitch and roll directions. Each knee arrangement was tried with both fixed and free hip configurations. There are some very clear trends worth investigating. Firstly though, the most efficient arrangements were E (free knee roll), H (fixed, knee roll), and J (fixed, knee both). They achieved losses all close to 66[J], with 45[J] due to resistive and 21[J] due to friction. All configurations that restricted the knee to act across the hip pitch axes only were less efficient. For example, F and I had losses of 78[J] and 83[J] respectively, no better than the arrangements A and B. Indeed, a closer look at the H matrices of F and I show that the 2,3 element, representing the lever arm of the knee actuator across the hip pitch, is close to zero. Interestingly, unlike biarticular actuation in humans, it is not beneficial to interconnect the hip and knee pitch axes.

The knee actuator was much more effective across the hip roll axis. Given full freedom across both the hip pitch and hip roll axes, the optimization still found the best results when the knee was allowed to act only in roll. This explains the similarity of results between E (free knee roll), H (fixed, knee roll) and J (fixed, knee both). A close look at their respective H matrices shows almost identical values, even though J was less constrained than H. Furthermore this indicates a true optimum was found. These results can be explained by the coupling between the hip roll torque and the knee torque, which during common humanoid motions are often loaded simultaneously.

It is worth mentioning that the knee roll biarticular configuration can incur more mechanical complexity, especially when using rotary or serial configurations as will be explained. The hip of a humanoid should be configured with the roll axis preceding the pitch axis to ensure that the hip does not reach a singularity when the pitch joint is at its extreme range of motion. A biarticular knee actuator must pass torque through the pitch joint to actuate torque. For rotary actuators, this can only be accomplished with a constant velocity joint, as mentioned above. The additional CV joint adds a significant amount of complexity. A linear actuator though can in fact all but negate this problem. The linear actuator should be mounted to the torso at a point that intersects the pitch axis. The knee actuator will then not exert any torque about pitch, while still affecting roll, except for the few cases during with the roll axis rotates considerably.

Hip as biarticular

Four iterations of biarticular hip arrangements were studied. They consisted of the one or two of the hip actuators acting biarticuarly across the knee in either the fixed or free hip configurations. As can be seen from the results table, these easily provided some of the best improvements in efficiency, especially for the nominal increase in complexity. All four arrangements lost only 62[J] or less energy. The best was configuration N (free hip, 2 hip biarticular) which resulted in 59[J] lost, 34[J] from resistive and 26[J] from friction. The energy savings amount to 29% decrease in the control configuration (A).

These particular arrangements are interesting because of their relatively low mechanical cost, especially when implemented in a parallel actuation fashion. When in parallel, both hip actuators terminate on the thigh, and can be made easily biarticular with the pulley arrangement shown in Figure 2-5. In the case with two biarticular hip actuators, there will simply be two pulley arrangements, one for each hip, that lead to the knee and terminate on the shin. The nature of the mechanism will ensure that loads are evenly distributed between actuators.

In a serial actuator arrangement these configurations are less enticing. Configurations K and L and M show that it is most beneficial to couple the roll axis, (or something close to roll in the case of M), to the knee. This coupling agrees with the results from above when the knee actuator was biarticular. However like above, the roll joint should be placed above the pitch to maintain the robots range of motion. To pass power from the first actuator through the second degree of freedom again requires the use of a constant velocity joint.

The importance of coupling the hip roll joints and the knee pitch are supported by these configurations as well. L(fixed, 1 hip) and M (fixed, 2 hip) have almost identical efficiencies and nearly identical matrix representations save for element (3,2). Even though this element is non-zero, and half of the magnitude of element (3,1), it only amounts to an improvement of 0.3[J]. Intuitively this indicates that the hip pitch actuator when biarticular across the knee, and has a not insignificant lever arm, has little effect on the efficiency. Furthermore, when the optimization is given the chance to optimize the hip axes and one hip biarticular actuator, it chooses to make the hip actuator most heavily loaded in roll also act across the knee. This implies that coupling the hip roll and knee pitch is most beneficial.

Allowing full freedom to design both hip biarticular levers and hip axes orientations (configuration N, free hip with 2 biarticular hip actuators), yields an interesting data point. First, this configuration produced the greatest increase in efficiency yet. From its G matrix, one can see that the axes about which the two hip actuators act are not close to the control configuration. Instead, both actuators act with similar leverage across the roll joint, and again nearly opposite across pitch. This configuration has the effect of loading both actuators nearly equally under large roll loads. Similar biarticular lever length (elements 3,1 and 3,2) maximize the amount of torque transferred from both actuators about the knee.

Because all efficiencies in this family of arrangements are similar, choosing a particular one should be done with regards to the mechanical cost. The author proposes that K (fixed, one hip) be implemented has the lowest cost. It maintains a hip actuator configuration that is closest to the classical style, and can therefor borrow the mechanical design from these. From there, only one mechanism needs to be added to couple the roll axis to the knee. The simplest implementation would be accomplished with parallel actuation across the hip, yet could be done serially for a slight increase in complexity.

Maximum design freedom

Two other configurations were tried with 6 degrees of freedom for the optimizer to vary. C (free, 1 hip and knee roll) and D (free, 1 hip knee pitch) were showed similar efficiencies as those configuration with only the hip actuators biarticular. Configuration C gave 62.4[J] of losses and configuration D gave up 61[J]. Oddly, it is only configuration D in which the knee actuator showed better efficiency when actuating the hip pitch axis as opposed to the hip roll. This configuration is very similar to that of M (free, 1 hip), which also posted similar efficiencies. In fact, configuration M is a subset of configuration D. It is possible that the optimization found a local maximum of fitness that achieved similar performance with a very different configuration. The fact that several different configurations offer very similar efficiency is a good indication that the problem has many local maximum. Fortunately, genetic algorithms are well suited to these types of problems, especially when compared to other strictly search based optimizations.

2-4-3 Other Considerations: Range of Motion

A significant difference between serial and parallel manipulators is the available range of motion. There has been much work in literature dedicated to the task of extending the range of motion while maintaining mechanical advantage of the parallel actuators over the manipulator. This is particularly important for the linearly actuated representation used here because of the inherent mechanical disadvantage that arises when a linear actuator is used to drive a rotating lever. Specifically, the angle between the lever and actuator is perpendicular in only one position, and decreases, along with the mechanical advantage at any other position.

Serial manipulators on the other hand do not have this characteristic. They have consistent mechanical advantage over their actuation range. With non-perpendicular axes though, the problem of range of motion crops up too for serial actuation. It is possible that to maintain a certain ROM about the x and y directions, very large rotations about the joint axes are necessary. This remains an open question and should be considered by robot engineers.

2-4-4 Biological Parallels

The optimization results showed that a strong coupling existed between the hip roll joint and the knee pitch joint of a humanoid robot. Coupling the two in several different ways allowed the respective actuators to operate in a more favorable region of their torque/velocity curve. Efficiency improvements upwards of 30% are possible. In some ways the operating characteristics of a human muscle is similar to that of an electric motor. Under isometric loads, metabolic energy is expended with no output work (analogous to the resistive losses). At very high rates of contraction, less force can be exerted (akin to frictional and viscous losses). Given the similarities in actuation, one might look for similarities in the actuator to skeleton configuration.

The human leg is a massively parallel system. Several muscles of the hip stand out as operating similar to the optimal arrangements described above. The tensor facia latae runs from the iliac crest of the pelvis to the a stretch of tendon called the iliotibial tract. The iliotibial tract inserts on the tibia (shin). The tensor facia latae is responsible for hip abduction and knee extension, analogous to a robots hip roll and knee pitch respectively. The other main connection to the iliotibial tract is the gluteus maximus, one of the largest muscles in the human body. It too contributes to hip extension and hip abduction while extending the knee through the iliotibial tract. The action of the gluteus maximus is most similar to that of configuration 13 or 14 above in which the hip pitch and roll are coupled with a strong biarticular action on the knee as well.

2-5 Parallel versus Serial Actuation of the Hip Joint

Two methods of actuating humanoid robot limbs have been introduced, parallel and serial. Serial robots or serial chains are defined as fixed base and end effector connected by an alternating chain of rigid links and actuators with no closed kinematic loops. Parallel robots on the other hand are defined as an end effector and base connected by a number of independent serial chains, wherein each chain typically has only one actuated degree of freedom. Many kinematic loops are present in parallel actuators. In this section we show that both types of actuation can feasibly implement the configurations above. Furthermore it is shown how to interpret the results above into actual actuator orientations for both serial and parallel configurations.

As this work pertains to humanoid robots, it will be focused on a couple specific types of manipulator, either the 2DOF or 3DOF rotational manipulator. Figure 2-11 illustrates both the revolute actuated serial and linear actuated parallel implementations of the 2DOF joint.

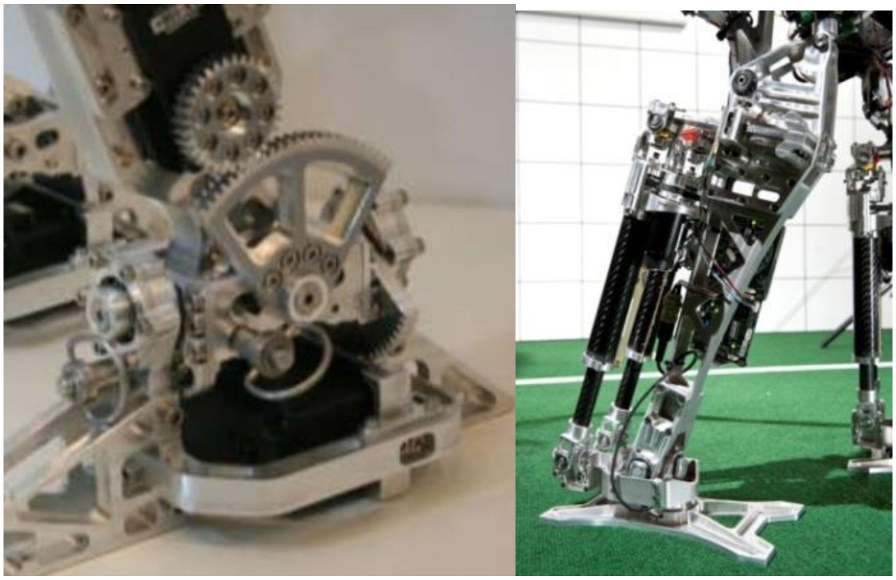

Figure 2-11. Serial actuation on left, and parallel actuation on right of a 2DOF humanoid ankle joint.

Parallel actuators are often chosen over serial robots because they have higher stiffnesses and lower inertia. It however comes at the expense of range of motion. It has been well documented that there exists a duality between the forces of parallel actuators and the velocities in serial actuators[refs]. This duality is most often expressed in terms of the Jacobian, which relates the actuator velocities to the end effector velocities. Using the principle of virtual work, a similar linear relationship can be derived to relate the actuator forces to the end effector forces. For non-redundant actuators, these can be expressed as in Equation 2-16 and 2-17:

$$V = H^t \dot{q} \tag{2-16}$$

$$F = Hf \tag{2-17}$$

where V is the end effector velocity, H^{-t} is the Jacobian, $\dot{q}_\iota$is the joint velocity, F is the end effector wrench, and f_i is the actuator torque or force. Each column of H thus represents the wrench applied to the end effector by one actuator. The focus of this work is constrained such that H is well conditioned, and therefore the inverse exists. It is useful to define the following relationships:

$$\dot{q} = G^t V \tag{2-18}$$

$$f = GF \tag{2-19}$$

where G is defined as the inverse of H.

For the 2 or 3DOF rotational parallel manipulators, each column of H can be given as in Equation 2-20:

$$H_i = \rho_i \times w_i \tag{2-20}$$

where ρ is the vector from the manipulator center to any point on the vector w_i which is the normalized vector along each actuator. For serial manipulators, it is easiest to formulate G row by row. In this case each row represents the normalized direction along which each actuator acts, as in Equation 2-21:

$$G^t{}_i = w_i \tag{2-21}$$

This remainder of this work will explore how these definitions relate to the physical design of the actuators and the effect their value has on the efficiency of a humanoid robot implementation. In particular the parallel and serial implementations of 2 and 3 DOF rotational manipulators will be compared especially how it affects actuator efficiency.

2-5-1 Analysis of 2DOF revolute manipulator

For simplicity, a 2DOF u-joint rotational manipulator is examined first. In the case of humanoid robots, this type of joint can be used in either the hip or the ankle joints. The hip joint experiences some of the highest loads, comparable only to the knee joint. With higher loads, it will consume more energy, and is therefore a good candidate for optimization. This work looks at two types of manipulators, serial and parallel. The axes of actuation are defined as the direction about which a single actuator generates a pure moment load on the end effector. It is hypothesized that by varying the orientation of the axes of actuation, the energy use of a 2DOF manipulator can be optimized for specific robot motions. These axes will be varied for both parallel and serial cases.

In [40], Alici optimizes a 3DOF system with a passive constraining joint in the middle (analogous to a humanoid hip). Dasgupta draws parallels between the kinematic redundancy of serial manipulators with that of the force redundancy in parallel manipulators [41]. Fang analytically determines all possible limb configurations of a 3 DOF rotational parallel manipulator in [42]. The seminal work of Gosselin describes several 3DOF parallel manipulators with both prismatic and rotary actuators that have been used in several robotic applications [43]. Some work has been done studying either hybrid serial/parallel manipulators [44]. Waldron illuminates the duality between serial and parallel manipulators [45], a work which closely follows the results of this work. The particular contribution of this work is to make specific conclusions from the results of Waldrons work as it pertains to multi DOF rotational manipulators and to further the study to include the characteristic of load distribution between actuators. Furthermore, the relationships described by Waldron are extended to include some hybrid configurations which come about as a product of biarticular actuation.

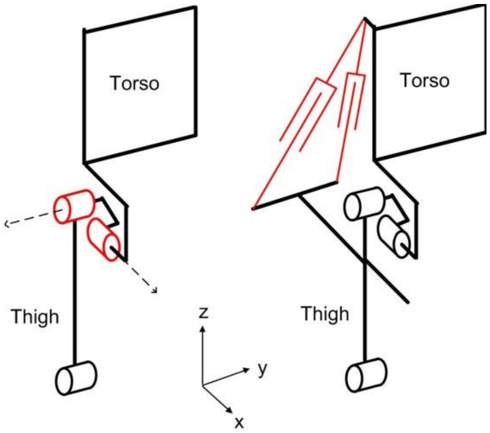

Figure 2-12. Serial configuration on right and parallel configuration on left. Powered actuators are shown in red, and passive joints in black.

Figure 2-12 above illustrated the two simple cases studied here. On the left, is the serial style manipulator, made up of two revolute actuators, the axes of which intersect at the manipulator center. With two actuators, there are four variables to optimize: the orientation of the actuator, as well as the gear ratio of the actuator. These four parameters correspond to the four elements of the Jacobian. The parallel implementation is show on right. Note the passive u-joint at the actuator center that constrains the end-effector (the thigh) to rotation only.

Analogies can be drawn between the parallel and serial configurations. In the parallel configuration, each linear actuator is a serial chain made up of a passive 2DOF u-joint on one end, and a passive spherical joint on the opposite. Along the axes connecting these two centers lies an actuated cylindrical joint. Therefore each actuator generates pure linear forces. A linear force transmitted at some distance from the manipulator center will create a moment load about the center. All linear reaction forces resulting on the base (torso) and manipulator (thigh) are transmitted through the u-joint at the hip center. They are of no consequence to this study. Furthermore the moments of interest are only the fraction that has a component in the plane created by the two axes of the manipulator u-joint (in this case the x-y plane). The vector along which this moment lies is analogous to the axis of action of a serial manipulator. The difference between the parallel and serial cases are the interactions of the two axes of action that result from the differences in architectures. Along with the orientation of the axes, it is this interaction that will be studied.

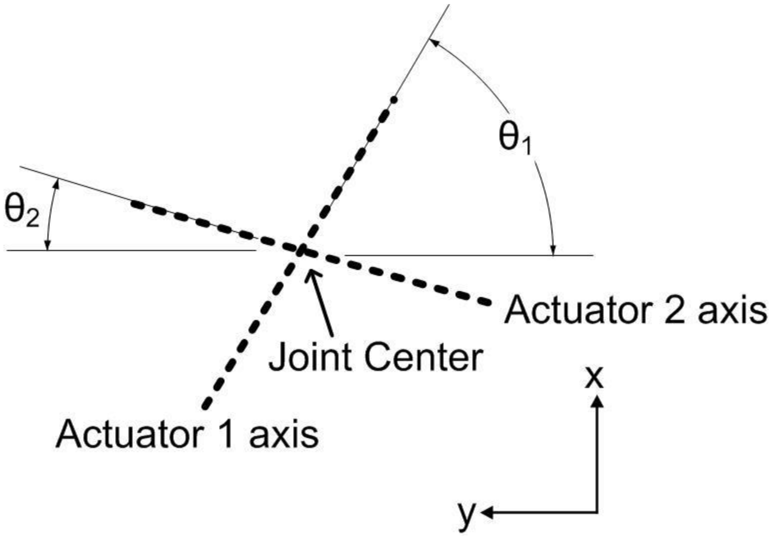

Figure 2-13. Joint orientations different than the traditional x, y directions are investigated.

A few properties of the Jacobian and its related forms will now be explored. The serial architecture will be covered first, and because the actuator forces are of primary interest, the matrix G and H are used as opposed to their transpose. To reiterate, each row of matrix G represents the unit vector about which the actuator rotates. This can result in interesting behavior when the axes are non-perpendicular. Take for example a manipulator with θ_1=0 and θ_2=1.3. G will take the form as given in Equation 2-22:

$$G = \begin{bmatrix} k_{g1}sin(\theta_1) & k_{g1}cos(\theta_1) \\ k_{g2}\,sin(\theta_2) & k_{g2}sin(\theta_2) \end{bmatrix} \tag{2-22}$$

where k_{g1} and k_{g2} are the gear ratios of the first and second actuator respectively. The matrix H defines the output torque applied to the end effector for one unit of input torque about the actuator axes. H is given by Equation 2-23:

$$H = \frac{1}{det(G)} * \begin{bmatrix} k_{g2}sin(\theta_2) & -k_{g1}cos(\theta_1) \\ -k_{g2}\,sin(\theta_2) & k_{g1}sin(\theta_1) \end{bmatrix} \tag{2-23}$$

where the determinant of G is given as in Equation 2-24:

$$det(G) = k_{g2}k_{g1}sin(\theta_1 + \theta_2) \tag{2-24}$$

It follows that:

$$\frac{1}{det(G)} > \frac{1}{k_{g1}k_{g2}} \tag{2-25}$$

when:

$$sin(\theta_1 + \theta_2) < 1 \tag{2-26}$$

which occurs whenever the axes of the two actuators are not perpendicular.

The interesting effect of the inverse process is an increase in the gear ratio of the actuator over the end effector. This is a result of the interaction between the two axes as will now be explained. First, notice that the first column of *H* is perpendicular to the second row of *G* due to the following property:

$$G * H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2-27}$$

Additionally the second column of *H* is perpendicular to the first row of *G*. The first row of *G* is the axes along the first actuator. Each column of *H* is the torque exerted on the end effector by the respective actuator. Therefore it follows that the torque exerted by the first actuator is perpendicular to the second actuator axes. Furthermore the torque of the second actuator is perpendicular to the first actuator axis. Intuitively, this can be explained by the nature of series architecture. Take for example the first actuator in the chain. It will exert a torque about its own axes. This torque must be transmitted through the second actuator to be applied to the end effector. However, any component of torque from the first actuator that is collinear with second is "registered" or must be generated by the second actuator as well. Therefore the component of the torque is eliminated from the effect of the first actuator. Conservation of energy can be applied and will show then that to make up for the "lost" component the effective torque must be increased. Similarly, the angular velocity of the end effector will be increased. A similar analysis can be applied to the second actuator by simply reversing the roles of the end effector and the manipulator base.

This is a very interesting feature of series actuators because the transmission on conventional electromechanical drives is a major source of losses. For some drive such as planetaries, a higher gear reduction results in higher losses. The flipside to this is that the gear ratio is built into the structure of the robot instead. For robots that require a large range of motion, the extra travel required from non-perpendicular actuator arrangements may not be structurally possible. It is however a benefit to series architectures.

The study of parallel architectures is more straightforward. In this case the columns of H are given as the axes about which each linear actuator exerts a torque. *H* then takes the form:

$$H = \begin{bmatrix} k_{g1}sin(\theta_1) & k_{g2}sin(\theta_2) \\ k_{g1}cos(\theta_1) & k_{g2}cos(\theta_1) \end{bmatrix} \tag{2-28}$$

The form of H shows that it is possible to design a parallel manipulator of the rotational type that behaves the same, (distributes loads between actuators in the same manner), as a serial manipulator. The primary difference being the effect of the determinant above. Notably, this means that a similar parallel actuator will need a higher gear ratio reduction than a series manipulator actuator.

2-6 Conclusions

In this chapter, two characteristics of a bipedal robot's hip and knee joints were considered. Those characteristics are the orientation of the hip axes, and the nature of biarticular actuation on the hip and knee. The goal of the work was to better understand the nature of actuation across three usually independent joint axes, the hip roll, hip pitch, and knee pitch axes. These three are the most prominent in the function of the leg, being responsible for the majority of the loading cases as well as the majority of the energy consumption. Notionally too, they are the most difficult for a designer, the hip because of large torque and tight packaging, and the knee because of yet larger torques.

Biarticular actuation was introduced as a possible means of improving the efficiency of humanoid robots, in particular when used with various hip configurations. Biarticular actuators are actuators that span distally separated joints, the hip and knee for example. The formulation from above was extended to include an additional joint, the knee. A genetic algorithm was programmed to find the most efficient of each configuration during the two most common and taxing motions undertaken by humanoids, stepping and standing up. Additionally the optimization took into account desired peak joint torques and velocities to ensure the motor velocity and power amplifier capabilities were not exceeded.

The results showed that dramatic improvement in efficiency of up to 30%. Interestingly, the majority of those savings can be traced to a coupling between the hip roll axes and the knee pitch. Specifically, nearly all the gains can be made by simply coupling the hip roll actuator of a conventional serial hip arrangement to exert torque about the knee as well. This is possible through the aforementioned cable pulley arrangement and parallel manipulators, or using a clever constant velocity joint described in the appendix. More complex arrangements showed slight improvement in efficiency but at higher mechanical cost in terms of complexity. Finally, parallels were illustrated between the human body’s gluteus maxiumus and tensor facia latae muscles in regard to their similarity with the optimum configurations.

The work concluded with a characteristic of a 2DOF revolute manipulator, analogous to the hip. The two DOF can be actuated either serially or in parallel. A formulation was adopted from literature to express the relationship between the actuators and their positions to their effect on the leg. The formulation was described for both serial and parallel cases and as literature would indicate, there is a unique duality between the two. It was shown that for a pure rotational manipulator, the behavior of either a serial or parallel manipulator can be made identical. However, in configurations with non-perpendicular axes, the serial manipulator gains a small mechanical advantage about some axes over a similarly geared parallel manipulator due to the interaction between the two serial degrees of freedom. This is beneficial for humanoids that typically have already large gear reductions. A designer’s choice then between serial or parallel need only be driven by packaging or other constraints such as efficiency.

3 SAFFiR Architecture

Humanoid robots are widely seen to be the next big leap in service robots, simply because they are well suited to operate in an environment already designed around the human form. In order to reach that end, much research and development remains and there are many humanoid robots in existence as research platforms, including, KAIST University's HUBO, Honda's ASIMO, Waseda University's Wabian-2, and Technical University of Munich's Johnnie and LOLA among others [12], [14], [15], [31]–[33], [46]. These existing robots share some similarities, primarily dictated by the compromises within the design of such a robot, namely, designing a powerful, lightweight, and accurately positioned revolute, universal, or spherical joint.

To accomplish such a task, nearly all these designs rely on a harmonic drive unit. They are relatively light, are capable of very high gear reductions, have a high torque capacity, and are free of backlash. But that is not to say that harmonic drives have no disadvantages. The flexspline, by its very nature is flexible and therefore will allow both torsional deflections and resonant vibrations.

Electrically driven ball screw linear actuators are presented as an alternative to harmonic drives for use in Virginia Tech's SAFFIR robot. These actuators use a threaded screw and nut to convert a rotational motion to a linear one. Recirculating ball bearings between the nut and screw reduce friction; the threaded rod allows for high gear reductions; and they are both stiff and widely available. A degree of design flexibility is also introduced with such actuators.

Implementing such reduction units allows a more musculo-skeletal relationship to be maintained between the actuators and joints and also allows for smaller, cheaper, and less powerful motors to be used through optimization strategies which maximize each motor's potential. With SAFFIR, Virginia Tech will introduce a new performance paradigm of humanoid robots.

3-1 Harmonic Drives in Humanoid Design

Harmonic drives have clearly made a niche for themselves in humanoid robotics, and for compelling reasons as will be seen, but they are not without their drawbacks. A drive consists of three parts, the wave generator, a ring gear situated on the periphery called the circular spline, and an internal and flexible spur gear called a flexspline.

This operation affords the harmonic drive with several advantages over more conventional planetary gearsets. Firstly, harmonic drives can achieve very high reduction ratios, up to 1:160 [47]. Also, because there are only three parts which are all mounted coaxially and collinearly, these harmonic drives are exceptionally small and light compared to planetary gearsets of similar reduction ratios.

Positional accuracy is vital for any robot and in particular, for humanoid robots. With that in mind, any gear reduction unit must have minimal backlash. Harmonic drives are known to have nearly zero backlash due to the high number of teeth engaged between the flexspline and the circular spline, up to 30% of the total number of teeth. The backlash on a higher end planetary gearset can be 1-2 [degrees], whereas the harmonic drives exhibit only 1-2 [arc-min]. The large number of teeth engaged also gives harmonic drives a large torque capacity.

Ball bearings between the flexspline and wave generator of harmonic drives allow efficiencies of 90%. This translates to more powerful joints as well as the ability to backdrive the motor.

3-2 Disadvantages

While harmonic drives at first appear ideally suited for humanoid robots, they are not without their disadvantages. Other actuator reductions exist which offer compelling reasons to reconsider harmonic drives. In this study, we suggest linear actuators match nearly all advantages of harmonic drives, while also minimizing the disadvantages.

Although harmonic drives exhibit almost zero backlash, the compliant nature of the flexspline means in can deform under load. This compliance is necessary to allow the flexspline to mesh with and at the same time remain concentric with the circular spline. Even moderate operating torques can cause a notable deflection within the flexspline resulting in positional errors [47].

Flexibility in the flexspline can also result in resonant vibrations. As the gear teeth mesh, torque ripples can interact and excite resonances. In some cases, high amplitude vibrations can be generated.

Nonlinearities also exist between the torque input and output as a result of both the compliance and the frictional losses within the transmission. Ball screw driven linear actuators are able to overcome all of these shortcomings. As will be seen, they exhibit comparable backlash, high efficiencies, light weight, and very high accuracy.

3-3 Linear Actuators in Humanoid Design

Industrial applications have utilized linear actuators for years to convert rotary motions to linear ones. Computer controlled milling machines, military aircraft, and as has been briefly shown, some humanoid robots all utilize linear actuators for their efficiency, precision, and strength.

3-3-1 Linear actuators

Linear actuators combine a rotary actuator and a power screw in a compact assembly sealed from the environment. The rotary actuators are most commonly an AC or DC motor. This motor drives a ball screw which is secured to the housing through one or more bearings. The ball nut is then attached to a telescoping rod which is fixed from rotating. As the screw rotates, it drives the nut and telescoping portion of the assembly towards one end of the housing, thereby lengthening or shortening the entire actuator. The ball bearings eliminate friction, both static and dynamic, resulting in frictional losses comparable to harmonic drives, or around 90%. With such efficiency, these linear actuators can also be backdriven

The ballnuts can be preloaded in a number of ways to eliminate backlash. Typical backlash values can be as small as 0.1[mm]. When used to rotate a lever arm 100[mm] long, this results in a backlash of the joint of only 3[arcmin]. This is a backlash greater than a harmonic drive, but still substantially less than a planetary gearbox. Ball nuts are susceptible to contamination but because the linear actuators are typically sealed units, this is not a major concern.

The weight of the linear actuators appropriate for a humanoid robot falls entirely in the range of harmonic drives. The harmonic drive models typically used weigh between 0.9 and 5 [kg] whereas an appropriate linear actuator weighs between 0.9 and 3.2 [kg][48].

Ball screws are typically precision ground. Precision grinding produces linear actuators that are accurate to 0.1[mm] per meter. Said differently, for an actuator with a 150[mm] stroke, the overall length of the actuator will be within 0.015[mm] of the expected length anywhere in the stroke, meaning the position of the limbs is also highly accurate.

The thread pitch of a typical precision ground ball screw can be anywhere between 3 to 12 [mm]. Using a hypothetical lever arm of 100[mm] again, a 3[mm] pitch screw will result in a maximum mechanical advantage of 200:1. Therefore even without additional belt reduction, a linear actuator can easily achieve the reduction ratios expected of harmonic drive designs.

Considering the weight of such an actuator matches or is lower than a harmonic drive, the backlash is only slightly greater, and the efficiency and accuracy are matched, the reasons for implementing linear actuators are quite compelling. In addition, the designer has more freedom with joint and actuator placement; the mechanical advantage is within a few percent of harmonic drives; and the flexibility and vibration flaws discussed earlier are avoided.

3-3-2 Biological inspirations

With a modest amount of research, one will see that the implementation of linear actuators for limb like structures is quite common. While these can only be actuated in tension, together they act as a linear actuator to rotate the forearm in both directions. In most animal limbs, a similar actuation strategy is employed but with extensive redundancy, that is several contractile muscles used in coordination to control a limb [49].

In comparing the strength of two individuals, an important consideration in determining their maximum strength is the attachment position of the tendon to the actual bone. Of course, the greater the distance between the pivot and the attachment point, the larger the mechanical advantage will be between the muscle and the limb. As a robot designer, this distance is a variable that can be easily changed, and therefore provides the operator an adjustable mechanical advantage should the motor prove not strong enough or the speed not high enough. Furthermore, optimization strategies which maximize each motors potential will be able to reduce the motor sizes while maintaining the robots performance.

Humanoid robots have rigorous power, strength, and accuracy specifications. Those specifications, up to now, have dictated the use of serial style manipulators in all the limbs of current designs, in part because of the performance available from harmonic drives. This paper investigated the use of linear actuators for use in SAFFIR, Virginia Tech's first full size humanoid robot. With such linear actuators, as opposed to harmonic drives, a parallel actuation strategy can be used to increase the power, accuracy, and strength of a robot limb. While both monoarticular and biarticular actuators were studied, only monoarticular actuators are used for simplicity. It was found that biarticular actuators would unnecessarily limit the range of motion. Because of the couple nature of parallel manipulators, the computation of joint torques and velocities is more involved. This paper presented a simple technique to compute both the actuator forces and velocities given joint torques and angular velocities. Such work is important in both a design and optimization algorithm, as well as for onboard joint control of the robot.

3-4 SAFFiR Design

SAFFiR was designed to break out of this mold and investigate and take advantage of both compliant control schemes [23], [50], [51]and a parallel architecture employing linear actuators. It is hoped that this new compliant parallel architecture will enable more robust performance in difficult environments in which SAFFiR and future emergency response robots are expected to operate. This section will discuss the design of the SAFFiR, with emphasis on the lower body, as well as experimental results from walking tests which validate the parallel actuator arrangement.

SAFFiR, as seen in Figure 1, is a humanoid robot that will stand 1.6[m] tall when completed. It features 33 DOF, which includes two 6 DOF legs and arms, a 1 DOF waist, 2 DOF neck, and 3 DOF hands, oriented as shown in Figure 2. The general proportions of the robot can be seen in Table 1 and the range of motion, power, and approximate reduction ratio of each DOF can be seen in Table 2. SAFFiR employs parallelly actuated joints in the lower body; therefore, the available power in each direction is the maximum available assuming the leg is loaded in only that direction. When completed, it is expected that the robot will weigh 40[kg].

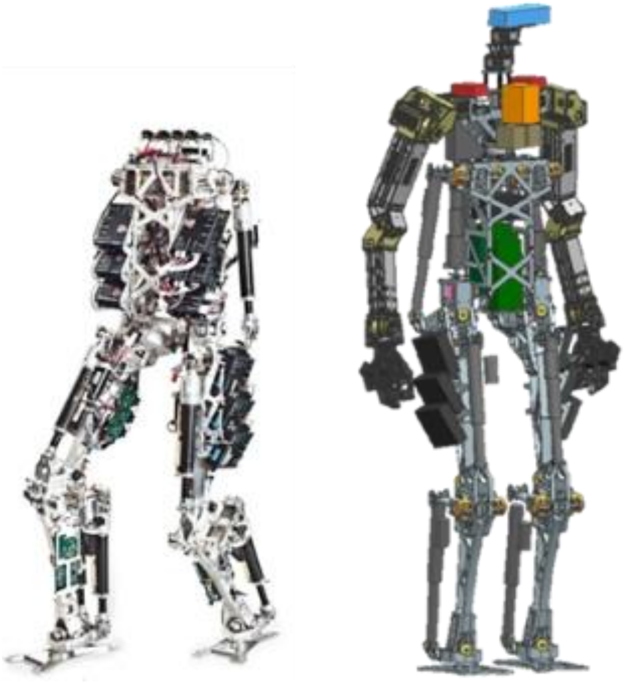

Figure 3-1. Currently state of completion of SAFFiR on left and proposed final CAD model.

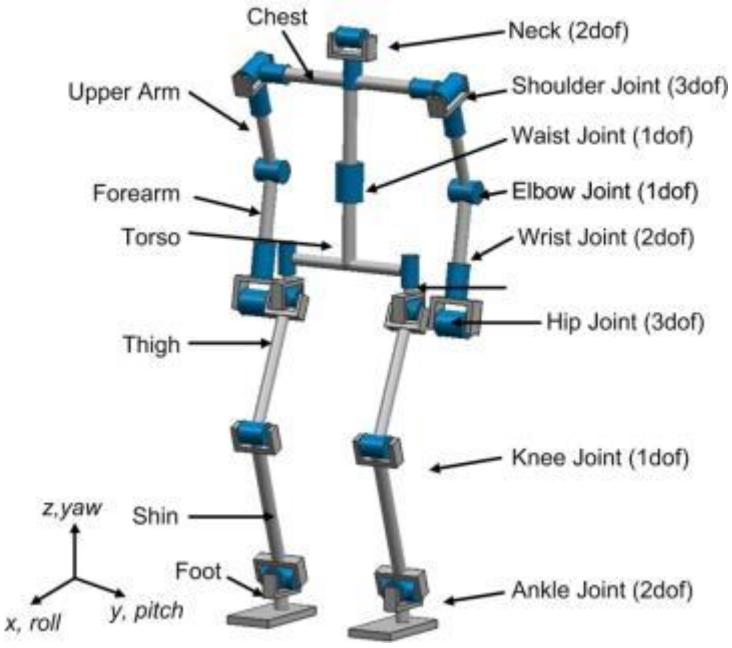

Figure 3-2. Kinematic arrangement of SAFFiR.

Table 3-1. Specifications of SAFFiR.

General	Height	160 cm
	Weight	40 kg

	Total DOF	33
Legs	Thigh length	380 mm
	Shin length	380 mm
	Foot height	40 mm
	Hip width	195 mm
Arms	Shoulder width	460 mm
	Upper arm	250 mm
	Lower arm	200 mm

Table 3-2. Range of motion, mechanical advantage, and power of SAFFiR's leg joints.

	ROM [deg]	**Peak power**	**Peak gear reduction**
Hip Yaw	-25, 10	100 W	230:1 Planetary
Hip Roll	-23, 23	400 W	320:1 Ballscrew
Hip Pitch	-45, 45	400 W	350:1 Ballscrew
Knee Pitch	0, 90	200 W	370:1 Ballscrew
Ankle Pitch	-40, 50	200 W	350:2 Ballscrew
Ankle Roll	-20, 20	200 W	180:1 Ballscrew

The robot is primarily fabricated from 6061 Aluminum because of the materials high strength to weight ratio and ease of machining as compared to other Aluminum alloys. Each skeletal joint of the lower body is supported by two preloaded angular contact bearings to ensure there is no play in the structure. The joints of the upper body rides on preloaded cross roller bearings in each DOF.

3-5 Design overview

Current humanoid designs are almost all serially actuated. This trend is in part due to the limits of high gear reduction revolute actuators based on the harmonic drive. It is worth investigating other architectures as the current trend has several drawbacks. In a serial configuration, actuators cannot share an applied load and must therefore be larger and more powerful. Position errors are summed across the joints which decreases accuracy. A serial architecture also has higher inertias and therefore lower accelerations. Harmonic drives also have a high stiffness and very little dampening which can create troublesome resonances in the robot [47][52] in addition to being counterproductive to the stiff position controlled motions generally employed.

The SAFFiR architecture employs both a unique actuator and actuator arrangement to achieve new gains in mobility and agility as compared to conventional humanoid robots. First, a parallel actuation architecture (in which one joint comprised of several DOFs is spanned by an equal number of actuators) is used to enable higher torques and power in certain motions, improve positional accuracy, and reduce backlash. The parallel architecture also has the added benefit of centralizing and simplifying the robot structure (which reduces weight) as well as consolidating more of the actuator mass nearer to the center of mass. The arrangement of actuators can be seen in Figure 3.

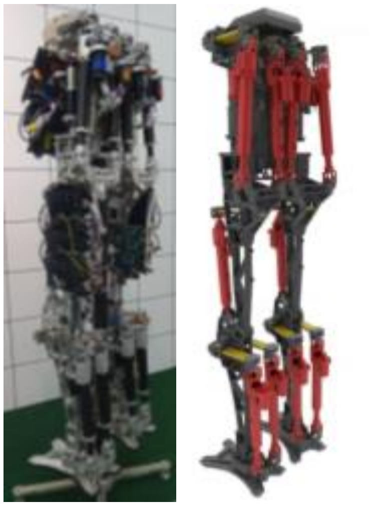

Figure 3-3. Linear actuators shown in red against SAFFiR's structure. Compliant beams shown in yellow.

This architecture drives the selection of actuators. Linear actuators are employed to make use of highly efficient, stiff, and low backlash ballscrews as the gear reduction mechanism and facilitate the addition of an elastic member in series. Titanium cantilevered beams, as seen in Figure 3-4, serve as the elastic members and can be individually stiffened or even locked out. This configurable compliance in conjunction with a very stiff and efficient actuator enable both position and force controllable modes to operate on the robot. Having both modes on SAFFiR gives us ability to experiment with the compliance and its appropriateness at each joint, a topic for future research.

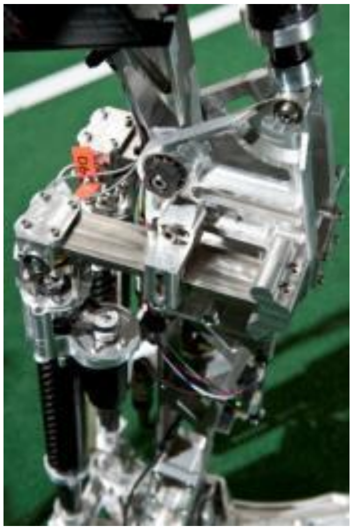

Figure 3-4. Titanium compliant members in series with the two ankle actuators.

Simple levers are used to transmit linear motion to a rotation about a joint. This configuration is limited by the fact that for any linear actuator and lever, the largest range of motion between singular positions (in which the mechanical advantage goes to zero) is 180 degrees. Therefore, in the case of SAFFiR, all the joints were designed to have maximum mechanical advantage at the middle of their ranges of motion so as to maximize the MA over the remaining range. Four bar mechanisms in which the driven and driving link are connected to ground were investigated, but any improvement was on the order of 10% and not worth the additional complexity. Other types of linkages, e.g. the Hoeken's linkage, are under investigation for future revisions.

3-6 Design details

Actuators

Custom linear serial elastic actuators capable of force control [53]–[55]were designed for SAFFiR to save both weight and volume. Lee presents a detailed analysis of the design in [56][57], but an overview is provided here. Each actuator is powered by either a 100 or 200 Watt Maxon EC 4-pole brushless DC motor running at 48 volts. The motor in turn drives a precision rolled ballscrew through a one-stage belt reduction. Power is delivered to the joint through the moving ballnut and attached carbon fiber tube. The ends of the actuator are affixed to the robot through u-joints, which both fully constrain the actuator as well as limit the ballnut rotation relative to the ballscrew.

Accurate and high bandwidth force control is achieved using feedback from Futek™ LCM-200 inline load cells in each actuator and Futek signal conditioning boards. Compliant titanitum beams serve both as the series elastic member, and the mounting point for the actuator on the robot [57]. The beams are relatively stiff at 150-500[kN/m]. Closed loop control of the actuator force is handled by custom motor controller code implemented on Maxon EPOS2 Controllers. In cases for which compliance is not needed, the beams can be fixed and high gain position controllers run for accurate trajectory control. The linear actuator can be seen in Figure 3-5.

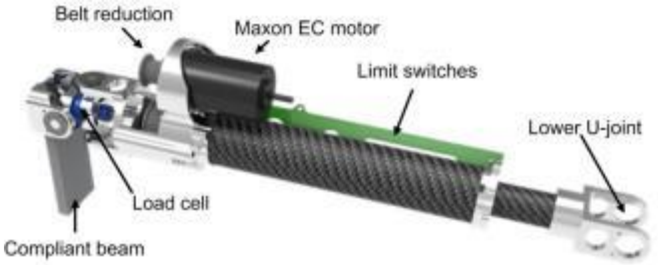

Figure 3-5. Rendering of custom electric linear series elastic actuator with Titanium compliant beam.

The yaw DOF in the hip is the only DOF in the lower body that is not actuated with a linear actuator nor has any compliance. Instead this DOF is controlled in position mode through a planetary gear reduction unit.

The upper body is driven by a new line of actuators from the ROBOTIS Corporation™ called the Dynamixel-Pro™. They are a complete servo-motor package, including motor, controller, and gear reduction. Furthermore the housing is structural, and the output supported by a robust cross roller bearing such that when implemented the motors form an integral part of the upper arm structure. They controllers support CAN and RS-485 communication protocols. A summary of the actuator characteristics can be seen in Table 3-3.

Table 3-3. Actuator specifications as used on SAFFiR.

Type	100W Linear	200W Linear	20W Rotary	100W Rotary	200W Rotary
Weight [g]	653	735	320	642	760
Cont. Force	300[N]	500[N]	5.5[Nm]	21[Nm]	32[Nm]
Max Force	1000[N]	1600[N]	11[Nm]	39[Nm]	48[Nm]
Max Speed	0.2 [m/s]		28[rpm]	35[rpm]	35[rpm]

3-6-1 Hip

As alluded to earlier, the hip joint employs a hybrid parallel/series arrangement. The yaw DOF is driven directly by one actuator, while the pitch and roll DOFs are parallelly driven by two linear actuators. The hip pitch and roll typically require more power, torque, and precision than the yaw, and so were chosen to be driven together to maximize the benefit of parallel actuation during a walking cycle. The highest power draw occurs about pitch during the accelerations of the leg swing, while maximum torque is needed for the stance roll joint to support the remainder of the robot. Because of the length of the leg, the precision of the roll and pitch have a greater effect on foot placement than yaw. The two associated actuators start at the top of the torso and terminate at the top of the thigh. They are nominally placed on the torso to minimize the leg inertia. The assembled torso and hip joints can be seen in Figure 3-6.

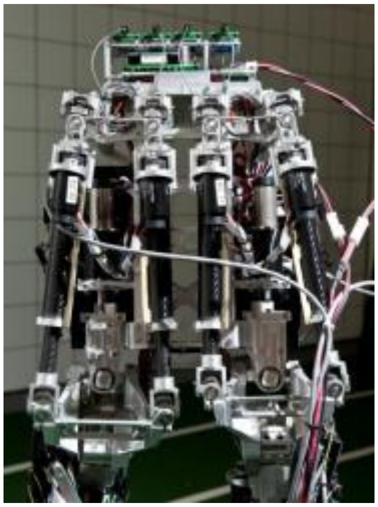

Figure 3-6. Parallel hip actuators as seen on the robot.

The two linear actuators of the hip drive the pitch and roll DOFs through two effective levers (defined as the perpendicular length between the actuator and the axis of rotation). The roll levers are maximized to provide a large degree of torque about this axis while keeping interferences from occurring. With a hip width of 190[mm], the roll lever was made 65[mm] long for a gear reduction of 320:1. The lever arm for pitch was made slightly longer as there were fewer packaging constraints. It is 70[mm] long for an overall reduction of 350:1.

3-6-2 Knee and Ankle

The knee and ankle are more straightforward than the hip. The knee is driven by a 200W linear actuator to handle the higher power and force output required by certain motions such as bent knee walking or squatting. As is, this motor operates at roughly 40% of its continuous capability, indicating the motor is properly sized considering more weight will be added to the robot as it develops.

The lever arm for the knee is 75[mm] long and is biased so that when the knee is bent, the mechanical advantage increases, and when straight, the advantage decreases. This allows for greater knee speed when walking, and more torque when bent for squatting. The knee can be seen in Figure 3-7.

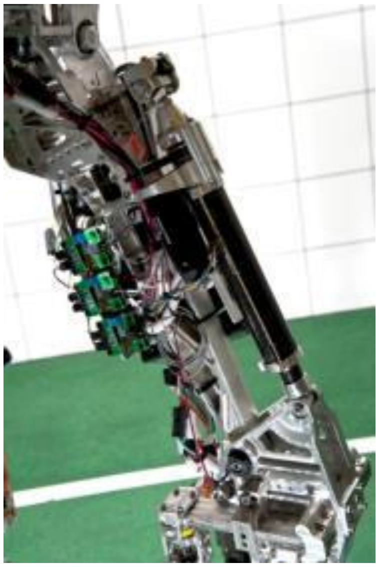

Figure 3-7. Details of knee actuator.

The ankle has two DOFs, pitch and roll, arranged in that order and also controlled in a parallel manner. The ankle joint is unique in that the dynamic requirements of the joint are less severe, i.e. the foot is lightweight and requires little torque to move when in the air, and when on the ground, can only exert as much torque on the ground as the support polygon and robot weight will allow. Therefore, the lever arms can be smaller; for roll they are 37[mm], and pitch they are 70[mm].

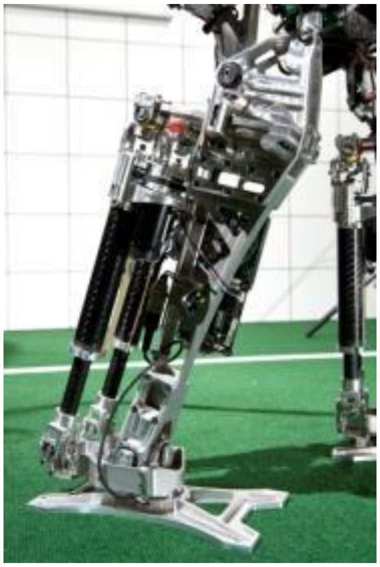

Figure 3-8. Parallel ankle actuation.

3-6-3 Upper body

The upper body consists of two six DOF arms, with a three DOF shoulder, one DOF elbow, and two DOF wrist. The current hands have three fingers, each with one DOF capable of grasping a variety of objects. The upper body is actuated solely with serially arrange revolute actuators in order to achieve the large range of motion necessary. This also ensures uniformity of communication protocols across the entire upper body. The orientation of each actuator can be seen in Figure 3-2. The first two actuators in the arm are of the 200[W] variety, the second two 100[W], and the final two 20[W].

The hand consists of three independent 1-DOF fingers. Each finger is made up of two links, where the second link position is dependent upon the first. The fingers are driven by ROBOTIS EX-106™ motors.

Finally, the neck includes two DOFs, a yaw followed by a pitch to allow the sensors onboard the head a wide field of view. Both utilize the 20W Dynamixel™ Pro motor.

3-7 Sensors and computation

SAFFiR utilizes a distributed control system. High level motion control is done on a dual core 1.6[GHz] FitPC3™ that also handles the kinematic calculations and communication using the lightweight Linux distribution, Arch™. Low level motor control, that which handles motor current, motor position, and actuator force, is done on the Maxon EPOS2™ motor controllers located on the front of the torso and side of the thighs. Communication between the PC and the controllers is achieved through a CAN network running the CANOpen protocol. Two CAN network cards, PEAK™ PCI-Express units, are used for a total of 4 channels. One channel is utilized per leg, one for the force torque sensors (as they run at a lower baud rate), and one channel remains for the upper body. The CAN network is the slow link in the control loop chain; however, cycle times are only 2[ms].

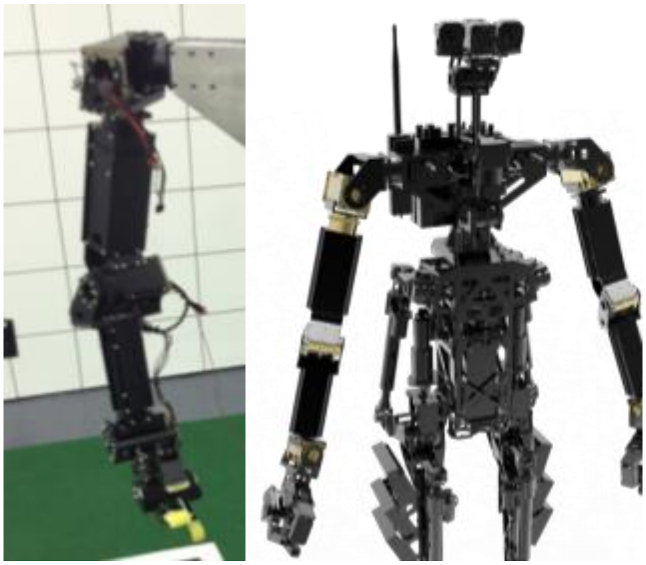

Figure 3-9. Physical arm on left and render of completed upper body on right.

SAFFiR includes an advanced instrumentation suite for both proprioceptive and exteroceptive sensing. The angular rates, positions, and linear accelerations are attained from a Microstrain™ 3DM-GX3-25 AHRS unit. The robots interactions with the ground are monitored via an ATI Mini-45™ 6-axis force torque sensor in each foot. As mentioned earlier, actuator positions and forces are sensed through the motor controllers, and processed into joint position and torques onboard the FITPC3™. Exteroceptive sensors include stereo FLIR™ cameras for navigation, UV sensors for fire detection, a webcam, and a RADAR proximity sensor for range estimation in smoke filled environments. A Hokuyo LASER range finder is located on the chest for navigation in clear air.

SAFFiR is designed to be operated without a tether, therefore all power is onboard. The power system is separated into two isolated subsystems. The first supplies the motors with power, and consists of two Lithium polymer batteries that provide 10 amp-hours of energy at 48 Volts. It is expected that during normal operation this will provide 30 minutes of power. The second power subsystem runs at 24 Volts and supplies all the computation and sensory devices. All batteries are contained within the torso. Separating the actuator and sensory power systems reduces noise and power interruptions that may occur during demanding motions.

3-8 Results

The lower body of the robot has been fabricated as seen in the figures above along with portions of the upper body. A preliminary walking algorithm has been implemented on the robot, capable of walking at 0.25[m/s]. The algorithm uses simple time based trajectories generated online in conjunction with ankle and hip feedback strategy based on the gyro rates to improve stability. The forces generated in each actuator from one leg were measured during two strides and are presented in Figure 10. In this figure, dual support and single support phases are denoted by DS, and SS respectively. The blue line represents the inner actuator, and the red line the outer actuator. Positive values signify the actuator is in tension.

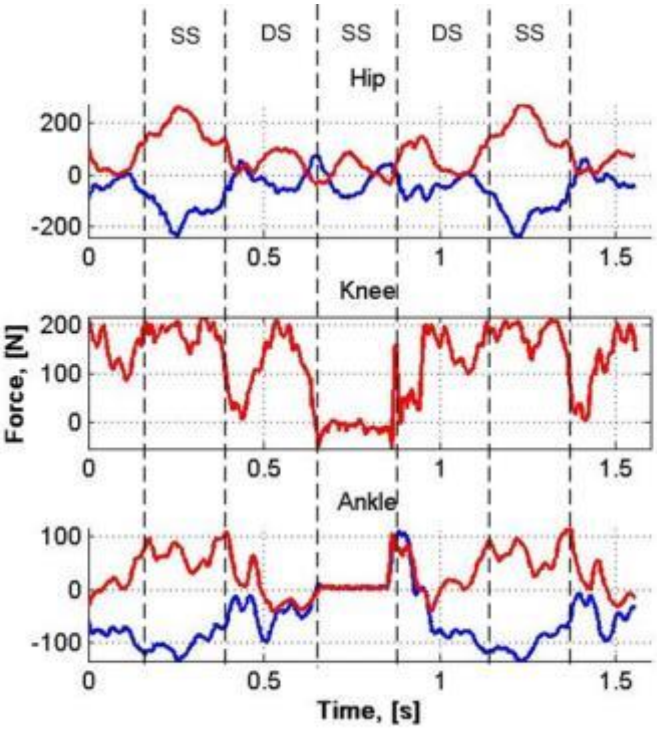

Figure 3-10. Actuator forces as a function of time for the right hip, knee, and ankle during a walking stride.

The data in Figure 3-10 is that of the right leg. Around 0.2[s], the left leg lifts from the ground and the robot goes into single support phase with the right leg acting as the stance leg. The left heel strikes the ground at 0.4[s] and momentarily unloads the right knee; the double support phase follows. Both the knee and ankle plots go to zero during the next single support phase when the right leg is in the air. The right heel strike is clearly visible at 0.9[s] in both the ankle and knee.

Most notable from this plot is the distribution of force between the two hip actuators during the first single support phase in which the right leg is the stance leg. Firstly, it is during this phase that the maximum loading of the hip actuators occurs. Secondly, the actuators are nominally acting in equal and opposite directions, meaning the hip is predominantly loaded about the roll axis. Therefore the parallel actuator configuration chosen complements the loading conditions present. The effort of each actuator to meet the loading requirements of the hip is minimized, which prolongs battery life and increases the load carrying capability of the robot.

4 Walking Algorithm

4-1 Introduction

As stated before, the approach to bipedal walking can be broken down into three approaches, those being the “ZMP”, “Passive-Dynamic”, and “Centroidal” approaches. At present, there is a great divide between the former two, with the Momentum control approach partially filling that gap. Robots utilizing a ZMP based approach are characterized by their rigid electro-mechanical actuation and stiff position controlled joint trajectories. Passive Dynamic bipeds can be almost entirely be grouped on the opposite end of the spectrum. These robots typically have force controllable actuators and one or more completely passive joints. Instead of predefined joint trajectories, they let the natural dynamics of the system dictate the motion of the robot.

Naturally, each approach has merit. Passive dynamic robots are very efficient and walk in a natural and visual appealing manner. ZMP robots though can walk faster, through more varied terrain, and can recover from larger disturbances. However, significant energy has not been spent understanding if and where an appropriate compromise should be made between the two. The walking algorithm presented herein aims to use the best characteristics and design methodologies from successful robots in both categories whilst developing a novel control architecture that encompasses it all. The approach here can be characterized within the Momentum control research area.

Above, the physical hardware that enables this approach, being capable of both position and force control modes, was detailed. This chapter will outline the unique walking algorithm that governs this mechanical system. First, the specific phases and characteristics of a walking cycle are discussed. Secondly, favorable attributes of both ZMP and passive dynamic robots are discussed, the most useful of which will be incorporated into the novel walking algorithm presented here. Finally, the specific approach of the algorithm will be presented.

4-2 Walking Cycle

Legged locomotion is a fundamentally easy task. The motion of a legged being need only achieve forward progress of the center of mass to be technically locomoting. This is achieved in nature through a countless variety of ways. Hopping, running, crawling, shuffling, and swimming can all examples of legged locomotion. Walking is a special case of legged locomotion, and although almost every person on the planet knows how to walk, science has yet to understand precisely how a human walks. There are some characteristics though of a walking gait that are important to be familiar with.

Walking is characterized by the motion of the center of gravity with respect to one or more feet on the ground. In particular, the center of gravity follows and inverted pendulum phase trajectory having maximum potential energy at the top and maximal kinetic energy at the bottom. The top and bottom in this case are with respect to one or more stiff legs which serve to support the center of mass, called the support or stance leg. Forward motion occurs when this inverted pendulum motion occurs in conjunction with an alternating of stance legs between two or more total legs. The legs not supporting the mass are called the swing legs. If the transfer of support between the legs occurs on average one in front of another, then forward progress will have been made. As this research concerns a bipedal robot, a walk is heretofore limited to mean specifically bipedal walking.

The walking cycle is characterized by a unit, the stride. The various stages of and events punctuating a stride can be seen in Figure 4-1. A stride consists of one cycle that begins at the time of contact between one leg and ends when this same leg again contacts the ground after having gone through both stance and swing phases. Two steps make up a stride, where one step consists of the time of contact following the swing phase of one leg with the ground and ending at the time of contact with the ground of the opposite leg. Thus in a humanoid form, wherein the legs are nominally on the left and right of the body, a stride consists of a left step and a right step. In practice, the left and right steps will be mirror opposites of each other.

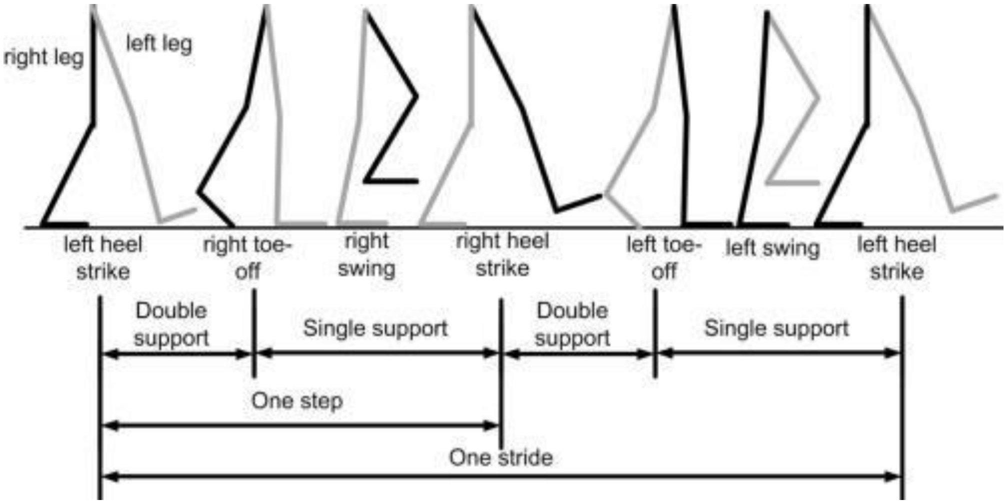

Figure 4-1. Components of a one stride.

Each step consists of several phases. Take for example the left leg. A left step begins the moment at which the right foot collides with the ground, denoted as a heel strike. In this collision, a small amount of energy is lost. For at least a moment, there are two feet in contact with the ground. This phase is denoted as the double support phase. During double support, the weight of the robot is transferred from the right leg to the left leg. In practice, this occurs over a finite about of time the length of which varies according to the design of the walking algorithm. Double support ends once support has been fully transferred to the left leg, which occurs as the right foot leaves the ground, most often following a “toe off” routine in which the right toe propels the robot upward and forward. At this point the single support phase begins.

During single support, the COG behaves as an inverted pendulum. Immediately after single support begins, the COG phase trajectory has its lowest potential energy because the pendulum is furthest from top dead center. Although it lost a small amount of energy during the stride to stride collision, there remains enough kinetic energy to raise the COG to the peak of the stride. After passing top dead center, the potential energy again returns to kinetic and the COG velocity increases. During this time, the right leg, now the swing leg, moves from a posterior position to an anterior one. As it does so, the COG will deviate slightly from an ideal pendulum as both the inertia of the swing leg causes the COG to momentarily increase its velocity, and therefore move further forward than otherwise. The ankle may also exert torques upon the ground to speed up or slow down the COG so long as the foot remains flat on the ground.

When the COG and swing leg have reached a certain position, the right leg collides with the ground. This collision is primarily inelastic, with energy being dissipated through negative work of muscles or actuators, heat, and noise. The angle at which the COG is travelling has a large effect on the amount of energy lost, and therefore plays a large role in the efficiency of a walking stride. In humans, it is shown that reducing this energy loss is the primary function of the magnitude and timing of toe off.

During double support since both legs are on the ground, and at all times one leg must always be on the ground, each leg is in contact with the ground for an average of at 50% of the time. As the time spent in double support decreases, the contact ratio approaches 50%. This ratio is different from that of running, which includes an aerial phase, causing the ratio of time spent in contact with the ground to drop below 50%.

The nominal walking stride has been described. However it is quite common for the length, timing, or magnitude of the events of a stride to vary between various robot walking algorithms. For example, many ZMP based robots deliberately reduce the effect of the stride to stride collision by closely matching the approach velocity of the swing foot to that of the ground. Passive dynamic robots do quite the opposite, in fact using the collision as a control parameter. The following section will catalogue generalizations of both approaches as they pertain to the walking stride.

4-3 ZMP walking stride

The origin of the ZMP based bipedal walking robots can be traced to industrial robots and thus share many characteristics. These industrial robots operate with powerful high gain motors and controllers that can quickly and accurately perform their task. ZMP robots have typically retained this style of actuation. Because they lack a fixed base though, special care is taken when such trajectories are generated to prevent the robot from falling over. This is accomplished by ensuring the center of pressure, or ZMP, of the robot does not leave the support polygon. Upon encountering disturbances, these trajectories are modified on-line so as to remain stable. Because of the nature of their actuation and control methodology, there are some unique characteristics of the ZMP controlled walking stride.

The most significant difference between a ZMP walking robot stride and that as described above, is the path of the center of gravity over time and space. For a number of reasons, ZMP robot designers seem to be quite cautious of singular leg configurations. In particular, while walking the knee is always in a bent position. This allows the controllers to be continuously defined in all three dimensions without the need to make exceptions for a singular configuration. As such, there is no compelling reason for the center of gravity to rise and fall within the stride. It is instead simpler to define a smooth or flat COG trajectory through space. Only a few examples, explicitly called “straight leg walking” punctuate the field of ZMP walking [refs]. Additionally, because the legs cannot be extended until the singularity, long strides are difficult simply due to a nominally shorter leg.

A secondary result of dictating the height of the COG explicitly is the tendency to disregard or eliminate the step to step collisions. A collision is difficult to handle with a conventional controller because they are not continuous events, but rather occur on theoretically infinitesimal time scales. Instead, joint trajectories are carefully calculated so as to both slow the foot to zero at the moment of “heel strike” and orient the foot so as to fall flat on the ground. The result is a much smaller discontinuity of the COG trajectory, and thus greater stability of the controllers. This, along with the near constant COG height and bent leg stance give ZMP based robots their unnatural appearance. This is most dramatically demonstrated by slow motion footage of ASIMO walking and running. Unfortunately, by negating the collision, these robots loose an effective means of control.

Furthermore, because there is no collision, there is no impulsive force to cause the robot mass to transfer support from one leg to another. Instead a long double support phase is used to move the COM forward over the new stance leg. This in and of itself is not a problem; however, because the legs are technically over-constrained during double support and the joints are run in position mode, it is quite easy to burn energy holding joint positions that are inconsistent with the true kinematics. Additional inefficiencies are a result of the bent knee configuration that requires large amounts of torque and therefore current to maintain.

The irregularities of the ZMP stride extend to the single support phase as well. Firstly, to maintain continuous control over the robot, it is necessary that the ZMP does not move outside the support polygon. Theoretically this means is that the ratio between the torque on the ground and the vertical force on the foot cannot exceed a fixed ratio. Practically though, this constraint is more difficult to maintain. No ZMP robot has the necessary hardware to effectively control the torque at the ankle. The impedance of the electro-mechanical actuation is simply not low enough. Therefore the ZMP is controlled through precise position controlled trajectories determined with a full model of the dynamics. Any disturbance unaccounted for will cause a deviation from the trajectories and the foot to roll up on an edge. Most worryingly, this type of control requires precise knowledge of and solid contact with the ground, the two requirements that are very difficult to meet.

The swing leg action of a ZMP robot is also carefully controlled for two reasons. One, the inertial effects of the swing leg must not force the ZMP to or past the edge of the foot. Secondly, a very effective method of ensuring the stability between steps is to actively place the swing foot prior to heel strike. Fortunately, the ability of ZMP robots to accurately position the swing leg gives them great control over the placement of the foot, and therefore the stability of the walking algorithm. While the ability demonstrated by these robots is compelling, it is clear that there are many deviations from an average stride. Those that result in an improvement are strong candidates for a hybrid force/position controlled scheme. The following section will address the opposite end of the spectrum: passive dynamic robots.

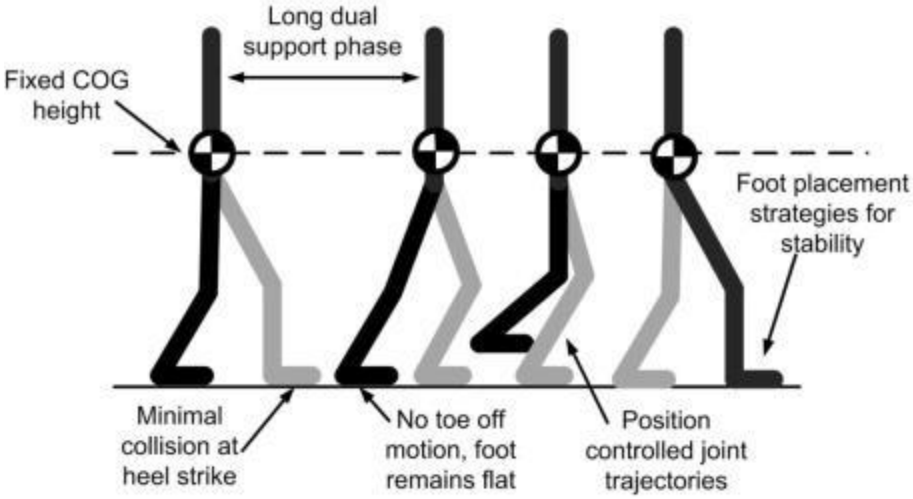
Long dual support phase
Fixed COG height
Foot placement strategies for stability
Minimal collision at heel strike
No toe off motion, foot remains flat
Position controlled joint trajectories

Figure 4-2. The characteristics of a ZMP based walking strategy.

4-4 Passive dynamic stride

There are a large array of passive dynamic based walking machines and robots. They range from completely unactuated versions of McGeer [ref], to the nearly fully actuated ones of Delft [ref]. Because of this range it is much more difficult to make generalizations; however, they do all share some useful guiding principles that can be gleaned from the bunch. This section will cover the general trends of passive dynamic walkers as they relate to the walking stride.

The most apparent difference between ZMP robots and passive dynamic ones is the degree of naturalism with which passive dynamic robots walk. There are two primary reason for this. The first is that a passive dynamic robot's COG is allowed to rise and fall during a stride as a human being would do. To force it to do otherwise would require a significant expenditure of both positive and negative work. As many passive dynamic robots have one or more unactuated joints, this is more often than not simply not possible. The result though is higher efficiencies. Allowing the COG to move as it would naturally, requires no energy. An additional characteristic then is that these robots walk with straight legs and therefore do not require any energy to maintain a bent leg configuration.

The realism with which passive dynamic walkers move is also due to the actuation typically found on such robots. In almost all the examples of passive dynamic robots that have actuators, force controllable actuators are used. These are quite different from the more conventional position controlled actuators found on ZMP style robots. Such actuators are simply those in which the output force of the actuator can be controlled accurately with sufficiently high bandwidth. They have been studied since the 1980's [ref]. This can be accomplished in a number of ways: pneumatic and hydraulic actuators can be very easily made to control their output force and have been used on several passive dynamic machines. Electric varieties are growing in popularity and are seen on Delft's Flame and MIT's M2V2. Importantly, this actuator enables the natural dynamics of the robot to guide the motion (as opposed to the actuators doing so) giving them a much more realistic gait.

Aside from an early toy [ref], the field of passive dynamic walkers originates with the work done by McGeer [ref]. Fundamental to his approach was the effects of the step to step collisions on the momentum of the robot. It was shown that an inelastic collision accurately predicts the velocities of the robot after collision, and goes a long way of explaining the importance of toe-off in a humans gait [ref]. Because the collision dissipates energy, it is a source of damping in the system and has a stabilizing effect[20], [58]–[62]. An increase in velocity causes more energy to be dissipated in the collision, and therefore the system tends to return to an equilibrium. Therefore, all passive dynamic robots utilize this effect to varying degrees. The most pronounced use of collisions as a stabilizing feature is the work of Hobbellen[23], [63]. His robot, Flame, amplifies the stabilization of the collision by actively retracting the swing leg just before heel strike. Low bandwidth actuation makes though makes more complex swing leg trajectories impossible.

This discussion also highlights the actuation of the swing leg. Many passive dynamic walkers actual have an unactuated swing leg. The natural dynamics of a double pendulum are sufficient to cause the swing leg to accelerate forward under the hip and to the correct position for heel strike if the mass properties are tuned correctly. When actuated though, the swing leg is utilized for an interesting effect, that of accelerating and raising the COG [63]. Enough energy can be input in this fashion to compensate for that lost during collisions. While ZMP style robots may also make use of this effect, it is from the passive dynamic robots that it was first fully understood. Because many passive dynamic robots have limited or no swing leg actuation, active placement of the foot for disturbance rejection is difficult or impossible.

Furthermore, because of the impulse of collision, the double support phase of passive dynamic robots is very short, and in many instances or simulations is assumed to be zero. Both the unactuated joints and the kinematic singularity of the straight leg walk further make a long double support difficult or impossible. If the efficiencies of these robots are any indication though, a short double support phase appears to be beneficial. Another source of efficiency is the curved foot often equipped on passive dynamic robots. A curved foot causes the ZMP to naturally progress from the heel to the toe during the stance phase. This has a tendency to smooth out the COG trajectory and result in less energy loss at heel strike. Humans naturally do this and more by rocking up on the ball and toe of the foot just before heel strike [62].

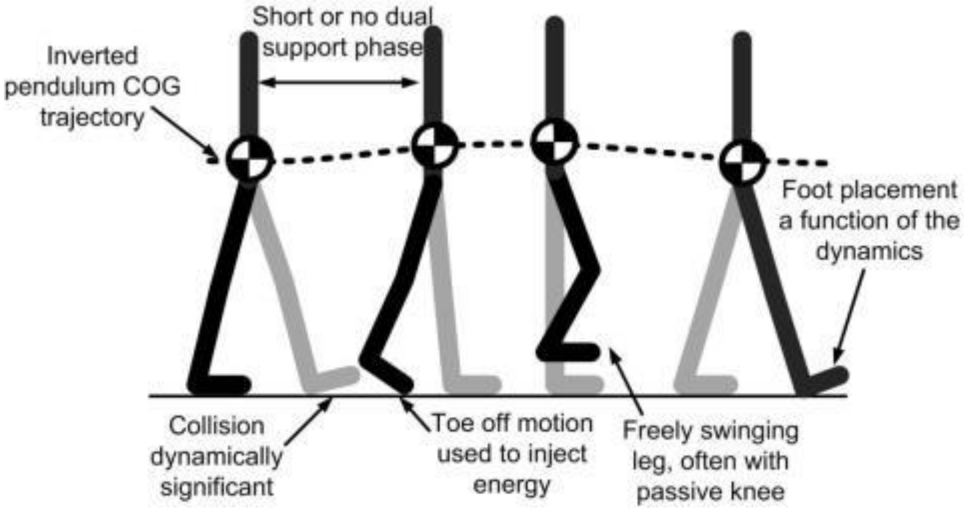
Inverted pendulum COG trajectory
Short or no dual support phase
Foot placement a function of the dynamics
Collision dynamically significant
Toe off motion used to inject energy
Freely swinging leg, often with passive knee

Figure 4-3 The characteristics of a passive dynamic walking stride.

Collisions play a large role in the energetics of each step much like passive dynamic bipeds. This is because at heel strike, as demonstrated by passive dynamic walkers, a significant portion of energy is lost. This must be accounted for when predicting the phase portrait for the latter half of the step. Specifically, there will be a discontinuity between the phase portrait at the moment before heel strike, and the moment after representing an instantaneous loss of velocity and energy, as can be seen in Figure 4-4. Collisions can also be utilized as a control input, as demonstrated by the work of both McGeer and Hobbelen explained above.

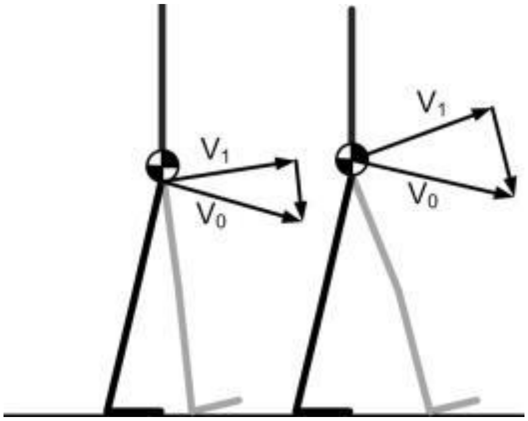

Figure 4-4. Variations in stride length affect the energy loss during stride to stride collisions.

4-5 Hybrid approach

Clearly much is to be learned from both ZMP and passive dynamic style walking bipedal robots. A primary goal of this work is to investigate a hybrid approach. The best attributes of both the mechanical and control systems of either style will be used with the hope that a robust walking algorithm be developed that has both the versatility of a ZMP robot and the simple and natural motion of a passive dynamic robot. In this section, such an algorithm and the hardware necessary to implement it will be proposed.

4-5-1 Controller Selection

As discussed, an actuator was specifically developed for SAFFiR to enable both position and force control. The actuator is an electromechanical unit composed of a Maxon brushless motor that drives a ballscrew reduction system. If mounted very rigidly to the robot frame, these actuators can provide very accurate positioning of the joints. When mounted to the robot through a series elastic spring, very accurate force control can be enabled. The method of control for each joint in the lower body must then be chosen such that it is appropriate for both its use and how it will be controlled.

The purpose of each joint in the leg drives the selection of the controller. From an understanding of the walking cycle and how it has been approached in the past, one can discern the purpose of each joint in that cycle and thus the most appropriate controller. Starting from the ground, and moving up the leg: the ankle serves to support the robot and apply torques to the ground. In doing so, these torques can produce or absorb energy during the stance phase of a step. The ankle must also absorb the collision forces during heel strike. It is desired that SAFFiR be able to walk on unstructured terrain, therefore the ankle must be able to quickly and easily adapt to the inclination of the terrain. The ankle serves little purpose during the swing phase. For these purposes, a force controlled ankle is more suitable. Energy can be controlled through the force times displacement curve. Collisions are unlikely to damage the actuators when set to zero force. Similarly, the foot can freely move into any position dictated by the terrain during heel strike.

The knee performs important functions during both the swing and stance phase. During swing, the knee bends to lift the foot clear of the ground and any obstructions. Just before heel strike, the knee extends. During a human gait, the knee momentarily flexes during collision to soften the impact. The knee stays locked in the extended position during stance phase to support the weight of the robot in passive dynamic walkers, but remains bent for ZMP based robots. Unlike the ankle, the case for either force or position control is not as clear. Force control could accomplish the tasks required and has been shown to work. However because this robot must also perform other tasks besides walking, position control was as a more proven method of control.

The hip joint serves to connect the legs to the torso. It is responsible for accurately positioning the foot for heel strike during the swing phase. Inaccurate placement of the foot will introduce disturbances into the system, and destabilize the walking algorithm. Furthermore, the hip joint has large inertia's on either side of the joint, the leg on one side and the torso and opposite leg on the other. While research shows a force controlled hip is possible, it required considerable feed forward control to do so accurately. Therefore the hip is most suitable for position control.

The hybrid approach here implements force control on the ankle, and position control elsewhere, combining portions of both ZMP and passive dynamic based robots. The distribution of the two controllers can be seen in Figure 4-5. The future of robotics though points to more widespread use of force control because of its compliant nature, making it safer when in proximity to humans. Should the above approach be successful, it can be easily extended with force control later when such controllers are more rigorously understood. With these selections in mind, the following section will identify specific control themes to be developed to solve the stepping stone problem.

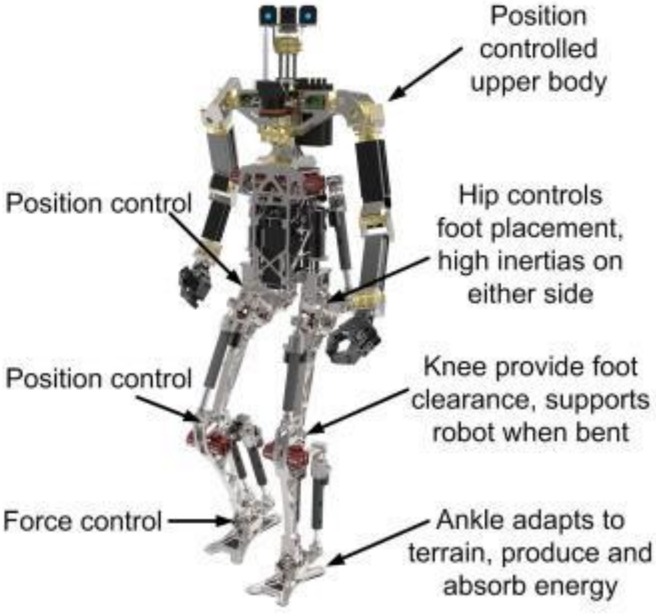
Position controlled upper body
Position control
Hip controls foot placement, high inertias on either side
Position control
Knee provide foot clearance, supports robot when bent
Force control
Ankle adapts to terrain, produce and absorb energy

Figure 4-5. SAFFiR runs a mixed force and position control architecture.

4-5-2 Walking algorithm features

As with any research, this project builds off of and extends the work of others in the field. In this section, the general features of this novel walking algorithm will be outlined. The individual features will be combined into a cohesive algorithm that both directs and stabilizes a bipedal walking gait.

To begin, the problem statement is reiterated. Specifically, there exists no walking algorithm that allows a robot to take a stride to a designated step location with arbitrary realizable initial and final velocities. The goal of this research is to solve the “stepping stone” problem. Literature and intuition makes it clear that a successful algorithm will have a strong balancing capability to handle disturbances and model error. It must also maintain the flexibility and extensibility to make arbitrary steps. That means that gaits generated as part of a limit cycle are not useful, as they cannot be easily extended beyond a small step size. ZMP based algorithms are generally computationally intensive, and a solution to arbitrary steps difficult to find. Therefore this work proposes a new intuitive over-arching control mechanism through which successful principles from can be incorporated.

The over-arching mechanism through which the robot will be controlled is simply the energy of the robot. In this light, stability of a step or walk can be ensured by carefully controlling the energy of the system. As long as the transfer of energy from potential to kinetic is limited, such that the robot does not gain more kinetic energy at the beginning of the step than it can lose at the end, the robot will not fall over. Other advantages include: trajectories through space can be generated through the phase portraits of the center of gravity. This method does also not exclude the use of more conventional controllers, such as those needed to move the swing leg, balance the torso, and move the upper body. As will be seen, this approach is amenable to ZMP stability criteria and aspects of passive dynamic walkers, yet has the advantage of being relatively simple to understand and all inclusive. The specific sub-controllers through which this architecture is implemented will now be addressed.

There are three primary levels of controllers in this algorithm. At the top level exists the center of gravity trajectory controller. The purpose of this controller is to guide the COG through the phase portrait of an inverted pendulum acted upon by a torque at the pivot. Three dimensional walking will be accomplished through the coordination of the phase portraits of both the sagittal and frontal planes. Because this approach allows and encourages the COG to rise and fall during a step, it shares some aspects of passive dynamic walkers and natural bipeds.

The primary means of controlling the trajectory of the COG will be through the torque generated by the force controllable ankle joints. An effective control law will be the devised to accomplish this. This approach has the benefit of also enabling the stability criteria for ZMP based algorithms is met by limiting the torque applied to the ground such that the center of pressure does not move beyond the support polygon. Doing so will allow a unique comparison of the two methods. For example, how the position and torque profiles of a force controllable ankle compares to that of a position based one. Effectively then, this work aims to separate the ZMP criteria into its two primary components: the COG and the torques about the ankle joint, as seen in Figure 4-6.

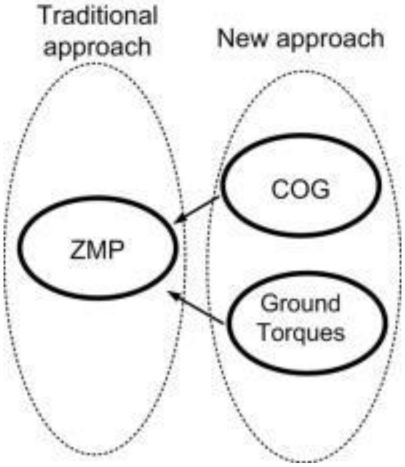

Figure 4-6 . Our approach to walking seperates the ZMP principle into its two components, the COG trajectory and corresponding ground reaction forces.

Utilizing the collision dynamics as a method of control, requires accurate swing foot placement. Any error in the position of the foot at heel strike will greatly affect the energy lost, and therefore the effectiveness of the controller. Historically, accurate foot placement is not a strength of passive dynamic walkers, but is accomplished successfully by stiff position controlled hips, such as those seen on ZMP robots. A unique benefit of this actuation though is the ability to easily modify the trajectory to account for model and sensor error in the terrain. Specifically, swing leg retraction can be extended to include lengthening such that the disturbance due to ground height variation is minimized.

Forward progress when walking is of course only achieved if one leg alternately swings forward of the other. Having a fully actuated leg results in a very large inertia of that leg, and therefore the swing phase can have a significant impact on the dynamics of the robot. This work aims to reduce the disturbances that happen as a result and more importantly utilize this effect to improve the stability of the walking algorithm. For example, by carefully timing the acceleration and deceleration of the swing leg, it can both accelerate the COG early in the stance phase, and decelerate it at the end, thereby increasing the control authority of the ankle. Similar strategies will also be employed to control the torso based on the effective strategy readily seen in humans.

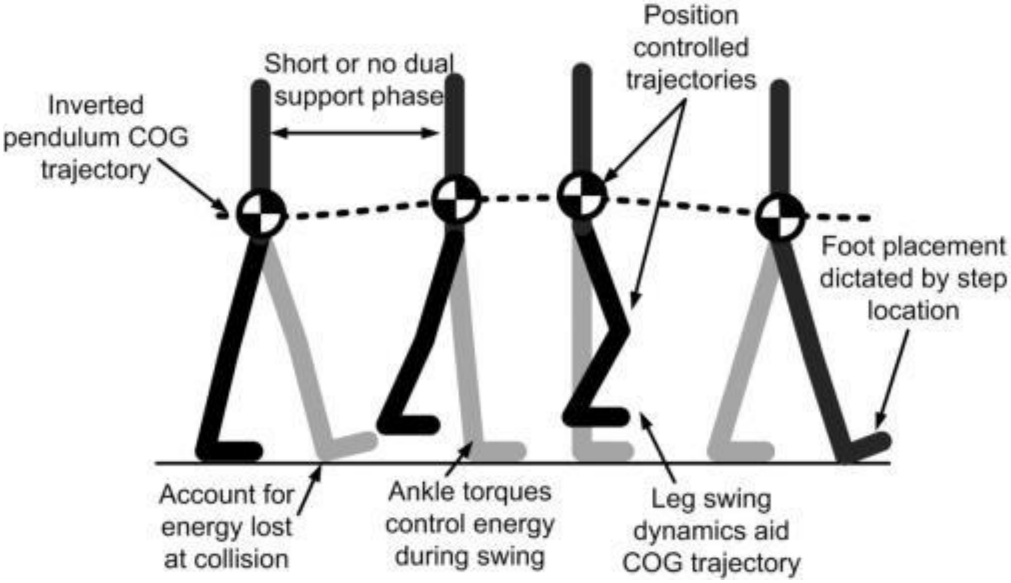
Position controlled trajectories
Short or no dual support phase
Inverted pendulum COG trajectory
Foot placement dictated by step location
Account for energy lost at collision
Ankle torques control energy during swing
Leg swing dynamics aid COG trajectory

Figure 4-7. Characteristics of the walking algorithm implemented here.

4-6 Balance Controller

Once the physical hardware is completed, the development of the walking controllers follows. In this case we use several simplified dynamic models to gain understanding into the underlying principles affecting a stride and how to observe and control them. In this case the control loops are implemented through the use of either position or force feedback loops. The following sections will describe one very important portion of the walking controller, the balance controller. Higher level controllers are then built on top of the balancing controller to implement robust walking.

A stable walking strategy is built around a robust balancing controller. The balancing controller is responsible for maintaining a certain Center of Mass (COM) position with respect to one or both feet. It is therefore if not the heart, at least a major organ of the walking algorithm. Its function is to stabilize the COM about either a stationary setpoint or desired trajectory against all disturbances. Therefor the same balancing controller is used to maintain a desired COM trajectory during the walking gait as well while the robot is stationary. The controller is able to accomplish this via two control signals to the system, the first being the ankle torques, and the second being torso or swing leg windmilling. Both methods will be discussed in detail.

The following sections will outline exactly how the balance controller is implemented beginning with the calculations necessary to accurately compute the COM. The COM velocity is less straightforward due to noise in the sensors, particularly the IMU, and thus requires an observer for accurate measurement. The design of the COM measurement and observer is covered in Section 4-7. Two controllers run on the COM signal. The first controls the torque delivered to the ground via the ankle joint. This is the most effective and direct way of controlling what is basically a tall inverted pendulum (without taking a step or moving the support polygon). The design of the Ankle torque controller is covered in Section 0.

Extending the analogy of an inverted pendulum, a torque about the pendulum pivot is not the only way of balancing if additional degrees of freedom are present. Many control students have balanced reaction wheel pendulums using the "flywheel" technique[64], in which torques can be generated by accelerating a reaction mass atop the pendulum. Humans do a very similar action, in which they rotate their torso and a free leg about the opposite hip joint to recover from a shove. A similar strategy is being developed for SAFFiR. The design of this controller and its performance is discussed in Section 4-13.

4-7 COM Estimation

The COM is the point in space about which the sum of mass of the entire body is balanced. The COM plays an important role in the stability of all robots, but in particular humanoids. Humanoids have a very high COM, on average positioned at roughly half of their height. In combination with the relatively small support polygon defined by one or two feet, this results in a robot that is often only millimeters away from falling. Therefor accurate estimation and control of the COM with respect to the support polygon can dramatically increase the stability of the robot.

On the face, the computation of the COM is not difficult. In reality, sensor noise complicates the problem. In this case, only the *x* and *y* position of the COM are computed, as they have the largest bearing on stability. Additionally, due to the limited range of motion of SAFFiR, the height of the COM actually varies very little, and thus the height is assumed to be constant. The *x* and *y* position of the COM are given by Equations 4-1 and 4-2.

$$x_{COM} = \frac{\sum_1^n m_i * x_i}{\sum_1^n m_i} \tag{4-1}$$

$$y_{COM} = \frac{\sum_1^n m_i * y_i}{\sum_1^n m_i} \tag{4-2}$$

where m_i is the mass of the i^{th} link, *n* is the number of links, and x_i and y_i are the location of the center of mass of each robot link. The mass of each link and its respective location on each link are computed using the CAD model. Then the position, x_i and y_i, are computed using the forward kinematics of the robot in the current configuration. Naturally, a concern when using this method is the accuracy of the estimated mass and its location from the model, as typically the model differs from the actual hardware is some way. Therefore the computed COM was checked against the center of pressure (COP) in a number of static configurations. Poses included both one footed and two footed stances. The results of which can be seen in Table 4-1.

Table 4-1. Measured COP location as compared to the estimated COG position in [mm].

Pose	Estimated and Measured Positions					
1	-0.1	Measured *x*	Error *x*	Estimated *y*	Measured *y*	Error *y*
2	25.8	-0.1	0	5.6	2.0	-3.6
3	11.2	30.9	5.1	2.2	0.5	-1.7
4	50.0	11.5	0.3	1.4	-0.9	-2.5
5	6.6	60.5	10.5	-0.8	-0.8	0
6	9.4	8.1	1.5	-17.7	-29.2	11.5
7	49.2	9.0	0.4	24.3	27.6	3.3
8	27.2	58.3	9.1	-24.2	-25.8	1.6
9	44.6	29.0	1.8	-30.9	-25.6	5.3
10	20.3	45.8	1.2	1.9	-0.8	-2.7
11	14.2	27.2	6.9	32.6	23.7	-11.1
12	18.0	21.0	5.8	-0.9	-1.2	-0.3

With a suitably good estimate of the position, the velocity of the COM can be computed using a simple first order digital derivative filter with a cutoff frequency of 30[Hz]. This however leads to a good deal of noise in the signal, as can be seen in Figure 4-8, which is problematic for any feedback signal. The low cutoff frequency also introduces more phase lag than is desirable.

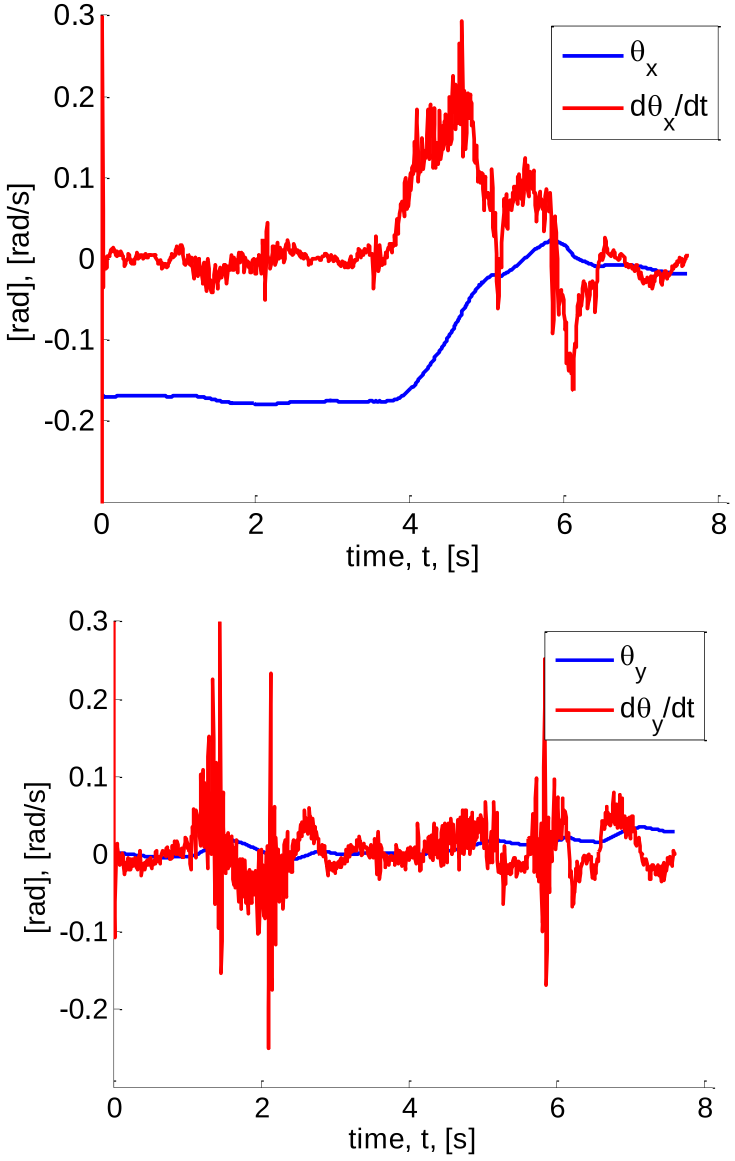

Figure 4-8. First order derivative filter with 30Hz cutoff frequency is used to measure the COG velocity. The plots above show the angle and rate of change about *x* and *y* respectively.

It is necessary to improve the COM velocity signal for better stability. Fortunately, with the multitude of sensors onboard SAFFiR, it is easy to replace the filter with a Luenberg observer. This technique eliminates spurious and noisy readings by comparing the measured values, with expected values computed using the commanded inputs and some representative model. Should the measurement fall too far from the expectation, its weight is decreased and thus the noise reduced. In this case, the robot is modeled as an inverted pendulum with mass *m*, and inertia *I* as seen in Figure 4-9. An observer also requires an input or control signal which in this case is the torque exerted about the pendulum axis by the COP. The observer estimates the angle between the ground and a line connecting the ankle to the COM. From here, its linear velocity, can also be computed as will be shown later. While in single support, the robot actually behaves as 2 dimensional pendulum, being able to move freely in two directions. For simplification, these two directions are assumed to be independent and computed separately. The derivation below though is similar for either direction.

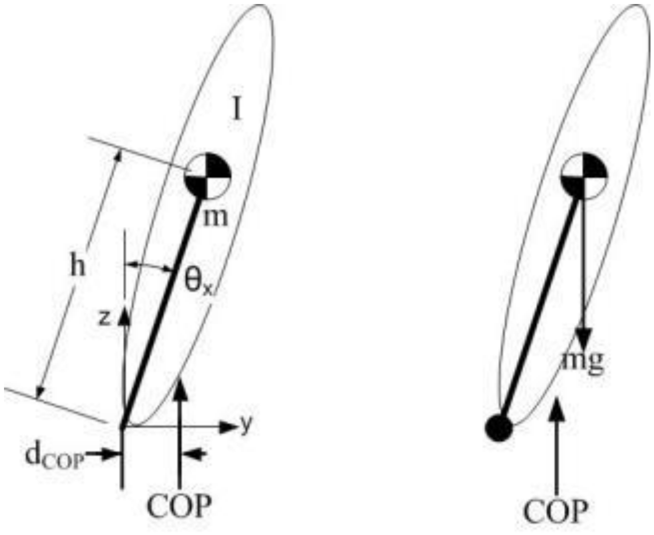

Figure 4-9. Representative model used in observer to improve velocity estimate.

The full dynamic model of such a system is given in Equation 4-3:

$$I\ddot{\theta} = mghSin(\theta) + u \tag{4-3}$$

where h is the height of the COM above the ankles, I is the inertia of the robot about the ankle, m is the mass of the robot, and g is acceleration due to gravity. Because of the limited range of motion of the robot neither the inertia nor the height can change significantly during the walking motion; therefor, both are assumed to be constant. Linearizing Equation 4-3 and putting into state space form yields Equation 4-4:

$$\begin{bmatrix} \dot{\theta_x} \\ \ddot{\theta_x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ mgh/I & 0 \end{bmatrix} \begin{bmatrix} \theta_x \\ \dot{\theta_x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/I \end{bmatrix} u \tag{4-4}$$

Or more simply as in Equation 4-5:

$$\dot{\theta} = A\theta + Bu \tag{4-5}$$

The remainder of the system is given as in Equations 4-6 through 4-8.

$$\rho = C\theta + Du \tag{4-6}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4-7}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4-8}$$

where u is the torque applied about the ankle by the COP. Both the location of the COP and its magnitude is computed from the force torque sensors in each of the feet. The input torque to the model can then be found as in Equation 4-9:

$$u = F_{COP}(d_{COP} - p_{foot}) \tag{4-9}$$

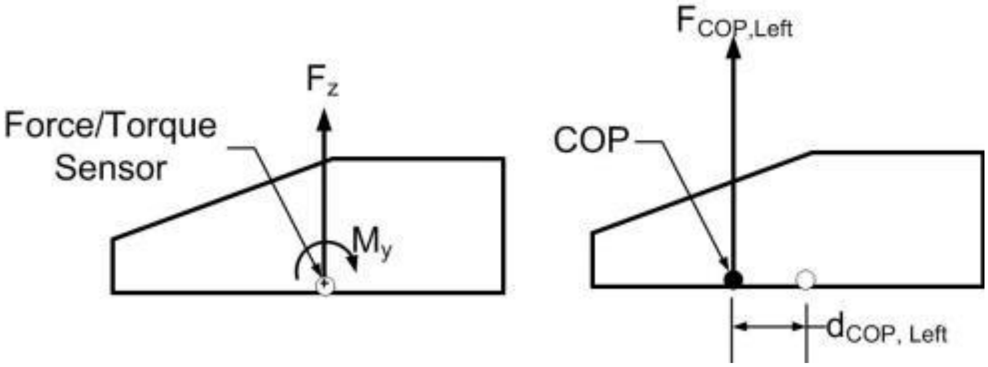

Figure 4-10 Each foot COP can be found by transforming the force torque reading to the point at which the moment goes to zero. The robot COP is a distance weighted average of each individual foot's COP.

Since this system is observable, the rank of [C, CA] is two, we can define a gain matrix **K**, such that the observer poles are significantly to the left of the system poles. In this case the system poles are fairly slow, and will be well damped due to the ankle torque. They are estimated with a natural frequency of 0.75[rad/s] and a damping ratio of 0.85. The observer poles are considerably faster (20x the system poles). The estimated state of the system is then given as Equation 4-10.

$$x_{est+1} = [A - KC]x_{est} + [KC]x_{act} + Bu \tag{4-10}$$

In Figure 4-11, one can see the difference between the filtered and observed signals. Interestingly, the quality of the estimated position is degraded due to the additional noise caused by the COP measurement. Therefore all controllers use as the state the measure position and observed velocity.

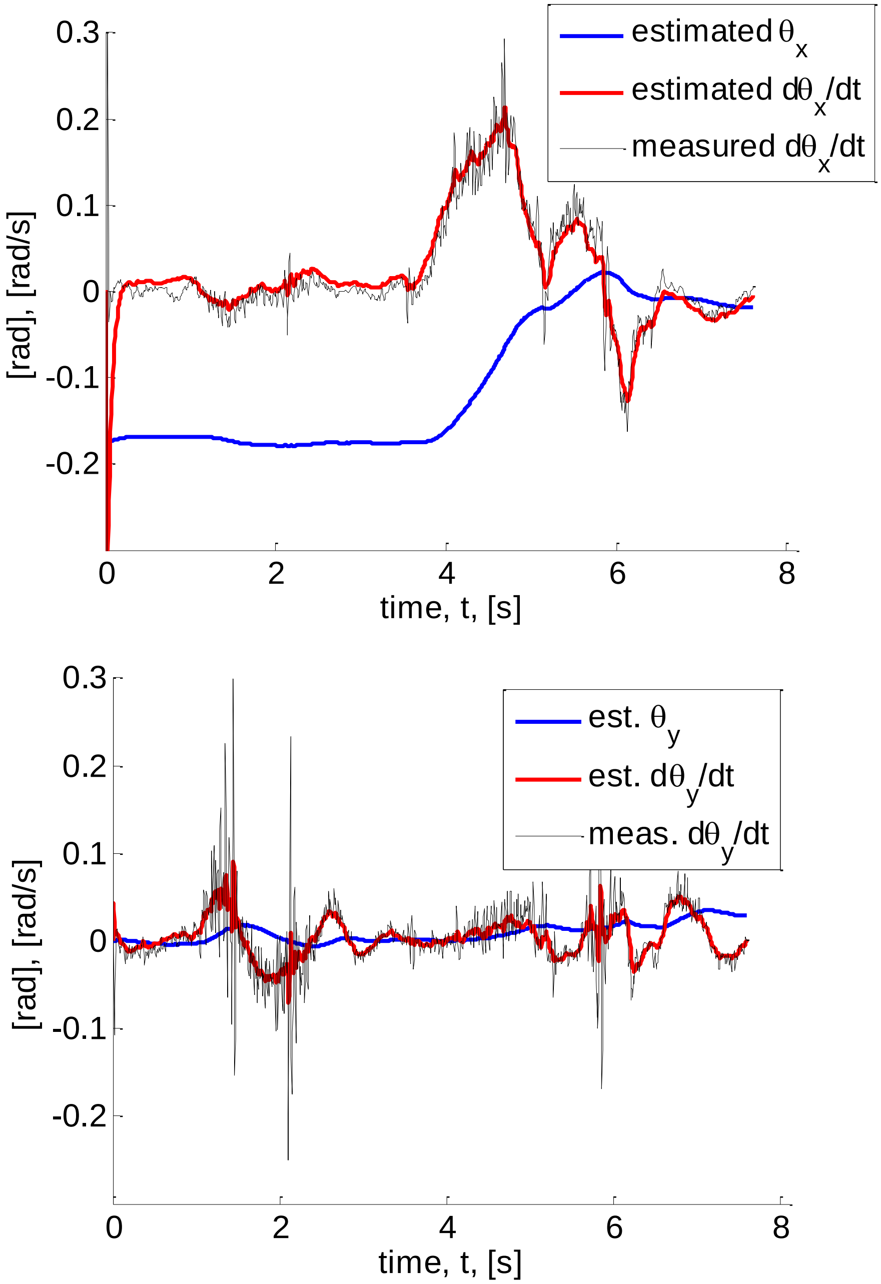

Figure 4-11. A Luenberg observer was used to improve the velocity estimate of the COM

A humanoid is not a simple two dimensional inverted pendulum, in the double support phase it acts as a 1-D pendulum. By using the COP as the control signal, ***u***, the observer is able to work correctly in both single support and double support. The only difference is which foot the pendulum pivot is associated with, either left or right but not both. The specific foot is chosen by a higher level controller in the walking algorithm that determines the appropriate foot at the time. Switching between the two states is accomplished by resetting the observer to the best measured position and velocity, thereby minimizing any spikes in the estimated signal. With this design, the same observer can be used continuously during and between any hybrid robot states.

A limitation of this design is its inability to correctly interpret the COP on non-level surfaces due to the COP calculation. These calculations assume that the plane of the foot is oriented perpendicular to gravity. As SAFFiR's ultimate goal is to traverse uneven and compliant terrain, it will be a priority to modify the COP calculations and likely the observer design to accommodate these situations.

In simulation and on hardware, this observer proved to be successful at estimating the robots state accurately in all but a few cases. An interesting complication occurred using the real hardware one some compliant terrain, particularly loose gravel and high pile synthetic grass. In some cases the COM measurement would oscillate at 4-5[Hz] usually accompanied by a vigorous vibration at a similar frequency of the foot. The cause was determined to be the result of some unmodeled dynamics within the robot exciting the IMU, specifically excessive flexibility between stance foot and the swing foot, as illustrated in Figure 4-12. A notch filter had to be added to eliminate this spurious mode of vibration; the particulars of the design are covered below.

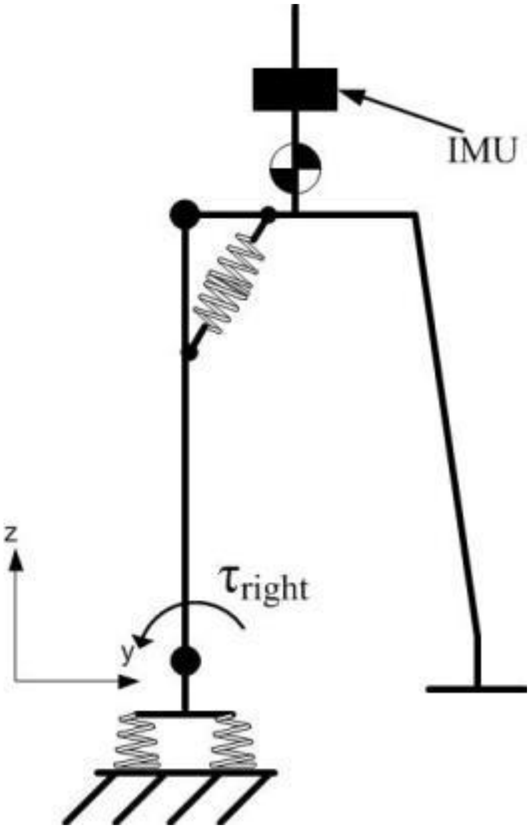

Figure 4-12. Unmodelled flexibility in the robot, as modeled in the figure, induced additional noise into the COM position.

Figure 4-12 shows a simple representation of the robot in single support with the right foot on the ground and the left in the air. In this configuration, both the inherent elasticity of any engineering material and the long and slender legs of a full size humanoid allow a good deal of flexibility between the position of the stance and swing foot. Put simply, the robot has considerable flex with little damping. This becomes a problem when such vibrations are fed back to the IMU. This is illustrated in Figure ?, wherein the flexibility in the robot’s right leg is modeled as a spring loaded revolute joint at the hip. Because of the IMU’s position relative to this flexibility, it is excited at the same frequency as the swing legs oscillations. This signal is interpreted by the controller as a rocking of the entire robot about the ankle joint. The problem is exacerbated by the feedback circuit, in which the controller oscillates the torque at the foot in response to this signal, further exciting the vibration of the swing leg.

While the observer does reduce the magnitude of this signal, it occurs close enough to the robots natural frequency that it is not sufficiently damped out. A notch filter specifically for this mode of vibration was added to further knock down the signal. The cutoff frequency for this filter was designed experimentally. The robot was made to stand on one leg while the swing leg was impacted, causing it to oscillate. The frequency of this oscillation was recorded, and the notch filter set accordingly. The positive effect of the notch filter can be seen in Figure 4-13.

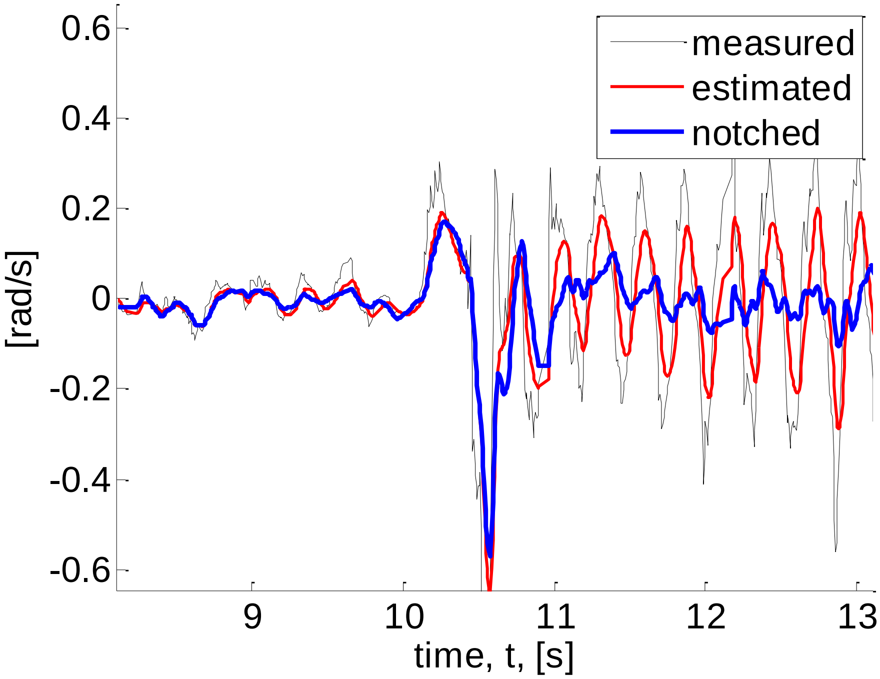

Figure 4-13. A notch filter was used to reduce magnitude of the oscillations caused by excessive flexibility.

Figure 4-13 through Figure 4-16 illustrate the effect of the notch filter. Figure 4-13 in particular illustrates the effect of filtering the data in a post processing step. For this experiment, the robot was made to stand on one leg, while an impulse was applied to the swing leg, thereby exciting the additional vibration mode in question. The estimated position and velocity of the COM is plotted. The large swings in COM velocity are due to the rotation of the IMU due to the mechanical flexibility and not from an inverted pendulum like motion about the ankle.

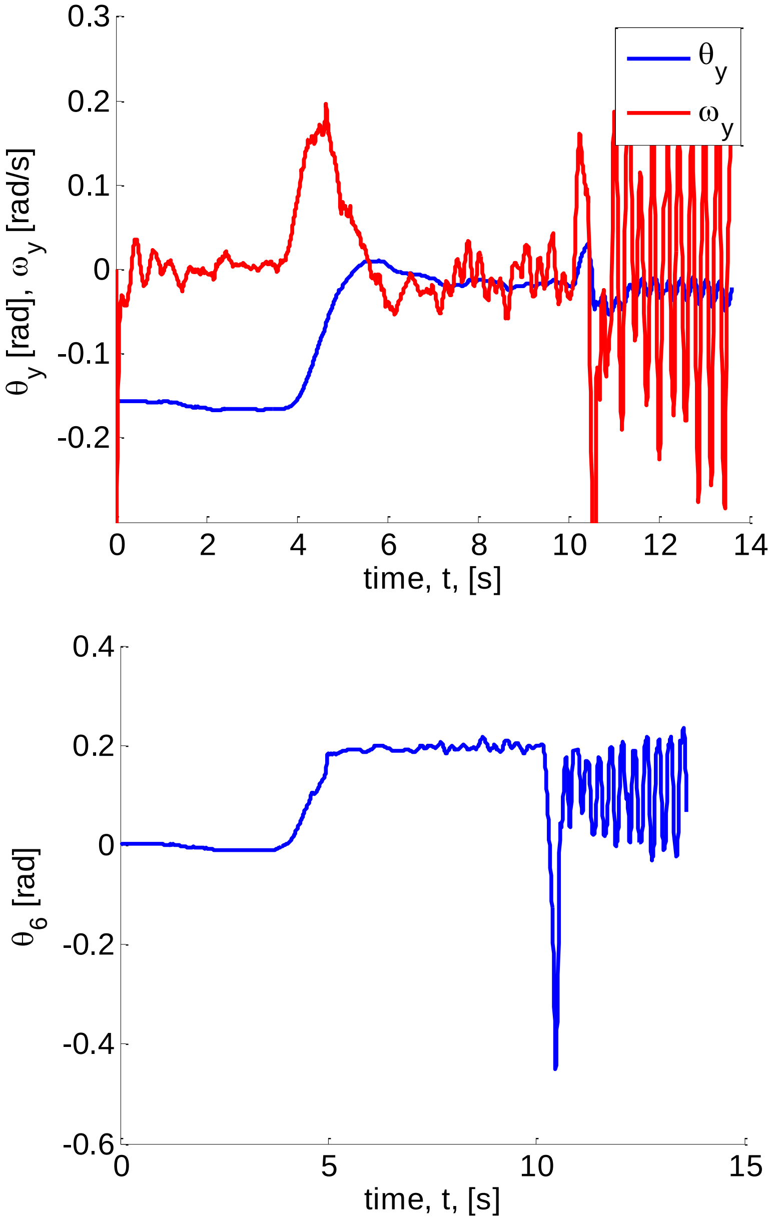

Figure 4-14. The effect of the coupling between the mechanical flex and the controls system is most pronounced on compliant terrain. The figures above show the response when the robot is perturbed (at 10[s]) while standing on one leg on a tall pile of gravel.

The effect of the coupling between the mechanical flex and the controls system is most pronounced on compliant terrain. Figure 4-14 show the response when the robot is perturbed (at 10[s]) while standing on one leg on a tall pile of gravel. In this experiment the pile was formed such that it was 50[mm] tall and had shear sides in very close proximity to the edge of the feet. An impulse was applied to the torso of the robot, and the recovery monitored. In all tests, the impulse was made strong enough that the torques exerted on the ground in response were large enough that the pile of gravel collapsed. The robot then fell down and to the side as it slid down with the collapsing gravel. This fall excited the swing leg's natural frequency, which when fed back through the IMU and feedback law, caused wild oscillations in the torque applied to the foot. The compliant nature of the gravel then allowed the foot to rock violently back and forth in response to the applied torque further exciting the swing leg. The motion of the swing leg can be inferred from Figure 4-15 which illustrates the measured torque on the roll joint of the stance hip. The torque shows an increasing amplitude oscillation as the feedback loop further excites the natural frequency of the leg.

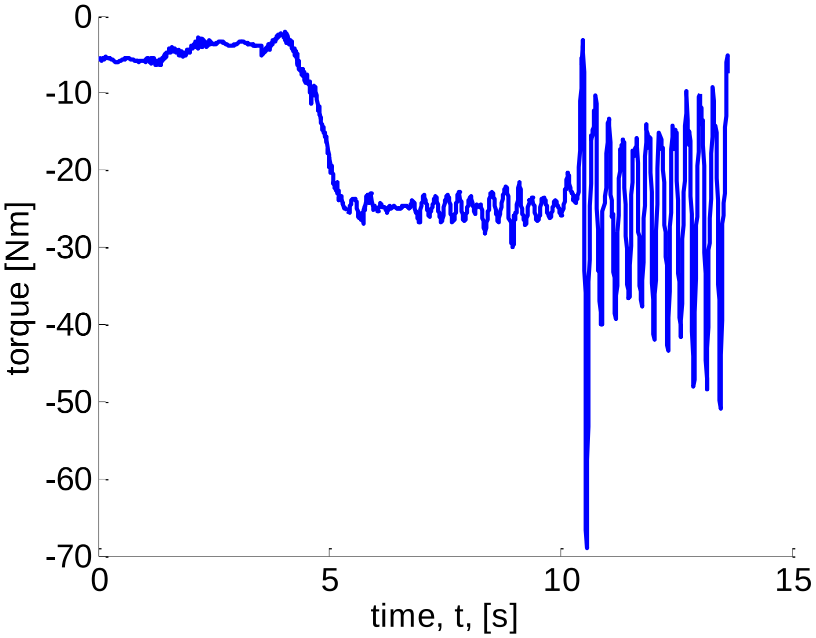

Figure 4-15. The torque on the roll joint of the stance hip shows an increasing magnitude oscillation as the feedback loop further excites the natural frequency.

The notch filter was designed to eliminate from the control loop the signal generated by the swing leg's natural frequency. The test illustrated in Figure 4-16 showed the same test scenario as above except with a notch filter added to the COM velocity estimation. The robot was again perturbed and the pile of gravel made to collapse in response. With the notch filter in place, there is almost no oscillations of the COM velocity or foot position at the characteristic 4[Hz] frequency. Even more telling is the torque and motion about roll joint of the stance hip, as seen in Figure 4-17. Instead of increasing in amplitude, the vibrations settle in time due to the natural dampening in the system.

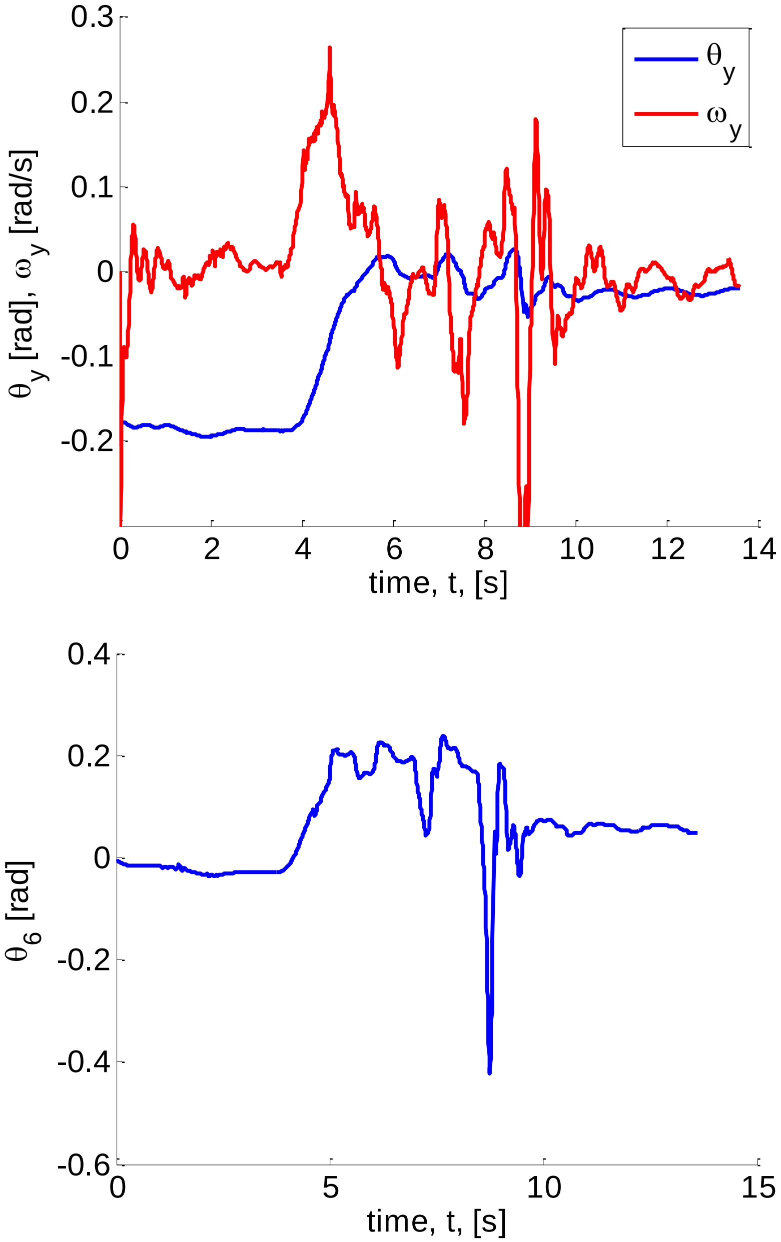

Figure 4-16. After the notch filter was added, the same test conditions result in significantly greater stability.

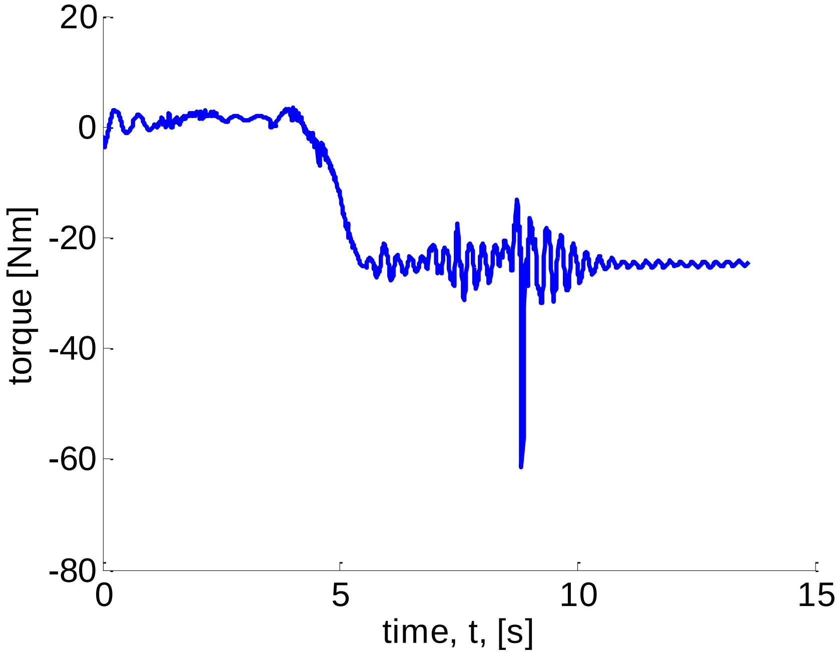

Figure 4-17. By eliminating the instability due to the feedback loop, the oscillations in the commanded torque is significantly reduces as compared to .

The preceding section outlined the design of the observers for accurate state estimation, specifically the position and velocity of the COM location. The sensors provide an accurate estimate of the position; the velocity however had to be estimated using a Luenberg observer. A notch filter was added on top of the observer to reduce the magnitude of extraneous oscillations originating from unmodelled modes of the robot. The following sections will detail the implementation of several control strategies using this state information.

4-8 Footfall sensing strategy

The footfall sensing strategy is the method by which the robot senses which feet are on the ground. Such information is needed by several control laws and similarly the walking algorithm such that the proper torques and motions can be applied. In theory and simulation this information is straightforward to come by. That is, it is sufficient to simply measure the vertical force on the foot (if it is assumed the robot is walking on flat terrain). If the force exceeds some threshold, then the foot is likely on the ground. On physical hardware though, there are some additional factors that increase the complexity. The most significant is the stiffness exhibited by the ankle when in zero force mode. Specifically, although zero force may be specified at the ankle, it still takes some amount of force to move the foot. This can cause issues when the robot is walking on non-flat terrain and miss-estimates the inclination of the ground. It has to force the foot down on the ground, and since the requires some force it is necessary to account for this in the sensing strategy. In this section, a robust algorithm to experimentally determine the number of feet on the ground is explained.

The footfall sensing strategy is still based on the premise that the weight bias of the robot dictates which feet are stance, and which ones are swing. The weight bias is calculated as in Equation 4-11.

$$bias = \max\left(0.1, \frac{F_{z\,left}}{(F_{zleft}+F_{zright})}\right) \tag{4-11}$$

The right bias is simply unity minus the left bias. Therefore in principle, if the weight bias does not exceed an upper and lower threshold, then both feet must be exerting forces on the ground and are therefore also in contact with the ground. Should the bias exceed or drop below the upper or lower thresholds, the footfall sensing strategy employs several checks to ensure that spurious readings do not interfere with the overall algorithm. A flowchart of this algorithm can be seen in Figure 4-18.

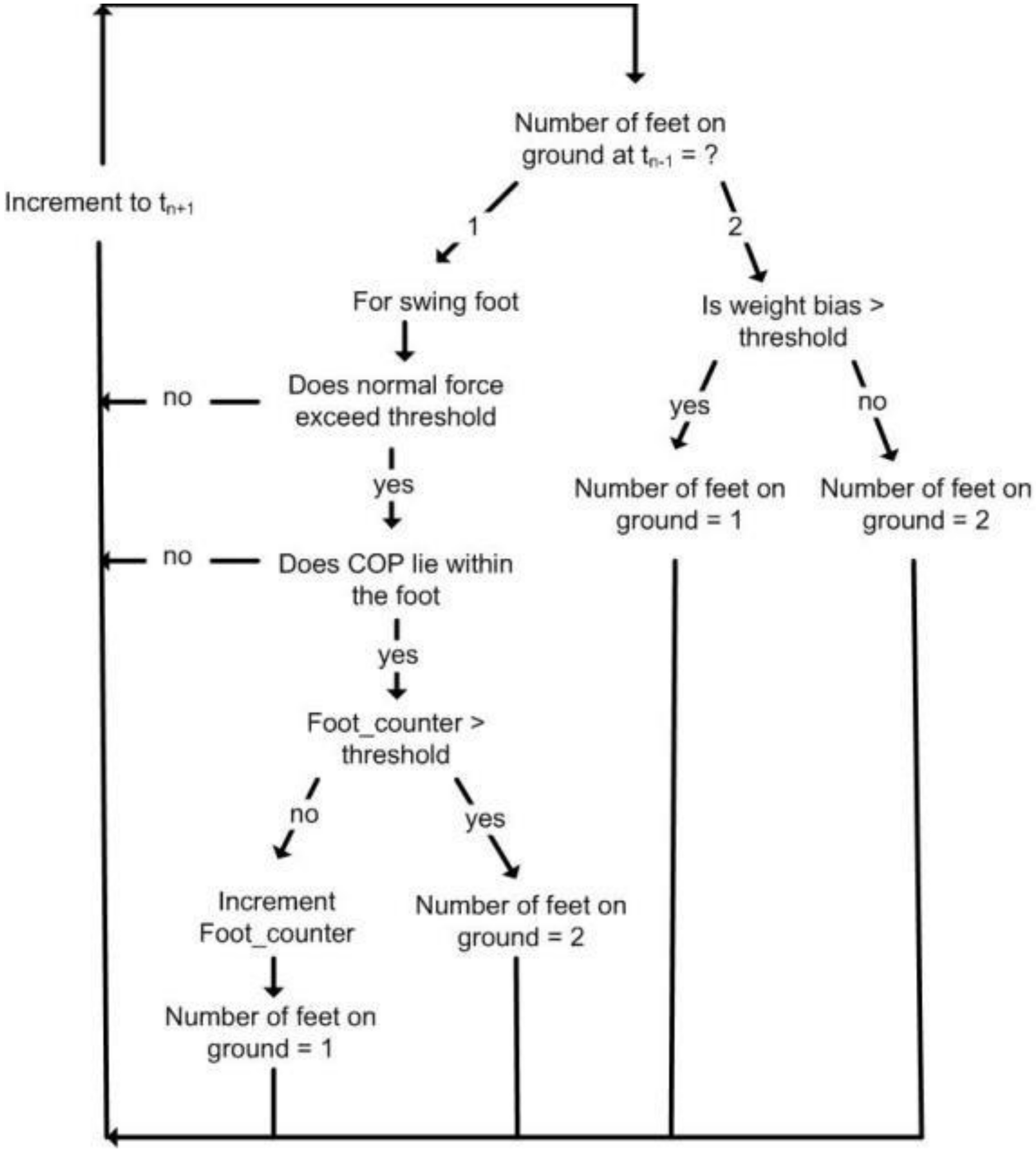
Number of feet on ground at t_{n-1} = ?
1
2
Increment to t_{n+1}
For swing foot
Is weight bias > threshold
no
Does normal force exceed threshold
yes
no
yes
Number of feet on ground = 1
Number of feet on ground = 2
no
Does COP lie within the foot
yes
Foot_counter > threshold
no
yes
Increment Foot_counter
Number of feet on ground = 2
Number of feet on ground = 1

Figure 4-18. The footfall sensing strategy uses three checks (bias, COP, and time) to ensure a foot if firmly on the ground when so indicated.

Aside from the weight bias, the footfall sensing strategy incorporates two additional checks to ensure the foot is flat on the ground before declaring so. Those two are the COP and the time elapsed since any expected touchdown occurred. The former is necessary to ensure the foot is flat, while the latter ensures spurious or noisy readings do not trip the footfall sensor.

As alluded to above, the COP measurements can indicate when the foot has at least three points of contact on the ground. This can be proven by looking at a moment balance of the foot about the ankle when zero torques are applied by the actuators. To balance the robots weight passing through the ankle, there must exist at least one contact point directly under the ankle, or two points of contact on either side of the ankle. The same logic can be applied to both the x and y axis. Any two points interior points on opposite sides of the ankle will generate a COP measurement that also lies within the foot. Therefore such a COP measurement indicates that the foot is statically supported by the ground.

The COP measurement is a good indicator that the foot is flat, however that alone is not a reliable as necessary. The reason being that as the foot is force onto the ground, it can have the tendency to move from heel contact to toe contact without becoming statically stable. As the COP shifts from back to front or vice versa, experience shows that the COP will sometimes be measured as being on the interior of the foot. Fortunately, the problem is easily solved by looking for two or more consecutive interior readings. In practice, a threshold of five readings was used with good results.

The footfall sensing strategy is a critical functionality of the robots proprioceptive software suite. It allows the robot to accurately and reliable know which feet are firmly on the ground and can apply useful torques. Without such information, erratic behavior will almost certainly result. The algorithm outlined above provides such reliable information by incorporating a three tiered check, including the magnitude of the normal force, the position of the COP, and the elapsed time at a suitable COP. With the basic proprioception algorithms described, the following section will detail how that information is used to deliver torques to the ground.

4-9 Ankle Torque strategy

The most straightforward and effective means of balancing an inverted pendulum is to apply torques at the pendulum pivot. A secondary method, one that is covered later, is to use a reaction mass such as the torso. For humanoid robots, the former is possible through the application of torques at the ankle. A position controlled robot is inherently and rapidly able to do this given the nature of the stiff control algorithms controlling the joint position. Any external disturbance is nearly immediately sensed as an error in the joint position, and the current adjusted accordingly. The limit to this policy is the maximum torque that can be applied by the foot to the ground (when the foot rolls or tips onto its edge). A force controllable ankle though must incorporate higher level states than just the joint position. But with that added complexity though, comes the ability to naturally balance on terrain that is compliant or irregular without specific knowledge of the terrain itself.

Since the ankle is free to move in a force control strategy as it adapts to the ground, local states are insufficient to control the joint. The control signal for ankle torque instead comes from the main controller which incorporates the COM state and the desired COM trajectory to generate a command signal. The sub-controller that modulates this torque is called simply the Ankle Torque controller (ATC). The control law itself used by the ATC is in and of itself a simple PID design. However, complexities arise due to the hybrid nature of a humanoid, it can exist in double and single support. The ATC is designed to both adapt to these two states, modulate the ankle torque accordingly, ensure the feet remain flat against the ground, and damp out unmodelled motions of the robot. These specific features and their implementation on hardware are discussed below.

The ATC can be divided easily into to independent, but similar controllers. The controller for single support is most straightforward, and so is covered first. It is dubbed the SSATC. The double support controller (DSATC) must take into account an over constrained system and some additional unmodelled dynamics, and is thus more complex.

The SSATC is fundamentally a PID controller with feed forward gravity compensation. The input signal for the controller is the desired *x* and *y* position of the COM with respect to one of the feet, as seen in Figure 4-19**.** This figure shows as an example the left foot support polygon. Overlaid are the projections of the ankle joint and the COM on the plane through the ankle joint perpendicular the gravity vector. The desired COM position, labeled as x_{com_d} and y_{com_d}, are prescribed by a higher level controller, as will be documented later. The right foot is controlled similarly with the only difference being the sign of the distances.

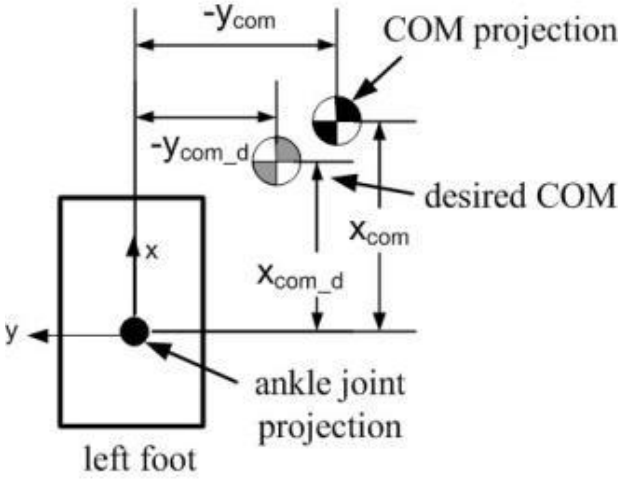

Figure 4-19. The error between the projections of the desired and actual COM position was used to drive the ankle torques.

The observer above provides the feedback signal for the controller; however, since the observer operates on the angle between horizontal and the line connecting the ankle and the COM, a quick conversion is necessary to compute the linear position of the COM. These are given as Equation 4-12 and 4-13:

$$x_{COM} = 0.66 * Sin(\theta_y) \quad (4\text{-}12)$$

$$\dot{x}_{COM} = 0.66 * Cos(\theta_y)\dot{\theta}_y \quad (4\text{-}13)$$

Similarly, in the *y* direction:

$$y_{COM} = 0.66 * Sin(\theta_x) \quad (4\text{-}14)$$

$$\dot{y}_{COM} = 0.66 * Cos(\theta_x)\dot{\theta}_x \quad (4\text{-}15)$$

where the height of the COM, 0.66[m], is assumed to be constant. This is a safe assumption because the SAFFiR robot does not have the range of motion to significantly alter the height of the COM. Rather, during all walking tests to date, the COM remains at a nearly fixed height. In the future, this height will be variable, and instead come from the kinematic solver as future tests or robots will be able to induce a larger height variation of the COM.

Given both a desired and actual COM position, the control law is calculated as follows:

$$\tau_x = k_p(x_{COM_d} - x_{COM}) + k_d(\dot{x}_{COM_d} - \dot{x}_{COM}) + k_i \int(x_{COM_d} - x_{COM})dt + mgx_{COM} \quad (4\text{-}16)$$

$$\tau_y = k_p(y_{COM_d} - y_{COM}) + k_d(\dot{y}_{COM_d} - \dot{y}_{COM}) + k_i \int(x_{COM_d} - x_{COM})dt + mgy_{COM} \quad (4\text{-}17)$$

where k_p, k_d, and, k_i, are the control gains, and τ_x and τ_y are the torques in the x and y directions respectively commanded about the stance ankle joint. The gains are chosen experimentally such that settling time, rise time, and steady state offset are satisfied. The specifics of the gain selection are covered in Section 4-11. The final term in the equation, mgx_{com}, is the feed forward gravity compensation term, there to improve the controller response and reduce steady state error. The remainder of this section will cover the double support controller with specifics as to how these control signal is applied to the ankles.

The double support ankle torque controller (DSATC) while similar, is more complex than the single support version. The DSATC must take into account both an over constrained system, as well as some unmodelled dynamics associated with the aforementioned flexibility in the SAFFiR's legs. As the name implies, the DSATC operates when the robot is in double support. With both feet on the ground, the degree of freedom is removed from the system. The robot is no longer able to freely tip in two directions. Instead, because each ankle has two DOFs, it can only move its COM in a direction perpendicular to the axis connecting the ankles. The DSATC is primarily responsible for controlling this DOF.

A representation of the desired and feedback signals for the DSATC are pictured in Figure 4-20. As mentioned earlier, in double support, SAFFiR is only able to rock about the axis that connects the two ankle joints, called *axis*. Therefore it is necessary to compute this axis and the distance of the COM from it. The controller also uses the rate of change of the distance of the COM along this axis to compensate for some unmodelled dynamics, as will be explained later. These two distance are computed every cycle during double support. The derivation is given below.

The axis is given as in Equation 4-18:

$$axis = \begin{bmatrix} lf_x - rf_x \\ lf_y - rf_y \end{bmatrix} * \frac{1}{\sqrt{(lf_x - rf_x)^2 + (lf_y - rf_y)^2}} = \begin{bmatrix} axis_x \\ axis_y \end{bmatrix} \quad (4\text{-}18)$$

where lf_x and lf_y are the x and y positions of the left foot respectively as measured from the projection of the torso frame. Similarly, rf_x and rf_y denote the same for the right foot.

The distance d_1 is then given as Equation 4-19:

$$d_1 = \frac{\frac{axis_x}{axis_y}(y_{COM} - lf_y) - x_{COM} + lf_x}{\sqrt[2]{\frac{axis_x}{axis_y}^2 + 1}} \quad (4\text{-}19)$$

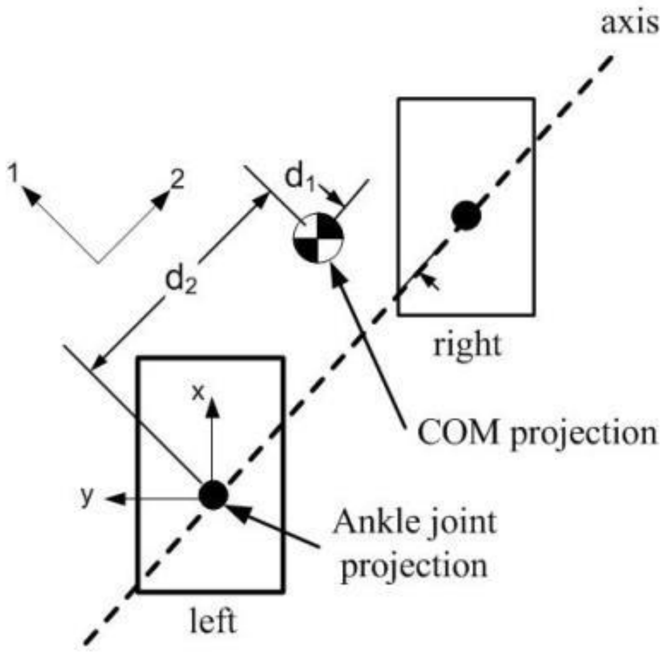

Figure 4-20. The desired and actual COM projections in double support are referenced to a frame whose one axis is parallel to the line connecting the ankles.

For the DSATC, it is necessary to add damping along the axis to damp out oscillations that occur as the flexibility in the structure allow the two legs and hip to parallelogram, much like an unconstrained four bar mechanism. This phenomenon is depicted in Figure ? which shows the torso pinned to two legs, each of which are pinned to the ground. Compliance is unfortunately introduced because of 1) the inherent flexibility of any long slender mechanism such as the legs a humanoid, and 2) flexibility and backlash in the actuators constraining each of the joints. The compliance is illustrated as two springs, one across each of the hip joints, and amounts to an undamped mode of vibration as the robot shifts its COM from one foot to the other.

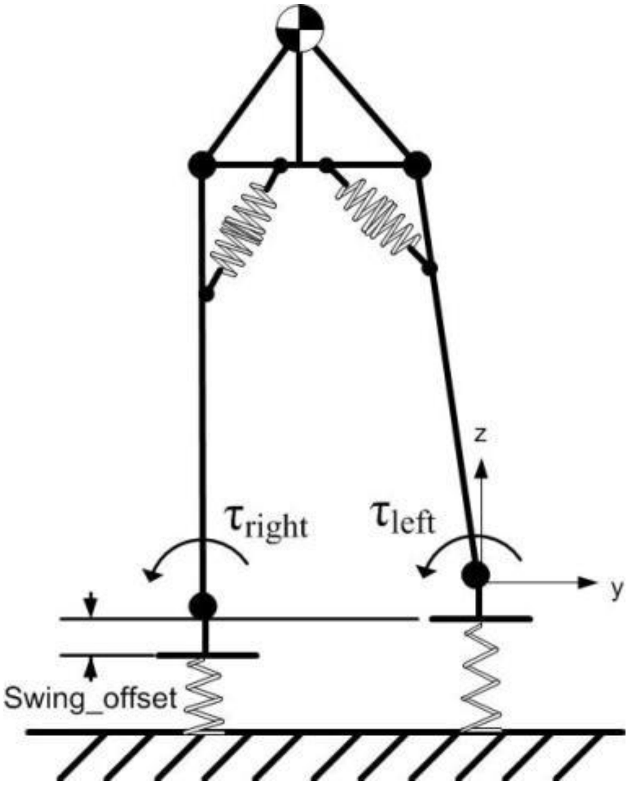

Figure 4-21. Unwanted flexibility is also exhibited in the double support phase and is compensated for by a velocity controller.

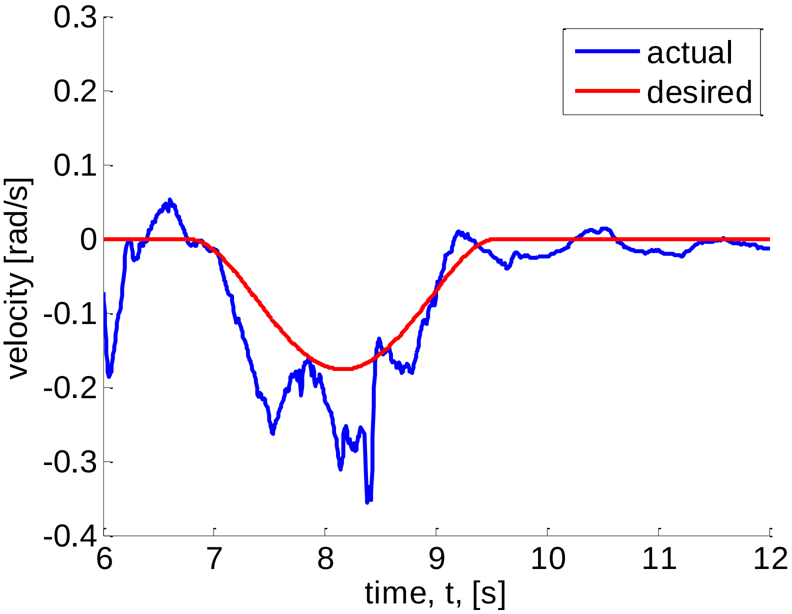

Figure 4-22. The mechanical elasticity of the robot produces undesirable motions as illustrated by the actual and desired velocities of the COM parallel to the line connecting the ankles. This motion is purely position controlled, and should theoretically exhibit good trajectory tracking.

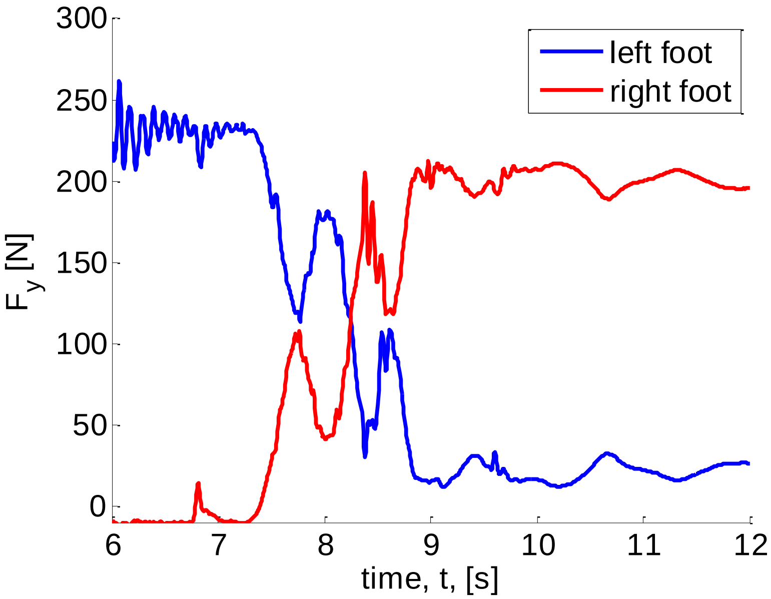

Figure 4-23 The mechanical elasticity can also be seen in the vertical reaction force at the feet. The highly undulating nature of the force indicates the robot is rocking back and forth between feet as it shifts its weight from the left foot to the right foot.

There is no straightforward way of mechanically eliminating this compliance save for redesigning the robot. Instead, a controller is used to damp out its effect. The controller does so through the use of the two ankle torques that act perpendicular to the *axis*. When the COM is moving faster than desired, an opposing torque is applied to both ankles which, acting through the legs, applies a retarding force to the COM. To compute the damping torque, it is necessary to compute the actual velocity of the COM along *axis* as in Equation 4-20:

$$\dot{d}_2 = \dot{x}_{COM} axis_x + \dot{y}_{COM} axis_y \qquad (4\text{-}20)$$

Where $\dot{x}_{COM}$ and $\dot{y}_{COM}$ are the same as above in Equation 4-12 and 4-14. The torques τ_1 and τ_2 are the found according to Equation 4-21 and 4-22.

$$\tau_2 = k_p(d_{1des} - d_1) + k_d(\dot{d}_{1des} - \dot{d}_1) + k_i \int (d_{1des} - d_1)dt + mgd_1 \qquad (4\text{-}21)$$

$$\tau_1 = k_d(\dot{d}_{2des} - \dot{d}_2) \qquad (4\text{-}22)$$

Where τ_1 and τ_2 are torques about the 1 and 2 direction as seen in Figure 4-20. The desired and actual COM projections in double support are referenced to a frame whose one axis is parallel to the line connecting the ankles. One can see that again a PID control law is used to control motion perpendicular to *axis* while a simple derivative controller is used for motion along *axis*.

As can be seen in Figure 4-24, the derivative controller makes a significant improvement in the tracking of the COM. It is made even clearer when observing the weight transfer of SAFFiR from one foot to another. Figure 4-25 shows one such example. With no controller, the robot tends to oscillate at 4Hz as it shifts to the right foot. With the controller applied, the weight shift is significantly smoother, as desired.

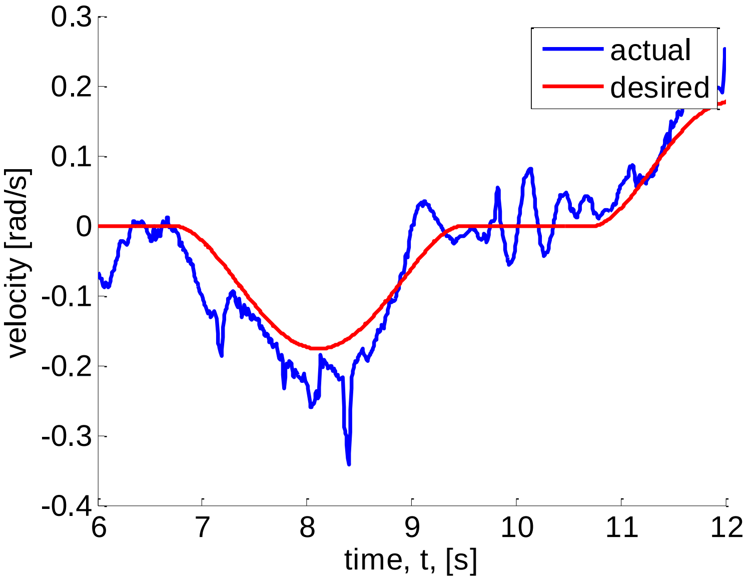

Figure 4-24. A derivative filter was added to control unwanted motion along the axis connecting the ankles, thereby increasing the tracking of the COM velocity as compared to above.

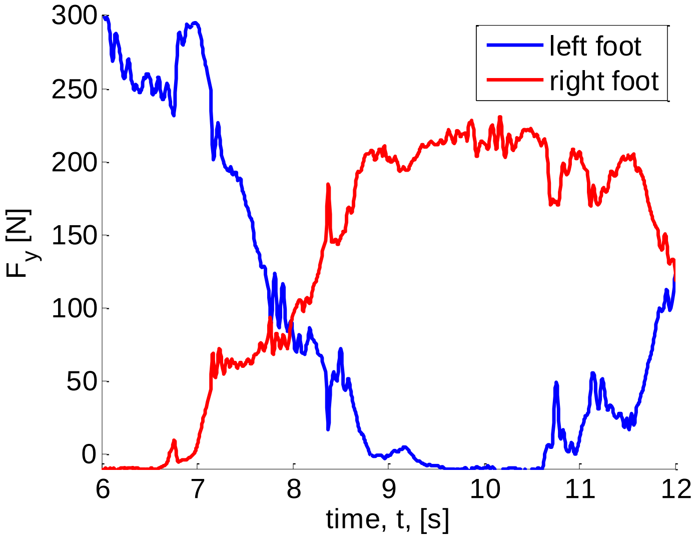

Figure 4-25. The vertical force on the ankle exhibits almost none of the rocking behavior as seen above. The derivative filter damps out nearly all the unwanted vibrations.

4-10 Torque Limiting

The preceding paragraphs detail the controllers governing two pair of torques, both (τ_x and τ_y) and (τ_1 and τ_2). These torques though are not simply applied to the ankles. Instead, the torque are first modulated according to the percentage of the total robots weight on each foot, as well as limited to a maximum and minimum also according to the weight bias. This former allows the maximum torques to be applied to the ground, thereby effecting the greatest change. The latter ensures that the foot does not roll up on its edge.

Doing so upsets the balance of the robot, as that leg is effectively lengthened, and also delays the application of the opposite torque. Both situations are ones which should be avoided for stability. The calculation of the applied torques are described below.

Each ankle of SAFFiR has two degrees of freedom, oriented in the *x* and then *y* direction when the robot is in the nominal configuration. Therefore it is most straightforward to compute a desired torque about these two directions. The SSATC results need no modification; the DSATC however is in the wrong coordinate frame. To convert from the *1-2* frame to the *x-y* frame the *axis* is used as in Equation 4-23:

$$\begin{bmatrix}\tau_x\\ \tau_y\end{bmatrix} = \begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} * \begin{bmatrix}axis_x & axis_y\\ axis_y & -axis_x\end{bmatrix} \tag{4-23}$$

Since the robot cannot be in both single support and double support at the same time, the same variables, τ_x and $\tau_{y,}$ can be used. In one case, they are set by the SSATC and otherwise set by the DSATC and the above equation. Next, the weight bias is calculated. This is simply the percentage of the robots total weight that is being applied to each of the feet. This term is computed from the force torque sensor data, and so reflects both the acceleration due to gravity, as well as any force due to the acceleration of the COM. The bias is calculated as in Equation 4-24, and is with respect to the left foot:

$$bias = \max\left(0.1, \frac{F_{z\,left}}{(F_{zleft}+F_{zright})}\right) \tag{4-24}$$

where F_{Zleft} and F_{zright} are the vertical force applied to the left and right feet respectively. The *max()* function is used to eliminate chatter when very little weight is applied to the foot and also to prevent the imperfect force fidelity from causing instabilities. The torques applied to each ankle can then be calculated as in Equation 4-25:

$$\begin{bmatrix}\tau_{lx} & \tau_{ly}\\ \tau_{rx} & \tau_{ry}\end{bmatrix} = \begin{bmatrix}bias\\ 1-bias\end{bmatrix}\begin{bmatrix}\tau_x & \tau_y\end{bmatrix} \tag{4-25}$$

where τ_{lx} and τ_{rx} are the torque about the roll joint of the left and right ankle respectively. The *y* subscript on the torques denotes a torque about the ankle pitch axis. Equations 4-23 through 4-25 are necessary to distribute one desired torque between two stance feet. It is still necessary to limit the magnitude of the torque applied to each ankle so that the foot does not roll up on an edge. Doing so results in two unwanted situations, 1) as the foot tips, the leg effectively lengthens, upsetting the robot, and 2) when the foot hits the range of motion hardstop, an incorrect torque will be applied to the ground. The following describes a simple algorithm for torque limiting the ankle.

$$\tau_r = \begin{bmatrix}\tau_{rx}\\ \tau_{ry}\end{bmatrix}, \tau_l = \begin{bmatrix}\tau_{lx}\\ \tau_{ly}\end{bmatrix} \tag{4-26}$$

By limiting the torques applied to each foot in this manner, the COP of each foot must remain within the support polygon. Using only the *z* reaction force though limits the usefulness of the equation to flat ground. Modifications will be made when the gradient of the ground becomes substantial. Other authors have proposed graphical means of regulating ankle torques in a balance strategy [65]. The method presented above has the advantage of being computationally efficient as well as being mechanically intuitive.

Once the torques have been distributed between the two ankles, it is then necessary to limit the torque such that the ankle does not roll up on edge. This is exactly analogous to keeping the COP within the support polygon. The method employed below though differs in that the distribution of forces is an independent problem from the limiting, which has the benefit of being computationally easier. To maintain a COP within a support polygon, it is necessary that the ratio of the moments on a body to the normal force between the body and the ground does not exceed the dimensions of the support polygon. Therefore any moments applied to the ankle should be limited by a constant scaled to the normal force, as in Equation 4-27 and 4-28:

$$\tau_{*x} = \min\left(\tau_{*x}, \frac{10F_{z*}}{220}\right) \tag{4-27}$$

$$\tau_{*y} = \min\left(\tau_{*y}, \frac{20F_{y*}}{220}\right) \tag{4-28}$$

where τ_{*x} is the x torque on the left and right foot as denoted by the asterisk and similarly, F_{z*} denotes the vertical force on the left and right foot. The constants are determined from a simple moment balance about the x and y axes of the feet. Each foot on SAFFiR is 0.200[m] long and 0.100[m] wide. Therefor if the robot were to have all of its weight (220[N]) on the edge of its foot, it would be exerting 22[Nm] and 11[Nm] about the y and x axes respectively. The constants in the equation above are slightly derated from these to ensure the foot does not roll even on compliant surfaces. The scalings by the normal force F_{z*}, has the effect of regulating for the current vertical load, which can of course change when the robot is accelerating or has shifted its weight from one foot to another.

In this section, the foot balancing controller was explained. This controller is responsible for computing and applying the appropriate torques to the feet in order to direct the COG. To frame it in the broader picture, this section described a method of calculating the appropriate ground reaction forces to control the COG and distributing those between the feet. In particular, this method is tailored to the unique control architecture of SAFFiR, which employs both position and force controllable actuators, which in this case greatly simplifies the solution.

The controller is broken down into two independent controllers, one for single support and one for double support. The controllers command certain ground reaction torques to reduce the error between the desired and actual COG position. The ground reaction torques are distributed between the two feet based on the normal force exerted by each foot as a way of maximizing the available torque applied to the ground. Finally, should the command be too great, the signal is clipped so as to prevent the feet from rolling on edge. The following sections will describe how the preceding gains were determined, as well as two augmentations to the balancing strategy including a foot height policy and torso windmilling as well.

4-11 Gain tuning

The performance of any control law is heavily dictated by the chosen gains. In this case, there are four separate but similar control laws dictating the torque applied to the ankle. All the gains were chosen experimentally based on the steady state error, rise time, overshoot and settling time to a step input change of the set point. The results of the tests for the y-axis PID double support controller can be seen in Table 4-2.

Table 4-2 PID gains were tuned experimentally. Proportional gain was increased until acceptable, and then derivative added until settling time and overshoot met requirements.

Test	P	I	D	Steady State	Rise Time	Overshoot	Settling time
1	50	1	10	0.008	-	-	-
2	75	1	10	0.031	-	-	-
3	100	1	10	0.05	7.514	0.053	8.61
4	125	1	10	0.06	7.275	0.069	-
5	150	1	10	0.062	7.12	0.081	-
6	125	1	20	0.05	7.391	0.053	7.866
7	125	1	30	0.05	7.41	0.050	7.85

Firstly the proportional gain was increased until the system reached something near the steady state solution. Low proportional gains were not able to overcome the inertia and stiction in the system to be effective. Larger gains however performed noticeable better. There is clearly a large envelope of acceptable gains, the upper end of which is determined by the maximum torque that the ankles can apply to the ground without lifting the robot. Within this envelope, the derivative gain was adjusted until the settling time and overshoot were within acceptable ranges. With the derivative gain it was also fairly self-evident when too high, as it would cause excessive noise in the command signal due to the remaining noise in the COG velocity estimate. The integral gain was kept low as the steady state errors were small enough without it.

4-12 Foot Pressure Strategy

Walking on compliant or non-stationary terrain requires auxiliary controllers to handle the discrepancies between a position controlled ankle location and the shifting terrain beneath the robots foot. This is best illustrated through Figure 4-26. This figure shows a biped in double support standing on compliant terrain, represented by two springs. As the robot shifts its weight from left to right, as seen in the figure, the ground will shift under foot. If left unaccounted for, this shift in terrain will cause an increased COG movement as the robot essentially rocks further in the direction it was moving. For stable walking on compliant terrain, this dynamic should be taken into account, or else it can be shown experimentally that the robot will fall over.

A simple but effective controller was implemented to account for these dynamics. This section will further outline the motivation for such a controller as well as specifics to its implementation. Furthermore, there are some interesting parallels between this controller and a fully force controlled leg, and these parallels will be explored.

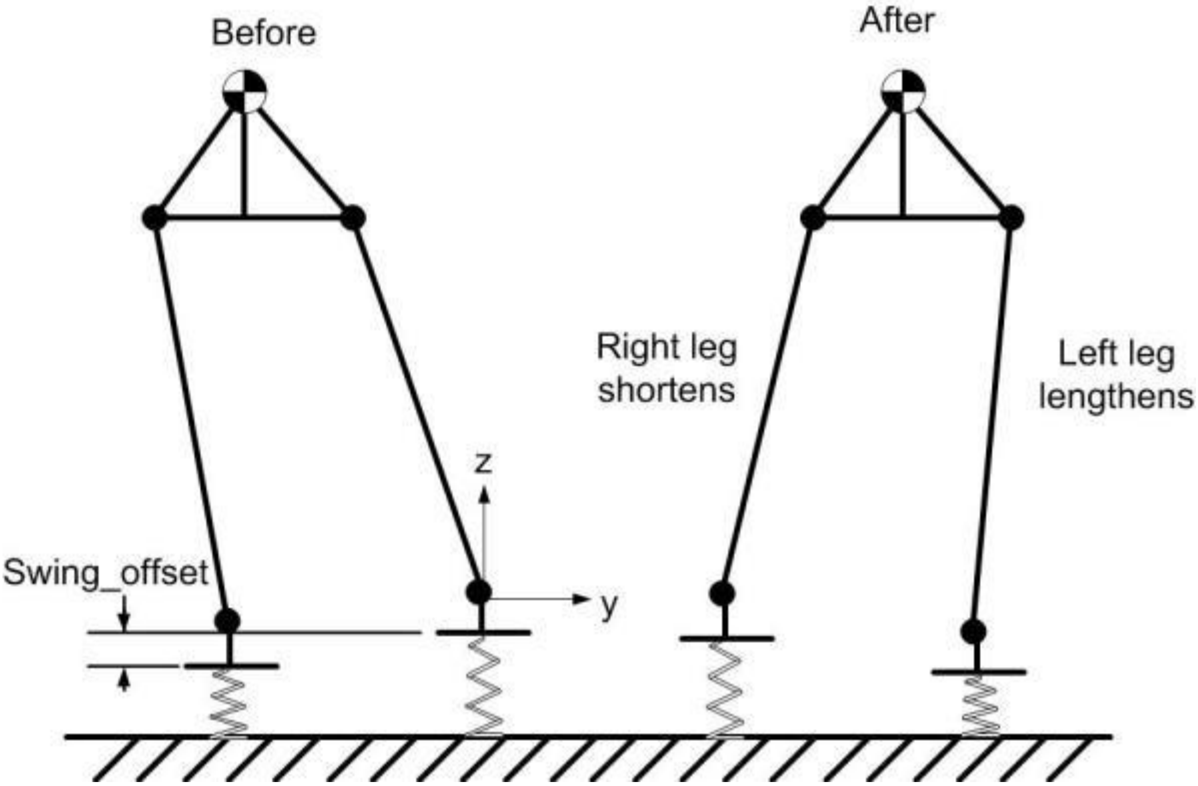

Figure 4-26. Compliant ground is accommodated for through the control of the vertical height of the foot. As weight is transferred to the foot, it is depressed further into the ground.

Natural ground is a highly non-linear surface [ref], and so remains a difficult challenge to bipedal locomotion. It is also an important challenge, given that only a minority of the Earth's surface is sufficiently stiff for traditional robots. Adding further difficulty is the complexity of any ground models, especially without apriori knowledge of the terrain. Sensors will only be able to gather a small fraction of the information necessary to create a usable model before the robot must take a step on the terrain. Therefore a reactionary technique was investigated and subsequently implemented onto SAFFiR.

The controller is also in part necessary because of the footfall sensing strategy described in Section 4-8. This algorithm assumes that the foot has made contact with the ground when the vertical force crosses some threshold. It makes no distinction between where the foot lands and where it was expected to land. While this is beneficial as a reactionary measure to sensor noise and ground miss-estimation, it also means that on compliant surfaces such as tall synthetic grass, the detection of foot fall will happen much earlier and at a higher height than the stance foot. The robot is programmed to interpret this as a step-up and will attempt t∆o move the torso upwards as it shifts from left to right when it should only move horizontally. The secondary motivation for developing a compliant terrain controller is to take advantage of the vertical reaction force under the foot. For balancing, this force can be used as an additional control input by which the robot can stabilize itself. The length of the leg can be varied according to the position and velocity of the COM and provide a restoring force.

Several assumptions were made when developing this controller. The first is that the ground is level, with no slope or height variation of any kind. Furthermore, it is assumed that the ground conditions are similar from step to step. An operator or additional controller is required to supply the parameters appropriate to the ground conditions. How this controller would handle large changes in the ground characteristics between steps is unclear, and remains an open but important question. Under these assumptions, the Equation 4-29 is used to determine the motion of the foot during weight transfer:

$$\Delta z = k_c Swing\_offset \tag{4-29}$$

where Δz is the change in height of the foot during the step, k_c is a gain value, and *Swing_offset* is the initial height difference between the swing and stance foot at touchdown as seen in Figure 4-26. The gain was experimentally found during several trial runs. At low gains the robot would continue to tip forward, and at large gains (k_c>1) the robot would tip backwards over the stance foot. Stable walking was achieved with $k_c = 0.8$.

On compliant terrain, each leg acts like a series elastic actuator. The ground underneath the robot compresses like a spring as force is applied. As the robot shifts from one leg to another and applies less and more weight respectively, it should lengthen and compress the "spring" under each leg to handle that load. This action is exactly analogous to a series elastic actuator, in which the force that is output is proportional the displacement of the spring. If each leg were fully force controlled, as on our newest humanoid THOR, the motion required to adapt to compliant ground would happen automatically. It is promising then that by mimicking a fully force controlled leg, SAFFiR can tackle compliant ground, and it bodes well for a fully force controlled robot.

4-13 Torso Windmill Strategy

The final component of the balancing controller is the Torso Windmill Controller (TWC). The TWC provides a restoring force in the x and y directions at the foot by accelerating the torso about the hip pitch and roll joints. It is very similar in principle to the action of a reaction wheel pendulum. The implementation though is more difficult because the torso has a very limited range of motion and it is desirable to have the torso upright as much as possible. Therefore, more complex controllers are required for a significant improvement in stability. Such a controller is proposed in Section 7D - 2. For SAFFiR a simpler controller was implemented as a proof of concept. This section will describe that controller.

SAFFiR's TWC commands a torso position proportional to the COM error. That is, the difference between the torso's nominal position and the actual position is proportional to the error in the COM position, as described by Equation 4-30:

$$\theta_{t,roll,actual} = \theta_{t,roll,nom} + k_t(d_1 - d_{1des}) \tag{4-30}$$

where $\theta_{t,roll,actual}$ is the actual torso roll angle with respect to the gravity vector, $\theta_{t,roll,nominal}$ is the nominal angle of the torso (usually zero), and k_t is the gain. An identical controller is also implemented for torso pitch. In some examples, this controller can closely approximate a more complex one. The optimal controller would servo the torso to its maximum position when a large disturbance is detected. Ideally the torso comes reaches zero velocity well before the COM reaches the boundary of the support polygon. When tuned correctly, the controller in Equation 4-30 will roll or pitch to its maximum allowable angle but do so slower than the optimal one, only stopping the torso the moment the COM reaches the boundary of the support polygon. However, this controller automatically returns the torso to its nominal position when the COM error is zero.

The performance of the TWC was appeared acceptable, but was never quantified and remains an area of study. The stability of the robot with the controller should be compared to that without and with a more complex one as well, such as that proposed in Section 7D - 2. However, visually, the results looked promising. SAFFiR reacted as one would expect a person to do when pushed or disturbed. It is possible that the simplicity of this controller has some merit in and of itself. A comparison between the several options will be explored as part of the continuing research.

4-14 Conclusions

In this Chapter the concept and implementation of a balance controller was detailed. Additionally, the controller was framed within the work of past researchers including ZMP, Passive Dynamic, and Momentum controlled robots. The balance controller aims to stabilize the robot COM on a desired trajectory. It does so through the application of ground reaction torques applied by the ankle. The controller operates in one of two states, single or double support. In both phases the state of the COM is estimated using a Luenberg observer operating with a linear inverted pendulum model. The observer serves to eliminate unwanted noise from the velocity estimates. An additional notch filter is applied to further reduce extraneous readings from the unmodelled flexibility within the robots structure and associated dynamics. In single support, the controller runs a PID loop with feedforward compensation for gravity and acceleration terms. In double support, the controller operates a similar loop to act across the axis connecting the ankles. Along that same axis though, there is a lone derivative controller to damp out unwanted oscillations stemming again from unanticipated flexibility.

Two additional controllers work to augment the stability of the balancing controller. The torso windmill controller is designed to apply restoring forces through the application of torques about the hip joint. A proof of concept system was implemented that moved the torso in proportion to the COM error. While experimental data is yet to be attained, qualitative results suggest that the balance is improved. This work suggests that further study is appropriate. Secondly, a step height controller was added to improve stability on compliant ground such as Astroturf. The similarities between this controller and a series elastic actuator were compared. They also indicate that a full force control architecture is amenable to compliant and non-stationary terrain. Finally, an experimentally successful proprioceptive module to determine foot contact with the ground was described.

This work demonstrates a successful balance controller as implemented on a mixed force and position controlled robot. The balance controller shares many characteristics with momentum based control approaches utilizing full force control architectures, and therefore bodes well for future projects.

5 Step Controller

With an effective balancing algorithm, the problem of taking a step is simplified. Walking is simply a matter of providing a suitable trajectory around which the balance controller can stabilize the robot. These trajectories can be any number of the robots states. Literature shows examples of joint, COM, ZMP, and Capture point trajectories; all of which have successfully demonstrated walking. It is worth reiterating that SAFFiR is a mixed force and position controlled robot. While most of the joints in the robot utilize rigid position controllers, the ankles operate in force control. A unique step controller is necessary then to utilize and take advantage of this characteristic. SAFFiR's step controller does that by jointly commanding the torso and COM position along with the feet positions to produce a stable walking gait.

The mixed control architecture of SAFFiR flexibly links the COM and torso positions with the torso orientation. To accurately control the torso orientation in the gravitational frame requires coordinated control of the COM, the torso position, and foot position in the robot's base frame. This behavior is a result of the feet and torso positions and orientations being specified in a non-inertial frame. Instead their states are defined only with respect to one another. The COM on the other hand is specified in a fixed inertial frame at the ankle.

The separation of COM and torso states allows the joint angles to be mostly independent of the COM feedback loop. The position controlled joint angles are specified by the relative position of the torso and feet and therefore operate free of the noise inherent in the feedback loop. The force controlled joints are modulated based on the balance controller described above. The step controller allows this strategy to work by specifying both COM and torso trajectories. When the balance controller drives the COM to the desired state, it in turn drives the torso to the desired position and orientation. Small errors in the coordination between the COM and torso orientations are eliminated at each footfall, when the step controller re-orients the independent coordinate frames.

All trajectories are specified with respect to the current footstep. As one footstep is completed, the controller uses the next step in the queue to generate goal positions from which it can compute the trajectories. All footsteps are stored in a queue with each footstep object containing all relevant information regarding that step. The information includes the x, y, z position of the step, the orientation of the step, height and position of any tripping hazards, the duration, and which foot will take the step. Footsteps are added to the queue by a higher motion planning control algorithm using exteroceptive data.

The step controller must also specify the foot trajectories and manage the various walking states. During each single support the step controller generates a trajectory to first lift, then move, and finally lower the swing foot. This ensures the swing foot clears any obstacles or tripping hazards and that the foot is placed gently back on the ground at the end of the swing phase with as little disturbance as possible. As part of that job, it is also responsible for using the proprioceptive data to determine when to transfer between the various walking states, including two double support phases and two single support phases.

This chapter will describe the function of the step controller. First the various states with the state machine that runs within the step controller will be illuminated. The entry, function, and exit of each state will be described. The state machine provides a convenient framework to describe the various components of one step. Within each component, the primary function of the step controller is to generate appropriate trajectories for the COM, torso, and feet. How these trajectories are generated in each state will be covered. Finally, experimental results from the implementation of the step controller will be discussed as it pertains to various terrain such as grass, gravel, and shallow steps.

5-1 State machine

The step controller state machine coordinates a handful of unique states required for a walking gait by using exteroceptive and proprioceptive data. There are a total of five different states within the step controller. They include, Double support 1, double support 2, single support, lowering, and double support 3. Each state is unique and contributes a particular function to the gait. The state machine itself is a “function state machine” and so each state has entry, run, and exit functions. Furthermore, multiple conditions can be set to exit a state or transfer between states. This section will outline those states and the transfer criteria.

5-1-1 Double support 1

Each foot step begins in Double support 1. SAFFiR will be coming from either the standing posture, or from a previous step into this state and so will be in the double support posture. Upon entry for the first time, several parameters are set. The coordinates of the torso and swing foot with respect to the stance foot are computed. All trajectories are defined with these offsets as their initial state. Furthermore, the first footstep is pulled from the footstep queue and initial goals for the torso position set. These goals are used by the trajectory generators described below to coordinate the motion of the torso, COM, and swing foot.

After entry, the Double_Support_1 state is run every iteration of the control loop until it is exited. At each iteration the state calls a function to update the torso and COM trajectory. The trajectory is computed using the current robot state and the elapsed time. In this state both feet remain stationary while the torso moves above them.

The criteria for exit from Double_Support_1 is examined at each cycle. The criteria depends on the relative position of the COM from the stance foot and the absolute velocity of the COM. These are both computed within this state. The criteria for exit is given as Equation 5-1:

$$vel > 3.6 * d - 0.092 \tag{5-1}$$

Where vel is the absolute COM velocity and *d* is the relative position of the COM. This criteria is called the liftoff point. The purpose of liftoff is to estimate when the COM has both the necessary velocity and is close enough to the stance foot that the swing foot can be unloaded and yet the COM will still reach the desired state above the stance foot. The constants in Equation 5-1 were found experimentally by examining the COM state and foot loading conditions through a variety of step durations. One the velocity of the COM exceeds a certain value dependent on its position, it will exit Double_Support_1 and go into Double_Support_2.

5-1-2 Double support 2

Double_Support_2 is responsible for unloading the swing foot and moving the torso and COM into the appropriate state to begin single support. The state continues to call the torso and COM trajectory generator as in Double_Support_1. In addition, it generates a simple trajectory for the swing foot which raises the foot several millimeters. Although the leg is predominately position controlled, it still flexes under load. By raising the foot a small fraction, it causes the leg to become unloaded without extending. An extending motion of the swing leg during this phase has a tendency to push the COM past its goal. Lifting the leg eliminates this tendency.

Double_Support_2 exits upon time out. The time is set by the foot step queue. Upon exit, SAFFiR removes the next footstep from the queue to generate the next goals for the COM, torso, and swing foot. Following Double_Support_1 is the first of the single support states.

5-1-3 Single support

During Single_support state the swing foot moves to the new footstep location. Upon entry the ankle of the swing foot is put into position control mode, which allows the foot to be oriented appropriately for the upcoming terrain. The swing foot trajectory generator is called every cycle to update the desired position of the swing foot. The generator takes into account the elapsed time since single support began to determine the trajectory of the foot. The particulars are described in Section 5-3.

Single_Support looks for two exit criteria. The first is a time out. If the duration specified in the footstep queue is reached, the trajectory generator has commanded the foot to the required location. The second exit criteria is an unexpected contact with the ground. This can happen if the either the ground estimate is incorrect or a disturbance has caused the COG to move and thus the foot is in a different position relative to the ground than expected. If the timeout criteria is met, the robot will go to the Lowering state. If contact occurs, the robot will go automatically to Double_Support_3.

5-1-4 Lowering

The lowering state is responsible as the name implies for lowering the foot until it makes firm contact with the ground. This state is handled separately than Single_Support for simplicities sake. At each update a function is called to update the swing foot trajectory which moves the foot in the z-direction towards the ground at a constant velocity. Exit from the Lowering state occurs after the foot has either made contact with the ground. The robot goes into Double_Support_3 upon exit.

5-1-5 Double support 3

Double_support_3 is responsible for the first half of the double support phase of walking. After touchdown, the robot has a tendency to lean backwards due to the foot impact. Therefore Double_Support_3 first waits until the balance controller recovers and the COM is moving in the appropriate direction again. This avoids the tendency for the robot, while it is rocking backwards, try to move the torso forwards, which can cause the swing foot to lift from the ground. When the proper state is reached, Double_Support_3 begins to move the torso and COM towards the swing foot. Again the feet are held stationary with respect to the inertial frame and only the torso moves above the feet.

Double_support_3 exits upon timeout when the torso has reached the midpoint of the two footholds. At exit, new offsets are updated to reflect the new feet positions. New goals are also calculated to reflect the change in offsets. The swing foot and stance foot are also switched. Double_Support_3 transitions to Double_Support_1 as long as there are footsteps remaining in the queue. Otherwise it exits out of the step controller entirely and returns to the standing state.

5-2 Trajectory of COG

The double support walk controller is built around two different and semi-independent trajectories, the COM and torso. All trajectories are generated using minimum jerk trajectory calculations which require minimum overhead but generate very smooth motions [66]. As stated above, the torso and the COM are closely related. To ensure the robot is in the correct configuration, the torso orientation in the base frame, torso position with respect to the feet, and the COM position must correspond to one another. The trajectory generator is responsible for making this happen. This section will cover the trajectories generated during the five phases listed in Section 5-1.

During Double_support_1 the COM moves from its current location (often the midpoint of the two feet) to over the stance foot. Because the robot is in double support, the COM motion is primarily driven by the movement of the torso with respect to the feet as opposed to the ankle torques. The step controller generates trajectories for the x, y, and z components of the torso’s position that move it to the goal location. The COM and torso of course do not move with a 1:1 relationship. Therefore goal positions for the torso are estimated as in Equation 5-2:

$$Torso_{goal} = 1.3 * COM_{goal} \tag{5-2}$$

where $\text{Torso}_{\text{goal}}$ is the distance the torso must travel in x, or y such that the COM moves a distance of COM_{goal}. For walking motions, the proportion of torso to COM movement can be assumed to be constant. This value was measured experimentally to be 1.3. Torso orientation was kept constant during Double_Support_1 and Double_Support_2. For these preliminary tests, the COM comes to a rest at the end of each trajectory. Example trajectories for the torso can be seen in Figure 5-1 which show the x, and y positions and velocities respectively. An example COM trajectory can be seen in Figure 5-2.

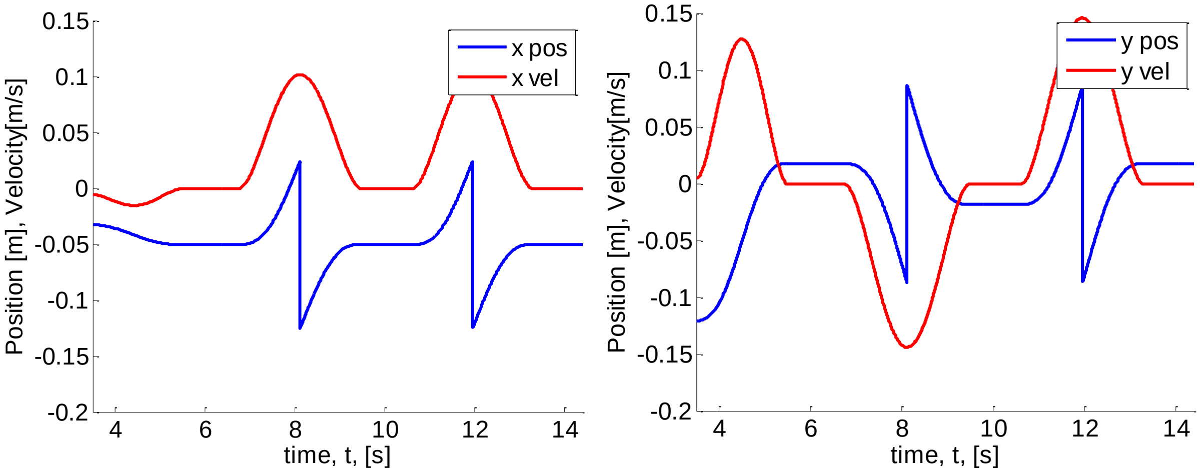

Figure 5-1. Trajectories for the torso in x and y directions are generated using minimum jerk criteria.

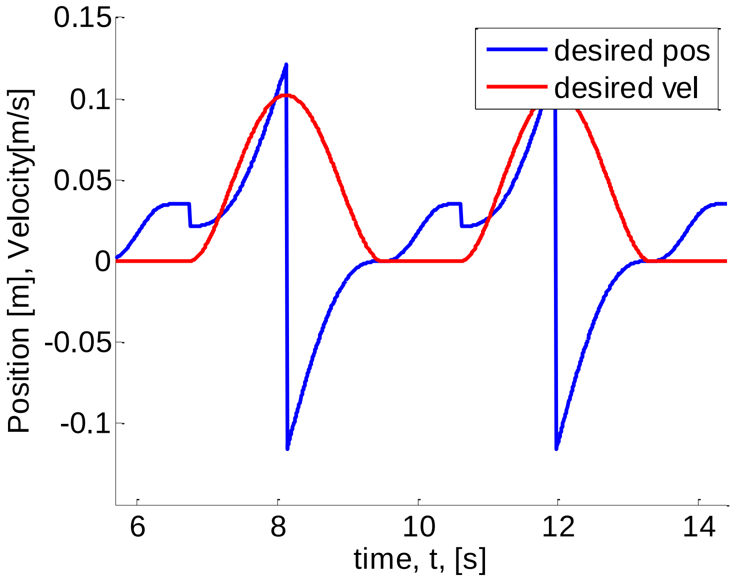

Figure 5-2. The COM trajectory in the x direction shows the single support (6[s] to 7[s]) and double support (7[s] to 9[s]).

During the single support phases, the COM naturally moves forward as the swing leg advances to the next foothold. To account for the shift in COM and ensure correct torso orientation, the desired COM position must be updated to match the expected motion. Therefore a desired COM trajectory is generated during single support to move the COM in the direction of the next foothold. This motion can be seen in Figure 5-1

5-3 Trajectory of Foot

The swing foot motion is defined in three different phases during single support: raising, lowering, and touchdown. The ankle operates in position control mode during the first half of the swing, and returns to force control during the latter half. In position mode the pitch and roll angles of the foot are able to be precisely controlled, ensuring the foot does not impact any obstacles and lands in an orientation approximating that of the ground normal vector. The duration of the majority of the swing phase is defined by the footstep queue. The touchdown sub-phase occurs outside of the Single_Support state and only terminates once ground contact has been made. The raising and lowering sub-phases of Single_Support are each half the duration of Single_Support.

Four trajectories are generated to describe four of the foot's six DOFs during the swing phase. Those four include the x, y and z position relative to the stance foot, and the pitch of the foot. The roll and yaw are zero for the duration. The trajectories are defined during the three different phases as follows:

X-direction) During the entirety of Single_Support the end goal of the x-trajectory is the next foothold's x position. The duration is the same as the length of Single_Support. There is no movement in the x direction during touchdown.

Y-direction) During the first half of Single_Support the goal of the y-trajectory is .15[m] to the side of the stance foot. The goal is the y position of the next foothold during the latter half. There is no movement in the y direction during touchdown.

Z-direction) During the first half of Single_Support the foot lifts in the z-direction to a height specified in the foothold queue to ensure any obstacles or tripping hazards are cleared. During the latter half the z goal is 0.015[m] above the expected ground location. The z trajectory is the only one that does not end with a zero velocity. Instead it ends with a downward velocity of 0.045[m/s]. It continues downward with a constant velocity during Touchdown until contact with the ground is made.

Pitch) The pitch trajectory is only defined during the lowering sub-phase of Single_support to prevent range of motion limits from being triggered. The goal state of the pitch trajectory is defined by the foothold queue.

Example foot trajectories can be seen in Figure 5-3.

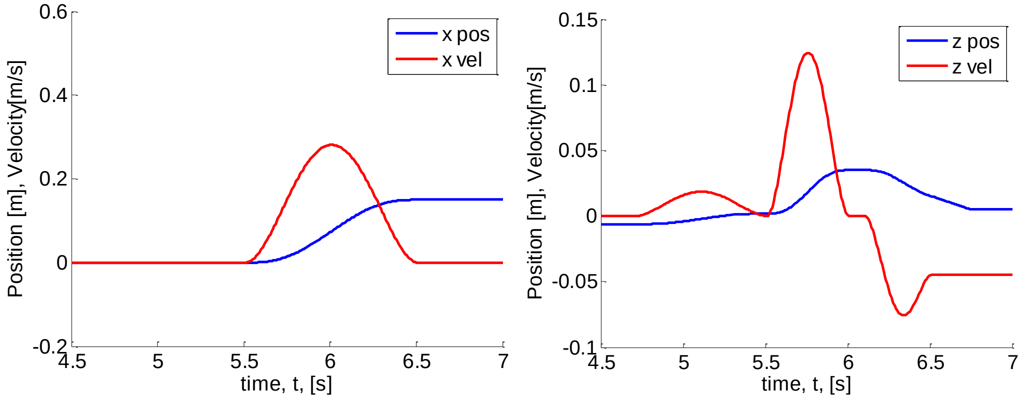

Figure 5-3. Minimum jerk trajectories were also used for the foot movement during the swing phase. The plots above show the trajectories for one step. Note the unloading phase between 4.5 and 5.5 seconds in the figure on right.

5-4 Results

Three different terrains were attempted with the walking algorithm described above. They included strewn plywood, loose pea gravel, and high pile synthetic grass. The particulars and figures from each terrain are shown below.

5-4-1 Gravel

Loose gravel was chosen as the first terrain because it was believed its damping characteristics would improve the stability of the robot. While this was found to be of little value, the terrain was still accomplished successfully. The gravel consisted of a 1.5[m] strip approximately .5[m] wide. A pit was fabricated from lumber and filled with 50[mm] of pea gravel. The gravel has an average diameter of approximately 10[mm]. At that depth, the consistency was such that excessive pressure from a point or edge would sink dramatically into the gravel.

One stride of SAFFiR into and across the gravel is pictured in Figure 5-4. The force control balancing algorithm worked well in this compliant terrain. As the gravel shifted under the feet due to changing pressures, the ankles would naturally react as they applied the commanded torque. This indicated that the force control algorithm detailed in Section ? was working correctly. Furthermore, small undulations in the gravel height had little effect on the stability of the robot due to the adaptability of the state machine in Section 4-8 to ground height errors.

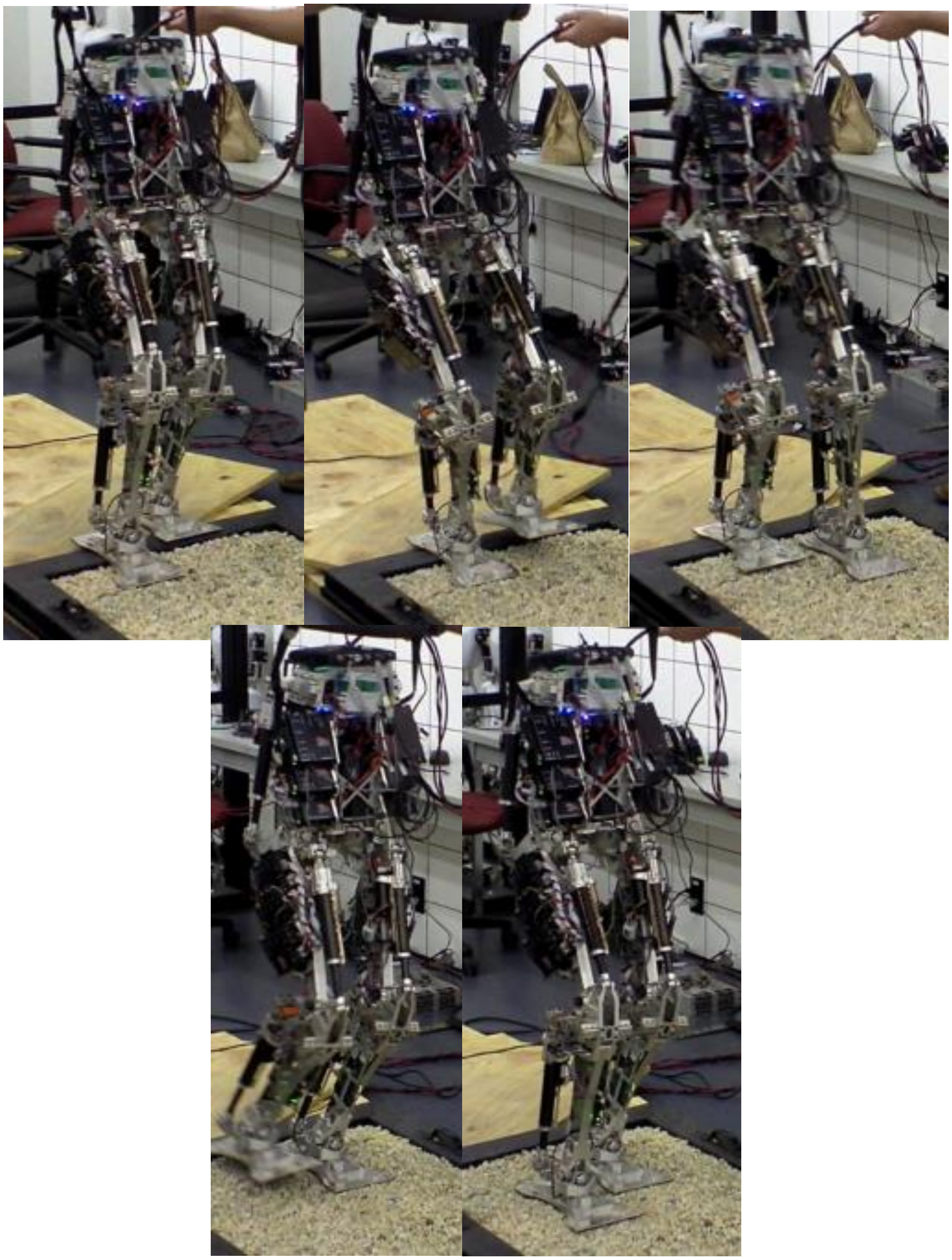

Figure 5-4. SAFFiR successfully handles uneven and non-stationary ground.

5-4-2 Grass

Grass presented the most challenging terrain for SAFFiR. The grass in this case was 50[mm] thick synthetic grass. The very tall nature of the grass acted like a very soft spring underneath of the feet as it walked. The results were twofold. First the robot would sense ground contact at a much higher position than on a stiffer surface. This effect was counteracted by the step-height controller in Section ?. Secondly, the low spring rate and relatively little damping within the grass required higher damping in the PID balance controller. The motion of the feet in the grass was vigorous. Every change in torque required a large change in displacement of the feet. The system as designed though was successful, as can be seen in Figure 5-5.

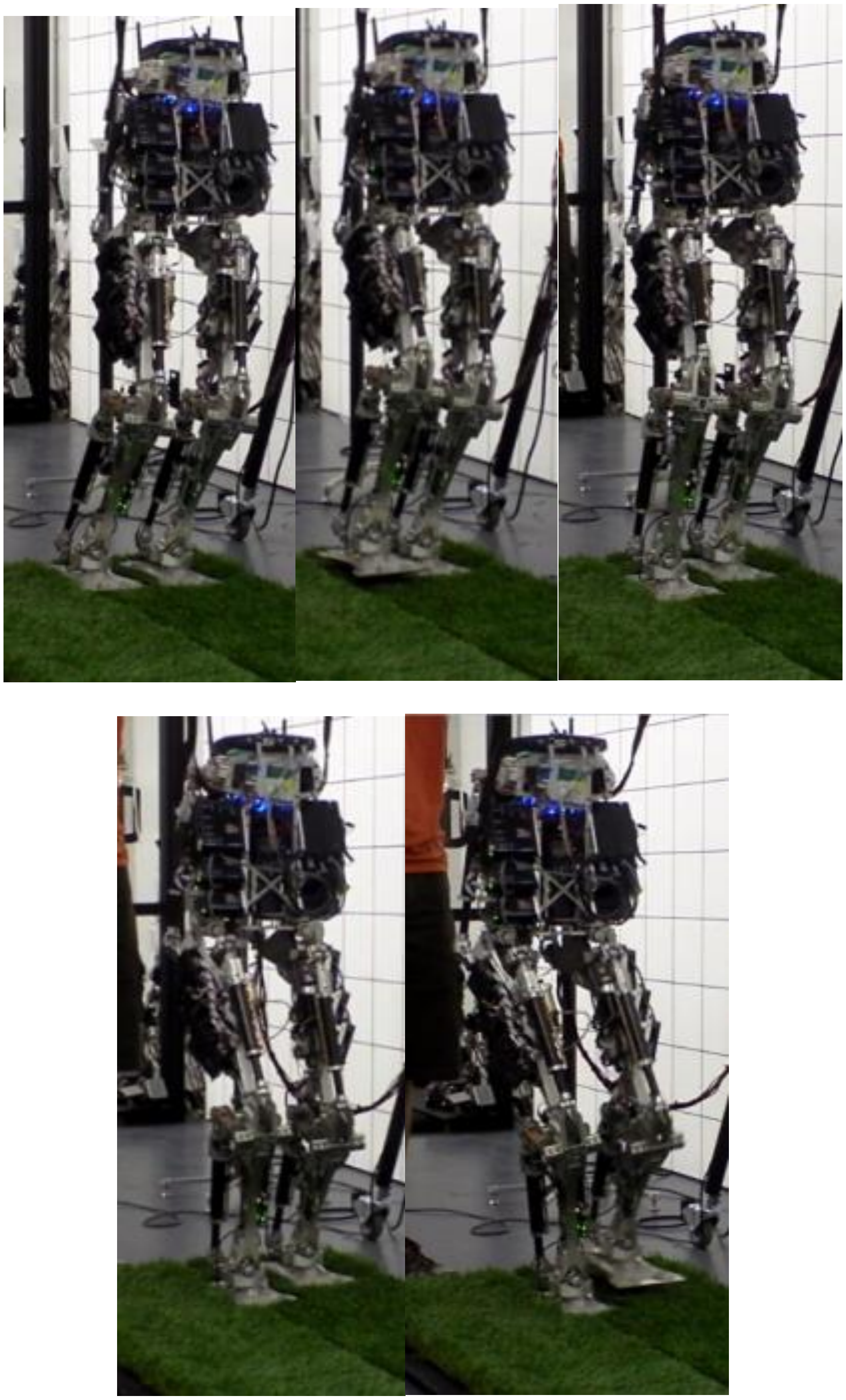

Figure 5-5. Grass constituted the most difficult terrain. The compliant nature required special behavior from the foot trajectory for stability.

5-4-3 Plywood

Strewn plywood demonstrated SAFFiR’s ability to walk up and down inclines on non-flat terrain. For this experiment, plywood sheets were cut down and placed at random angles to each other in the robots path. Each piece was laid atop the one preceding it to create a stepped inclining surface. The odd angles at which the plywood was laid emphasized the importance of force control. When stepping on an edge, SAFFiR’s feet were usually supported by only three points of contact. As torques were applied, the feet would rock back and forth between triangular support polygons. The inclined nature of the plywood also emphasized the positioning of the swing foot during the swing phase. Correct positioning and orientation of the foot allowed for gentle foot placement that minimized the disturbance to the robot. Step height in this case averaged 20[mm]. Much higher steps were not possible due to range of motion limitations in SAFFiR’s ankles. Step information was approximated to +/- 10[mm] for these steps and given apriori to the robot. It incorporated no exteroceptive sensing for this terrain. A successful stride can be seen in Figure 5-6.

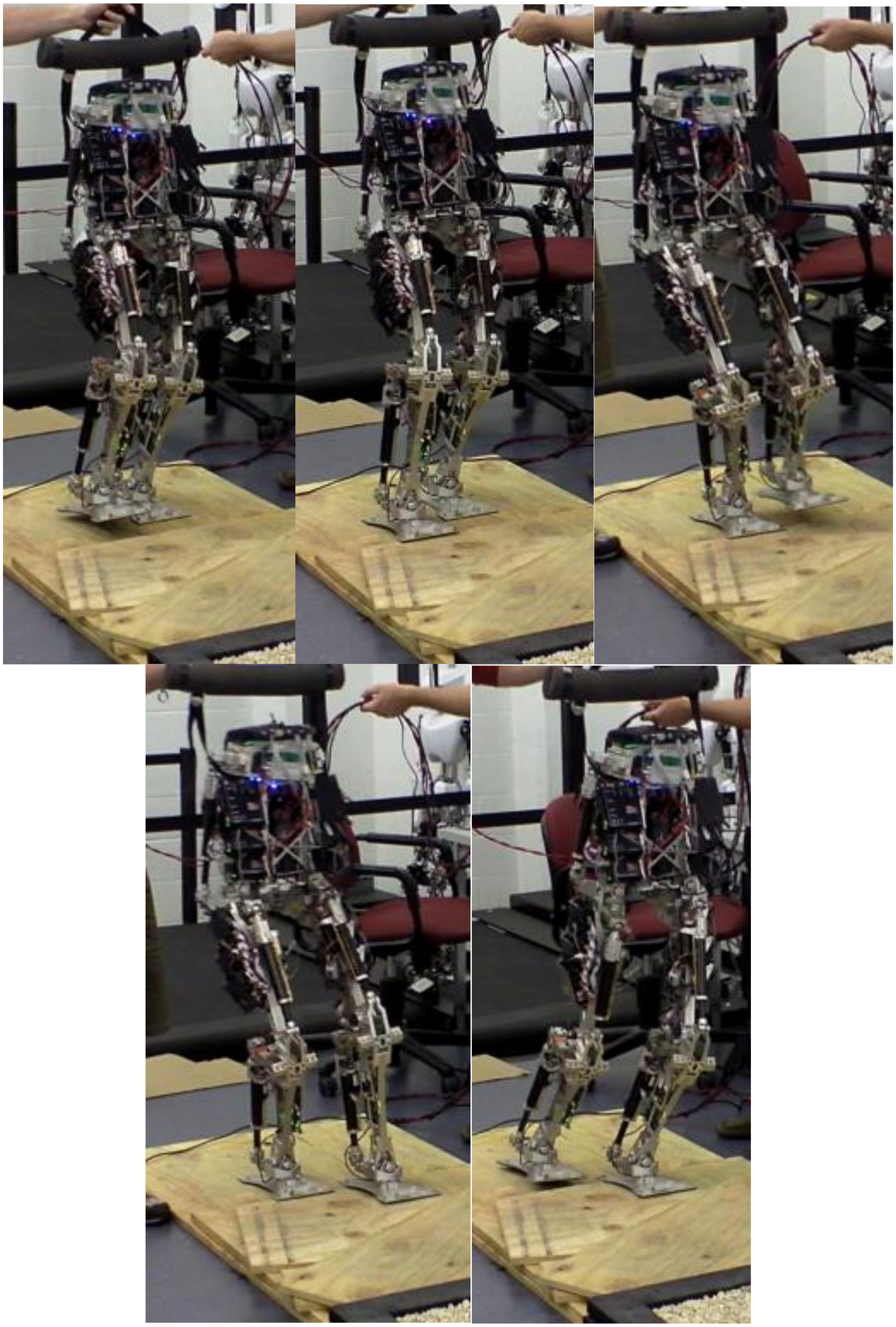

Figure 5-6. Strewn plywood demonstrated the ability of the walking algorithm to handle inclined and uneven terrain.

5-4-4 Sand

Sand was successfully attempted at the 2013 DARPA Robotics Challenge exposition area in December 2013. The sand was typical “playground” sand trucked in by NIST for their “Robot Playground,” a challenging obstacle course for all types of robots. The sand was roughly 100[mm] deep. Several textures of sand were tried to investigate the limits of the controller, varying from smooth to highly irregular. The smooth sand provided a much firmer surface for the robot, while the more irregular sand shifted under the robot more. SAFFiR successfully demonstrated walking on this sand under a number of conditions. It is believed SAFFiR is the first biped in the world to walk across sand. Again, SAFFiR incorporated no exteroceptive sensing for this terrain. A successful stride can be seen in

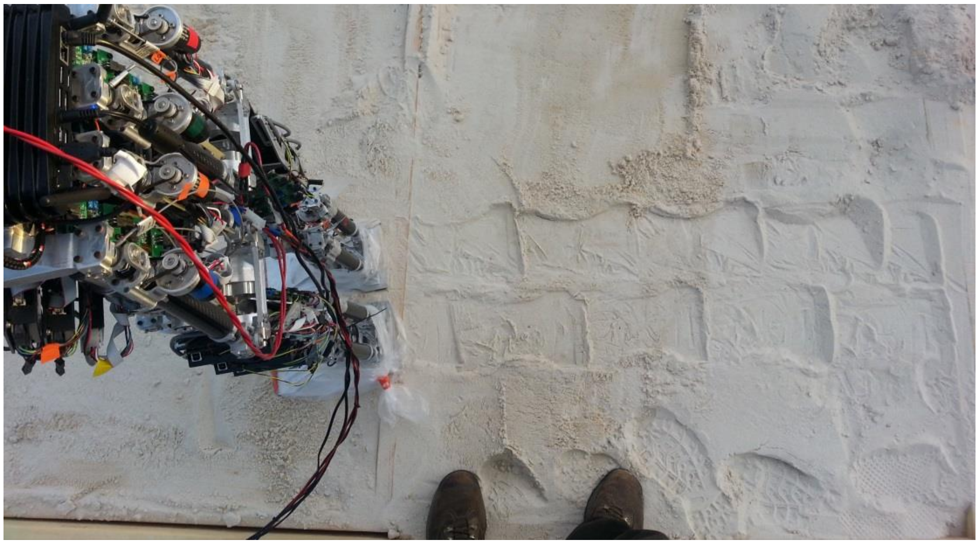

Figure 5-7. The first recorded bipedal robot footsteps in sand next to the authors.

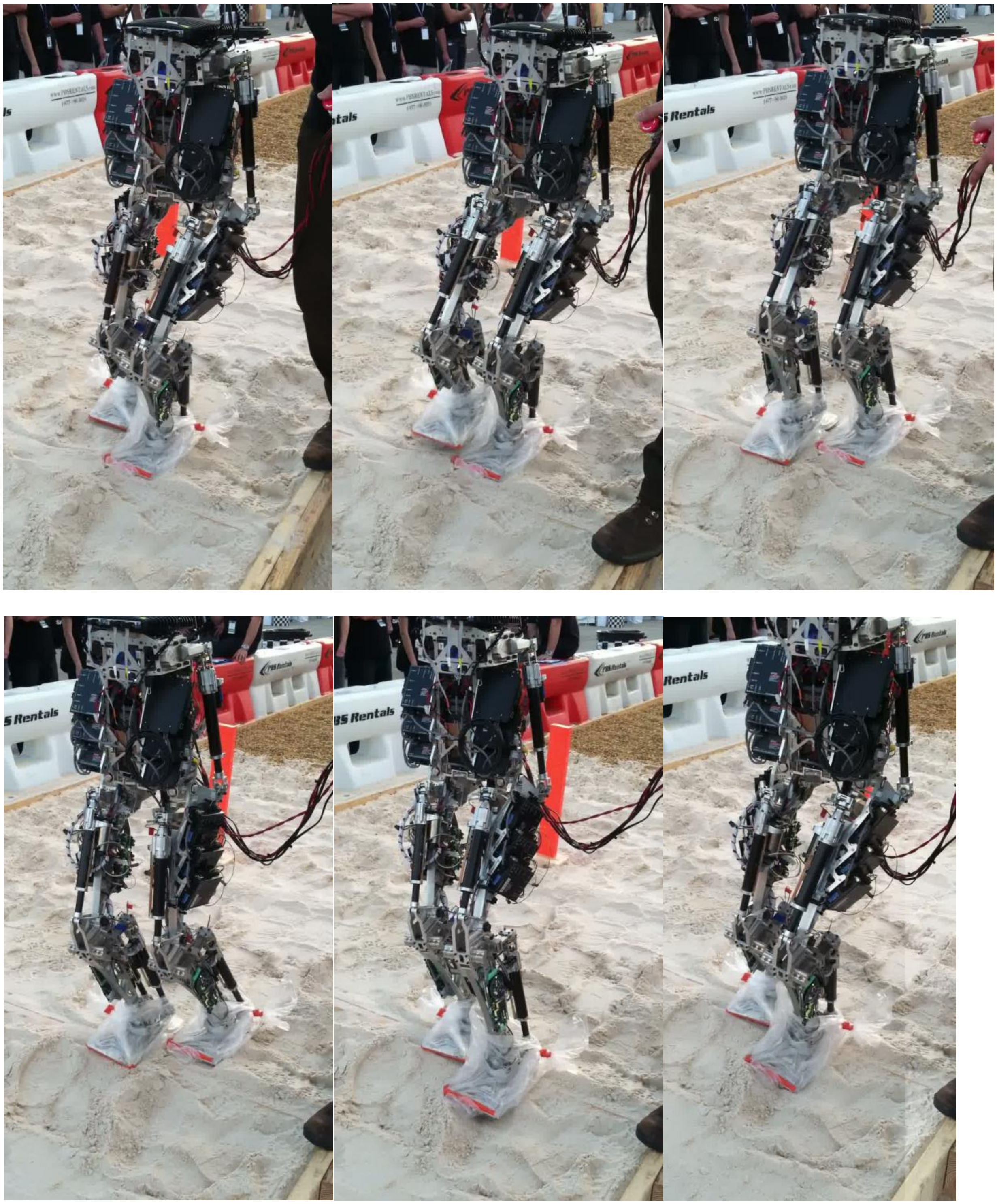

Figure 5-8. SAFFiR is the first bipedal robot to demonstrate walking in loose sand.

5-5 Conclusions

In this Chapter, a successful walking algorithm has been implemented on SAFFiR and proven to work on a variety of terrain. The walking approach is built around a robust balancing controller, as outlined in Chapter 5. As such, the walking controller need not to be complex to ensure stability. Instead the balance controller maintains stability, while the walking algorithm ensures locomotion is achieved. A state machine forms the basic structure of the walking controllers. Five states make up the state machine, including Double_Support_1, Double_Support_2, Single_Support, Lowering, and Double_Support_3. The transitions between states are controlled by either timing or proprioceptive based events.

Within the state machine, torso, COM, and foot trajectories are computed to generate the motion of the position controlled robot joints. Trajectories computed online using footstep information from a queue that itself is filled by higher level behavioral/planning software. The trajectories themselves are computed using minimum jerk criteria. The torso and COM positions are computed jointly to ensure correct torso orientation. Foot trajectories are segmented into three phases to account for liftoff, positioning, and touchdown.

The walking and balance controller were operated on four different terrains successfully, strewn plywood, gravel, high pile synthetic grass, and deep sand. The combination of balance controller and walking algorithm were able to maneuver over all four. Most notable from the experiments were the effectiveness of the ankle force control balance strategy that performed regardless of terrain conditions. The results bode well for the application of a well-designed force controlled balance strategy for rough and non-stationary terrain.

6 Improved Walking Trajectory

Trajectory planning is an important and lively area of research for bipedal locomotion. It is important because humanoids are inherently unstable machines. Care is required to generate appropriate trajectories that are both feasible and stable. Feasibility is determined by both the joint range of motion constraints as well as the constraints imposed by the reaction forces generated by the feet. Among mobile robots, these constraints are particularly tight for humanoids. They walk upright, with the knee often very near a singularity, and thus joint velocity limits can be easily exceeded[67]. The support polygon of a humanoid is as small as or smaller than any other mobile robot and so the forces they can exert on the ground are similarly limited. It is therefore important to generate trajectories intelligently such that the stability to disturbances is maximized given these constraints.

The problem of trajectory planning is also a rapidly progressing one. The problem is quite difficult (and unsolved) due to the complex dynamics of a humanoid, and thus many solutions are continuously being proposed. The approaches generally can be divided into two categories, those that use the full dynamics in a large optimization to root out a feasible solution[68]–[70], and those who derive trajectories from simplified dynamics[67], [71]–[74]. The techniques described below fall into the latter category.

It is widely held that in nature, the center of mass trajectory during bipedal locomotion takes on an inverted pendulum form[62]. Similarly, the simplified dynamics of humanoids are derived from the inverted pendulum. The most popular and widespread simplification is that of the Linear Inverted Pendulum model. This method was popularized in the 90’s by Kajita and Tani [75][72].Their approach allows for closed form analytical solutions, but constrains the dynamics of the center of mass to a plane which in some cases is detrimental. Even so, the simplicity of this approach has spurred numerous other results. Morisawa demonstrates how such a method can be extended to an arbitrary surface, to account for uneven terrain or alleviate kinematic constraints [67]. Tang applies the 3D inverted pendulum problem to gait generation in [74]. Tsuji extends the LIPM model to include virtual support points, akin to using ankle torques for stabilization.

There is some work in extending the LIPM model to include additional bodies. Park adds point mass to the dynamics of the swing leg in [76], but went on to reduce the model to include only the effects of gravity. Albert picks up from this point and prescribes and models accelerations of the point mass swing leg [71]. His method is continued to include multiple models. The work presented here differs in that the effects of the inertia and the mass of the swing leg are studied, and shown to be a more accurate representation of the dynamics with no additional complexity. Additionally, this work develops a method for 3D gait generation using the extended formulation.

6-1 Background

There are three main mechanisms for disturbance rejection in humanoid robots: ankle torque, flywheeling, and step placement[29]. During walking the center of mass trajectory most impacts the effectiveness of ankle torques. If all available control authority is required to maintain a trajectory, there is none left for disturbance rejection. It is desirable then to generate a trajectory that minimizes use of ankle torques.

The foot generates a 6 DOF wrench on the ground. The wrench can be interpreted as 3 forces and 3 torques exerted at the ankle. The second representation is as a twist vector that passes through the support polygon. To maximize the disturbance rejection capability, this twist should pass through the ankle. Any disturbance can be then be controlled by moving the twist to a different location within the support polygon, which is akin to exerting torques on the ground about the ankle joint. An ideal COM trajectory will then be one that can be completed using only twists passing through the ankle. A simple telescoping pendulum with point mass can be shown to meet this criterion.

The premise of this contribution is that the forces on the COM can be decoupled into those generated by gravity (including those generated by robots as a reaction to gravity), and the forces generated by the internal dynamics of the robot moving through space. In this section, it will be shown that the dynamics of a LIPM with multiple masses can be devolved into two separate forces on the COG with a change of variable. Figure 6-1shows the system in question. For simplicity and to relate to literature, the inertias of the two links are omitted; however, the results still hold true.

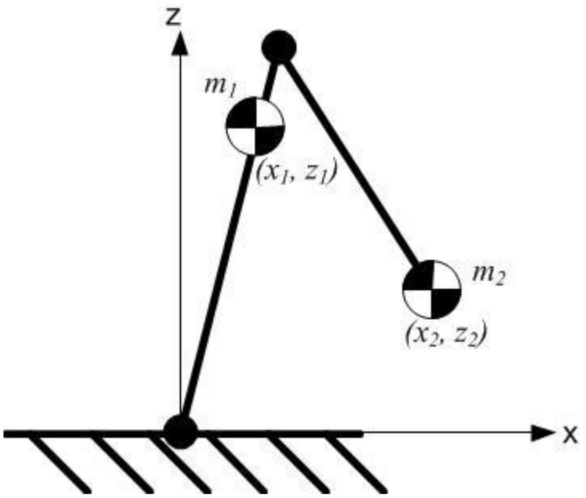

Figure 6-1. Two link pendulum with point masses and no inertia.

The derivation begins with the results seen in [71]. In this work, Albert quantifies the forces acting on the point mass m_1 with its linearized dynamics. The relationship is given as in Equation 6-1:

$$m_1\ddot{x}_1 z_1 - g x_1 = m_2(g x_2 + x_2 \ddot{z}_2 - \ddot{x}_2 z_2) \qquad \textit{(6-1)}$$

where g is the acceleration due to gravity. For this analysis, the acceleration in the z-direction is assumed to be zero for mass m_2. This is a safe assumption because the height of the swing legs center of mass in practices moves only 20-30[mm], and therefore the acceleration during the swing phase is very small. To isolate the center of mass motion from the motion each mass with respect to one another, the following substitution is made:

$$x_2 = x_1 + \alpha \tag{6-2}$$

$$\ddot{x}_2 = \ddot{x}_1 + \ddot{\alpha} \tag{6-3}$$

The following definitions are also made:

$$x_{cog} = \frac{m_1 x_1 + m_2 x_2}{m_1 + m_2} \tag{6-4}$$

where x_{cog} is the x location of the COM. Similarly:

$$z_{cog} = \frac{m_1 z_1 + m_2 z_2}{m_1 + m_2} \tag{6-5}$$

where z_{cog} is the height of the center of mass. Because the accelerations in the z direction is zero, the height of the COG is also fixed.

Substituting Equation 6-3 into 6-1 and rearranging yields:

$$\ddot{x}_1(m_1 z_1 + m_2 z_2) = g(m_1 x_1 + m_2 x_2) - m_2 l_2 \ddot{\alpha} \tag{6-6}$$

Solving for x_1 and substituting Equations 6-4 and 6-5 yields:

$$\ddot{x}_1 = g\frac{x_{cog}}{z_{cog}} - \frac{m_2 l_2 \ddot{\alpha}}{(m_1 z_1 + m_2 z_2)} \tag{6-7}$$

Equation solves for the acceleration m_1 given a prescribed motion of m2. A simple force balance on the COM and substitution of Equation 6-3, will result in the following equation:

$$m_{cog}\ddot{x}_{cog} = F_x = m_1\ddot{x}_1 + m_2\ddot{x}_2 = m_{cog}\ddot{x}_1 + m_2\ddot{\alpha}_2 \tag{6-8}$$

where F_x is the force generated by the ground to accelerate the center of mass. Similarly, this same equation can be derived by taking the second derivative of Equation ?.

Substituting Equation 6-7 into Equation6-8 and simplifying yields:

$$m_{cog}\ddot{x}_{cog} = g m_{cog}\frac{x_{cog}}{z_{cog}} - \alpha\frac{m_2(z_1 - z_2)}{m_1 l_1 + m_2 l_2} \tag{6-9}$$

Equation 6-9 shows that the dynamics of the COG are driven by two forces, the one due to gravity, and that due to the internal dynamics. Importantly, in the linearized example, the dynamic forces are independent of the position of either mass. As will be seen, this is important in that it allows for the analytical integration of the COM differential equation.

An alternate derivation is now presented in which the LIPM is extended to include the swing leg dynamics, but in a way as to abstract the forces generated by swing leg dynamics. Afterwards, the force generated by the swing leg dynamics is derived separately. The free body diagram in Figure ? is used to derive the equations of motion for a LIPM. The center of mass, M, is constrained to lie on a line. To maximize the available ankle torque, all forces must pass through the ankle.

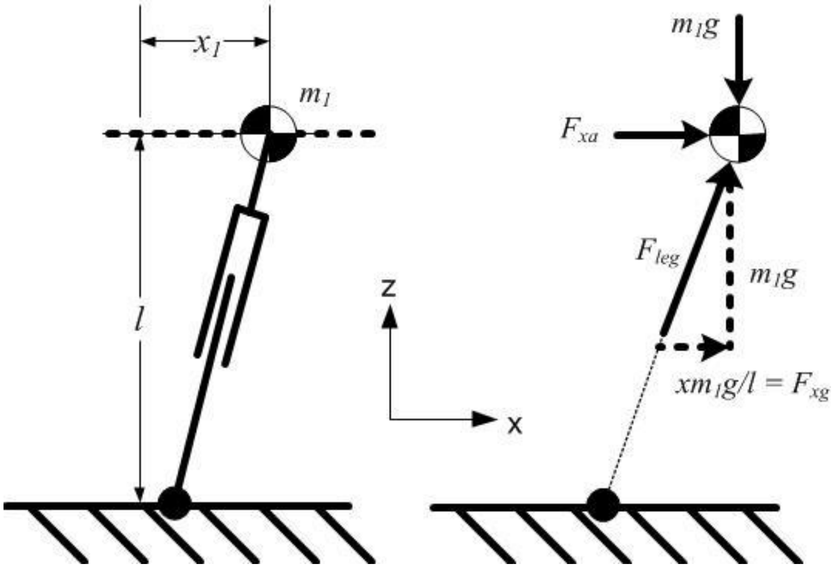

Figure 6-2. Free body diagram and variables used to derive the LIPM with swing leg dynamics.

There are two forces at play, one to counteract gravity, Fz, and Fx, which is generated by changes in angular momentum about the center of mass or as a component of the force required to support the COG. Specifically, Fx can be written as in Equation 6-10

$$F_x = F_{xa} + F_{xg} \tag{6-10}$$

Where F_{xa} is the component of the x-force that generates any angular momentum, while the F_{xg} is the component of the force along the pendulums length required to counteract gravity. Using the similarity of triangles, the following relationship can be written:

$$\frac{F_{xg}}{F_z} = \frac{x}{l} \tag{6-11}$$

where *l* is the height of the COG. Sum of the forces in the x direction yields:

$$M\ddot{x} = F_x = \frac{Mgx}{l} + F_{xa} \tag{6-12}$$

Or:

$$\ddot{x} = \frac{gx}{l} + \frac{F_{xa}}{M} \tag{6-13}$$

As mentioned, the force F_{xa} is that due to changes in the angular momentum of the center of mass. One such change in angular momentum is that due to the acceleration of the swing leg during single support. This has been shown to be a significant force affecting the dynamics of the center of mass[albert]. However, Albert makes the simplification that the swing leg is a point mass. This however is actually a fairly course representation, and can significantly change the outcome of the model simulation for the worse. For example, in the field of passive dynamic walkers, Weiss has shown that the swing leg can be used to propel the COG of an underactuated biped forward.

In practice though, the important matter is the force generated by this motion on the COG, not the stance leg. Weiss explains that if the inertia of the swing leg is large enough, the torque on the stance leg necessary to accelerate the inertia of the swing leg is greater than the torque due to the force required to swing the legs center of mass forward. Put simply, the inertia of the leg is a significant piece of the dynamics that is useful to model. In this work, a representative model of the swing leg dynamics will be derived and added to the LIPM model presented in Equation 6-13.

6-2 Motivation: Estimation and significance of swing leg dynamics

The question remains: how important is this force due to the swing leg. In this section the relative magnitude of this force for a realistic biped will be compared to that of gravity, the other most significant force on the COG. As such, this section is primarily a “back of the envelope” justification for the remaining work. The estimation is made using the simulation described in Appendix ?. Pertinent data such as step time and step length is gathered from experience. The estimation takes into account an average stride length and step time to determine the minimum acceleration necessary to move the swing leg the appropriate distance. The simulation is used to determine the force generated at the feet by this acceleration. The magnitude of this force can be compared to that of gravity during the nominal single support phase.

The biped model is based off of the SAFFiR robot. It has a nominal COG height of .7[m], leg length of .7[m], and steps 0.2[m] for an average step. One second is given for the single support phase, which agrees with the step time of many similar biped in literature. For this step length and leg length, the leg must rotate the amount given in Equation:

$$\theta = 4 * asin(\frac{d_{step}}{2l_{leg}}) \tag{6-14}$$

which is essentially twice the intra-leg angle when the robot is in double support at that step length. The minimum magnitude angular acceleration profile of the leg is a square wave of amplitude:

$$\alpha = 4 * \theta / t_{step}{}^2 \tag{6-15}$$

where t_{step} is the step time. This acceleration profile has infinite jerk, and so would likely not be appropriate for real life, but serves as a minimum baseline to estimate the force generated by this acceleration. Substituting the appropriate values, an angular acceleration of ~4[rad/sec^2] are required.

The simulation was run with the robot in a static pose, with both legs straight, one on the ground, the other free to swing. The acceleration was applied to the swing hip while the reaction force at the stance foot was measured. For an acceleration of 4[rad/sec^2], a reaction force of ~7[N] is required. This is analogous to a 7[N] force on the COG accelerating it forward. Due to the symmetry of bipedal locomotion, the single support phase of the nominal biped occurs for roughly half of the step length, or

0.1[m]. With a mass of 20[kg] at a COG height of 0.7[m], the maximum force exerted by gravity to accelerate the COM in the horizontal plane is:

$$F_{xg} = mglx \tag{6-16}$$

which when substituted for, produces forces on the order of 14[N].

In conclusion, the force generated by the swing leg can be a significant portion of the total forces acting on the COM. It is noteworthy that this force is opposite the force of the swing leg on the stance leg. Furthermore, the proposed method includes the inertia of the swing leg, which can have a large effect on the ground reaction force, and more important, generalizes changes in the centroidal angular momentum to only their effect on the COM position. The method presented below then allows for the use of a lumped mass model that features the simplicity of a COG model, but incorporates the additional dynamics. This has the benefit of allowing direct trajectory generation of the COM which is useful when the field of humanoids is moving towards centroidal dynamics as a control method.

6-3 Derivation of reaction forces

A simple relationship between the angular acceleration of the leg and the reaction force on the ground is desired. Because the LIPM equations of motion are linear with respect to the position of the COM, the relationship of F_{xa} to α should also be at most a function of x, ideally one that is analytically solvable. In this section, this relationship will be explored.

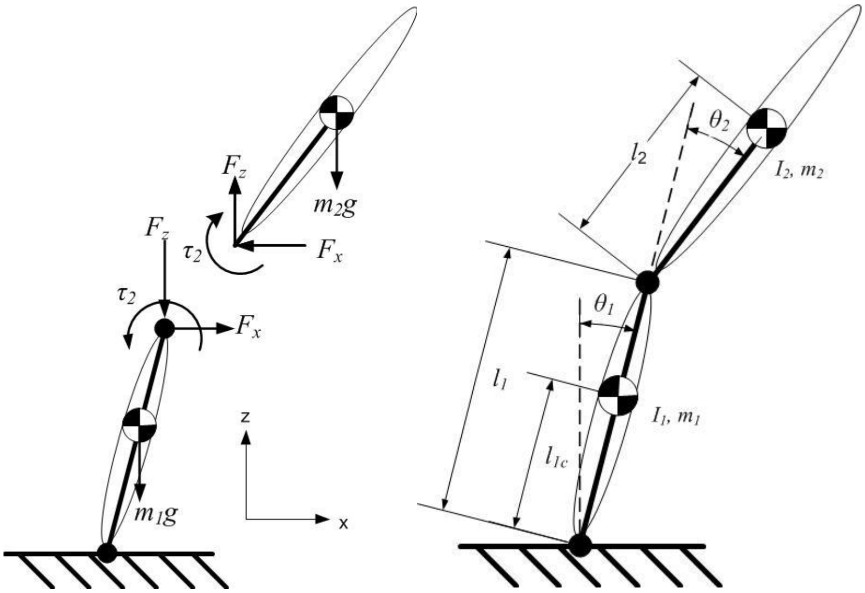

Figure 6-3. Free body diagram of a two link pendulum.

The Free body diagram of the system in question is shown in Figure 6-3. This figure depicts a two link pendulum. One link represents the stance leg and torso, while the second represents the swing leg. Both links have masses and inertias. The linear acceleration of swing leg is derived first. The coriolis and given as:

$$F_x = m_2\ddot{x} = -l_1\alpha\,cos(\theta_1) + l_2(\alpha_1 + \alpha_2)cos(\theta_1 + \theta_2) \tag{6-17}$$

$$F_z = m_2\ddot{z} = -l_1\alpha\,sin(\theta_1) + l_2(\alpha_1 + \alpha_2)sin(\theta_1 + \theta_2) \tag{6-18}$$

where the variables are shown in Figure 6-3. Sum of the torques on the swing leg is given as:

$$\tau = -(\alpha_1 + \alpha_2)I_2 - m_2\ddot{x}l_2\,cos(\theta_1 + \theta_2) - m_2\ddot{z}l_2\,sin(\theta_1 + \theta_2) \tag{6-19}$$

Sum of the torque on the stance leg is given as:

$$\left(I_1 + m_1{l_{1c}}^2\right)\alpha_1 = -F_x l_1\,cos(\theta_1) - F_z l_1 sin(\theta_1) + \tau \tag{6-20}$$

Sum of the forces in the X direction on the swing leg can be written as:

$$m_1(-l_{1c}\alpha_1\,cos(\theta_1)) = F_r + F_x \tag{6-21}$$

Equations 6-17 through 6-21 can be combined, solved for F_r and linearized about $\theta_1 = 0$ and $\theta_2 = 0$.

$$F_r = \frac{\alpha_2 l_2 m_1 m_2 {l_{1c}}^2 + \alpha_2 I_2 m_1 l_{1c} + \alpha_2 {l_2}^2 m_1 m_2 l_{1c} - \alpha_2 l_1 l_2 m_1 m_2 l_{1c} + \alpha_2 I_2 l_1 m_2 + \alpha_2 I_1 l_2 m_2}{m_2 {l_1}^2 - 2 l_2 m_2 l_1 + I_1 + I_2 + {l_{1c}}^2 m_1 + {l_2}^2 m_2} \tag{6-22}$$

As is apparent, the reaction force Fr is linearly related to the angular acceleration of the swing leg.

6-4 Trajectory

The trajectory generation for the COG is now considered. For this work, the footsteps are assumed be well defined, meaning their location is known before hand and they cannot be altered. These conditions represent a rough terrain walking scenario in which rubble debris dictates very strictly where a foot can be placed. It is necessary then to alter the COG trajectory to fit these footsteps. To adjust the trajectory, two parameters have been identified that correlate nicely with heuristics that can guide the selection of a trajecoty. Those parameters are the forward velocity when the horizontal velocity is zero, and the x and y location at which there is no horizontal velocity. It will be shown that these parameters make the problem easy to solve by hand, or with a rudimentary search program.

A suitable trajectory is one in which the path taken by the COG intersects both the prior and approaching footsteps, as seen in Figure?. There is no one solution, and thus the best trajectory is chosen that most closely matches the desired forward velocity and yet still allows enough time in single support to reposition the swing leg for footfall. This latter criteria generally has the effect of reducing the forward velocity, and so in some cases a compromise must be made. In this work it is the desired velocity that has a lower priority, because in rough terrain, any forward motion is desirable to falling.

This work also uses a piecewise continuous model of the dynamics, a distinction from prior work. During leg swing, the model in Equation 6-22 is used to represent the dynamics of the leg swing. However, the trajectory before and after leg swing are needed to find the footstep intercepts. The conventional LIPM is used before and after the leg swing phase to calculate these intercepts. The time interval of leg swing is thus important to the dynamics, as well as the relative position of the COG at which point the leg swing occurs. TO make the problem tractable, the assumption is made that the leg swing occurs when the COG position along the y-axis with repect to the stance foot is less than 0.07[m]. The width of the foot is

0.05[m] in that direction and the typical step width is .2[m], so a swing phase at 0.07[m] places it firmly in between a statically stable walk and the point of footfall.

Finally, the motion in the x-direction and y direction are decoupled in terms of the dynamics. This is a fairly conservative assumption which greatly simplifies the problem. However, because of the parameterization of the problem, it is not much more difficult to use a coupled model. The motion in the x and y direction is coupled though in that the COG must pass through the foot intercept in both directions at the same time. Therefore although they are dynamically independent, they must be considered together. The differential equations governing the motion of the COM are given by Equation 6-23:

$$\ddot{x} = \frac{gx}{l} + \frac{F_{xa}}{M} \; when \; |y| < 0.07$$

$$\ddot{x} = \frac{gx}{l} \; when \; |y| \geq 0.07 \tag{6-23}$$

$$\ddot{y} = \frac{gy}{l}$$

The parameterization is a unique contribution to the determination of the COM trajectory. As mentioned before, there are three main parameters to alter that completely and define the motion. They are unique in that the relationship between trajectory features and parameters is nearly linear and thus the solution is quite intuitive. The parameters are the x-velocity when the y-velocity is zero, and the x and y location of the point at which the y-velocity is zero. For convenience, this point will be called top dead center, or TDC. The effect of these parameters can be seen in Figure 6-4 below. The x-position of TDC adjust the angle of the two asymptotes in the same direction. The y position of TDC adjusts the angle between the two asymptotes. Similarly but not pictured is the effect of the x-velocity at TDC, which like the y position, affects the angle between the asymptotes.

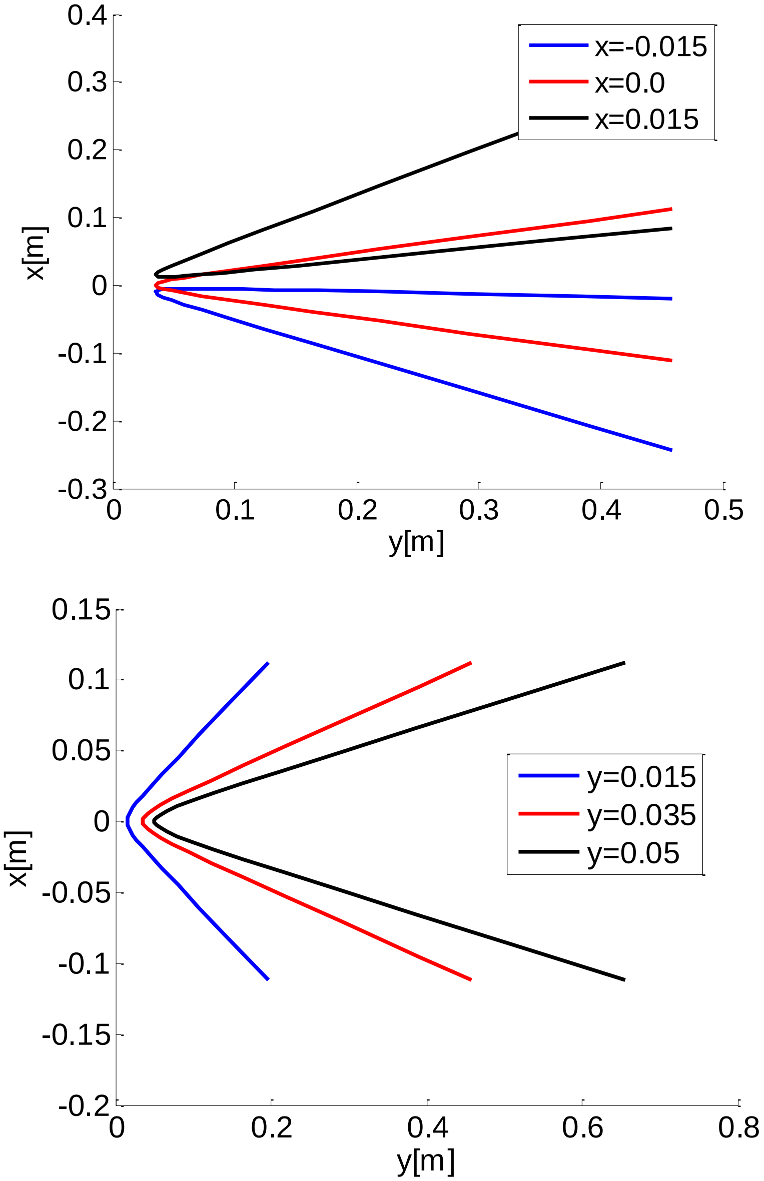

Figure 6-4. Variation in x-position at top dead center alters the asymptotes together, while variation in y-position varies the angle between asymptotes.

In practice it is advisable to regulate the x-velocity instead of the y-position at top dead center to alter the angle between the asymptotes. The y-position has a direct correlation to the time for moving the swing foot forward. The closer the y-position is to zero, the longer the swing period is and vice versa. Given a foot step length and some nominal swing velocity, the y-position is set such that it provides enough time to accelerate and decelerate the leg. Any variation in the angle between the COG trajectory asymptotes should then be achieved using the forward velocity at top dead center.

A series of three steps and four footholds can be seen in Figure 6-5. The figure highlights how multiple steps can be interconnected. Furthermore, foot impact can occur at any point along the curves when the COG is within 0.075[m] of the foothold. Any disruption due to the collisions occurring at foot strike can be accounted for during double support. The benefit of this method over that of the work of Albert is twofold. Firstly, by using the COG instead of the hip position, trajectories can be more easily found. The hip may be behind or ahead of the COG depending on the position of the legs and torso and so the hips trajectory may not intersect with the foothold. By using the COG instead, extra calculations are avoided. Secondly, this method is more suitable for the centroidal dynamic control methods being developed for humanoids, which as the name suggests, focus on the position and velocity of the COG. Again, this method produces trajectories directly for the COG, and thus avoids costly calculations.

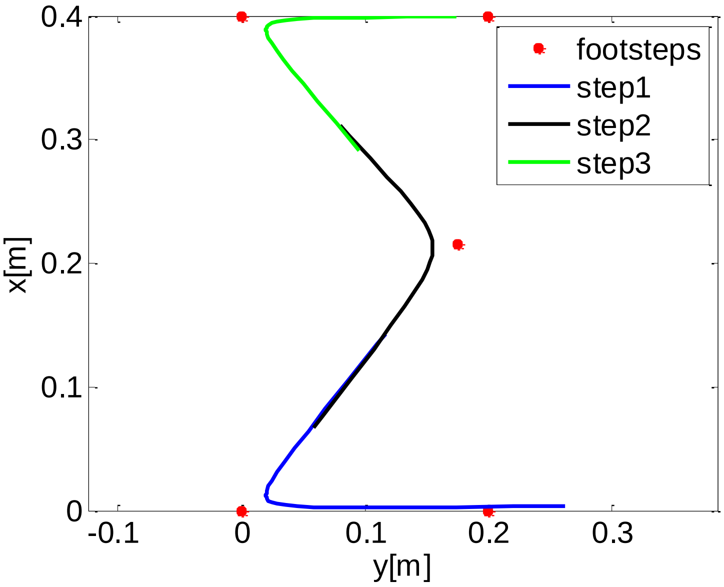

Figure 6-5. Three interconnected steps between four footholds showing the continuity in COG position.

The force profile generated by the swing leg dynamics has a large effect on the trajectory. To best replicate the actual force profile from simulation, a *Sin* function is used. The actual trajectory is of the minimum jerk type, but this can be closely approximated with a *Sin* trajectory. The force F_{xa} is then given as in Equation 6-24:

$$F_{xa} = 0 \; when \; |t| > \left|\frac{t_{ss}}{2}\right| \qquad (6\text{-}24)$$

$$F_{xa} = F_{peak} Sin\left(\frac{\pi t}{t_{ss}}\right) otherwise$$

where F_{peak} is the peak force generated by the leg swing and t_{ss} is the duration of the swing phase. The force is then piecewise continuous, but not differentiable at *t=t_ss/2*.

There are some unique characteristics of the generated trajectory. The COG trajectory in the x-y plane is shown in Figure 6-6. It shows a single forward footstep between three footholds. Two trajectories are shown, one that incorporates the swing leg dynamics and one that does not. An enlarged portion of the trajectory is shown on right that focuses on the area around the support foothold. Near the support foothold, the trajectory differs significantly from the COM trajectory that does not take into account the swing leg dynamics. This is more easily seen in Figure 6-7 which shows the difference in position and velocity from the two trajectories as a function of time. There is a significant velocity difference especially around top dead center. The trajectory incorporating leg swing has a higher velocity, and therefore reaches the final position sooner as seen in Figure 6-7 left.

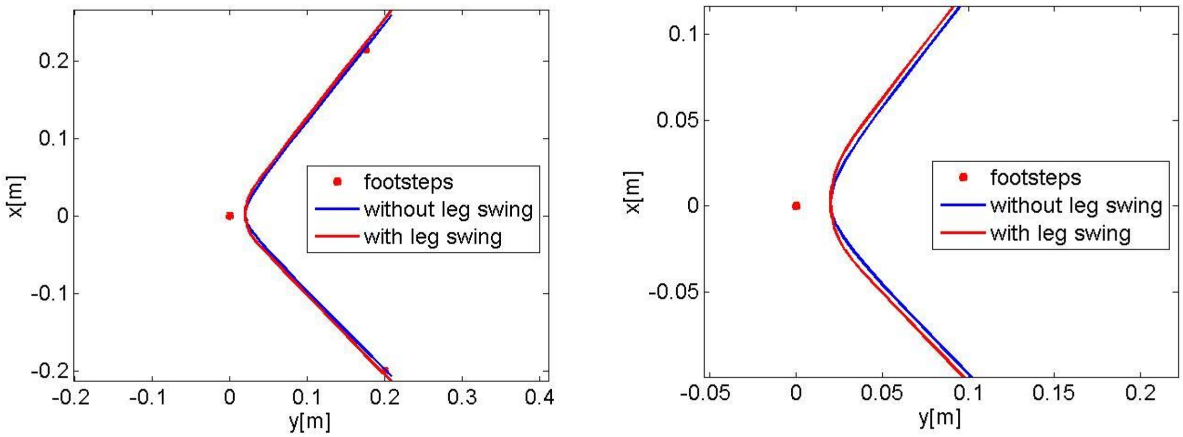

Figure 6-6. X and Y position of the COM using a LIPM with no swing leg action (blue), and with the swing leg(red). Enlarged area near the foothold is shown on right.

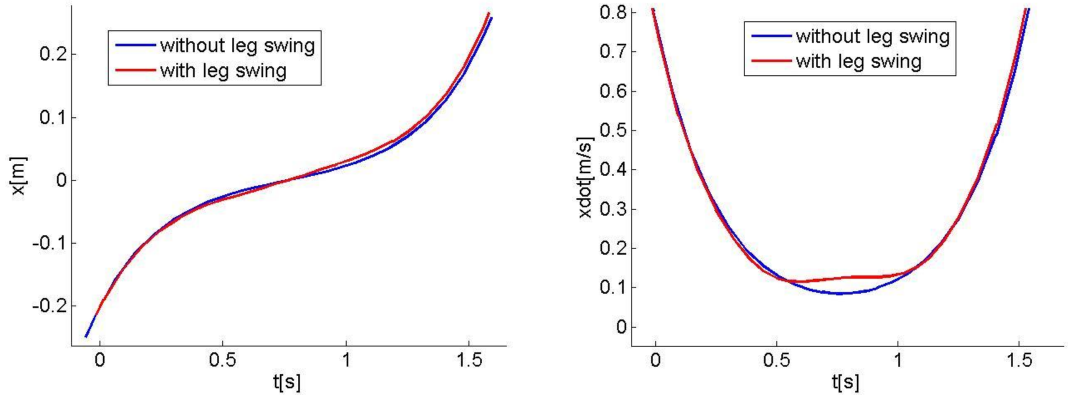

Figure 6-7. The x-position(right) and x-velocity(left) over time of the COG using a LIPM with no swing leg action (blue), and with the swing leg(red).

The phase portraits of both trajectories are shown in Figure 6-8. As expected the velocity most closely matches at the extremes when the models are the same. Another way of thinking about it is that the velocity at x=0[m] is the most different, and the forces generated by the leg swing actually drive the COG back to the nominal trajectory on either side of x=0[m]. This is an interesting conclusion because it shows that the simpler model can actually be used to determine two out of the three parameters necessary to specify the trajectory. This fact is illustrated in Figure 6-9 which shows a similar feature: the location of the top dead center is the same for both trajectories using a sinusoidal force profile. Differences in trajectory are accounted for in only the x-velocity at top dead center.

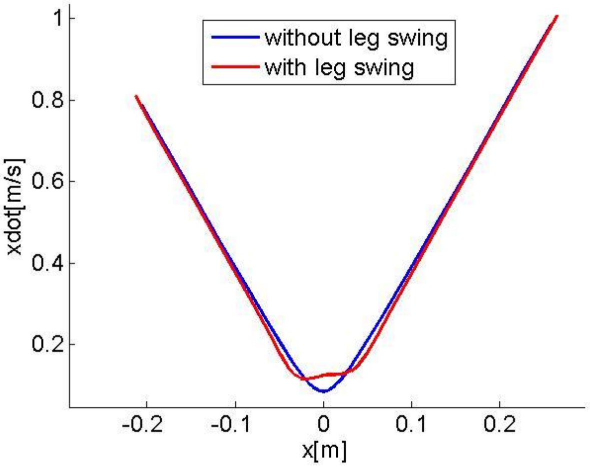

Figure 6-8. Phase portrait of the x position and velocity for one single support step.

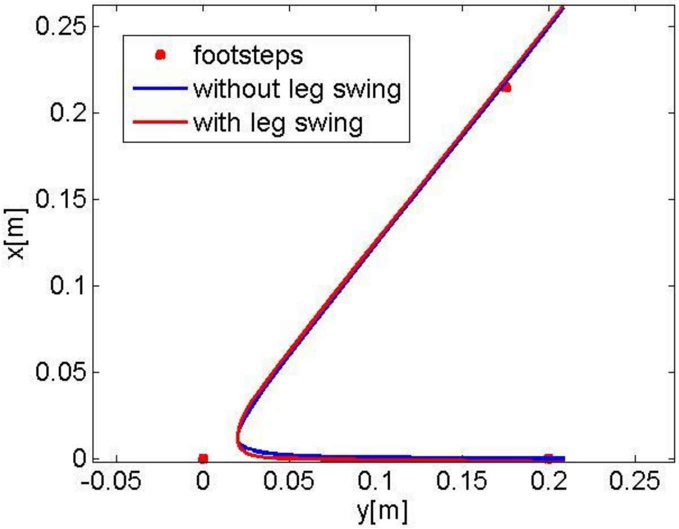

Figure 6-9. Demonstration of a trajectory with different initial footstep.

In the next section the two trajectories will be implemented in Simulation and the control authority required to maintain the trajectory will be measured and compared.

6-5 Simulation

A simulation was run in the modeling environment Gazebo. The robot model has the same degrees of freedom and mass properties as the actual robot. For this test, the robot was given an initial velocity that simulates the initial conditions coming out of double support. Once in single support, the robot moves its swing leg to its new location over a predetermined amount of time. Ankle torques are used at the base of the support leg to drive the actual COG position and velocity to the desired positions and velocities. Two trajectories are tried. The first is the LIPM trajectory. The LIPM trajectory with leg swing dynamics is included second. The two trajectories attempted were that shown in Figure 6-9. The results are shown below in Figure 6-10 through Figure 6-13.

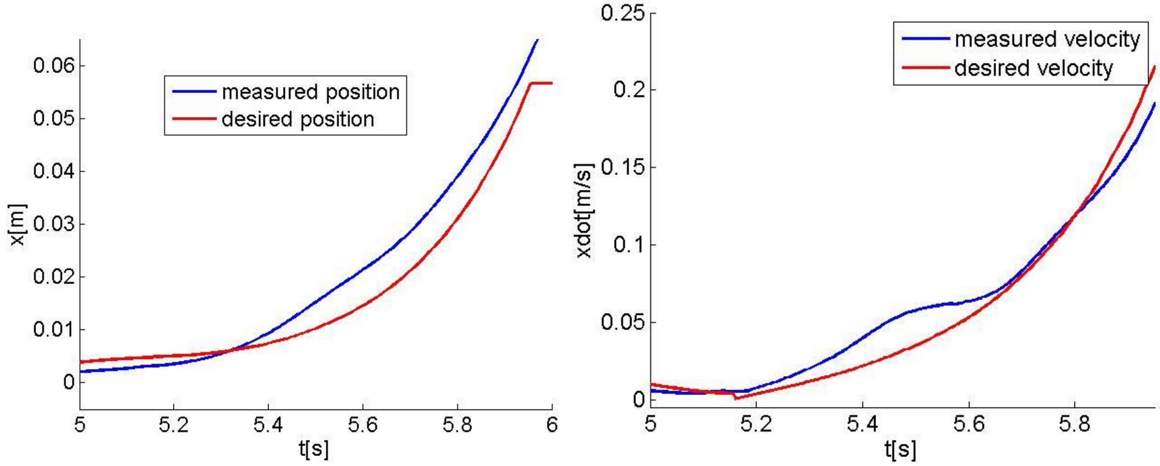

Figure 6-10 Desired and actual position (left) and velocity (right) during simulation using the LIPM trajectory.

Figure 6-10 shows the desired and actual positions and velocities of the COM during single support. As the plots indicate, single support begins at 5[s]. Prior to 5[s], the robot is in double support, where it has more control authority to reach a desirable initial condition for single support. When the initial conditions are met, the robot lifts the swing foot, and begins to accelerate it to the desired foot hold. It is at this point that the robot begins tracking the inverted pendulum trajectory, shown in red in Figure 6-10. Feedback signals are generated about the ankle pitch and roll joints and implemented using torque sources. Both position and velocity feedback is used on the desired trajectory.

As can be seen above, the trajectory following was satisfactory, but by no means perfect. There is a maximum deviation in the desired and actual positions of almost 10[mm]. For a step length of 0.2[m], this is a 5% error of the nominal stride length. Interestingly, the characteristics of the leg swing can still be seen in the final trajectory: the position and velocity rise more quickly than the desired as the swing leg is accelerated. Halfway through the motion, around 5.5[s], the COG begins to slow relative to the desired trajectory as the dynamics of the swing leg cause the COM to decelerate. The maximum error in the velocity is 0.025[m/s].

The trajectory is controlled by a simple linear PID controller that compares the desired position and velocity with the actual position and velocity. The controller can be written as in Equation 6-25.

$$\tau = -k_p(x_{des} - x_{act}) - k_d(\dot{x}_{des} - \dot{x}_{act}) \quad \textbf{(6-25)}$$

where k_p is the proportional gain and k_d is the velocity gain. Torques computed from this feedback law are applied to the ankle joint. By exerting torques about the ankle and thus onto the ground, the robot can control its COM much like a torque at the base of a pendulum controls the mass at the pendulum end. The torques generated by this feedback law can be seen in Figure 6-11. All feedback starts near zero when the robot leaves double support for single support. The peak torques applied to the ground are around 2.5[Nm], fully 10% of the available torque before the foot would roll up on edge.

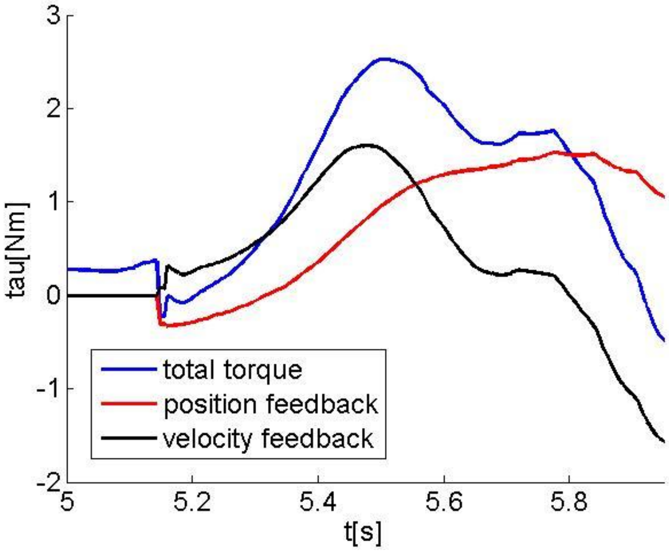

Figure 6-11. Response of the ankle torques required to maintain LIPM trajectory.

Next the same simulation was attempted with the improved LIPM that incorporated the leg swing dynamics. The initial conditions were adjusted such that if left uncontrolled, the inverted pendulum model would pass through the desired foothold. This is most evident in the negative velocity required at the beginning of single support. The results of the test are shown below in Figure 6-12 with single support beginning at 5.15[s]. As compared to above, note the significant improvement in position and velocity tracking. The maximum error in position is only 2[mm] and velocity 0.01[m/s], both roughly 20% of the error when using the traditional LIPM trajectory. The feedback torques for the LIPM with swing leg dynamics in Figure 6-13. The maximum torque required to maintain the improved trajectory during single support peaked around 1[Nm], or roughly 40% of the required torque as compared to the traditional trajectory.

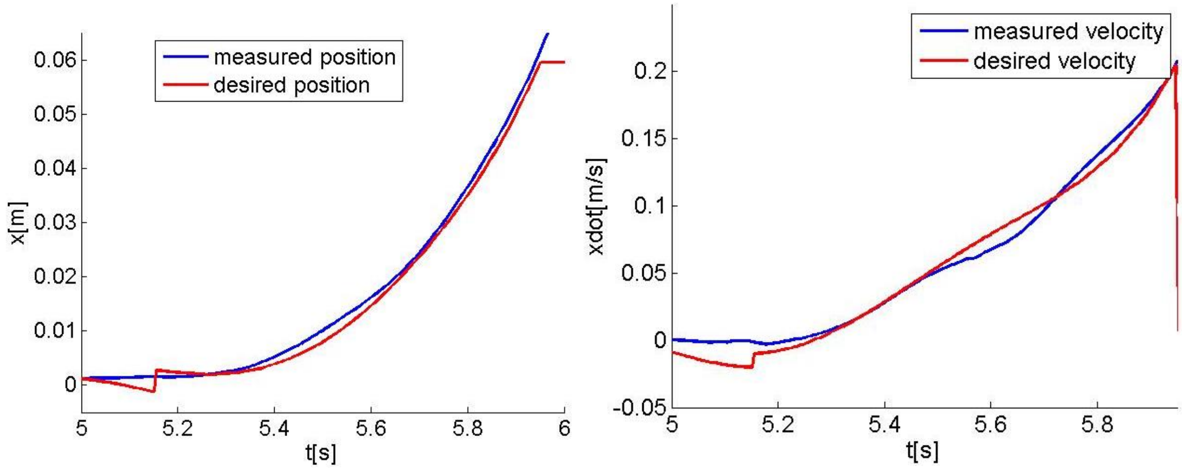

Figure 6-12. Desired and actual position (left) and velocity (right) during simulation using the LIPM trajectory with leg swing dynamics.

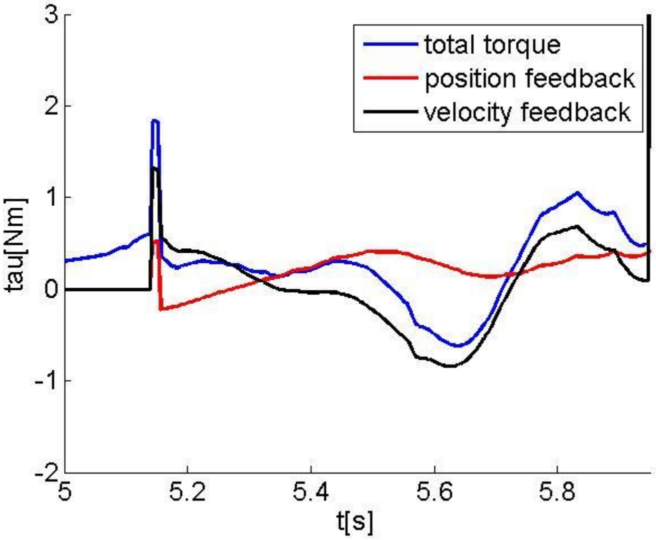

Figure 6-13. Control torques at the ankle required to maintain the improved COG trajectory. Note that much lower torques are required to maintain the trajectory than above, indicating that more torque is available for stabilization.

6-6 Conclusions

In this section a new linear inverted pendulum trajectory was proposed. The trajectory is designed to require a minimum amount of control input to follow in order to provide the maximum amount of torque for disturbance rejection. It was shown that a significant model discrepancy of the traditional LIPM was that it neglected the swing leg dynamics. A model was derived which separated the effects of gravity on the COM and the forces on the COM due to the internal robot dynamics. It was shown that this decomposition is valid for linear systems. The enhanced LIPM model is therefore just as easy to use as the traditional model while being more accurate as well.

A main advantage of this method over others that include the dynamics of multiple bodies is that the position and velocity of the COG are computed directly with this approach. It is therefore very straightforward to generate COG trajectories to interconnect footsteps. A method was demonstrated that parameterized trajectories into only three variables. It was shown that using these variables the swing time, and foot step intercepts could be easily achieved. The enhanced LIPM with swing leg dynamics was applied to a simulation for one step. The results demonstrated improved trajectory tracking with significantly lower control inputs.

The proposed model is useful for humanoids walking on flat ground, but can be easily extended to rough terrain using a number of methods in literature. Furthermore, the parameterization presented offers a useful guide to applying more complex models to the problem of footstep planning.

7 Conclusions

This work presented several research avenues in the field of humanoid robotics. The project was motivated by the potential application of humanoids to the disaster response and rescue robotics areas. Among many capabilities a robotic first responder will need, locomotion is a very important one. The research presented above was thus applied to this problem.

The contributions of this work to the field include: 1) novel actuation architecture to improve the efficiency of bipedal robots, 2) demonstrate how such architecture can be applied to serial or parallel actuated joints, 3) develop a fully electro-mechanical and linearly actuated humanoid robot based off of the theoretical results above, 4) develop an ankle torque based balance controller suitable for compliant and non-stationary terrain, 5) derive a simple but comprehensive dynamic model of the robot's center of gravity for the single support phase that includes swing leg and torso dynamics. The value of this work is that it is not applicable to any one specific robot, but can be used in pieces to improve the performance of any existing or future humanoid through simple hardware or software modifications.

The efficiency of a robot is an important determining factor in its usefulness. In this dissertation, a simple biarticular actuation method is proposed that can reduce the energy consumption of the lower body during common tasks by 30%. This is most easily accomplished by interconnecting the hip roll axis (adduction/abduction) with the knee pitch joint (knee extension and flexion). It was shown that this can be accomplished on both serial and parallel actuated robots and is therefore of great value to the field.

SAFFiR is a lower body platform built to demonstrate advanced humanoid robot technology. Its primary contribution was in demonstrating the suitability and limits of ballscrew equipped linear actuators as the primary transmission mechanism. A force control based balance controller was developed and verified through extensive experimentation. This controller is particularly valuable in that it utilizes only the ankle as pure force sources, and is therefore easily applied to existing robots (for example using the force torques in the ankle as feedback or by simply replacing the two ankle actuators). SAFFiR can walk on high-pile turf, gravel, sand and strewn plywood. It is believed to be the first robot to demonstrate bipedal locomotion on loose sand (in front of a live audience no less).

Besides hardware and controls, trajectory planning is the third component of bipedal locomotion. In this work, the linear inverted pendulum model was extended to include internal dynamics that were previously neglected. The model retains the simplicity of a lumped COM and thus is very appropriate for the state of the art centroidal momentum based controllers. The improved model was implemented in simulation and shown to reduce control input and tracking error.

Bibliography

[1] B.-K. Cho, I.-W. Park, and J.-H. Oh, “Running Pattern Generation of Humanoid Biped With a Fixed Point and Its Realization,” *International Journal of Humanoid Robotics*, vol. 06, no. 04, pp. 631–656, Dec. 2009.

[2] J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, “Realization of Fast Walking with High-Power Biped Robot,” in *IEEE International Conference on Intelligent Robots and Systems*, 2011.

[3] K. Hashimoto, Y. Sugahara, A. Ohta, H. Sunazuka, C. Tanaka, M. Kawase, and A. Takanishi, “Realization of Stable Biped Walking on Public Road with New Biped Foot System Adaptable to Uneven Terrain,” in *IEEE International Conference on Biomedical Robotics and Biomechatronics*, 2006, pp. 226–231.

[4] J.-H. Kim and J.-H. Oh, “Realization of dynamic walking for the humanoid robot platform KHR-1,” *Advanced Robotics*, vol. 18, no. 7, pp. 749–768, Jan. 2004.

[5] J. Kim, I. Park, and J. Oh, “Experimental realization of dynamic walking of the biped KHR-2 using ZMP feedback,” *Advanced Robotics*, vol. 00, no. 0, pp. 1–30, 2006.

[6] B.-K. Cho, S.-S. Park, and J. Oh, “Controllers for running in the humanoid robot, HUBO,” *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pp. 385–390, Dec. 2009.

[7] D. Stonier, “Modifiable Walking Pattern of a Humanoid Robot by Using Allowable ZMP Variation,” *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 917–925, Aug. 2008.

[8] J. Park, “General ZMP Preview Control for Bipedal Walking,” in *IEEE International Conference on Robotics and Automation*, 2007, no. April, pp. 10–14.

[9] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, and N. Higaki, “The intelligent ASIMO: System Overview and Integration,” in *IROS*, 2002, pp. 2478–2483.

[10] H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, and A. Takanishi, “Development of a new humanoid robot WABIAN-2,” in *IEEE International Conference on Robotics and Automation*, 2006, no. May, pp. 76–81.

[11] S. Lohmeier, T. Buschmann, and H. Ulbrich, “Humanoid robot LOLA,” in *IEEE International Conference on Robotics and Automation*, 2009, pp. 775–780.

[12] S. Lohmeier, T. Buschmann, H. Ulbrich, and F. Pfeiffer, “Modular joint design for performance enhanced humanoid robot LOLA,” in *IEEE International Conference on Robotics and Automation*, 2006, no. May, pp. 88–93.

[13] T. Buschmann, S. Lohmeier, H. Ulbrich, and F. Pfeiffer, “Dynamics simulation for a biped robot: modeling and experimental verification,” *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, no. May, pp. 2673–2678, 2006.

[14] M. Gienger, K. Loffler, and F. Pfeiffer, “Towards the design of a biped jogging robot,” in *International Conference on Robotics and Automation*, 2001, vol. 4, pp. 4140–4145.

[15] K. Löffler, M. Gienger, F. Pfeiffer, and H. Ulbrich, “Sensors and Control Concept of a Biped Robot,” *IEEE Transactions on Industrial Electronics*, vol. 51, no. 5, pp. 972–980, 2004.

[16] A. D. Kuo, “Choosing Your Steps Carefully,” *IEEE Robotics & Automation Magazine*, vol. 1, no. June, pp. 18–29, Jun-2007.

[17] T. McGeer, “Dynamics and Control of Bipedal Locomotion,” *J. Theor. Biol.*, no. 163, pp. 277–314, 1993.

[18] M. Srinivasan and A. Ruina, “Computer optimization of a minimal biped model discovers walking and running.,” *Nature*, vol. 439, no. 7072, pp. 72–5, Jan. 2006.

[19] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, “Efficient bipedal robots based on passive-dynamic walkers.,” *Science*, vol. 307, no. 5712, pp. 1082–5, Feb. 2005.

[20] A. Ruina, J. E. a Bertram, and M. Srinivasan, “A collisional model of the energetic cost of support work qualitatively explains leg sequencing in walking and galloping, pseudo-elastic leg behavior in running and the walk-to-run transition.,” *Journal of Theoretical Biology*, vol. 237, no. 2, pp. 170–92, Nov. 2005.

[21] M. Strasberg and W. Seidel, “Mechanical Hip Actuation Of A 2-D Passive-Dynamics Based Walker,” 2005.

[22] C. G. Atkeson, “Learning Biped Locomotion,” *IEEE Robotics & Automation Magazine*, no. June, pp. 41–51, Jun-2007.

[23] D. Hobbelen, “Limit Cycle Walking,” Delft University, 2008.

[24] J. Pratt and B. Krupp, “Design of a bipedal walking robot .,” in *SPIE Defense and Securtity*, 2008.

[25] J. E. Pratt, B. Krupp, V. Ragusila, and J. Rebula, “The Yobotics-IHMC Lower Body Humanoid Robot,” in *International Conference on Intelligent Robots and Systems*, 2009, pp. 410–411.

[26] Karssen and J.G., “Design and construction of the Cornell Ranger , a world record distance walking robot .,” Ithaca NY, 2007.

[27] a. V. Voronov, “The Roles of Monoarticular and Biarticular Muscles of the Lower Limbs in Terrestrial Locomotion,” *Human Physiology*, vol. 30, no. 4, pp. 476–484, Jul. 2004.

[28] F. Iida, J. Rummel, and A. Seyfarth, “Bipedal walking and running with spring-like biarticular muscles.,” *Journal of Biomechanics*, vol. 41, no. 3, pp. 656–67, Jan. 2008.

[29] S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee, “Learning full body push recovery control for small humanoid robots,” in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2047–2052.

[30] J.-K. Han, “Bipedal Walking for a full size humanoid robot utilizing feet trajectories and its energy consumption,” Virginia Tech, 2012.

[31] I. Park, J. Kim, S. Park, and J. Oh, “Development of Humanoid Robot Platform KHR-2,” *Journal of Humanoid Robotics*, vol. 2, no. 4, 2005.

[32] I.-W. Park, J.-Y. Kim, J. Lee, and J.-H. Oh, “Mechanical design of the humanoid robot platform, HUBO,” *Advanced Robotics*, vol. 21, no. 11, pp. 1305–1322, Jan. 2007.

[33] Y. Ogura, H. Aikawa, H.-O. Lim, and a. Takanishi, “Development of a human-like walking robot having two 7-DOF legs and a 2-DOF waist,” *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004,* pp. 134–139 Vol.1, 2004.

[34] L. Chaitow and J. Walker DeLany, *Clinical Application of Neuromuscular Techniques: The Lower Body*. Elsevier Health Sciences, 2000.

[35] T. Oshima, K. Toriumi, T. Fujikawa, and N. Momose, “Effects of the Lower Leg Bi-Articular Muscle in Jumping,” *Journal of Robotics and Mechantronics,* vol. 16, no. 6, 2004.

[36] V. Salvucci, Y. Kimura, S. Oh, and Y. Hori, “BiWi : Bi-Articularly Actuated and Wire Driven,” in *IEEE International Conference on Mechatronics*, 2011, pp. 827–832.

[37] F. Gosselin and J.-P. Lallemand, “A new insight into the duality between serial and parallel non-redundant and redundant manipulators,” *Robotica*, vol. 19, no. 04, pp. 365–370, Jul. 2001.

[38] “Maxon Motor Data and Operating Ranges,” 2010. [Online]. Available: www.maxonmotorusa.com. [Accessed: 03-Jun-2014].

[39] “Maxon Motor Catalogue,” 2013. [Online]. Available: www.maxonmotorusa.com. [Accessed: 03-Jun-2014].

[40] G. Alici and B. Shirinzadeh, “Topology optimisation and singularity analysis of a 3-SPS parallel manipulator with a passive constraining spherical joint,” *Mechanism and Machine Theory,* vol. 39, no. 2, pp. 215–235, Feb. 2004.

[41] B. Dasgupta and T. S. Mruthyunjaya, “Force redundancy in parallel manipulators: theoretical and practical issues,” *Mechanism and Machine Theory,* vol. 33, no. 6, pp. 727–742, Aug. 1998.

[42] Y. Fang and L.-W. Tsai, “Structure Synthesis of a Class of 3-DOF Rotational Parallel Manipulators,” *IEEE Transactions on Robotics and Automation,* vol. 20, no. 1, pp. 117–121, 2004.

[43] C. M. Gosselin and E. Lavoie, “On the Kinematic Design of Spherical Three-Degree-of- Freedom Parallel Manipulators,” *The International Journal of Robotics Research,* vol. 12, no. 4, pp. 394–402, Aug. 1993.

[44] T. K. Tanev, “Kinematics of a hybrid (parallel–serial) robot manipulator,” *Mechanism and Machine Theory,* vol. 35, no. 9, pp. 1183–1196, Sep. 2000.

[45] K. J. Waldron and K. H. Hunt, “Series-Parallel Dualities in Actively Coordinated Mechanisms,” *The International Journal of Robotics Research,* vol. 10, no. 5, pp. 473–480, Oct. 1991.

[46] M. M. Pedersen, A. A. Nielsen, and L. F. Christiansen, “Design of Biped Robot AAU-BOT1,” Aalborg University, 2007.

[47] H. D. Taghirad, “Robust Torque Control of Harmonic Drive Systems,” McGill University, 1997.

[48] THK, “Low Price Actuators.” THK Industries Specification Sheet, 2008.

[49] B. R. So, “Design of a Redundantly Actuated Hip Mechanism,” in *International Conference on Robotics and Automation,* 2006, vol. 49, no. 1, pp. 253–259.

[50] J. E. Pratt, “Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots,” MIT, 2000.

[51] J. E. Pratt and G. A. Pratt, “Exploiting Natural Dynamics in the Control of a 3D Bipedal Walking Simulation,” in *International Conference on Climbing and Walking Robots,* 1999, no. September.

[52] M.-S. Kim and J. H. Oh, “Posture Control of a Humanoid Robot With a Compliant Ankle Joint,” *International Journal of Humanoid Robotics,* vol. 07, no. 01, pp. 5–29, Mar. 2010.

[53] J. E. Pratt and B. T. Krupp, “Series Elastic Actuators for legged robots,” in *Defense and Security, International Society of Optics and Photonics*, 2004, pp. 135–144.

[54] G. A. Pratt and M. Williamson, “Series Elastic Actuators,” *IEEE Control Systems Magazine*, pp. 399–406, 1997.

[55] G. a. Pratt, P. Willison, C. Bolton, and A. Hofman, “Late Motor Processing in Low-Impedance Robots: Impedance Control of Series Elastic Actuators,” in *American Control Conference*, 2004, pp. 3245–3251.

[56] B. Lee, D. Lahr, V. Orekhov, and D. Hong, “Design and Measurement error Analysis of a Low Friction, Lightweight Linear Series Elastic Actuator,” in *ASME Design Engineering Technical Conferences*, 2013.

[57] V. Orekhov, D. Lahr, B. Lee, and D. Hong, “Design for Distributed Compliance in Humanoid Robots,” in *ASME International Design Engineering Technical Conferences*, 2013.

[58] M. Garcia, a Chatterjee, a Ruina, and M. Coleman, “The simplest walking model: stability, complexity, and scaling.,” *Journal of biomechanical engineering*, vol. 120, no. 2, pp. 281–8, Apr. 1998.

[59] A. Goswami, B. Espiau, and A. Keramane, “Limit Cycles in a Passive Compass Gait Biped and Passivity-Mimicking Control Laws,” *Autonomous Robots*, vol. 4, pp. 273–286, 1997.

[60] M. W. Spong, “Passivity Based Control of the Compass Gait Biped,” in *Proc. of IFAC World Congress*, 1999.

[61] A. D. Kuo, “Energetics of Actively Powered Locomotion Using the Simplest Walking Model,” *Journal of Biomechanical Engineering*, vol. 124, no. 1, p. 113, Sep. 2001.

[62] A. D. Kuo, J. M. Donelan, and A. Ruina, “Energetic consequences of walking like an inverted pendulum: step-to-step transitions.,” *Exercise and sport sciences reviews*, vol. 33, no. 2, pp. 88–97, Apr. 2005.

[63] M. Wisse, a. L. Schwab, R. Q. van der Linde, and F. C. T. van der Helm, “How to keep from falling forward: elementary swing leg action for passive dynamic walkers,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 393–401, Jun. 2005.

[64] M. W. Spong and D. J. Block, *Reaction Wheel Pendulum Control*. Morgan and Claypool, 2007.

[65] J. Pratt1, 2 Twan Koolen1, T. de Boer2, J. Rebula3, S. Cotton1, J. Carff1, M. Johnson1, and P. Neuhaus1, “Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid,” *International Journal of Robotics Research*, vol. 31, no. 10, 2012.

[66] K. . Kyriakopoulos and G. N. Saridis, “Minimum jerk path generation,” in *IEEE International Conference on Robotics and Automation*, 1988, pp. 364–369.

[67] M. Morisawa, S. Kajita, K. Kaneko, and K. Harada, “Pattern Generation of Biped Walking Constrained on Parametric Surface,” in *International Conference on Robotics and Automation*, 2005, no. April, pp. 5–10.

[68] M. Posa and R. Tedrake, “Direct Trajectory Optimization of Rigid Body Dynamical Systems through Contact,” in *Algorithmic Foundations of Robotics X*, vol. 86, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 527–542.

[69] T. Erez and E. Todorov, “Trajectory optimization for domains with contacts using inverse dynamics,” in *International Conference on Intelligent Robots and Systems*, 2012, pp. 4914–4919.

[70] L. Yang, C. M. Chew, and A. N. Poo, “Adjustable Bipedal Gait Generation using Genetic,” in *IEEE Intelligent Robots and Systems*, 2006, pp. 4435–4440.

[71] A. Albert and W. Gerth, “Analytic Path Planning Algorithms for Bipedal Robots without a Trunk,” *Journal of Intelligent and Robotic Systems*, vol. 36, no. 1994, pp. 109–127, 2003.

[72] S. Kajita, F. Kanehiro, K. Kando, K. Yokoi, and H. Hirukawa, “The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation,” in *IEEE International Conference on Intelligent Robots and Systems*, 2001, pp. 239–246.

[73] N. Motoi, T. Suzuki, and K. Ohnishi, “Linear Inverted Pendulum Mode,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 1, pp. 54–61, 2009.

[74] Z. Tang and M. J. Er, “Humanoid 3D Gait Generation Based on Inverted Pendulum Model,” in *International Symposium on Intelligent Control*, 2007, no. October, pp. 1–3.

[75] S. Kajita and K. Tani, “Study of Dynamic Biped Locomotion on Rugged Terrain,” in *Robotics and Automation*, 1991, pp. 741–746.

[76] J. Park and K. Kim, “Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control,” in *International Conference on Robotics and Automation*, 1998, pp. 3528–3533.

[77] W. M. Szydlowski and C. A. Nelson, “Synthesis of adaptive pliers mechanism using genetic algorithms,” in *ASME Design Engineering Technical Conferences*, 2007.

[78] D. E. Goldbert, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[79] F. Schillebeeckx and J. Peirs, “Compact Zero-Backlash Tilt-Pan Mechanism Based on Differential Gear Technology,” in *International Conference on New Actuators*, 2004.

[80] K. . Hunt, “Constant Velocity Shaft Couplings: A General Theory,” *Journal of Engineering for Industry*, vol. 95, pp. 455–464, 1973.

[81] F. . Myard, “Homokinetic Transmission Joint,” U.S. Patent US20337331936.

[82] P. Fenaille, “Universal Coupling for Machine,” U.S. Patent US19217371933.

[83] M. Clemens, “Improvement in Apparatus for Transmitting Rotary Motion,” U.S. Patent US963951869.

[84] M. Clemens, “Improvement in Universal-Joint Couplings for Connecting Shafts,” U.S. Patent US1258801869.

[85] M. [Clemens, “Improvement in Universal-Joint Couplings for Connecting Shafts,” U.S. Patent US1258811869.

[86] D. M. Wallace and F. Freudenstein, “Displacement Analysis of the Generalized Clemens Coupling , The R-R-S-R-R Spatial Linkage l,” *Journal of Engineering for Industry*, vol. 95, no. 2, pp. 575–580, 1975.

[87] P. Milenkovic, “Nonsingular Spherically Constrained Clemens Linkage Wrist,” *Journal of Mechanisms and Robotics*, vol. 3, 2011.

[88] P. Milenkovic, “Triangle Pseudocongruence in Constraint Singularity of Constant-Velocity Couplings,” *Journal of Mechanisms and Robotics*, vol. 1, no. 2, p. 021006, 2009.

[89] “Petman.” [Online]. Available: http://www.bostondynamics.com/robot_petman.html.

[90] "Meka M1 A Dexterous Mobile Manipulator." [Online]. Available: http://mekabot.com/products/m1-mobile-manipulator/.

[91] G. A. Robinson, D.W., Pratt, J.E., Paluska, D.J., Pratt, "Series Elastic Actuator Development for a Biomimetic Walking Robot," in *Proc. Of IEEE Intl. Conf. on Adv. Intelligent Mechatronics*, 1999.

[92] "DRC Task descriptions." [Online]. Available: http://www.theroboticschallenge.org/local/documents/DRC Trials Initial Task Descriptions DISTAR Case 21473.pdf.

[93] S.-H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Autonomous Robots*, vol. 33, no. 4, pp. 399–414, 2012.

[94] D. E. Orin and A. Goswami, "Centroidal Momentum Matrix of a humanoid robot: Structure and properties," in *International Conference on Intelligent Robots and Systems*, 2008, pp. 653–659.

[95] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2–3, pp. 161–176, Jun. 2013.

[96] A. Goswami, "Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground," in *International Conference on Intelligent Robots and Systems*, 2010, pp. 3157–3162.

[97] A. Goswami and V. Kallem, "Rate of change of angular momentum and balance maintenance of biped robots," in *International Conference on Robotics and Automation*, 2004, pp. 3785–3790.

[98] Y. Choi, D. Kim, Y. Oh, and B. You, "Posture / Walking Control for Humanoid Robot Based on Kinematic Resolution of CoM Jacobian," vol. 23, no. 6, pp. 1285–1293, 2007.

[99] J. H. Choi, "Planar Bipedal Walking with Foot Rotation," in *Proceedings of the American Control Conference*, 2005, pp. 4909–4917.

[100] K. Nishiwaki, "Toe Joints that Enhance Bipedal and Fullbody Motion of Humanoid Robots," in *IEEE International Conference on Robotics and Automation*, 2002, pp. 3105–3111.

[101] H. Takemura, "Study of the Toes Role in Human Walk by Toe Elimination and Pressure Measurement Sytem," in *IEEE International Conference on Systems, Man and Cybernetic*, 2003, pp. 2569–2575.

Appendix A Genetic Algorithm Coding

Genetic algorithms have in recent years been used for a wide variety of type and dimensional synthesis and optimization problems. They are unique among search algorithms in that they combine elements of both random and directed search routines. The genetic algorithm described here is applied to the joint orientation optimization problem. Genetic algorithms use the breeding and evolutionary mechanisms of natural systems to root out an optimal solution, where the optimal is determined by a user defined fitness function. The strength of these algorithms stems from the combination of elements from both random walk type searches and broader and more computationally intensive enumerative type algorithms. The result is a rigorous search which can be used for many types of optimizations of multi-criteria problems [77].

The algorithm used here is applied to a dimensional type synthesis problem. In this approach, the various parameters of significance, (elements of the matrix G) area coded into a series of bits which make up a chromosome. A large population is then built up of many different chromosomes. Each individual in the population is then scored for a fitness level using the fitness function. Members of the population are then combined and crossed using a mating function which gives priority to the individuals with higher fitness. Mutations are also allowed to occur in every generation. With each new generation, the population is rescored by the fitness function, and the cycle continues. The result is a powerful search algorithm built upon very basic and easy to program functions. A flowchart representing the structure of the genetic algorithm can be seen in Figure A-1.

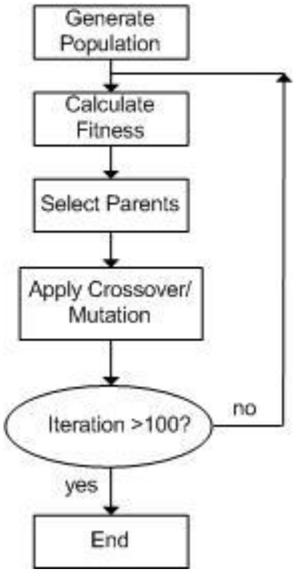

Figure A-1. Flowchart of the Genetic Algorithm process.

The coding of the population chromosomes is done as a binary representation. Each chromosome is made up of one or more genes, where each gene is a coding representing a transmission parameter. For this transmission, there are five different parameters which are varied which are summarized in Table A-1.

Table A-1: Parameter bounds for the numerical search and genetic algorithm.

Parameter	Symbol	Lower bound	Upper bound	Units
Lever length	r_1, r_2	0	0.1	Meters
Lever angle	θ_1, θ_2	0	π	Radians
Biarticular lever	p_1- p_4	0	0.05	Meters

Each gene's parameter is transformed into a five bit number. The parameter value is determined by the user defined limits. Each value of the gene, a 0 or a 1, is randomly defined by a random number generator with equal probability given to each outcome. The region defined by the upper and lower limits of each parameter value are divided into 31 steps, (2^5-1), where each step is represented by an six bit binary number, the gene. For example, if for some parameter value the upper limit was 10, the lower limit was 5, and the gene was randomly generated as 01101, then the actual parameter value of that gene would be given as in Equation A-1:

$$value = 5 + \frac{(10-5)}{2^5-1} * 01101_2 \qquad \textit{(A-1)}$$

where the subscript "2" indicates that it is a binary number. It is worth mentioning that the program deals strictly with the coded parameter value, only decoding the gene when it is necessary to use this value in the fitness scoring function. It is therefore easy to increase the resolution without greatly affecting the program runtime.

The upper and lower limits for each parameter were chosen based on experience from previous work with the transmission. This work showed a general range of acceptable values which would produce a viable transmission. The limits could be adjusted from run to run if it was found the algorithm chose the extremes more often than not. For the five parameters with a gene length of 6 bits, each member of the population has a chromosome length of 30. The first operation then performed by the program is to initialize the population. For this optimization, we used a population size of 100.

A - 1 Fitness Scoring

Once the population has been established, it is necessary to assign a fitness value to each of the members of the population. The fitness values will then be used by the mating functions to determine the fittest specimens and give them a higher probability of mating. For the optimization, the scoring function was based first on reducing the contact stress, and secondly on maintaining the transmission limits described above.

Once the parameters were decoded, they are sent to the scoring function. The energy lost by the leg is then calculated using Equation 2-7

In application, the lowest fitness value was 0.01 in order to allow even the poorest performing individuals to be allowed to reproduce, in case there was some valuable genetic information within the chromosome. The fitness value and the contact stress are related by Equation :

$$f(c) = \frac{-E}{350} + 1 \; when \; E \leq 350 \; and \; f(c) = 0.01 \; when \; E > 350 \qquad \textbf{\textit{(A-2)}}$$

where *E* is the energy lost by the motors. This function was chosen such that a desirable level of energy, 350[J], represented a unity fitness value.

It is necessary for the fitness function to also enforce the parameter limits. This is done by reducing the fitness value of the offending individual, either by a fixed value, or a percentage reduction. In this paper, we use a fixed value of 0.2 for each offense. Should there be too many individuals outside the limits, this value can be increased so as to inhibit reproduction further.

A - 2 Fitness Scaling

Fitness scaling is an important part of the genetic algorithm. Its purpose is to modify the fitness values of the population to prevent the domination of reproduction by a few individuals in the early generations, and also to increase competition among the population in later generations. A linear scaling routine was implemented here. Given an original fitness value, the scaled value can be found from Equation A-3:

$$f' = a \cdot f + b \qquad \textbf{\textit{(A-3)}}$$

where *a* and *b* are chosen such that the maximum original fitness is mapped to the twice the original average fitness but the original and scaled average fitness values remain the same. A graphical representation of this mapping can be seen in Figure A-2.

In this way the fittest individual will reproduce on average twice as often as the average individual. Thus a healthy competition is maintained in both the early and later generations. In order to prevent negative fitness values, which may occur when using a scaling function, any scaled fitness value less than zero is mapped to a value of 0.1.

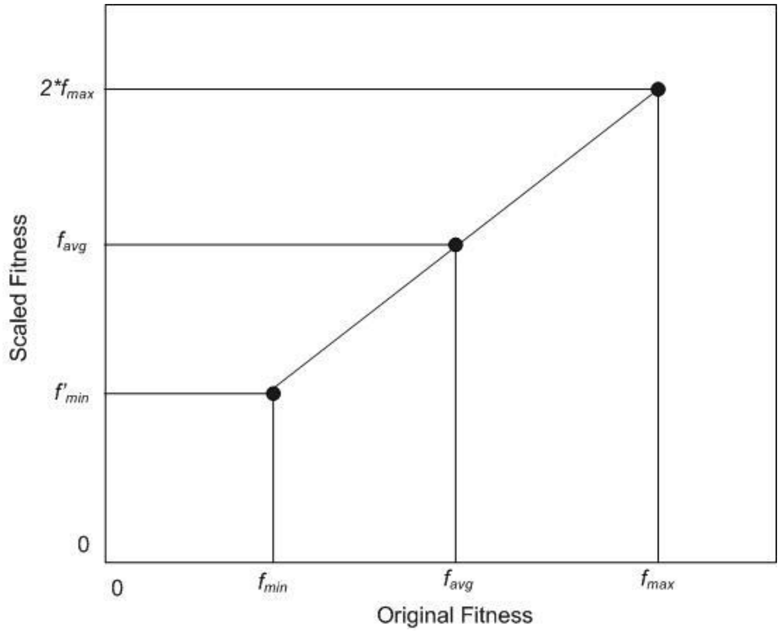

Figure A-2. Fitness scaling used to promote healthy competition.

A - 3 Selection and Mutation rate

Between each generation it is necessary to mate the population. A selection function chooses two members of the population for crossover. The probability that any member for reproduction is proportional to their fitness value. This routine is generally referred to as a "roulette wheel" style selection [78]. For every two selected members of the population, two offspring are generated. If the same individual is selected twice, it can be beneficial to cancel the mate and send the member back to the population through the mutation algorithm. It has been shown that this diversity monitor increases the number of acceptable solutions at the end of the program without lowering the maximum fitness level.

For every pair of mating individuals, a random number between one and the chromosome length is chosen. This is the crossover site. For reproduction, the two chromosomes are split at the crossover site, and the latter portions of the chromosomes are swapped between individuals. Put another way, each bit of the chromosome that occurs after the crossover site are swapped between the individuals.

Before the offspring are sent back to the population, they are passed through a mutation function. The purpose of this routine is to switch the value of one randomly selected bit within the chromosome. As implemented here, this mutation occurs with probability of 0.033. There is equal probability that any bit on the chromosome will be selected for mutation.

A - 4 Stop Criteria

There is one stop criteria built into the genetic algorithm. This stop limits the maximum number of generations that can be produced, essentially a run-time limit on the program. No more than 300 generations were produced during this study. In addition, the member with the highest fitness throughout all generations is stored such that it can be retrieved when the program terminates. The can be seen in the figure below:

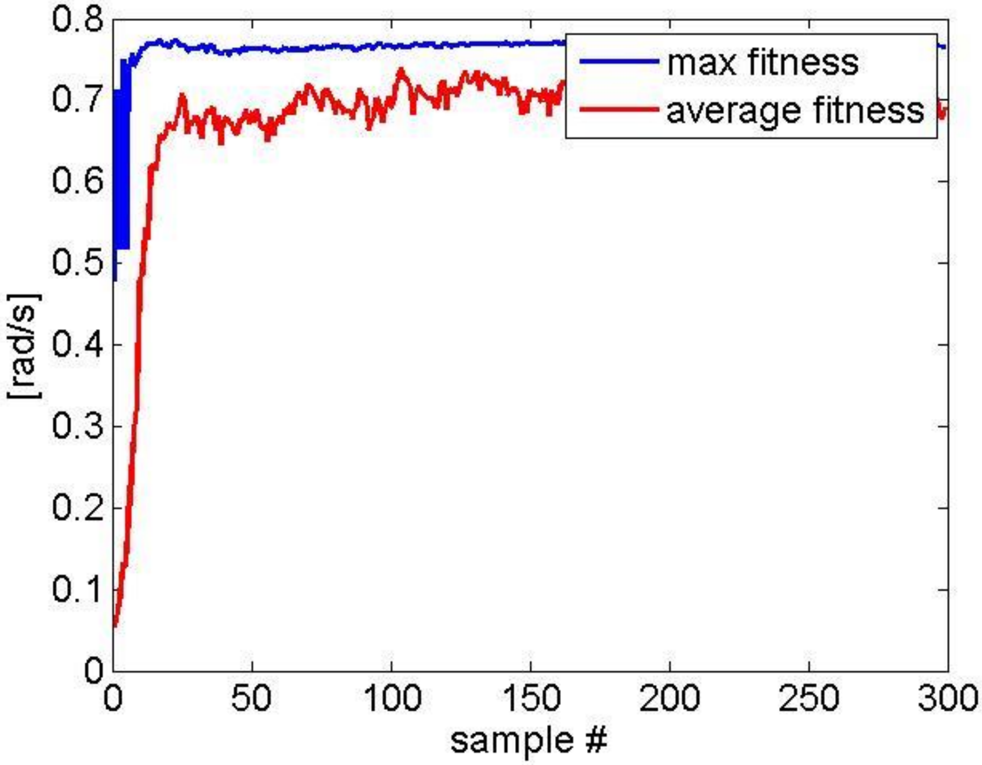

Figure A-3. Maximum and average fitness values over the course of 300 generations. The average tends to converge, indicating a near optimum is found.

Appendix B Clemens type constant velocity linkage

Robotic manipulators can be categorized as either parallel, serial, or in some cases a combination of the two. Among others, a notable drawback of serial manipulators in dynamic applications is the large inertia created by typically heavy electromechanical actuators at the distal end of the manipulator. In addition, compact packaging of multiple actuators in a multi-degree of freedom (DOF) joint, as is often necessary with serial manipulators, can be difficult. These difficulties can be alleviated should a means be found to relocate actuators across one or more degrees of freedom. In this paper, we investigate a constant velocity (CV) linkage, the Clemens linkage, that may be used to relocate an actuator across a one DOF revolute joint to an adjacent link while maintaining a serially actuated architecture. This can be very advantageous in some applications such as a humanoid robot ankle. The linkage is analyzed for both its range of motion and torque capacity for such applications given limitations of currently available bearing hardware.

B - 1 Introduction

There are two primary robotic hardware architectures from which a designer must select before embarking on a new design, those of parallel and serial architectures. Serial manipulators and robots are those in which each limb is comprised of a series of alternating rigid links and actuated joints. A parallel architecture is that defined by a number of serial chains arranged in parallel to control an end effector. Traditionally, serial architectures have been used with great success in industrial and research robots due to their large range of motion and straightforward design. However, they are not without disadvantages, which include increased inertias due to the necessary distal location of actuators, increased backlash, and increased positional error due to the compounding effect of the design. In this paper, we look at a constant velocity mechanism, the Clemens Linkage, which allows actuators to be located in more favorable locations within a serial architecture design to improve upon the inertial drawbacks of serial manipulators.

Humanoid robots have been traditionally designed with each limb acting as a serial manipulator in large part due to the required range of motion typical of most humanoid joints. Examples of this include KAIST University's HUBO, Honda's ASIMO, Waseda University's Wabian-2, and Technical University of Munich's Johnnie and LOLA among others[10], [31], [32]. As an example of the range of motion, the two degree of freedom ankle joint of HUBO, has 135 degrees of motion in pitch, and 90 in roll[31]. With the proper mechanical design, this range of motion is very feasible with a serial chain. However, it requires a large ~600[g] actuator to be placed on a cross gimbal centered in the ankle joint.

More generally, a typical humanoid robot ankle joins three rigid bodies, the shin, cross gimbal, and foot. The ankle has two DOFs, usually a roll and pitch (both oriented in the aeronautical sense). Up until now, serially actuated robots have utilized one of four actuator placements: 1) one actuator in the shin, and one in the foot, 2) in the shin and gimbal, 3) in the gimbal and foot, and 4) both in the gimbal. All four configurations incur a significant inertial penalty from at least one of the actuators being located at the extreme distal end of the limb. This causes both a significant increase in the legs inertia as well as presents a packaging challenge to designers. To be able to remotely position this actuator (for example near the proximal end of the shin) can therefore have a significant effect on the design and dynamics of the robot.

There are a number of ways to accomplish the relocation of actuators across DOFs. The humanoid LOLA has a very interesting system of pinion gears in the knee that drive a parallel system of linear transmissions[12]. Another alternative is a differential style gearbox that drive both ankle DOFs in a parallel fashion. Differential joints have been used before [79] but require a larger range of motion from each actuator and additional complexity in the control software due to the necessity for coordinated motion between actuators. We have investigated linear actuators in a parallel arrangement on the humanoid SAFFiR, but found compromises still arise [Lahr].

On the other hand, a constant velocity linkage placed within the joint adds little complexity, allows for the actuator to be repositioned, and maintains a serial architecture for software simplicity. To illustrate this point, take the example of the HUBO ankle mentioned above. A CV joint allows the actuator on the gimbal to be moved to the shin, and instead of driving the foot directly, does so instead through a CV joint. In this paper we investigate the feasibility of a Clemens linkage for use in the joint of a humanoid robot. A static analysis is performed to determine proper link and joint sizing, and a displacement analysis of the spherical joint provides a means of optimizing the linkage for the desired joint range of motion.

B - 2 Background

There are many types of Constant Velocity joints in existence, and Hunt offers a list of characteristics required of CV joints to maintain constant velocity [80]. The most common joints include those seen in automotive applications such as the Rzeppa, Tripod, or Cardan joint and less recently the Myard[81] and Tracta[82] joints. For robotics applications, simplicity, compactness and high articulation is necessary, and it will be shown that the Clemens linkage satisfies these criteria.

The Clemens linkage is a spatial linkage with a RRSRR topology as seen in Figure B-1[83]–[85], although the precise configuration can vary. For example the location of the revolute axes connecting *shaft1* and *link1* may be located anywhere in the plane perpendicular to the axis of rotation of the input shafts. The Clemens linkage has been studied before. A displacement analysis by Wallace showed that the linkage is in fact a constant velocity one [86]. Milenkovic showed an alternative proof of constant velocity, but also addressed some singular configurations [87]. He goes on to describe some uses for the linkage in robotic wrists [88].

B - 3 Analysis

Our goal is to size the linkage to be suitable for use in a humanoid robot ankle. In this role, there are several criteria that should be met. The first is that the linkage minimize the number of bearing interfaces to reduce the weight, size, complexity, and possibility of backlash in the joint, the last point being particularly important to maintain high positional accuracies of the ankle and therefore robot. The linkage must be placed inside the joint, such that rotations can be transmitted through one DOF to actuate a second DOF. Therefore the size must be reduced to the point that it does not interfere with the structure of the joint itself.

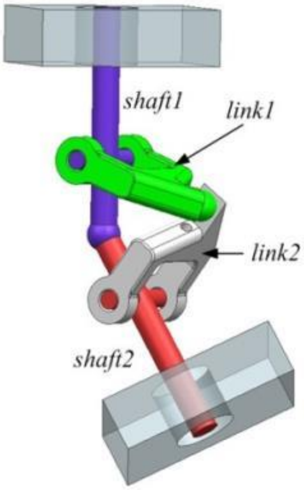

Figure B-1: Representative rendering of Clemens Linkage.

To that end, using a spherical bearing as the spherical joint in the Clemens linkage is advantageous in this application over the use of three intersecting revolute axes acting as a spherical joint. It can be shown though that this latter configuration (three intersecting revolute axes) offers some advantages in terms of range of motion and efficiency, but this is not addressed here. The single bearing offers a small package, generally large load carrying ability, and minimizes backlash. While there is incentive to use a spherical bearing, they are limited in both their capacity and articulation. Therefore, it is necessary to analyze the Clemens linkage in order to determine their feasibility in such a role.

In this analysis we assume the linkage is configured with the two revolute axes on the input and output side of the linkage intersects each other, as seen in Figure. Furthermore, because the structure of the robot fully constrains the two shafts to intersect, there is no need for further constraints on them (the Clemens linkage is often configured with a spherical joint at the intersection of the input and output shafts).

The ankle joint of a humanoid sees a wide range of motion, in this case it is desired that the ankle pitches +90 and -45 degrees about the *y* axis to allow the robot to both kneel and take large strides. The leg configuration can be seen in Figure B-2. It shows that the ankle roll joint is placed after the ankle pitch to allow the foot/shin to roll even at large angles of pitch. Because the roll is located distal to the pitch DOF, this is the axis that will be actuated with the Clemens linkage, and it is therefore necessary that the linkage transmit torque through +90/-45 degrees of shaft deflection. It is desirable to get as much range out of this roll axis as possible, so that will be the focus of this investigation.

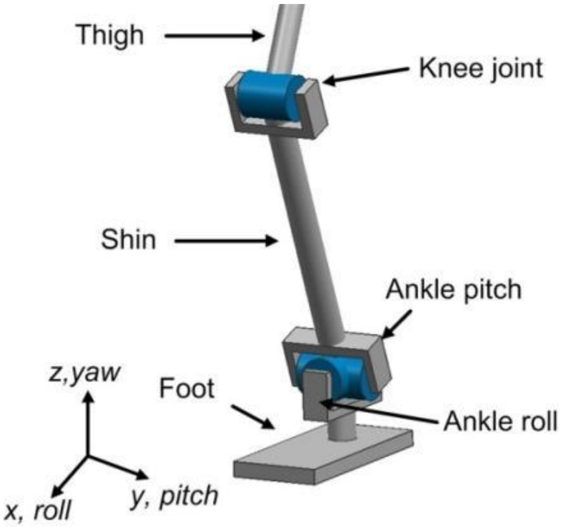

Figure B-2 Configuration of humanoid robot ankle pitch and roll DOFs

To perform a static analysis, it is first necessary to determine the spatial relationship between all the links in any position. The necessary parameters of the Clemens linkage are shown in a nominal configuration in Figure B-3.

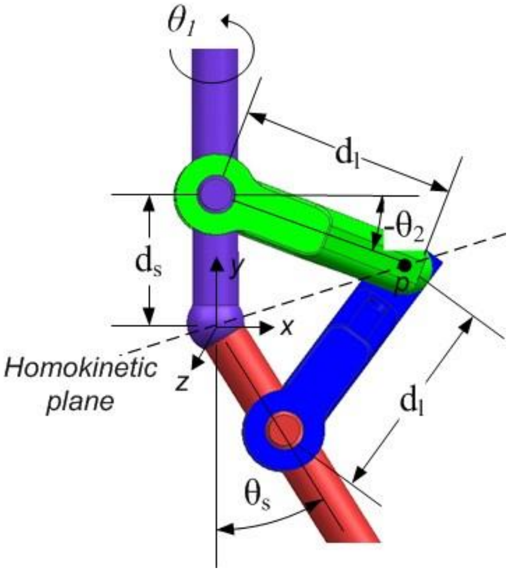

Figure B-3: Clemens linkage parameters

The angular position, θ_2, must be determined for any angle, θ_1 and θ_s. Firstly, the *x*, *y*, and *z* coordinates of endpoint, *p*, are given by Equation :

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} d_l \cos(\theta_1) \cos(\theta_2) \\ d_s + d_l \sin(\theta_1) \\ -d_l \sin(\theta_1) \cos(\theta_2) \end{bmatrix} \qquad \textbf{\textit{(B-1)}}$$

This point must lie in the homokinetic plane to ensure shafts A and B are driven at constant angular rate with respect to each other. The equation for this plane is given by Equation B-2:

$$0 = sin(\theta_s/2)\, x - cos(\theta_s/2)\, y \tag{B-2}$$

Solving for θ_2 yields Equation B-3, seen below.

$$\theta_2 = cos^{-1}\left(4\,\frac{\sqrt[2]{d_l{}^2 sin^2(\theta_s)(-(d_l{}^2-2d_s)cos(2\theta_s)+2d_l{}^2cos(2\theta_1)cos^2(\theta_s))+3d_l{}^2-2d_s{}^2)}+d_l d_s\,cos(\theta_1)sin(2\theta_s)}{d_l{}^2(cos(2(\theta_1-\theta_s))+cos(2(\theta_1+\theta_s))+2\,cos(2\theta_1)-2\,cos(2\theta_s)+6)}\right) \tag{B-3}$$

The forces that must be transmitted through the spherical bearing at point, *p*, must now be computed. Figure B-4 shows the force that acts through the spherical bearing. This force will be calculated from system of equations based on the sum of torques about various components.

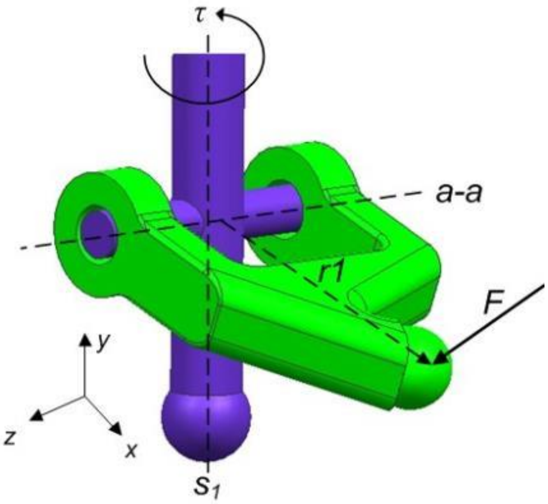

Figure B-4: Free body diagram of *link1* and *shaft1*

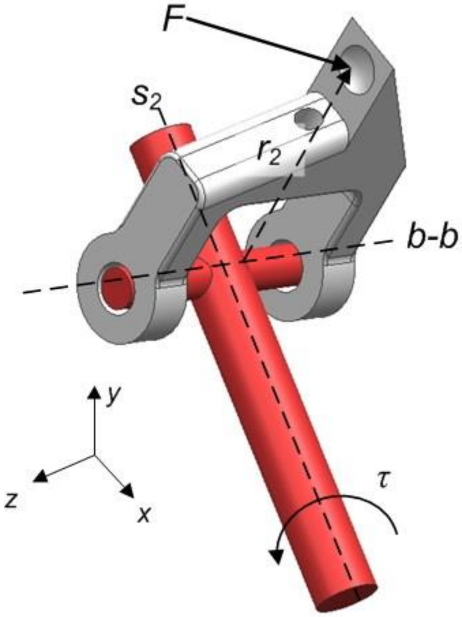

Figure B-5: Free body diagrams of *link2* and *shaft2*

The sum of the torques on *link1* about axis *a-a* must be zero if the dynamic effects are assumed to be small and the revolute joint is frictionless. This is a safe assumption for the case studied here, in which the velocities of robotic links are small and ball bearings are used. This yields Equation B-4.

$$0 = \left(\vec{F} \times \overrightarrow{r_1}\right) \cdot \hat{a} = \left(\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \times \left(\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} - \begin{bmatrix} 0 \\ d_s \\ 0 \end{bmatrix}\right)\right) \cdot \begin{bmatrix} sin(\theta_1) \\ 0 \\ cos(\theta_1) \end{bmatrix} \tag{B-4}$$

Simplifying Equation above gives us the first equation of three necessary to solve for *F*, Equation B-5:

$$0 = \begin{bmatrix} cos(\theta_1)\,(p_y - d_s) \\ sin(\theta_1)\,p_z - cos(\theta_1)\,p_x \\ -\,sin(\theta_1)\,(p_y - d_s) \end{bmatrix} \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{B-5}$$

Furthermore, force *F* must also generate torque τ on *shaft1* and *shaft2* as in Equations below

$$\tau = (\vec{F} \times \overrightarrow{r_1}) \cdot \hat{y} \tag{B-6}$$

$$\tau = (\vec{F} \times \overrightarrow{r_2}) \cdot \hat{s}_2 \tag{B-7}$$

Where $\hat{y}$ is the unit vector in the y direction, and $\hat{s}_2$ is the unit vector collinear to *shaft2* as in Equations B-8 and B-9:

$$\hat{s} = \begin{bmatrix} sin(\theta_s) \\ -cos(\theta_s) \\ 0 \end{bmatrix} \tag{B-8}$$

$$\overrightarrow{r_2} = \left(p - \begin{bmatrix} sin(\theta_s)\,d_s \\ -cos(\theta_s)d_s \\ 0 \end{bmatrix}\right) \tag{B-9}$$

Solving simultaneously yields Equation B-10:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} cos(\theta_1)\,(p_y - d_s) & sin(\theta_1)\,p_z - cos(\theta_1)\,p_x & -sin(\theta_1)\,(p_y - d_s) \\ sin(\theta_1) & 0 & cos(\theta_1) \\ sin(\theta_1)cos(\theta_s) & sin(\theta_1)\,sin(\theta_s) & cos(\theta_1) \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \frac{\tau}{d_l cos(\theta_2)} \\ \frac{\tau}{d_l cos(\theta_2)} \end{bmatrix} \qquad \textbf{\textit{(B-10)}}$$

For a nominal set of linkage parameters(d_l = 0.12[m], d_s=0.1[m], τ=10[Nm]) the magnitude of the force, *F*, is plotted in Figure B-6 for shaft angles, θ_s=0 and θ_s=π/2. When θ_s=0, point *p* traces a circle centered about the origin in the x-z plane, and therefore intuitively the force, *F*, is constant. In the case of θ_s=π/2, point *p* again traces a circle, but one that lies in the homokinetic plane and is not centered about the origin. One can see that the force, *F*, is maximum on either side of θ_1=π, when a combination of two effects (point *p* is closest to the origin and the constraint forces peak) cause the force *F* to peak at 6.7[N]. Increasing the ratio of d_l *to* d_s has the effect of lowering the magnitude of the forces across all values of θ_s by increasing the radius of the circle traced by point, *p*. Therefore, a Clemens linkage should be designed to maximize this ratio. Furthermore, the larger the link lengths d_l *to* d_s, the smaller the force *F*. Therefore as large a link as can be packaged should be used to minimize the force through the linkage.

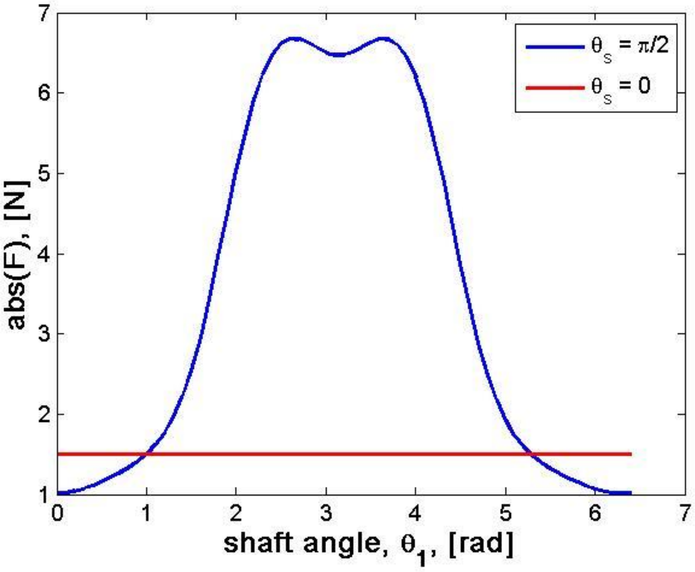

Figure B-6: Magnitude of force, *F*, for one complete revolution of shaft1 at two different shaft angles, θ_s.

Articulation analysis

To reiterate, a spherical bearing was chosen in this application for their high load carrying capacity, light weight, and low backlash. However, commercially available spherical bearings have limited articulation, defined as the total angular range of motion of the inner race with respect to the outer race. With the use of a spherical bearing it is therefore necessary that the movement of *links1*and *links2* do not exceed this articulation. In order to maximize the available articulation and therefore the range of motion of the Clemens linkage, it is beneficial to align the neutral axis of the race and bearing (defined as an axis perpendicular to the plane of the outer bearing race and parallel to the hole through inner sphere respectively) such that it minimizes the required articulation, as seen in Figure 7.

To determine the optimal tilt, it is necessary to find the 3D angle between r_a and r_b. This can be done by first solving for the unit vector in the direction of both r_a and r_b as in Equations B-11 and B-12:

$$\hat{r}_a = R_y(\theta_1)R_z(\theta_2)R_z(\theta_a)\begin{bmatrix}1\\0\\0\end{bmatrix} \tag{B-11}$$

$$\hat{r}_b = R_z(\theta_s)R_y(\theta_1)R_z(\theta_2)R_z(\theta_b)\begin{bmatrix}1\\0\\0\end{bmatrix} \tag{B-12}$$

where *R* is the rotation matrix about the subscripted axis. The angle is given simply as:

$$\alpha = acos(\hat{r}_a \cdot \hat{r}_b) \tag{B-13}$$

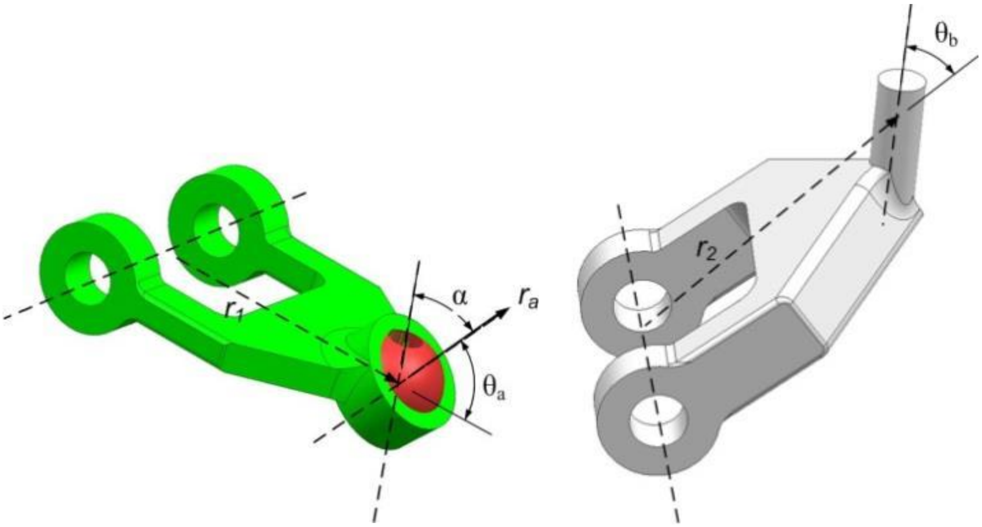

Figure B-7: Illustration of spherical bearing inclination, θa and θb to improve articulation.

Two angles can be varied, the angle between the axis of the outer race and *link1*, θ_a, and the angle of the inner race with *link2*, θ_b. Figure and Figure illustrates the articulation angle for several cases.

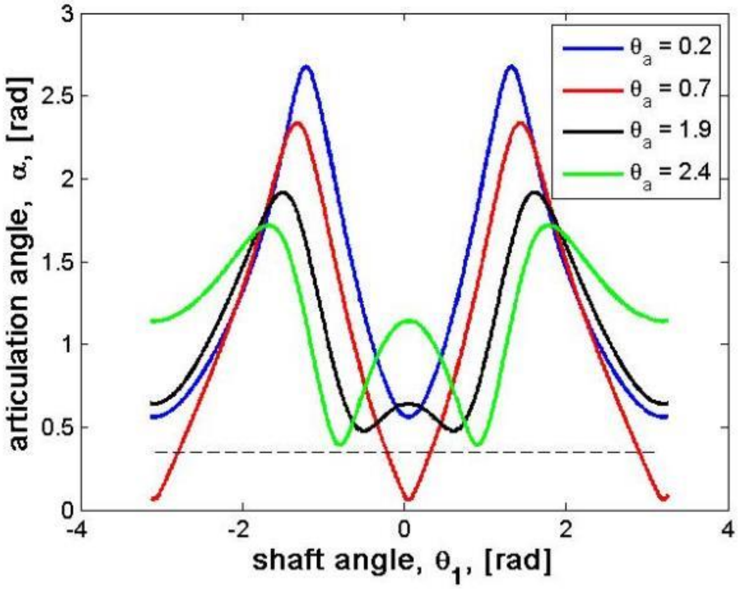

Figure B-8: Articulation angle when θ_b=0.0[rad] for various shaft angles and outer race angles, θ_a. θ_s is equal to $\pi/2$ for these plots.

The typical maximum articulation angle of most commercially available spherical bearings is 40 degrees, or +/-20 degrees from center (which directly correlates to α). This maximum is represented by the hashed line in Figure 8 and 9. With almost no tilt in either the inner or outer race, it is clear from the blue line in Figure that no spherical joint will have the necessary articulation (the blue line never drops below the hashed line). However by adjusting just outer race tilt, it becomes possible to achieve a range of motion of +/- 0.3[rad] when the shaft angle is $\pi/2$ [rad]. Furthermore Figure shows that by adjusting both the inner and outer angles, it is possible to further improve upon the range of motion. In this case, with θ_a=2.4 and θ_b=0.8, (the green line), a range of motion of +/-0.75[rad] is possible about the origin.

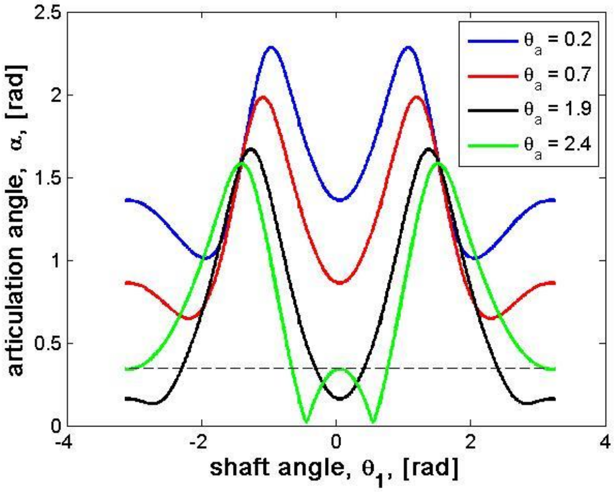

Figure B-9: Articulation angle when θ_b=0.8[rad] for various shaft angles and outer race angles, θ_a. θ_s is set to $\pi/2$ for these plots.

Therefore, spherical bearing races should be carefully aligned so as to provide the necessary range of motion in the θ_1 direction over the entire range of angles seen by θ_s. Furthermore, this tilt represents an important tool to optimize the linkage in future applications. In our scenario of a humanoid robot ankle, at these large shaft angles, the necessary range of motion of the joint does not exceed that of the linkage, and therefore the linkage is suitable for this joint. That is, at $\pi/2$[rad] of positive pitch in the ankle, it is desirable to be able to roll +/- 0.75[rad]. With the correct bearing inclinations, this range of motion is possible with the Clemens linkage.

B - 4 Implementation in a robotic joint

It has been shown that the Clemens linkage has both the necessary load carrying capacity and the range of motion for some robotic applications including that studied here, a humanoid ankle. Figure 10 shows a representative model of how a two DOF joint would be configured around a Clemens linkage to allow both actuators to be placed on one side of the joint.

There are two links, the shin (grey) and the foot (brown), rotatably attached to each other through a cross gimbal (green). Two actuators (blue) actuate the joint. The pitch actuator acts directly on the cross gimbal. The roll actuator acts on the red link, which serves as the input shaft to the Clemens linkage, analogous to *shaft1*. The grey and purple links of the linkage are the implementations of *link1* and *link2* respectively. The output of the linkage is attached to the foot through a perpendicular revolute axis. In this way, both ankle actuators may be placed well up the shin in order to reduce the total inertia of the leg and eliminate the need to package a large actuator in either the gimbal or on the foot. Because of the range of motion constraints of the linkage, it is preferable to use the linkage to actuate the DOF of a 2DOF joint with a smaller range of motion such that the articulation of the spherical joint is not exceeded.

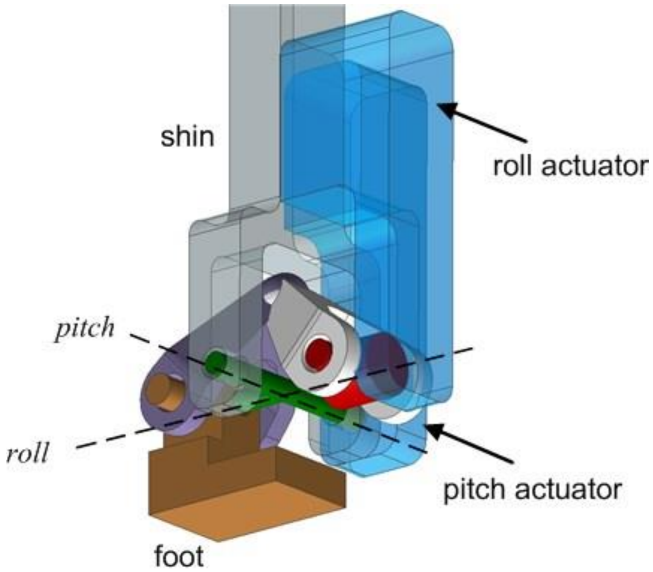

Figure B-10: Representative CAD model of how a Clemens Linkage could be implemented in a 2DOF robotic joint (humanoid ankle).

B - 5 Discussion

There are two popular methods for actuating a humanoid ankle, as a true serial joint and as a parallel mechanism. Examples can be seen in Figure B-11. The Clemens linkage does increase the complexity as compared to a serial chain; however, it offers the improved mass distribution of a parallelly actuated ankle by placing the massive actuators closer to the knee. A common disadvantage of the parallel style is significant loss of mechanical advantage at large ranges of motion. The Clemens style ankle avoids this tradeoff having constant mechanical advantage over its entire range of motion.

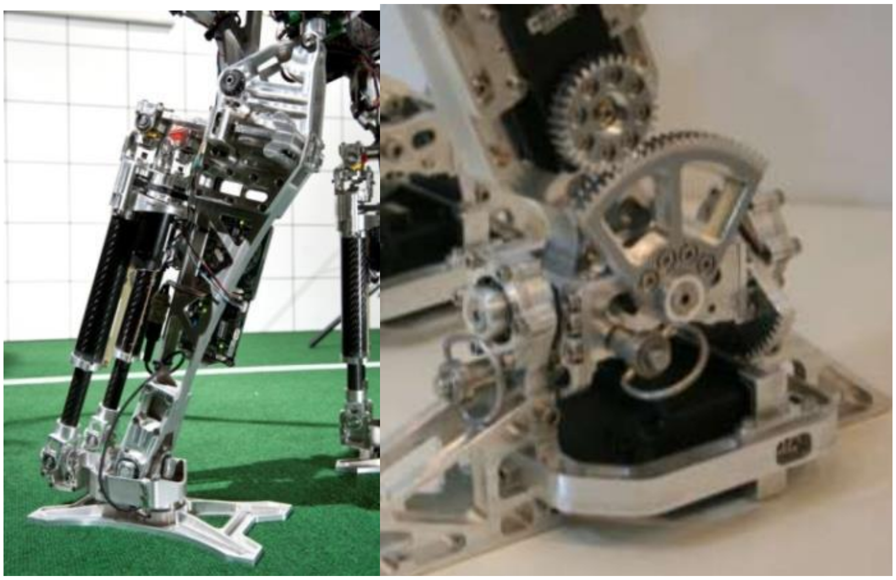

Figure B-11: Parallel actuated ankle on left and serially actuated on right.

B - 6 Conclusions

A notable drawback of serial manipulators in dynamic applications is the large inertia created by typically heavy electromechanical actuators at the distal end of the manipulator. Additionally, designers must often solve difficult packaging constraints at these distal joints. For example in the case of humanoid robots, joints and links around the ankle have conflicting requirements, low inertia and high torque. A constant velocity joint, the Clemens linkage, was presented here as a means to relocate actuators across a one DOF revolute joint to an adjacent link while maintaining a serially actuated architecture. To reduce backlash, complexity, and weight, a spherical plain bearing is utilized within the linkage. The forces transmitted by the spherical bearing were calculated with respect to the input torque and linkage parameters. Furthermore, the range of motion of the linkage was shown to be limited by the articulation of the spherical bearing, and parameters which can be used to optimize this range were identified. In conclusion, Clemens linkage is a useful design element for serial robotic applications and may possibly extend the dynamic ability of manipulators in the future.

Appendix C Motor Control

Existing motor controller solutions work quite well for traditional robotics applications. For example, in an industrial setting, precise speed and position control capabilities have a direct correlation to the quality and economy of production robots. However, there is a developing trend in the research community to deliver not only fast and precise robots, but also robust, efficient and interactive solutions. These robots will break outside of the existing robotic paradigm, and allow them to work closely with humans, perform delicate tasks, and operate in unstructured terrain with greater energy autonomy. The capability that must be developed to allow this paradigm shift is torque/force controllable, low impedance actuators.

Torque control allows for the accurate control of the torques delivered through the actuator through use of a torque feedback sensor. (This technique is also referred to as Force control when implemented in a linear actuator delivering forces.) When implemented with an appropriate control system, these actuators can be made inherently passive and compliant, thereby allowing such actuators to be used in tasks requiring human-robot interaction (such as rehabilitation robotics), unknown environments (such as legged locomotion), and energy efficiency (such as passive dynamic motions).

Force and torque controlled actuators can be implemented using either high bandwidth, high torque direct drive motors, or alternatively, by placing a spring in series with a more conventional geared motor (series elastic actuator). The latter approach is the one we have taken, and is shown in Figurewith an accompanying schematic. This approach results in improved impact tolerance, as the mechanical structure acts as a low pass filter, and also improved energy efficiency through energy storage [54]. In the robotics research community, there are an increasing number of robots using force or torque control. These include Boston Dynamics Petman[89], IHMC's M2V2[25], Meka robotics M1 Mobile[90], DLR LWR-2 [4], Delft's Flame[23]. There numerous other labs that utilize torque control in rehabilitation robotics and actuator development.

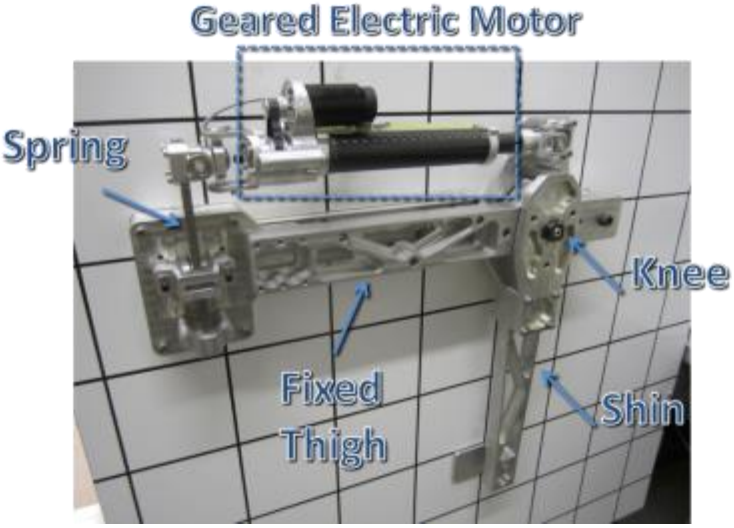

Figure C-1. Picture of the test stand above, and schematic of the same on bottom

There are currently no commercially available motor drives capable of force or torque control. Due to this, research labs are forced to develop their own motor controllers to gain the functionality that they need. More specifically, the ability to program custom control loops for implementing more advanced and distributed control architectures. We believe that a motor drive with force or torque control functionality would be an important contribution to the field and would be attractive to many research and development laboratories. Furthermore, a programmable motor drive would open the potential market to a wide variety of laboratories and developers.

C - 1 Architecture

The desired architecture is required to control the mechanical setup seen in Figure 1, with the major difference being the equations below are derived for Torque control rather than Force (as we have implemented mechanically). The plant shown in Figure is shown as a schematic in Figure.

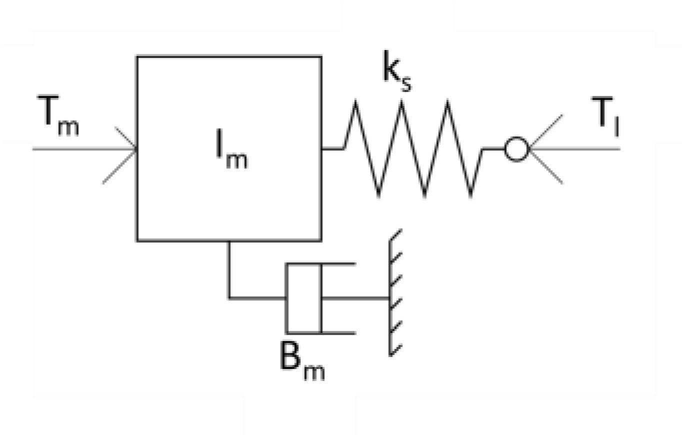

Figure C-2. Schematic of the Plant as derived from the mechanical setup in Figure 1.

The transfer function of the plant is given in Equation C-1:

$$\frac{T_l(s)}{T_m(s)} = \frac{1}{\frac{I_m}{k_s}s^2 + \frac{b_m}{k_s}s + 1} \qquad \textit{(C-1)}$$

where I_m is the reflected inertia of the motor (that is the inertia as seen through a gear reduction), b_m is the damping of the motor and drivetrain, and k_s is the spring constant. We wish to use the controller to achieve a torque on the load, T_L equal to the desired torque, T_d. The feedback controller will use proportional and derivative control on the error along with a feedforward term to control the torque (or current) on the motor, T_m to achieve this goal. Feedback will be accomplished through a torque sensor attached to the spring to measure the output load. The transfer function describing both the controller and plant is given in Equation C-2.

$$\frac{T_l(s)}{T_d(s)} = \frac{K_d s + (K_p + 1)}{\frac{I_m}{k_s}s^2 + \frac{b_m + k_s K_d}{k_s}s + (K_p + 1)} \qquad \textit{(C-2)}$$

where K_p is the proportional gain, K_d is the derivative gain, For a derivation of Equation C-2, please see [91].

The controller architecture is shown in Figure 3. This architecture includes several additional terms that are required of the controller to handle the unit conversions of the various signals. Of those terms: A_O is the analogue voltage offset, A_s is the analogue scaling term, and C_{ff} is the conversion between desired torque and amperage.

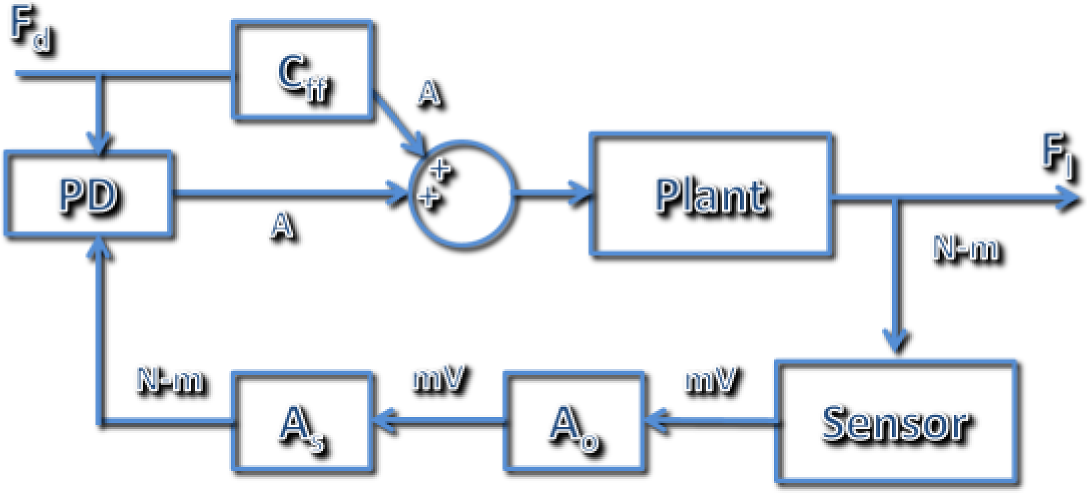

Figure C-3. Implementation of the controller for the Maxon Epos Firmware showing the units of each of the signals.

The constants necessary for this implementation are given in Table 1 along with the respective units. Because this is the first implementation of this control scheme, it is desirable to have as much flexibility in both the range and the resolution of the parameters. Therefore, we wish to store each parameter as a numerator and denominator, an example of the proportional gain can be seen in Equation C-3:

$$K_p = \frac{K_{p\dots num}}{K_{p\dots den}} \tag{C-3}$$

where $K_{p\dots num}$ and $K_{p\dots den}$ are both 16 bit integer numbers in this case. The desired data types and ranges are also included in Table C-1.

Table C-1. Parameter values and units

Parameter Name	Abrv.	DataType	Min	Max	Units
Proportional Gain	$K_{p\dots num}$	Int16	0	65535	A/Nm
	$K_{p\dots den}$	Int16	0	65535	
Derivative Gain	$K_{d\dots num}$	Int16	0	65535	A·s/Nm
	$K_{d\dots den}$	Int16	0	65535	
Desired Torque	$T_{d\dots num}$	Int16	-32767	32767	Nm
	$T_{d\dots den}$	Int16	0	32767	
Feed Forward Scaling Constant	$C_{ff\dots num}$	Int16	-32767	32767	mA/Nm

	$C_{ff\ldots den}$	Int16	0	32767	
Analogue Scaling	$A_{s\ldots num}$	Int16	-32767	32767	Nm/V
	$A_{s\ldots den}$	Int16	0	32767	
Analogue Offset	$A_{O\ldots num}$	Int16	-32767	32767	mV
	$A_{O\ldots den}$	Int16	0	32767	

We have estimated the values of the parameters as well as seen in Table 2. The gains listed are equivalent to the “$K_{P\ldots SI}$” listed in the Application notes in that they use SI units. Should the gain storage scheme describe above and represented in Table C-1 not be possible, we would like to defer to the values in Table C-2.

Table C-2. Parameter values

Parameter Name	**Abrv.**	**Min Value**	**Max Value**	**Units**
Proportional Gain	K_p	0	1000	A/Nm
Derivative Gain	K_d	0	50	A·s/Nm
Desired Torque	T_d	-200	200	Nm
Feed Forward Scaling Constant	C_{ff}	-500	500	mA/Nm
Analogue Scaling	A_s	-50	50	Nm/V
Analogue Offset	A_O	-5000	5000	mV

Appendix D DRC work

The DARPA Robotics Challenge is a real-world competition designed to accelerate the development of rescue robots and the semi-autonomous behaviors that govern them. The competition is based upon the Fukishima reactor disaster response and similar scenarios. Often environments such as these become extremely hazardous to humans, to the point that they are actually impossible to operate it. Therefor much like the impetus for the SAFFiR project, it is in exactly these situations that a robot may act as a powerful surrogate for human first responders.

Empowering robots to serve as supervised first responders will allow for faster, safer, and more persistent service in the aftermath of natural and man-made disasters, but doing so will require revolutionary advances in technology. Seemingly mundane tasks, such as traversing rubble, wielding common tools, or climbing into a vehicle are state-of-the-art capabilities for mobile robots – extremely challenging tasks, even in the controlled environment of a sophisticated laboratory. Creating a single platform capable of performing all of these tasks in austere environments requires leap-ahead advances in hardware and software.

It is the goal of the DRC then to spur the advancement of robotic technologies that can make a measurable impact on the effectiveness of first responder robots. These technologies include: mobility, manipulation, localization and mapping, autonomous behavior, and teleoperation. In that vein, DARPA has contrived seven individual challenges that make up the greater challenge. Specifically, these include 1) Rough terrain walking, 2) climbing a ladder, 3) operating a valve, 4) operating a firehose, 5) removing debris, 6) driving a vehicle, 7) opening doors, and 7) breaking through a wall. The proposed work outlined below pertains directly to the rough terrain walking task.

This chapter outlines the work lying ahead. It is framed within the scope of the rough terrain walking portion of the DARPA Robotics Challenge. Therefore, both the task and the specific challenges that have been identified within that task will be discussed first. The robot taking part in the Challenge, THOR, is similar to SAFFiR, but incorporates many improvements. The relevant portions of THOR's architecture will be described. Finally, the balancing and walking algorithm that will be developed for THOR will be proposed. Similar characteristics between THOR and SAFFiR's algorithms will be outlined.

D - 1 DARPA Robotics Challenge

The DARPA Robotics Challenge will force robotics technology to overcome the difficult tasks presented[92]. One of the most demanding tasks that if accomplished will represent a significant improvement in the state of the art, is the rough terrain walking. As has been mentioned previously, bipeds today can only traverse a limited variety of terrain. On the other hand, the DRC sets forth a demanding walking course comprised of 8 levels of increasing difficulty. This section will discuss the eight sections and the expected difficulties THOR will have with each.

The walking challenge is comprised of eight different sections arranged consecutively to form a course 80[ft] long and 8[ft] wide. The sections are constructed of wood or cinder block features that serve as the obstructions. In total, the sections address certain walking techniques such as stepping on uneven terrain, traversing inclines and declines, stepping up and down, and crossing over stepover obstacles.

1) The first section amounts to flat terrain made up of paving stones and Astroturf. As shown above, this type of terrain can be handled by the current generation of humanoids such as SAFFiR. No significant challenge to THOR is expected here.

2) This section includes two short ramps, made of plywood that rise 6[in] in 4[ft]. Similarly, the decline assumes the same slope.

3) The tripping hazard section is designed to test the step planning and underfoot terrain capabilities of the robot. That is, whether or not the robot is sensitive to height variations under the foot. The obstacles are constructed of first 2x4's and then 4x4's arrange diagonally across the length of the walking challenge as seen in Figure ?. This is the first sub-challenge in which perception can improve the robots chance of success. Intelligent step planning and foot placement may allow the robot to avoid all the obstacles. Being tolerant of uneven conditions underfoot though will increase the likelihood of success.

4) Two rows of cinderblocks make up the first of the step-over obstacles as in Figure ?. There are two individual obstacles in this sub-challenge. The first is a straight line of blocks while the second forms a "zig-zag" pattern. Because of the height and width of the blocks, the robot faces significant difficulties to both step over and maintain balance across the walls. The height also precludes any attempt to step up onto the wall and then down. Therefore this challenge will likely be handled using a whole bodied manipulation technique.

5) The next three sub-challenges are continuous terrain features made from cinder-blocks. They all challenge test the robots ability to perceive terrain and act accordingly. Sub-challenge five is made from a varying checkerboard pattern of blocks. In addition to perception, the robot will need to maintain balance as it steps up and down the blocks.

6) This section builds on the one before it by incorporating consecutive step ups and step downs to form a "hill." The declined portion also includes blocks on the side, which necessitate very accurate foot placement to avoid slipping and getting caught in one of the holes. Should sub-challenge five be accomplished, sub-challenge six requires few if any additional capabilities.

7) Section seven is much like six except that each step is inclined (i.e. not horizontal). Additionally, each step is at a different angle to its neighborThe steps again increase in height and then decrease to ground level. Perception will need to be used to estimate the ground normal so the feet can be placed accurately. The step algorithm and step planner will also need to be carefully designed to handle terrain that would otherwise cause the foot to exceed the range of motion of the ankle. Because of its significance

8) The final sub-challenge includes a variety of step-over obstacles including scaffolding and artificial rocks. Perception will play a large role as well as whole body motions over obstacles that would otherwise be impossible to traverse.

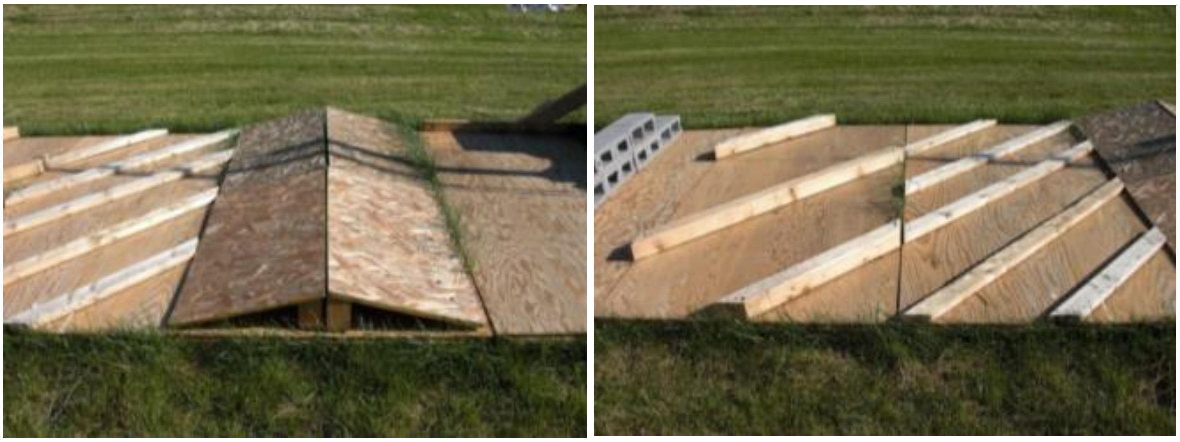

Figure D-1. The second and third portions of the DRC walking course. On left is the inclined ramps (2), and on right are the tripping hazards (3).

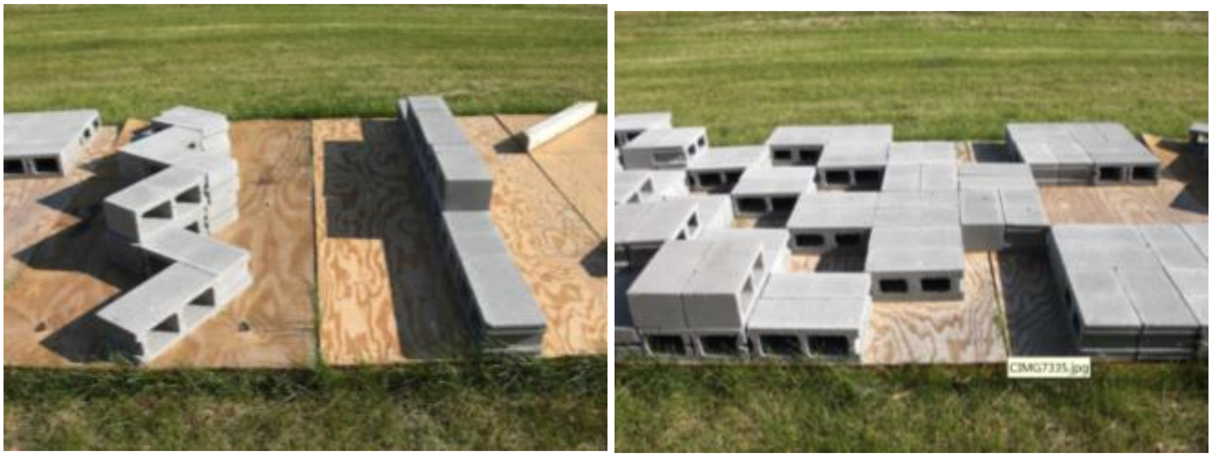

Figure D-2. The hurdle obstacles on left, and the footfalls and holes section on right.

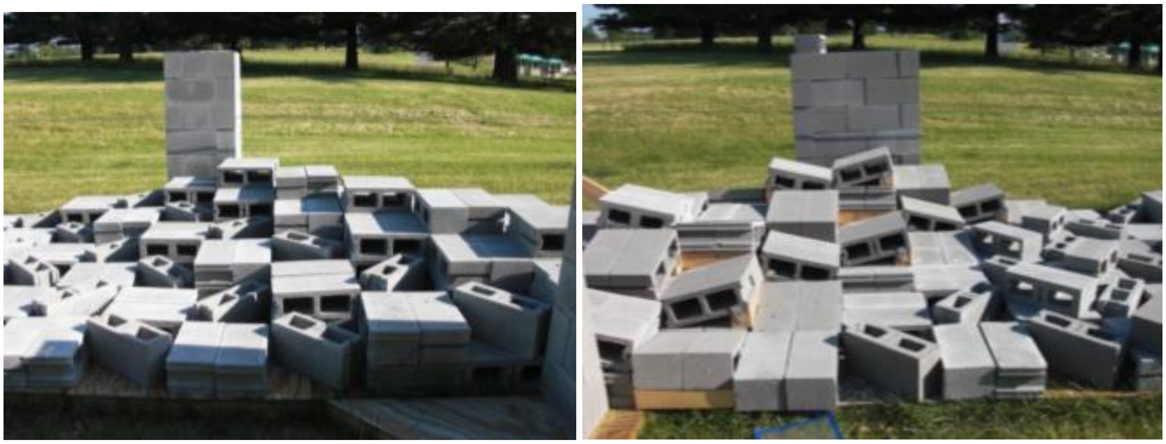

Figure D-3. Declined flat blocks on left and inclined Pitch/Roll blocks on right.

Our approach for the DRC will closely follow the control themes developed for SAFFiR but extended to a full force control architecture. As discussed above, the most recent advances in humanoid walking have been focused on a full force controlled walking approach. Very promising results for balancing on moving terrain have been shown [93]. Some early walking optimization based experiments also show promise. Likewise, our experience with SAFFiR using partial force control has demonstrated the advantages therein. For the DRC then, we will be implementing a full force control in the general framework that the literature explains. This framework is outlined in [94]–[98]. Given an understanding of the current state of the art, our particular contribution to the field will be in two primary areas:

1) Heel-to-toe walking and toe-to-toe walking techniques will be used to increase the support polygon during motions that approach the limits of the range of motion. This is particularly important when stepping down off an obstacle due to the limits of the ankle pitch joint.

The following sections will outline the proposed research. In particular, the method of determining the appropriate COM wrench from the torso accelerations will be described. The COM wrench must then be intelligently distributed between the two feet. The proposed tension based method for this will be outlined next. A preliminary set of torso control laws will also be proposed. Finally, to incorporate toe-to-toe walking, special considerations must be given to the ground reaction forces and extra joint constraints. Some thoughts on these issues will be laid out.

D - 2 Torso balance controller

In the framework provided above the six degrees of freedom of the torso are directly controlled. The linear accelerations and thus positions are derived from the desired COG trajectories. Similarly the yaw of the torso is also controlled as part of the robots heading. The pitch and roll of the torso remain as useful control inputs to aid in balancing and trajectory following. A controller was implemented on SAFFiR that used a rudimentary control law. It essentially geared it to the COM position such that it pitched and rolled in proportion to the error in the COM position. While functional, it was less than optimal. The following section outlines a proposed state-space control scheme for the torso acceleration in pitch and roll that will outperform the controller on SAFFiR.

The behavior of the torso on top of the legs acts like a two link inverted pendulum. To develop a state space model, it is necessary to derive the dynamics of this system. The linearized results can be seen in Equation D-1 and D-2:

$$\theta_1''(I_1 + l_1^2 m_2 + l_m^2 m_1) + \theta_2'' l_1 l_2 m_2 = \theta_1 g(l_1 m_2 + l_m m_1) + \tau_1 - \tau_2 \tag{D-1}$$

$$\theta_1'' l_1 l_2 m_2 + \theta_2''(I_2 + l_2^2 m_2) = \theta_2 g l_2 m_2 + \tau_2 \tag{D-2}$$

Summing of forces in x direction on body two yields:

$$F_x = m_2(-l_1 * \theta_1'' Cos(\theta_1) + l_1 \theta_1'^2 Sin(\theta_1) - l_2 \theta_2'' Cos(\theta_2) + l_2 \theta_2'^2 Sin(\theta_2)) \tag{D-3}$$

Summing of torques on body one yields:

$$(I_1 + m_1 l_m^2)\theta_1'' = -F_x l_1 Cos(\theta_1) - F_y l_1 Sin(\theta_1) + \tau_1 - \tau_2 \tag{D-4}$$

Equations D-1 through D-4 can be solved and rearranged in the form of Equations D-5:

$$\dot{\theta} = A\theta + Bu$$

$$\rho = C\theta + Du \tag{D-5}$$

The system has many constraints that the control law must obey, or damage will result. Firstly, the position and acceleration of the torso, θ_2 must remain within a bound. Too high an acceleration is not possible, and too large of a roll or pitch will exceed the physical limits of the hardware. Additionally, the ankle torques are clipped at a relatively low level. These factors make using traditional state space tools less effective for tuning. For preliminary results, a heuristic method was used instead. The following will describe how the system was tuned and the early results that were achieved.

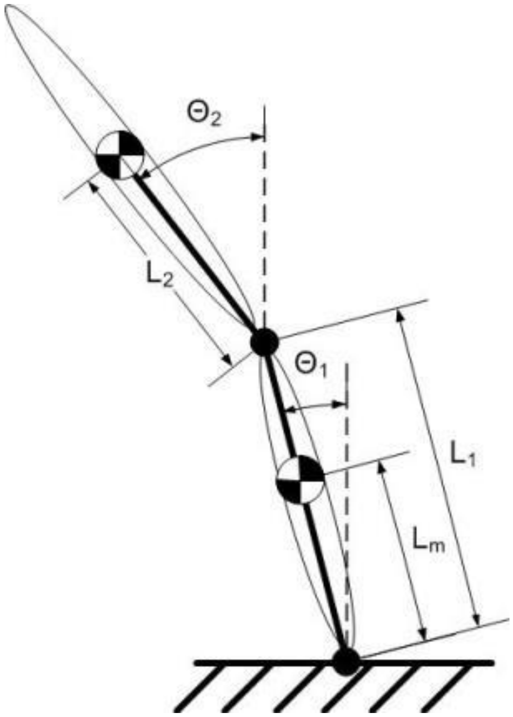

Figure D-4. The double link pendulum was studied as a means of improving balancing by servoing the torso (represented by the link 2).

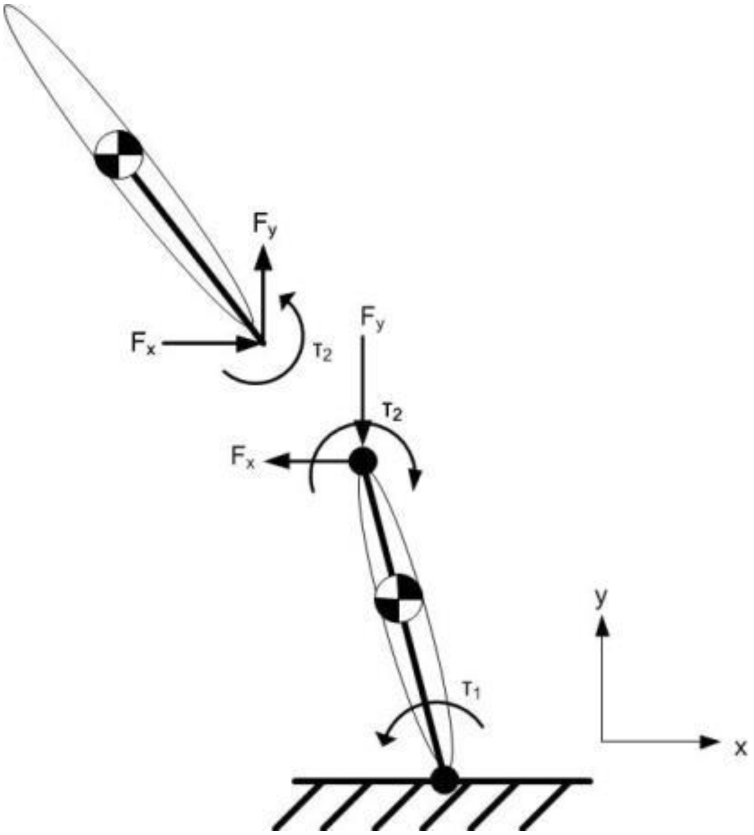

Figure D-5. A freebody diagram for a two link pendulum is used to complete the state space formulation.

There exists a maximal initial velocity from which this two link pendulum can "recover" from, where recover is defined as returning to zero initial conditions (at the zero position with zero velocity). Furthermore, the torso can move only a limited amount with some maximum acceleration. A step acceleration profile is defined that moves the torso to its maximal position with maximum acceleration. Several experiments are run using a physical simulator, Matlab Simulink, to solve the ODE's given a variety of initial velocities. Newton's method is used to find the maximum velocity from which the pendulum can recover when the torso acceleration is applied. At this velocity, the state of θ1 and θ2 are used to solve a least squares regression problem that seeks to match the actual acceleration to a calculated acceleration where the latter is found by a linear combination of the state. The coefficients that are solved for are used as the gains in the state space system in Equation . With these gains, preliminary results were better than expected.

The state space controller increased the magnitude of the disturbance from which the pendulum could recover by 50%. In addition, the gains chosen worked with both desired set points and trajectories. Within the velocity limits, the torso stayed within the desired angle limits. Therefore these gains show to promising results. Firstly, torso servoing is a valuable control input to the system and can make a significant impact on stability. Secondly, while unconventional, the heuristic method for determining gains at least offers a valuable data point from which to apply a more rigorous methodology. Future work will require that the torso control law be integrated into the balance controller and safely employed during the walking cycle, specifically ensuring that when there is a risk of over accelerating or over extending the torso. Steady state errors in the system due to signal conditioning or otherwise may also case the controller to behave unexpectedly and will have to be studied.

D - 3 Foot toe off

The DRC’s walking course is designed to challenge many different aspects of rough terrain walking. In particular, stepping over, stepping up, and stepping down are all critical abilities to complete the course. They are critical both in that they are required techniques to just traverse the course, but also critical in the sense that they present several challenge to THOR. These include power limitations, range of motion limitations, and dynamic control limitations. The particulars of these limitations and the proposed solution will be discussed in this section.

Humanoids have a long way to go until they can climb stairs and ladders with as much poise and such ease as a human. The difficulty they face is a confluence of limitations of a humanoid’s power, torque, controls, and range of motion. This is best illustrate by way of a step climbing attempt. Figure Figure D-6 shows an representative humanoid poised to climb from the left to the right up a single step. We make an assumption that the humanoid walks with its feet flat on the ground. It will be shown that this is a limitation, but represents the current state of the art. Overlaid in a thin black line is the range of motion of either ankle with respect to the hip if the foot is held flat. Said differently, the foot can be placed anywhere in that area and maintain a horizontal orientation. The hashed line represents the path the feet would take through that range of motion as the hip joint moves through space from above one foot to the other. In grey is the final configuration of the robot with the hip directly above the right step.

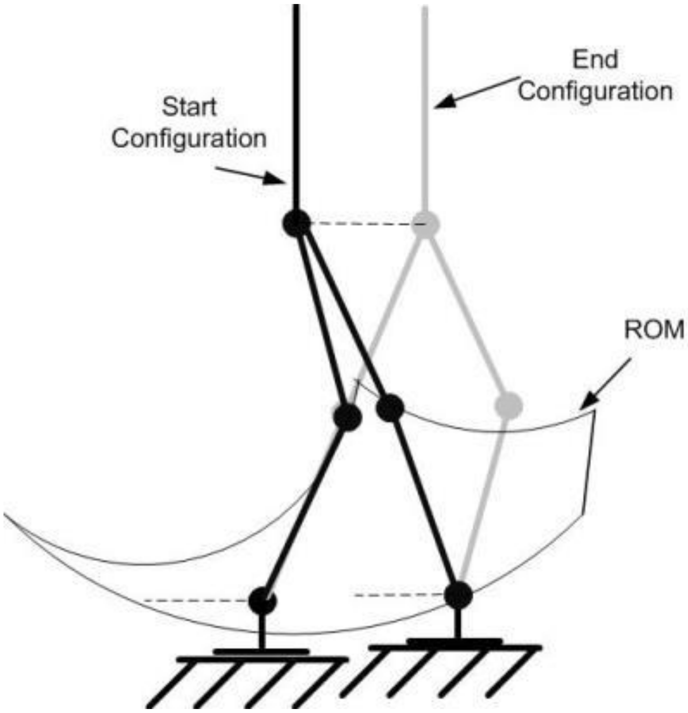

Figure D-6. THOR can step up a small height with static stability if no limitations are exceeded.

Note that several criteria are met: 1) Both feet can be placed flat on the steps and still be in the range of motion. Therefore, the step is dynamically possible with no aerial phase. 2) The hashed line is fully enclosed within the ROM envelope. Therefore the robot can remain in double support during the entire duration of the step, and thus will be statically stable as well. Meeting these two criteria is ideal, as the step can be made very deliberately and with great stability. Difficulties arise when one or both criteria are not met, as is expected to be the case during the DRC. Take for example the step seen in Figure .

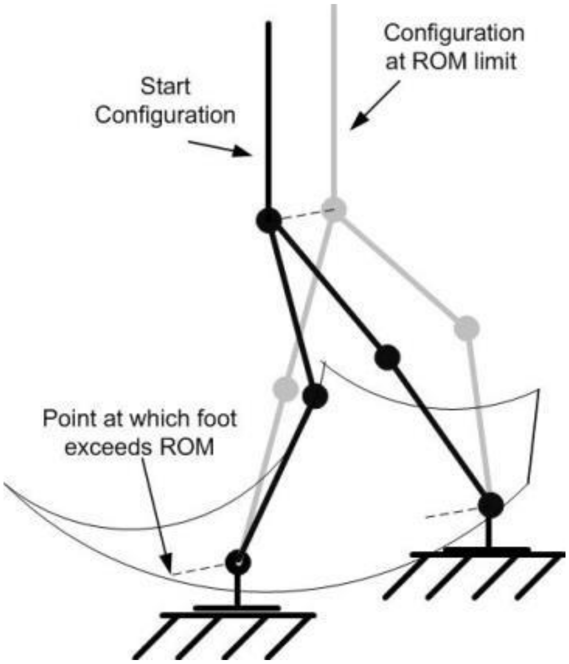

Figure D-7. The available power, range of motion, and torque limit statically stable strategies on large steps.

The step height and distance is increased in this instance until the robot can no longer make the transition with static stability guaranteed. Again the robot is able to reach both steps at the same time because their location is within the ROM envelope. However, as the hip moves from left to right, it reaches a configuration in which it can no longer maintain both feet in contact with the ground without either exceeding the ROM or maintaining a horizontal foot orientation. This is illustrated by the hashed lines, and in particular, when that line intersects the ROM envelope. There are several solutions to this challenge as picture below in Figure through Figure . They include: 1) altering the hip trajectory until it can be completed with static stability, 2) dynamically completing the step, or 3) expanding the ROM so that it encloses a greater area.

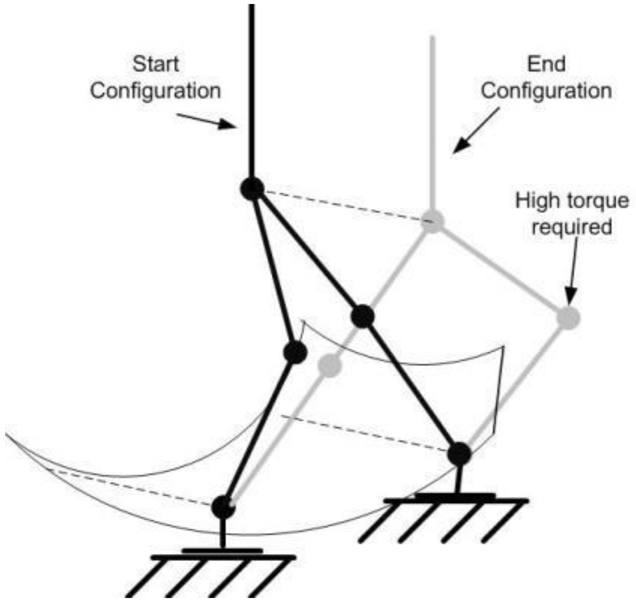

Figure D-8. A static step up will fit within the ROM but requires very high knee torques and does not solve the issue of stepping down.

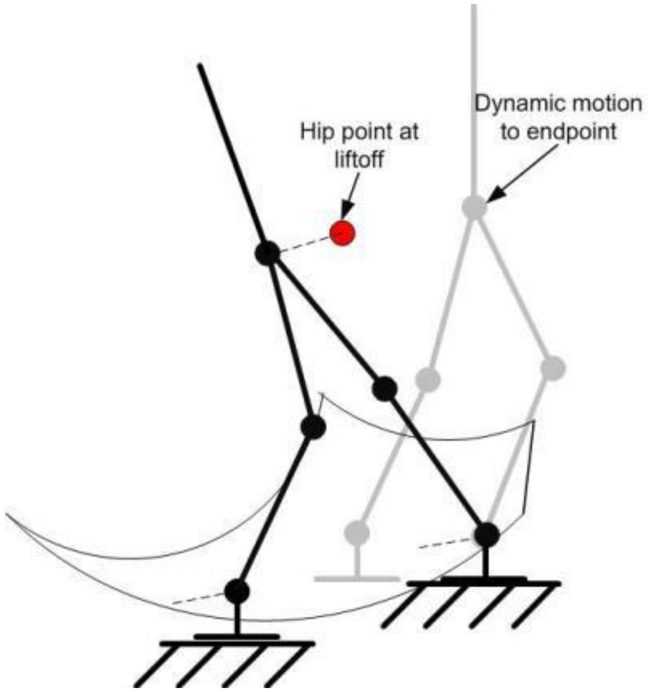

Figure D-9. A dynamic motion is possible. The projects timeline and the available power to dynamically step down made this approach too risky.

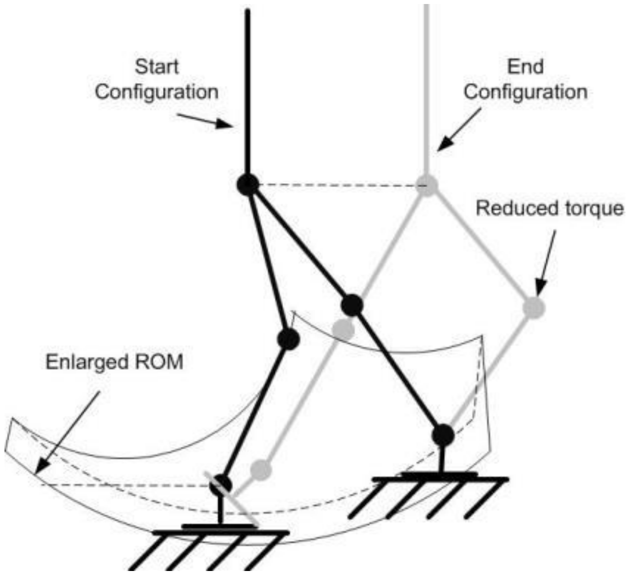

Figure D-10. Allowing the foot to rotate on edge will increase the ROM envelope, thereby enabling a long double support phase.

To take large steps such as those in the DRC, we propose to develop solution “C”, to increase the ROM by using the toe edge of the foot for support such that the ROM can be enlarged. Figure D-8 and Figure D-9 express some limitations of humanoids mentioned above. The controls required for a dynamic motion such as in solution A are riskier and more difficult to develop. Furthermore, it is likely that power limitations will present themselves during the acceleration or deceleration, particularly when stepping down. The time frame of the project also does not permit this approach. Solution B on the other hand exceeds some torque limitations of the robot. In its near final configuration, the robot has most of its weight on the one leg which is severely bent, resulting in a large knee torque. Solution C on the other hand falls more within the controls realm already explored with SAFFiR. By expanding the ROM the robot can stably make the step without nearing a torque limit.

This toe-off motion fits well within the framework outlined above and requires only a few additional considerations. Directing forces through the toe is the same as through the entire foot, except fewer moments can be applied. It shows promise in literature as well [99]–[101]. That is, the available area over which the COP can be applied is reduced from an area to a line. The static problems, outlined above can still be solved and thus incorporation will be straightforward. An additional constraint will be added to gradually move the foot’s COP to the toe during the step so a smooth transition to the toe can be made. When the heel lifts from the ground, the leg becomes underactuated. A second joint constraint will be added so that the leg kinematics and statics can be solved for the joint torques. Stepping down will be implemented as the reverse of stepping up, with the toe or heel contacting first, until the ROM allows the foot to fall flat, at which point the foot’s COP will gradual shift away from the toe.

The toe off strategy designed for THOR runs independently of higher level locomotion code. It will be primarily used only during double support, but will operate similarly in single support. Being independent, this allows the higher level code to command a torso position relative to a foothold with no regard for the kinematic peculiarities of that configuration. The toe controller is then responsible for four primary parameters: the center of pressure under the foot, the position, velocity, and acceleration of the foot relative to the foothold. Therefore, while the high level controller commands torso positions relative to the footholds on the ground, the toe controller commands the foot positions relative to the ground in order for the torso to achieve its desired position. The specific rules of motion of the toe controller will be discussed herein.

The premise of the toe controller operation is rooted in Figure 6. This figure shows a representative leg with hip, knee and ankle joint and foot placed flat on the ground. The range of motion of the hip with respect to the foot is that below the arc labeled: “Hip limit w/o toe”. The hip cannot move across this line without lifting all or part of its foot from the ground due to the kinematic singularity at the knee. If however the foot is allowed to rock up on its front edge, called the “Virtual Toe Joint”, the range of motion of the foot is expanded up to the “hip limit with toe” line before the range of motion of the ankle reaches its limit. The toe controller is therefore responsible for modulating the angle of the foot with respect to the ground about the virtual toe joint as the torso moves within the “Expanded operational region”

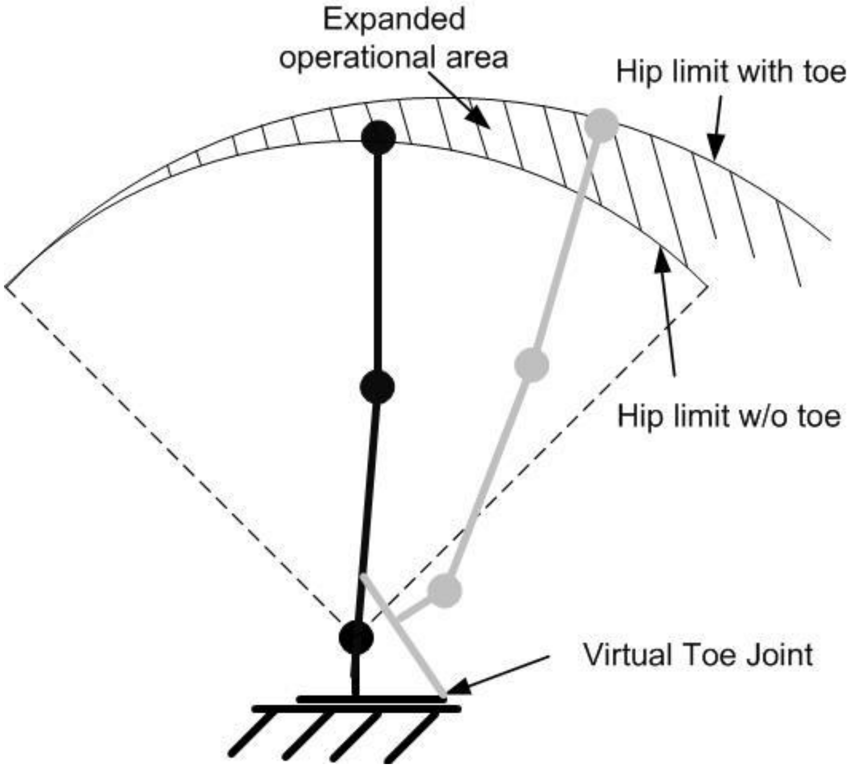

Figure D-11. The torso’s range of motion with respect to a foothold is extended by rotating on the front of the foot.

A similar premise can be developed as the torso lowers past where the range of motion of the ankle and knee would normally be exceeded. This is pictured in Figure 7. The hashed area and below is normally inaccessible due to the range of motion of the knee and ankle being exceeded. They cannot bend far enough to reach this region. If however the foot is allowed to rotated about the virtual toe joint, the torso is able to move within the expanded operational area without exceeding joint range of motion limits.

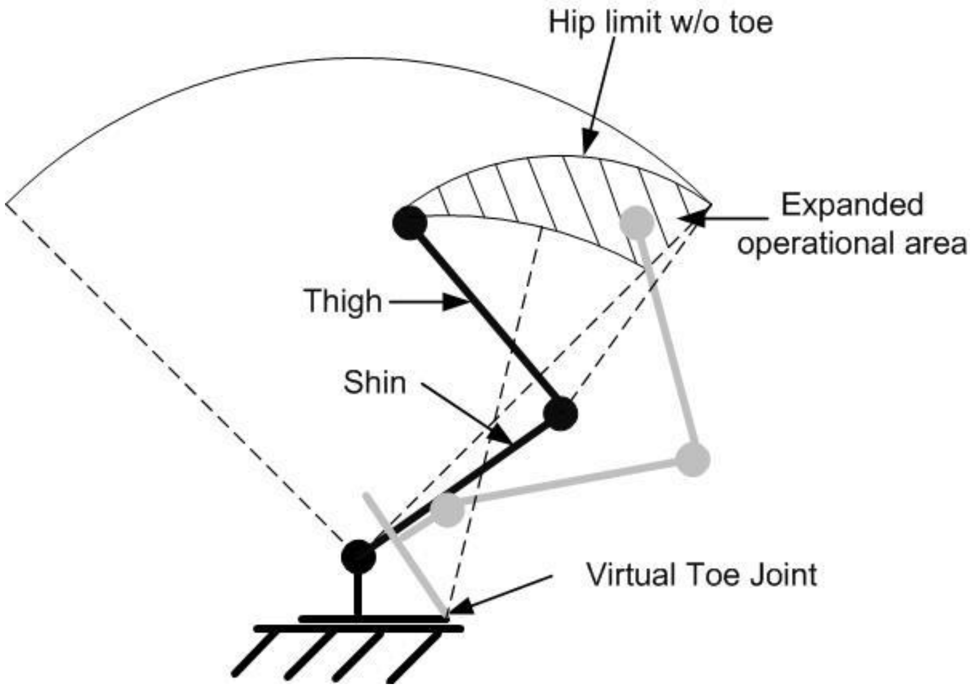

Figure 7

Figure D-12. Allowing the foot to rotate on edge will increase the ROM envelope, thereby enabling a long double support phase.

Rotating the foot about the virtual toe joint introduces an additional degree of freedom. The limb between the robot torso and the ground now effectively has seven degrees of freedom as opposed to the normal six. It is thus underconstrained, and an additional constraint must be added to solve the inverse kinematics. In this case, a simple rule is applied to prescribe the angle of the foot with respect to the ground. That rule is given by Equation D-6,

$$alpha/alpha\_\mathrm{max} = x\_1/x\_2 \tag{D-6}$$

where *x_1, x_2,* and *alpha* are as shown in Figure 8 and *alpha_max* is the farthest the ankle can tip on its toe before exceeding its range of motion. The actual values of *alpha* and *alpha_max* can be found through simple geometry given the foothold position and orientation and the torso position. Code has been developed to compute and command the foot position relative to the ground for all cases of toe or heel rotation including both cases. Testing was completed in simulation.

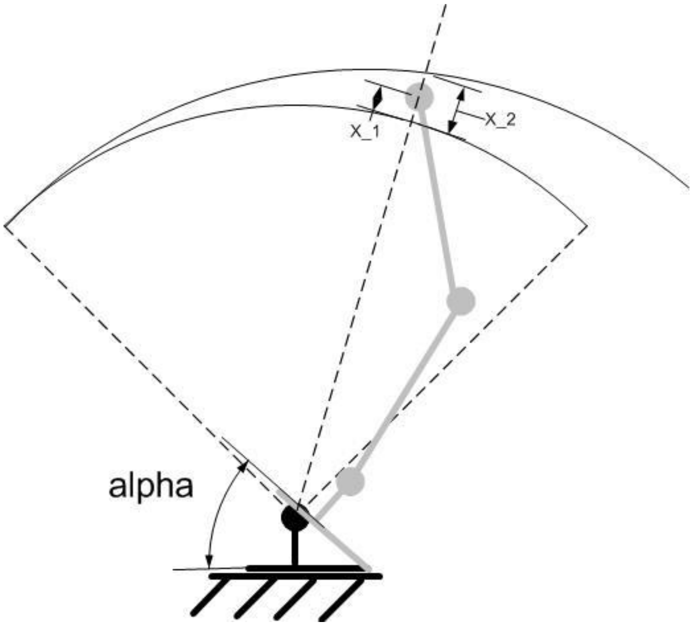

Figure D-13. Allowing the foot to rotate on edge will increase the ROM envelope, thereby enabling a long double support phase.

D - 4 Conclusion

The DARPA Robotics Challenge will drive the development of new robotic technologies that enable complex tasks in dangerous and degraded human-engineered environments. As part of that challenge, DARPA has developed a complex rough terrain locomotion course the robots must cross. Terrain includes inclines, tripping hazards, hurdles and both flat and inclined steps. The capability that is most critical to success is stepping up and down large distances.

This section presented the challenges and potential solutions to stepping up and down large distances. Power, torque, and dynamic control limitations must be solved to accomplish such steps and thus the rough terrain portion of the DRC. A novel toe-off approach using the force control framework was proposed as a solution. This approach increases the available ROM of the robot while minimizing the development time and increasing potential stability over other solutions. The approach will fit within the framework with the addition of only two unique constraints.