A Foot Placement Strategy for Robust Bipedal Gait Control

by

Derek L. Wight

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis introduces a new measure of balance for bipedal robotics called the foot placement estimator (FPE). To develop this measure, stability first is defined for a simple biped. A proof of the stability of a simple biped in a controls sense is shown to exist using classical methods for nonlinear systems. With the addition of a contact model, an analytical solution is provided to define the bounds of the region of stability. This provides the basis for the FPE which estimates where the biped must step in order to be stable. By using the FPE in combination with a state machine, complete gait cycles are created without any precalculated trajectories. This includes gait initiation and termination. The bipedal model is then advanced to include more realistic mechanical and environmental models and the FPE approach is verified in a dynamic simulation. From these results, a 5-link, point-foot robot is designed and constructed to provide the final validation that the FPE can be used to provide closed-loop gait control. In addition, this approach is shown to demonstrate significant robustness to external disturbances. Finally, the FPE is shown in experimental results to be an unprecedented estimate of where humans place their feet for walking and jumping, and for stepping in response to an external disturbance.

Acknowledgements

A big thanks to Mike MacLellan, Kathleen Denbeigh, Heather Lillico, and Jeremy Noble for all their assistance helping to get our human gait experiments up and running.

Thank you to Vicky Lawrence, our graduate secretary. You have been a pillar of reliability. Every student should be so lucky to have such a dedicated individual as yourself in their school administration.

To my committee members, Dr. Jan Huissoon, Dr. Glenn Heppler, Dr. John McPhee, and the late Dr. Aftab Patla. Your input and feedback for this project has been invaluable.

To Matt Millard, thank you for your considerable assistance and expertise. You have undoubtedly helped to shape the course of my research.

Thank you to my supervisors, Dr. Eric Kubica and Dr. David Wang of whom have provided priceless guidance and support. I would not have pursued a PhD without your confidence in me.

Finally, to my family for their love and encouragement and last, but certainly not least, to my wife, Heidi, who's love and support has seen me through my degree's many trials and tribulations.

To Heidi.

My friend.

My muse.

My love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With thousands of publications on the subject, the field of bipedal robotics is encompassed by a wide body of research, and it remains very active with a growing number of research groups. The aim of the author is to build on this foundation to produce a more robust approach to bipedal control using a foot placement strategy.

A bipedal robot is a difficult control problem. Not only is a bipedal robot nonlinear and naturally unstable in most circumstances, but the dynamics are also discontinuous with both open and closed-chain mechanical models and limited ground contact. Chapter 2 begins the thesis with a review of the current state of the art of bipedal robotics with a focus on the control systems pertaining to gait and the inevitable comparison to human motion.

At the core of the list of contributions of this research is a new control approach called the foot placement estimator (FPE) which is presented in Chapter 3. Beginning with fundamental physics, this measure of balance is derived along with a formal definition and proof of stability in a controls context for a simple biped. The FPE theory is then extended to more complex and realistic bipeds along with the introduction of a simple state machine to utilize this new measure of balance and coordinate the actions of a biped to produce a walking gait.

The use of the FPE in a real application begins with simulation in Chapter 4. The complexity of this project is revealed with all the component and environmental interactions that must be

modeled mathematically. Accurate models can only be made by characterizing physical hardware, but the hardware can only be built once the requirements are known. Engineering judgements must be made on less than complete information to resolve the circular logic. Thus, the sterile segregation of the mechanical, electrical, and software design presented in this thesis is the product of many engineering judgements to deal with the tightly woven interrelations of these subsystems.

Even in simulation, the gait patterns of the highly nonlinear, chaotic system demonstrate that a point-foot biped is highly sensitive to initial conditions and produces some variability from step to step. Thus, another contribution to the field is made in the form of a comprehensive, non-dimensionalized approach to testing the robustness of the system to external disturbances. This testing produces a probabilistic graph of the chance of a biped staying upright following a single impact at different points in the gait cycle.

The extensive design effort comes together in physical form in Chapter 5. The environmental interactions that consumed much of the development time in simulation come for free in the physical system, but now the design has to deal with noise, backlash, mechanical flexibility, and other unmodeled aspects of the system. The simple act of the physical robot walking provides the final validation needed to prove that the FPE can provide closed-loop gait control for a bipedal robot, and the robot's ability to withstand significant external disturbances demonstrates the FPE's potential advantage over other approaches.

One of the principal arguments for studying bipedal robotics is to help understand human locomotion. Researchers of kinesiology can demonstrate understanding of human motion by dissecting it to find all the underlying mechanisms and control systems that we inherit genetically and tune through experience. Researchers of bipedal robotics demonstrate understanding of human motion by constructing a biped with electronic control systems that results in behavior similar to humans. The human study conducted in Chapter 6 brings this research full circle. The study shows that the FPE measure is an unprecedented predictor of human foot placement.

This thesis concludes with a number of expansion possibilities for future work in Chapter 7 and a summary of the contributions in Chapter 8. The complete mechanical and electrical systems are included in the appendices along with the core code used for the control algorithms.

Chapter 2

Background

The bipedal form is a practical mobility solution for the structured environments created for humans, and tends to be one of the most flexible for unstructured, natural environments. Recent research even suggests that human bipedal gait was an evolutionary mechanism that resulted in a lower metabolic cost over other forms of legged locomotion [1]. However, bipedal gait is also a very difficult control problem. Not only is a biped nonlinear with large unstable regions, but the dynamics are also discontinuous with both open and closed-chain mechanical models and limited ground contact.

If the number of new publications is any indication, the field of bipedal robotics is a very active field of research. From the 1980's through to the early 1990's, only a couple dozen papers related to gait control of bipedal robots were published each year. Beginning around 1995, the field started an exponential growth with over 500 papers published on the subject in 2006 alone. The problem of making a bipedal robot walk has been solved. The current challenge is making a bipedal robot that is efficient, fast, and robust.

2.1 Applications for Bipedal Robotics

There are many potential applications for bipedal robotics, but for the most part they are still years from commercial production. The original inspiration for this work came from the author's Masters work [2] investigating powered orthoses. The majority of the powered orthoses being developed are oriented towards enhancing the abilities of healthy individuals [3] so the author's work investigated a control system aimed at assisting injured or limited mobility patients. However, this orthotic approach was not suitable for paraplegic patients who have absolutely no lower limb movement. To extend powered orthoses to these patients, the orthosis would need to be capable of walking on its own with a limited bandwidth input such as the direct neural interface [4], or a simple joystick [5]. Some groups have explored a variation of the walking orthosis idea with a walking chair [5–8], and task specific assistance such as walking up stairs [9]. These approaches may deal with some of the physical limitations of wheel chairs, but they do little to alleviate the social stigma associated with such devices. A self-walking, lower limb orthotic may provide even higher functionality, with less visual detraction. At the very least, self-walking orthotics may find an application in a spinal cord training process for paraplegics that is currently performed by therapists manually [10].

Outside of directly enabling mobility-limited individuals to walk, bipedal robots will likely find applications in environments designed for humans, in military and law enforcement, for rescue missions, and for entertainment. In environments designed for human, the use of humanoid robotics can automate tasks without replacing the vast infrastructure designed around the human form [11]. This could include maintenance or inspection of industrial plants, operation of machinery designed around the human form, assisting medical staff in hospitals or in homes for assisted living [12, 13], assisting humans at construction sites, guarding or monitoring homes, or acting as proxies for medical or technical expertise in remote locations [12].

In military or law enforcement, bipedal robots would likely find applications where remotely operated ground vehicles are already in use. Bipedal robots may have a significant mobility advantage over their wheeled counterparts for hostile encounters in urban environments. Alternatively, a humanoid robot could be used to operate vehicles interchangeably with humans

either autonomously or by teleoperation without the cost of outfitting each individual vehicle with additional sensors and actuators.

In unstructured, outdoor environments, bipeds have the ability to cover a wide range of terrain. A humanoid form can also dramatically change its size by walking sideways, crouching, crawling, and climbing, thus allowing it to fit into narrow spaces or navigate slopes that would be impossible for wheeled or multi-legged robots [14]. The legs also provide an opportunity to efficiently navigate areas with obstacles by simply stepping over them, as well as navigate "stepping stone" problems where only limited footholds are available. These abilities would give a bipedal robot a distinct advantage over other robotic forms in geographic surveys [15] or rescue missions involving scenarios such as collapsed buildings or mines where the structural stability is questionable [16].

Entertainment is the first field to see bipedal robots realized. Beginning in 1997, the RoboCup Federation[1] has been hosting robotic soccer competitions [17, 18]. The federation states that the goal of the organization is to "develop a team of fully autonomous robots that can compete against the World Soccer Champion team by the year 2050". Another competition that is developing in Japan and Korea is the RoBo-One series[2] that began in 2002. The sumo-like challenges between bipedal robots aim to "spread the joy of robots to many people". An unofficial version of these games also take place at the annual RoboGames[3] in San Francisco in addition to their numerous other bipedal robot challenges.

A more abstract realm of entertainment robotics is the "companion robots" [19]. These robots do not have a specific task, but solely act to entertain through user interaction. Following the success of Sony's robotic dog AIBO, Sony began demonstrating their humanoid QRIO, also known as SDR-II. Although they were never sold, QRIO enjoyed significant media coverage until Sony canceled their entire home robotics line in 2006. The newest media darling among the bipedal entertainment robots is the Alderbaran Humanoid [19] more commonly known as Nao[4]. It was recently selected by the RoboCup Organization to replace the Sony AIBO in the Standard Platform

[1] www.robocup.org

[2] www.robo-one.com, en.wikipedia.org/wiki/RoBo-One

[3] www.robogames.net

[4] www.aldebaran-robotics.com

League in the 2008 edition of the competition.

Finally, bipedal robotics helps to advance the understanding of the control systems involved in human locomotion [20]. There is no shortage of researchers constructing robots to emulate human gaits [21–30], however, it is the author's opinion that some of these attempts are misguided. Humans are a natural source of inspiration for control approaches, and there are some legitimate applications for emulating human gaits including assistive devices, or robots that intentionally mimic humans for aesthetics or kinesiologic study. Unfortunately, the performance of a biped is often subjectively determined based on its resemblance to human motion. The human gait evolved to achieve the lowest metabolic cost [1, 31]. Unless a robot is constructed with the same mass distribution as a human body and with actuators that share the same force, fatigue, and energy consumption characteristics, then it is unfair to expect that the optimal gait for a bipedal robot should look like the gait of a human.

2.2 Control Approaches

The control approaches used to regulate robotic bipedal gait vary widely. It is difficult to provide absolute separation between the methods because many of the approaches are hybrids of multiple control strategies, but they can be roughly categorized by their feedback mechanism.

2.2.1 Open-Loop Control

The possibility that a simple control solution exists to this complex problem is best demonstrated in bipeds that have no control system at all. Open-loop control has been used for years in toys, and it can be a reasonable approach even in more complex applications like the RoBo-One series [32] and RoboCup [18] where the robots are permitted relatively large feet, which creates a sufficient stability margin to deal with uncertainty. In general though, many open-loop controlled robots are limited to static balancing and slow motions [23, 33, 34].

Dynamic open-loop approaches are based around careful parameter selection to produce a system that is locally stable without feedback mechanisms. Passive bipeds, starting with McGeer [35,

36], demonstrated how an appropriately tuned mechanism could walk down an inclined plane in a balanced fashion with no input of energy other than the force of gravity. In fact, they are so efficient that they can even use less energy for walking than humans [37]. McGeer's work has also been extended to use passive knees [38, 39], a passive torso [37, 40], and even in a full 3D implementation [41]. The passive models have also found some use modeling the gait dynamics of humans [31].

Various works in recent years have sought to apply the principles learned from the passive mechanisms to the domain of active bipeds [31, 42–52]. The passive devices are excellent models for minimizing energy use and suggesting structural arrangements that are more conducive to walking [53], but as Anderson noted, the passive nature of the mechanical design can be easily inhibited or modified by the actuator dynamics [47]. A purely passive approach, or an active-passive approach that solely mimics an altered gravity field, is sensitive to initial and operating conditions and it can only handle minimal external disturbances [45, 54]. In active designs that go beyond simply mimicking a gravity field, Wisse noted that the active design can improve stability [51] and Kuo found that hip torques can potentially improve the overall energy efficiency of the system by reducing the step length and thus reducing the impact losses during heel strike [48].

In other open-loop approaches, Ringrose is credited with the first open-loop monopod that hops in place using a large spherical foot, but is capable of recovering from disturbances [55]. Mombaur further developed the theory to demonstrate the first open-loop actuated running [56] and flipping robots [57], and Aoi recently demonstrated open-loop walking in simulation [58]. In all cases, the basin of attraction is very small, and although the simulations demonstrated that these approaches could withstand some disturbances, none have been tested in practice. However, Mombaur points out that the purpose of these open-loop designs is to create systems that are naturally conducive to walking, which provides a solid foundation to improve the robustness with active control theory.

2.2.2 Biologically Inspired

Drawing inspiration from biological examples is another popular area of control, though the specific methods vary widely. There are numerous implementations of neural networks [46, 59–67], neural-fuzzy networks [68–70], central pattern generators [71–76] or similar oscillators [77–80], and various combinations [81–83]. Katic conducted a literature survey in the area of "intelligent control" for bipedal robots [84] which is an excellent review of this subfield. These techniques often do not use an explicit measure of balance, but rather a more abstract use of height, velocity, joint angles, step length and other parameters. This is not always the case though, as zero moment point is also a popular feedback variable (see Section 2.2.3).

In general, self-learning or adaptive control systems have the advantage that they can be applied to a system with little or no knowledge of its architecture. The disadvantage is that the process of training or parameter searching is often time consuming, specific to the hardware, and typically cannot provide any useful insight as to why it works. In addition, these methods can be difficult to train initially because until a robot starts to walk, the controllers can only evaluate if the biped succeeds or fails. There is also the problem of delayed reward as it may not be possible to evaluate if a present action is good or bad until some future time [82].

2.2.3 Zero Moment Point

Zero moment point (ZMP) is by far the most popular approach currently in the literature. Since the introduction of ZMP by Vukobratović to the robotics world in 1968 [85], it has been adopted by many researchers and applied to numerous successful implementations including the first dynamically balanced biped at Waseda University in 1986 [86, 87], the two well known robots by Honda [88–90] and Sony [91–97], and the more recent Nao [19]. Vukobratović argues that the introduction of ZMP was the driving factor behind much of the development of bipedal robotics over the last 40 years [86]. Of the thousands of papers reviewed for this thesis, approximately 15% to 20% percent were using ZMP in some way. ZMP is a measure of balance, not a control methodology, but many different control systems have been built exclusively on this measure as a feedback mechanism and are therefore grouped together here as ZMP control strategies.

Defining the Measure of Balance

One of the most basic measures of balance is the vertical projection of the center of mass (COM) also known as the center of gravity (COG). If the system moves slowly enough that the dynamic forces are negligible, then the system will be balanced if the COG lies within the base of support, or more technically, the convex hull of contact points. The problem with a COG measure is that it does not account for the dynamic forces of faster motions and it has a limited ability to deal with external disturbances. As a result, only a few systems [98–100] have been based on this measure.

A more suitable measure that takes dynamics into account is called the center of pressure (COP). The COP is basically a weighted sum of the vertical forces applied to the foot to find the location of the net applied force. Another way of describing the COP is the location where a single force vector could be applied without creating a moment about the foot, hence the zero moment point. A more rigorous definition is given in [101].

Figure 2.1 compares the center of pressure with the center of gravity. For slow motions, the COP and COG coincide. The COP and COG remain within the base of support and thus the biped remains balanced. For fast motions, however, as the COM accelerates forward, the COP moves behind the COG. Then as the COM decelerates, the COP moves in front of the COM until it hits the edge of the foot and cannot move any further forward. The COM is still within the base of support, but the COP has moved to the boundary of support, indicating that foot rotation is about to begin and a fall is imminent.

It should be noted that there is some debate in the literature about the equivalence of ZMP and COP, however, the differences are semantics. On a flat walking surface, it has been shown that the ZMP is mathematically equivalent to the COP [102], but according to Vukobratović, COP and ZMP only coincide in a dynamically balanced gait. When the gait is not dynamically balanced, the ZMP does not exist [86]. Several recent papers published by Vukobratović are dedicated to pointing out the errors and inaccuracies that have worked their way into the definition of ZMP over the years [86, 101].

Although the ZMP measure of balance can work with a wide variety of gait patterns, there

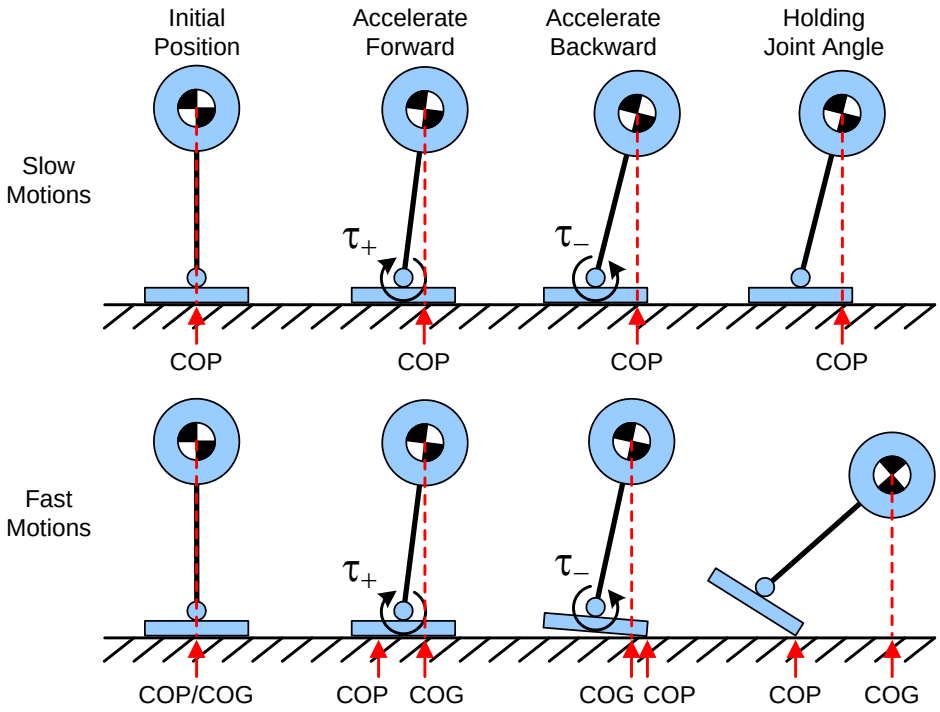


Figure 2.1: The use of the center of gravity as a measure of balance is only acceptable when the motions are slow and the dynamic forces are negligible.

are a few key limitations. First, the robot must have a full foot (as opposed to a point-foot) and at least one foot must be flat on the ground. Perhaps more importantly, if the robot becomes unbalanced and begins to rotate about the edge of its foot, then the measured ZMP cannot provide any useful information on how to recover balance so the robot must change control strategies [103] or employ emergency recovery approaches [86, 101, 102, 104]. In part, this is addressed by the foot rotation indicator [102, 103, 105, 106] which provides some indication of the severity of unbalance based on internal models. However, a control methodology that makes use of this information outside of the valid ZMP region has not yet been reported. Another suggested approach is to augment the ZMP measure with another measure called the centroidal moment pivot [102]. This regulates the angular momentum to improve the ability to reject external disturbances, but still cannot deal with the loss of foot contact.

Utilizing ZMP for Feedback

In the context of ZMP, a biped's stability is quantified by what is known as its stability margin. This is defined as the minimum distance from the location of the ZMP to the boundary of support [105]. Many systems attempt to maximize the stability margin by maintaining the ZMP near the center of the foot [107–110] while others move the ZMP forward continuously to mimic the process observed in humans [22, 107, 111, 112].

The simplest closed-loop implementations of ZMP utilize offline computations to determine the necessary joint trajectories to maintain a minimum stability margin and optimize secondary objectives such as step length or gait speed [29, 111, 113–115]. These trajectories are then fed open-loop to the local joint controllers. To close the balance control loop, a number of approaches are used including altering the torso position [104, 116–120] or modifying the ankle torques [89, 104, 110, 121, 122]. More advanced solutions involve various forms of time shifting or scaling to adjust the temporal evolution of the precalculated ZMP patterns [89, 121, 123–125], preview control [108], and scaling the trajectory geometry to alter the foot placement [89].

Another alternative is to use soft computing approaches including neural networks [126–130] and neural oscillators [72] to learn a gait pattern using the ZMP to evaluate the performance and guide the training. As with all soft computing approaches, they tend to be specific to the hardware on which they were learned, and it is difficult to gain insight as to how they work, but they have been demonstrated some successful results.

The ultimate solution to dealing with disturbances and parameter variations is realtime trajectory generation. A number of groups have made progress in this direction [22, 110, 131–137] and it appears that computing power is the only limitation before this approach is fully realized. However, all robots are subject to the bandwidth and torque limitations of their actuators. Even for a ZMP-based robot operating under realtime trajectory generation, it is possible that a disturbance of sufficient force could move the robot at a rate that exceeds its ability to maintain one foot flat on the ground. The instant that happens, the robot will lose its feedback mechanism and have no information about how to restore balance. The ability to deal with severe disturbances is the primary limitation of robots solely using ZMP for feedback.

2.2.4 Angular Momentum

A lesser known control methodology is to use the total angular momentum of the biped as a feedback variable. One of the first researchers to utilize this approach on a physical robot was Sano [138]. Although the use of angular momentum or its derivatives as a measure of balance is not widespread, there has been some renewed interest in it lately, though the best means of use is still in debate [139–143]. Goswami generalized the approach to create his zero rate of change of angular momentum (ZRAM) stability measure [140]. Popovic showed that the angular momentum of a human is tightly regulated [144] and used this knowledge to create his zero spin control point [141] and centroidal moment pivot [102]. Popovic argues that angular momentum should be regulated to a constant, though that constant should not necessarily be zero as in Goswami's approach. Popovic also noted that Goswami's ZRAM is the same as the centroidal moment pivot, though independently developed. Angular momentum is a significant component of the proposed measure shown in Chapter 3.

2.2.5 Hybrid Zero Dynamics

The zero dynamics of a nonlinear system are the internal dynamics of the system for a set of inputs u_0 and initial states x_0 such that the output $y(t) = 0$ for all t [145]. Zero dynamics can be useful to show the existence of local asymptotic stability for a nonlinear system [146]. In the pursuit of a stability analysis for bipedal robots based on zero dynamics, Westervelt incorporated an impact model to produce hybrid zero dynamics [147]. The development of this theory and several variations have been applied to the point-foot Rabbit biped to demonstrate underactuated walking [148–152] and running [153]. In contrast to ZMP, which requires the use of feet, hybrid zero dynamics does not work with feet [64]. Hybrid zero dynamics has also been criticized for the computational resources required for realtime trajectory generation and the difficulty to extend this concept to 3D [64].

2.2.6 Virtual Model Control

An approach called virtual model control developed by Pratt takes a more abstract approach to bipedal control [44, 154–156]. Rather than explicitly providing joint trajectories, the biped is connected to a "virtual granny walker" which is described in terms of spring and damper components. This is used to calculate the leg joint torques necessary to implement the walker, which indirectly regulates the balance of the biped. This can be extended to almost any virtual component including adaptive [157, 158] or learning elements [159–161]. This approach was successfully implemented on the Spring Flamingo and Spring Turkey at the MIT Leg Lab. Pratt notes that some intuition on the part of the designer is needed to determine which virtual components should be selected and where they should be used [156].

2.2.7 Foot Placement

The foot placement approach taken in this thesis is inspired from the fundamental ankle, hip, and stepping strategies that humans employ in response to external disturbances [20, 162, 163]. The ankle and hip strategies are well suited to control methods such as ZMP which inherently regulate the center of pressure within the base of support. Stepping strategies tend to be a byproduct of other control systems, but there are a few notable methodologies designed around foot placement. Foot placement can either be interpreted as finding a suitable foot placement to achieve balance, or adjusting the biped's balance to achieve a particular foot placement. The research in this thesis adopts the former approach.

Raibert, widely regarded as one of the pioneers in this field, used foot placement to control the acceleration, and indirectly, maintain the balance of his hopping robots. While his heuristic approach is substantially different from the analytical solution presented in this thesis, there are some philosophical similarities in seeking out a "neutral point" to maintain the current velocity [164]. A number researchers have built directly on Raibert's work [165] including Dunn who developed a walking variation built on the same principle [166]. Pratt's recent work operates on a similar principle of velocity control, but provides an analytical solution [167].

Unlike most other control techniques, relatively little has been done with soft computing

approaches using foot placement. Morimoto explored learning methods based on Poincaré maps that resulted in a stable gait pattern, though gait initialization patterns needed to be manually constructed [168, 169].

Townsend [170], Chew [171], van Oort [172], and Pratt [44] used foot placement for lateral control of a biped based on inverted pendulum models. In particular, Pratt's approach to lateral foot placement shares a similar mechanism to the approach used for the sagittal plane in this thesis. The work by van Oort also shares some similarities in terms of modeling the impact. Van Oort's work is one of the few approaches centered around improving the robustness to external disturbances.

The linear inverted pendulum model has been used to calculate foot placements to satisfy a desired COM trajectory [173–175]. The basic model assumes a massless leg and point-mass body with a telescoping leg to drive the mass along a "constraint line" which results in the linear equations. Kajita physically implemented this approach in 2D using inverse kinematics [174], and 3D using ZMP [176] in conjunction with ankle adjustments to help maintain the idealized model assumptions. Kudoh extended the inverted pendulum model to include angular momentum and also used it in conjunction with ZMP [177].

For stepping stone or obstacle avoidance, Kuffner explored a brute force foot placement approach by generating a search tree of the safe stepping locations using static balancing [178]. Hodgins achieved mixed results when she attempted to regulate specific foot placement using Raibert's approach by altering the forward speed, flight duration and stance duration of each step [179]. She noted that her approach could get precise foot placement for one step with the expectation that balance could be recovered on subsequent steps.

2.2.8 Other Approaches

There are a handful of other approaches including partial and full feedback linearization, or simply linearization about certain operating points, which are then driven by various controllers to track precalculated trajectories [180–187]. In another variation, Chevallereau [188–190], Doi [191, 192], and Fu [193] used the current geometric state of the robot to drive the evolution of precalculated

trajectories independent of time. Finally, there are a few other techniques that are not easily categorized [184, 194–200].

Sometimes a change of structure can make the control problem easier. Neville made use of a robot that is much more akin to an inverted pendulum than most and was able to make use of simple proportional-derivative control [201]. Figliolini avoided the dynamic balance problem altogether by placing pneumatic suction cups on the feet of his robot [202].

2.3 Discussion

The literature overwhelmingly supports closed-loop control methods for bipedal control. It is the opinion of the author that the foot placement approaches show the most promise for realtime trajectory generation and the highest flexibility beyond regular, undisturbed walking to deal with unknown terrain and external disturbances. It is also the opinion of the author that balance should be the primary objective of the control approach. Alternative objectives such as stride length and gait speed are only useful if the biped remains upright, and therefore, should be made secondary. A robot maintaining its balance is particularly important in applications for human assistance or interaction to avoid potential injuries.

Chapter 3

Control Approach

This chapter introduces the foot placement estimator (FPE) to answer the key question of where a biped's foot needs to be placed in order to restore balance. After developing this measure for a single step, Section 3.5 extends this principle to create complete gait cycles. This approach is philosophically different from most other control methods because instead of constantly trying to *maintain* a dynamic balance, this approach focuses on how to *restore* balance. This allows the control system to create a dynamically balanced gait in the presence of external disturbances without any pre-computed trajectories. The contents of this chapter have been accepted for publication [203].

3.1 The Stability of a Biped

A number of groups have managed to show bipedal stability to various degrees, or at least the existence of limit cycles, using a variety of simplified models and methods. One common approach, also used in this work, is to model the system as an inverted pendulum. Applying traditional control theory to bipeds requires revisiting the fundamental definitions of robustness and stability. Before the foot placement estimator is introduced, stability for a biped must be defined. This will form the basis for how the foot placement is computed. Doi developed a similar, though slightly

simpler, stability analysis independently of this work [191].

Consider the simple biped with massless legs and feet as shown in Figure 3.1. Assume mass m, inertia about the center of mass I_{COM}, leg length L, leg separation angle β, and the gravitational constant g. Assume the legs angles are fixed relative to the torso, $\beta > 0$, and the terrain is flat and level. If the biped is modeled as an inverted pendulum rotating about point A ($\theta_A = 0$ when leg A is vertical), then the dynamic equation using Newton's second law about point A is:

$$\circlearrowleft \sum \tau_A = I_A \ddot{\theta}_A \tag{3.1}$$

$$mgL\ sin\left(\theta_A\right) = \left(I_{\text{COM}} + mL^2\right) \ddot{\theta}_A \tag{3.2}$$

$$\ddot{\theta}_A = \frac{mgL\ sin\left(\theta_A\right)}{I_{\text{COM}} + mL^2} \tag{3.3}$$

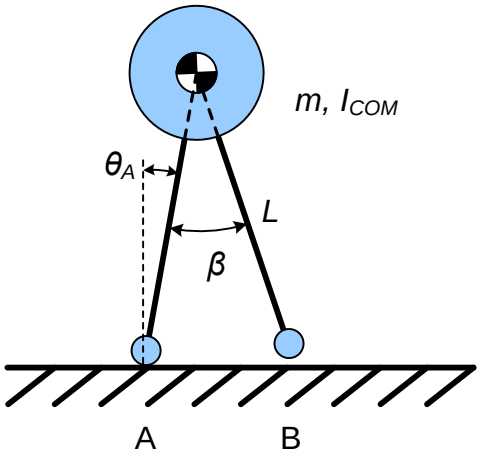


Figure 3.1: The parameters for the equations of motion for rotation about point A.

If the biped is rotating about point A clockwise, then at $\theta_A = \beta/2$, the impact of leg B occurs. The linear velocity of the COM is directly coupled to the angular velocity of the body by means of the tangential velocity v_{ti}. Assume the impact is plastic, gravity is ignored during the impact, and there is sufficient friction that the biped does not slip. Also assume that point A leaves the

ground the instant point B impacts the surface. Referring to Figure 3.2, conservation of angular momentum about point B ($H_i = H_B$) can be used to find the angular velocity immediately following impact ($\dot{\theta}_2 = \dot{\theta}_B$) as a function of the angular velocity just prior to impact ($\dot{\theta}_1 = \dot{\theta}_A$):

$$(H_i)_1 = (H_i)_2 \tag{3.4}$$

$$mLv_{t1}\ cos(\beta) + I_{\text{COM}}\dot{\theta}_1 = mLv_{t2} + I_{\text{COM}}\dot{\theta}_2 \tag{3.5}$$

$$mL(L\dot{\theta}_1)\ cos(\beta) + I_{\text{COM}}\dot{\theta}_1 = mL(L\dot{\theta}_2) + I_{\text{COM}}\dot{\theta}_2 \tag{3.6}$$

$$\dot{\theta}_2 = \frac{\left(L^2 m\ cos(\beta) + I_{\text{COM}}\right)\dot{\theta}_1}{L^2 m + I_{\text{COM}}} \tag{3.7}$$

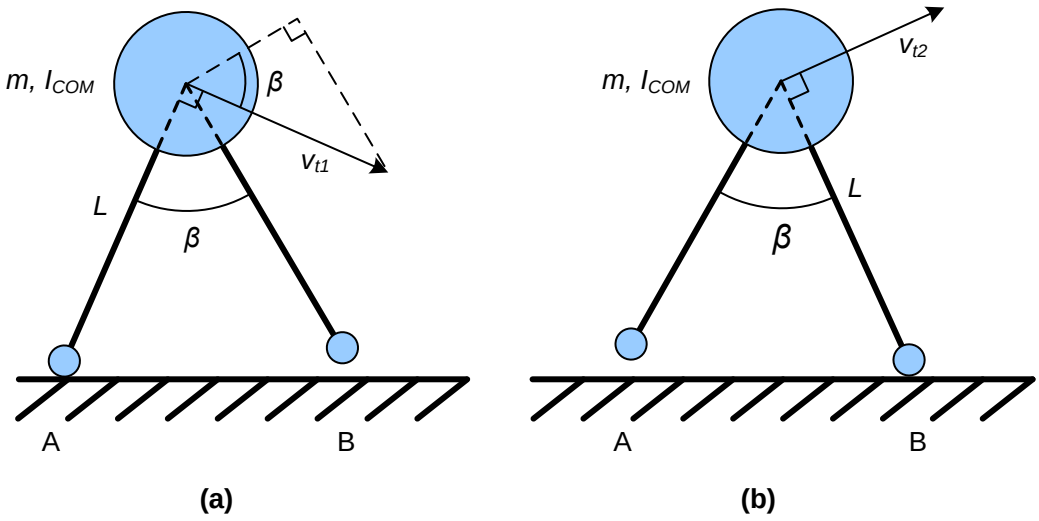


Figure 3.2: Modeling the impact using conservation of angular momentum (a) pre-impact and (b) post-impact.

The angular velocity $\dot{\theta}_2$ after the impact of the other leg is used as the initial angular velocity for a second inverted pendulum rotating about point B. The angle $\theta_B = 0$ when leg B is vertical (see

Figure 3.3). The dynamic equation about point B beginning at $\theta_B = -\beta/2$ follows the same form as Equation 3.3:

$$\sum \tau_B = I_B \ddot{\theta}_B \tag{3.8}$$

$$mgL\ sin\left(\theta_B\right) = \left(I_{\text{COM}} + mL^2\right) \ddot{\theta}_A \tag{3.9}$$

$$\ddot{\theta}_B = \frac{mgL\ sin\left(\theta_B\right)}{I_{\text{COM}} + mL^2} \tag{3.10}$$

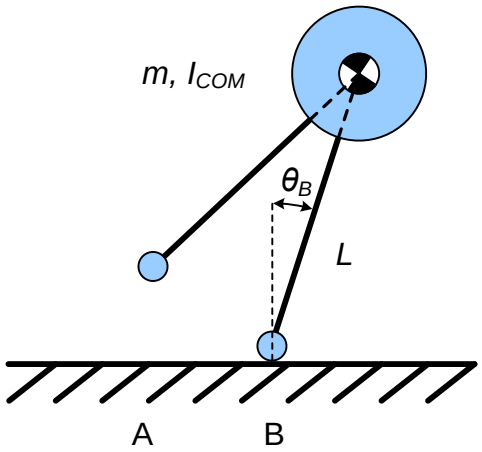


Figure 3.3: The parameters for the equations of motion for rotation about point B.

If the biped rocks back to point A again, then at $\theta_B = -\beta/2$ Equation 3.7 is again used ($H_i = H_A, \dot{\theta}_1 = \dot{\theta}_B, \dot{\theta}_2 = \dot{\theta}_A$) to calculate the angular velocity immediately following impact and the system switches back to Equation 3.3 at $\theta_A = \beta/2$.

To unify these equations into a single coordinate, let θ be the absolute angle of the biped relative to the vertical where $\theta = 0$ is the angle when both feet are in contact with the ground (see Figure 3.4). Therefore:

$$\theta_A = \theta + \frac{\beta}{2} \tag{3.11}$$

$$\theta_B = \theta - \frac{\beta}{2} \tag{3.12}$$

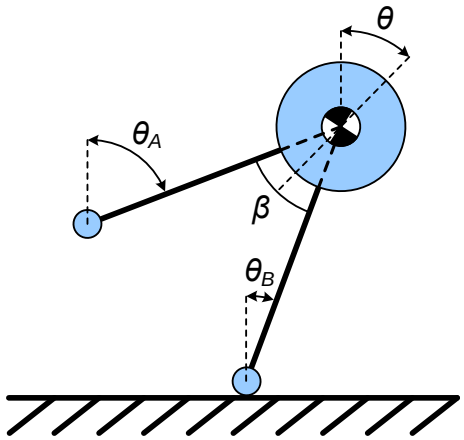


Figure 3.4: The single variable θ is used to unify the equations of motion which are functions of θ_A and θ_B.

Thus, Equations 3.3 and 3.10 can be collected into a single unified set of equations of motion for the system which will be defined as function $F(\theta, \dot{\theta})$:

$$\ddot{\theta} = \begin{cases} \frac{mgL\ sin\,(\theta + \beta/2)}{I_{\text{COM}} + mL^2} & \theta < 0 \quad (a) \\ \frac{mgL\ sin\,(\theta - \beta/2)}{I_{\text{COM}} + mL^2} & \theta > 0 \quad (b) \\ \frac{mgL\ sin\,(\beta/2)}{I_{\text{COM}} + mL^2} & \theta = 0,\ \dot{\theta} < 0 \quad (c) \\ \frac{mgL\ sin\,(-\beta/2)}{I_{\text{COM}} + mL^2} & \theta = 0,\ \dot{\theta} > 0 \quad (d) \\ 0 & \theta = 0,\ \dot{\theta} = 0 \quad (e) \end{cases} \tag{3.13}$$

$$= F(\theta, \dot{\theta}) \tag{3.14}$$

where the initial velocity following an impact (change of equation) is governed by Equation 3.7. The first two components of Equation 3.13 come directly from Equations 3.3 and 3.10. The next two components deal with the ambiguity of equation selection at $\theta = 0$ when both points A and B are in contact. The last component is also at $\theta = 0$, but if $\dot{\theta} = 0$ as well then there is no change in the angular acceleration.

3.2 Defining the Stable Region

Definition: The biped has *fallen* if $\dot{\theta} = 0$ and any point other than the feet is in contact with the walking surface.

Definition: The biped is *statically balanced* if $\dot{\theta} = 0$ and it has not fallen.

Definition: The biped is *stable* if, for a given set of initial conditions and no further input of energy to the system, the biped eventually comes to rest in a statically balanced, upright position. Once at rest, a sufficiently small, impulsive, nonzero external disturbance to the biped should result in motion that will eventually return to the same statically balanced position.

A stable equilibrium point does not strictly require that both feet be on the ground, but there must be multiple points of contact for it to hold. For instance, if the simple biped is standing still on one foot (and therefore a single point of contact), the system would be balanced, but not stable because any perturbation would move the biped away from its original stance. The only stable position for the simple biped occurs when both point feet are on the ground. A biped with a full foot (and therefore multiple points of contact) can have a stable equilibrium point on a single foot.

First, it will be proven that the standing position is a stable equilibrium point using Lyapunov's and Barbashin's theorems [204]. Putting Equation 3.14 in state space form results in:

$$\Theta = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \tag{3.15}$$

$$\dot{\Theta} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ F(\theta, \dot{\theta}) \end{bmatrix} \tag{3.16}$$

Let the total system energy U be the sum of the kinetic energy T and the potential energy V at height h (with an offset in the potential energy datum h_{datum}). Using the total system energy as

a Lyapunov candidate for $\theta < 0$:

$$\begin{aligned} U &= T + V && (3.17)\\ &= \frac{1}{2} I_A \dot{\theta}^2 + mg(h - h_{datum}) && (3.18)\\ &= \frac{1}{2}\left(I_{\text{COM}} + mL^2\right)\dot{\theta}^2 + mgL\ cos\left(\theta + \frac{\beta}{2}\right) - mgL\ cos\left(\frac{\beta}{2}\right) && (3.19)\\ \dot{U} &= \left(I_{\text{COM}} + mL^2\right)\dot{\theta}\ F(\theta, \dot{\theta}) - mgL\ sin\left(\theta + \frac{\beta}{2}\right)\dot{\theta} && (3.20) \end{aligned}$$

Using the first component of Equation 3.13 for $F(\theta, \dot{\theta})$ in the defined range of $\theta < 0$:

$$\begin{aligned} \dot{U} &= \frac{\left(I_{\text{COM}} + mL^2\right)\dot{\theta}\ mgL\ sin\left(\theta + \frac{\beta}{2}\right)}{I_{\text{COM}} + mL^2} - mgL\ sin\left(\theta + \frac{\beta}{2}\right)\dot{\theta} && (3.21)\\ &= 0 && (3.22) \end{aligned}$$

Likewise for $\theta > 0$, $\dot{U} = 0$. Essentially, this confirms that conservation of energy applies to the dynamic equations $F(\theta, \dot{\theta})$ outside of the impact conditions because the energy is constant. However, recall that the transitions from Equation 3.13a to 3.13d, and from 3.13b to 3.13c are given by the impact condition described by Equation 3.7. Therefore, the change in energy from pre-impact to post-impact will be:

$$\begin{aligned} U_2 - U_1 &= T_2 - T_1 && (3.23)\\ &= \frac{1}{2} I {\dot{\theta}_2}^2 - \frac{1}{2} I {\dot{\theta}_1}^2 && (3.24)\\ &= \frac{1}{2} I {\dot{\theta}_1}^2 \left[\left(\frac{L^2 m\ cos(\beta) + I_{\text{COM}}}{L^2 m + I_{\text{COM}}}\right)^2 - 1\right] && (3.25)\\ &< 0 && (3.26) \end{aligned}$$

It has now been shown that $\dot{U} \leq 0$ everywhere in the state space. If the Lyapunov candidates are limited to the range of $-\beta/2 < \theta < \beta/2$, then U is positive definite. From Equation 3.26, the set $T = \{\Theta | \dot{\theta} \neq 0, \theta = 0\}$ has $\dot{U}$ strictly less than 0. Any trajectory beginning in the set $S = \{\Theta | \dot{U}(\Theta) = 0, U < mgL[1 - cos(\beta/2)]\}$ must pass through T except at the equilibrium point.

Since the equilibrium point is the largest invariant set S, it can be concluded using Barbashin's theorem [204] that $\theta = 0$ is an asymptotically stable equilibrium point in the sense of Lyapunov. This is graphically illustrated in the phase plot of Section 3.3.3. The exact boundaries of this local stability will now be defined.

3.2.1 Stable Region 1

If the initial angle θ_0 of the biped is such that $-\beta/2 < \theta_0 < \beta/2$ and the total system energy is less than the peak possible potential energy of the system, then the biped will be stable. Figure 3.5 shows the full range of motion for the center of mass. Under the conditions of stable region 1, the biped will not be able to escape the energy well (the white region) between the stable standing position at $\theta = 0$ and the peak potential energy at h_{peak}. Mathematically, if:

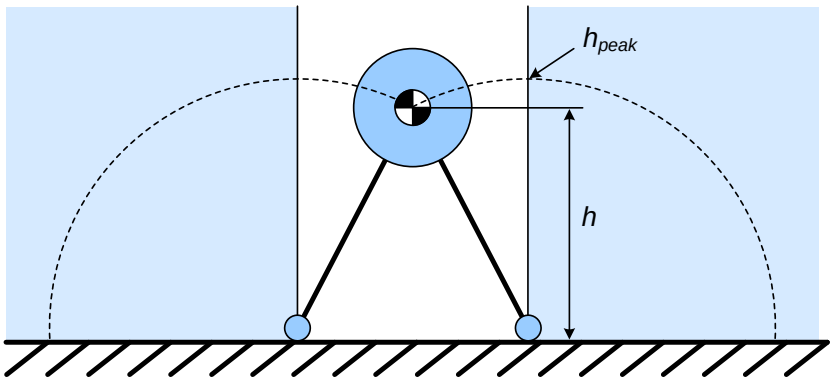


Figure 3.5: The dotted line traces out the path of the COM. Stable region 1 defines the necessary conditions to keep the COM within the energy well bounded by the shaded regions.

$$U_0 = T_0 + V_0 < mgh_{peak} \tag{3.27}$$

$$mgL\ cos(|\theta_0| - \beta/2) + \frac{1}{2}(I_{\text{COM}} + mL^2)\dot{\theta}_0{}^2 < mgL \tag{3.28}$$

$$\dot{\theta}_0{}^2 < \frac{2mgL\,(1 - cos\,(|\theta_0| - \beta/2))}{I_{\text{COM}} + mL^2} \tag{3.29}$$

then θ will have a decaying orbit towards the equilibrium point $\Theta = 0$.

3.2.2 Stable Region 2

If the energy loss due to impacts is taken into account, then the biped can start with a slightly higher energy than is allowed by stable region 1 as long as the initial velocity is towards an impact ($\theta = 0$). For $-\beta/2 < \theta_0 < 0$ and $\dot{\theta}_0 \geq 0$, after impact:

$$T_2 + V_2 < mgL \tag{3.30}$$

$$mgL\ cos(\beta/2) + \frac{1}{2}(I_{\text{COM}} + mL^2)\dot{\theta}_2^2 < mgL \tag{3.31}$$

The angular velocity $\dot{\theta}_2$ (just after impact) is known in terms of $\dot{\theta}_1$ (just before impact) from Equation 3.7:

$$\frac{\frac{1}{2}(I_{\text{COM}} + mL^2)(mL^2 cos(\beta) + I_{\text{COM}})^2\dot{\theta}_1^2}{(I_{\text{COM}} + mL^2)^2} + mgL\ cos(\beta/2) < mgL \tag{3.32}$$

$$\frac{(mL^2 cos(\beta) + I_{\text{COM}})^2\dot{\theta}_1^2}{2(I_{\text{COM}} + mL^2)} < mgL[1 - cos(\beta/2)] \tag{3.33}$$

$$\dot{\theta}_1^2 < \frac{2mgL[1 - cos(\beta/2)](I_{\text{COM}} + mL^2)}{(mL^2 cos(\beta) + I_{\text{COM}})^2} \tag{3.34}$$

From the inverted pendulum model, conservation of energy can be used to find $\dot{\theta}_1$ (prior to impact) in terms of the initial angular velocity $\dot{\theta}_0$:

$$T_0 + V_0 = T_1 + V_1 \tag{3.35}$$

$$\frac{1}{2}(I_{\text{COM}} + mL^2)\dot{\theta}_0^2 + mgL\ cos(\theta_0 + \beta/2) = \frac{1}{2}(I_{\text{COM}} + mL^2)\dot{\theta}_1^2 + mgL\ cos(\beta/2) \tag{3.36}$$

$$\dot{\theta}_1^2 = \frac{2mgL\left[cos(\theta_0 + \beta/2) - cos(\beta/2)\right] + (I_{\text{COM}} + mL^2)\dot{\theta}_0^2}{I_{\text{COM}} + mL^2} \tag{3.37}$$

Substituting into Equation 3.34, the maximum $\dot{\theta}_0$ can be solved in terms of θ_0:

$$\frac{2mgL\left[cos(\theta_0+\beta/2)-cos(\beta/2)\right]+(I_{\text{COM}}+mL^2)\dot{\theta}_0^2}{I_{\text{COM}}+mL^2} < \frac{2mgL[1-cos(\beta/2)](I_{\text{COM}}+mL^2)}{(mL^2cos(\beta)+I_{\text{COM}})^2} \tag{3.38}$$

$$(I_{\text{COM}}+mL^2)\dot{\theta}_0^2 < \frac{2mgL[1-cos(\beta/2)](I_{\text{COM}}+mL^2)^2}{(mL^2cos(\beta)+I_{\text{COM}})^2} - 2mgL\left[cos(\theta_0+\beta/2)-cos(\beta/2)\right] \tag{3.39}$$

$$0 \le \dot{\theta}_0^2 < \frac{2mgL}{I_{\text{COM}}+mL^2}\left[\frac{[1-cos(\beta/2)](I_{\text{COM}}+mL^2)^2}{(mL^2cos(\beta)+I_{\text{COM}})^2} - cos(\theta_0+\beta/2)-cos(\beta/2)\right] \tag{3.40}$$

Likewise for the range of $0 < \theta_0 < \beta/2$ and $\dot{\theta}_0 \le 0$:

$$0 \ge \dot{\theta}_0^2 > \frac{2mgL}{I_{\text{COM}}+mL^2}\left[\frac{[1-cos(\beta/2)](I_{\text{COM}}+mL^2)^2}{(mL^2cos(\beta)+I_{\text{COM}})^2} - cos(\theta_0-\beta/2)-cos(\beta/2)\right] \tag{3.41}$$

The results of Equations 3.40 and 3.41 still constrain the initial position of the biped to be between the shaded regions in Figure 3.5, but the initial energy level can be higher than in stable region 1 as long as initial motion of the biped is directed towards an impact.

3.2.3 Stable Region 3

Finally to extend the defined stable region outside of $-\beta/2 < \theta < \beta/2$, a minimum energy constraint must be added. Referring to Figure 3.5 again, the biped is now starting in one of the shaded regions. There must be enough energy to rise over the peak potential energy and get into the stable region, but not so much as to pass right through and fall out the other side. The upper energy bound is the same as Equations 3.40 and 3.41. The lower, minimum energy bound for $\theta_0 \le -\beta/2$:

$$U_0 = T_0 + V_0 > mgL \tag{3.42}$$

$$mgL\ cos(\theta_0+\beta/2) + \frac{1}{2}(I_{\text{COM}}+mL^2)\dot{\theta}_0^2 > mgL \tag{3.43}$$

$$\frac{1}{2}(I_{\text{COM}} + mL^2)\dot{\theta}_0^2 > mgL[1 - cos(\theta_0 + \beta/2)] \tag{3.44}$$

$$\dot{\theta}_0^2 > \frac{2mgL[1 - cos(\theta_0 + \beta/2)]}{I_{\text{COM}} + mL^2} \tag{3.45}$$

Combining with the upper bound of Equation 3.40:

$$\frac{2mgL[1 - cos(\theta_0 + \beta/2)]}{I_{\text{COM}} + mL^2} < \dot{\theta}_0^2 <$$
$$\frac{2mgL}{I_{\text{COM}} + mL^2}\left[\frac{[1 - cos(\beta/2)](I_{\text{COM}} + mL^2)^2}{(mL^2 cos(\beta) + I_{\text{COM}})^2} - cos(\theta_0 + \beta/2) - cos(\beta/2)\right] \tag{3.46}$$

Similarly, for the range of $\theta_0 \geq \beta/2$:

$$\frac{2mgL[1 - cos(\theta_0 - \beta/2)]}{I_{\text{COM}} + mL^2} < \dot{\theta}_0^2 <$$
$$\frac{2mgL}{I_{\text{COM}} + mL^2}\left[\frac{[1 - cos(\beta/2)](I_{\text{COM}} + mL^2)^2}{(mL^2 cos(\beta) + I_{\text{COM}})^2} - cos(\theta_0 - \beta/2) - cos(\beta/2)\right] \tag{3.47}$$

Equations 3.46 and 3.47 now allow the biped to start in the shaded regions of Figure 3.5. The lower energy bound ensures there is enough energy to enter the stable region while the upper energy bound ensures that the biped does not leave the stable region.

3.3 A More Realistic Biped and the Region of Validity

The stability results of Section 3.2 can be extended to a more general bipedal architecture. An arbitrary biped configuration, such as the one shown in Figure 3.6a, can be simplified to the form shown in Figure 3.6b. The arbitrary torso is bound by a circle with the origin at the center of mass (COM). The arbitrary leg configuration is replaced by two straight legs originating from the COM and terminating at the same points on the ground as the original biped. If it is assumed that $L_1 = L_2 = L$, $\beta > 0$, and the joints remain immobile, then the previous stability results can be applied to this arbitrary biped.

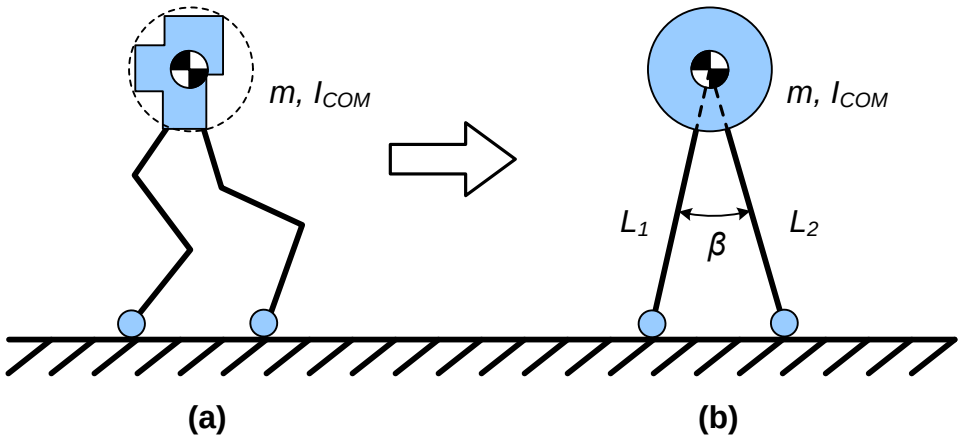


Figure 3.6: Simplifying an arbitrary biped configuration.

3.3.1 Minimum Normal Force

The dynamic equations for the simple biped were derived as if the standing foot was pinned to the ground. In a real biped, the standing foot will lift off the ground when the normal force F_n of the pinned model is less than or equal to zero. Therefore, the dynamic equations are only valid when the normal force on the foot is positive. Referring to the free body diagram in Figure 3.7, consider the biped as a simple inverted pendulum rotating about point A again.

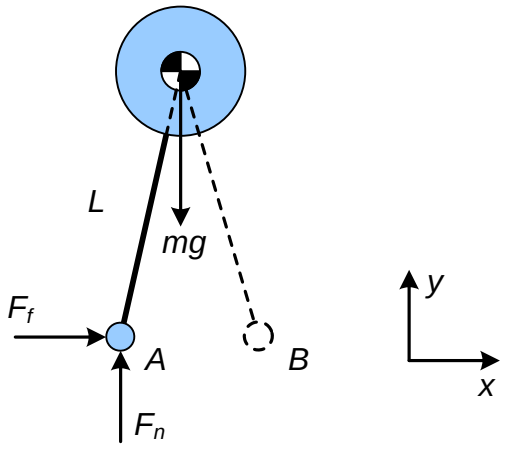


Figure 3.7: The free-body diagram when Point A is in contact with the surface.

The kinematic equations for the body at the COM:

$$y_{\text{COM}} = L\ cos\left(\theta + \frac{\beta}{2}\right) \tag{3.48}$$

$$\dot{y}_{\text{COM}} = -L\ sin\left(\theta + \frac{\beta}{2}\right)\dot{\theta} \tag{3.49}$$

$$\ddot{y}_{\text{COM}} = -L\ cos\left(\theta + \frac{\beta}{2}\right)\dot{\theta}^2 - L\ sin\left(\theta + \frac{\beta}{2}\right)\ddot{\theta} \tag{3.50}$$

Dynamic equations:

$$\sum F_y = m\ddot{y}_{\text{COM}} \tag{3.51}$$

$$F_n - mg\ cos\left(\theta + \frac{\beta}{2}\right) = m\ddot{y}_{\text{COM}} \tag{3.52}$$

Let $F_n > 0$:

$$m\ddot{y}_{\text{COM}} + mg\ cos\left(\theta + \frac{\beta}{2}\right) > 0 \tag{3.53}$$

$$\left(g - L\dot{\theta}^2\right)\ cos\left(\theta + \frac{\beta}{2}\right) - L\ sin\left(\theta + \frac{\beta}{2}\right)\ddot{\theta} > 0 \tag{3.54}$$

Substitute Equation 3.3 for $\ddot{\theta}$ and solve for $\dot{\theta}^2$:

$$\left(g - L\dot{\theta}^2\right)\ cos\left(\theta + \frac{\beta}{2}\right) - \frac{mgL^2\ sin^2\left(\theta + \beta/2\right)}{I_{\text{COM}} + mL^2} > 0 \tag{3.55}$$

$$\dot{\theta}^2 < \frac{1}{L}\left[g - \frac{mgL^2\ sin^2\left(\theta + \beta/2\right)}{\left(I_{\text{COM}} + mL^2\right)\ cos(\theta + \beta/2)}\right] \quad (for\ \theta \leq 0) \tag{3.56}$$

Likewise for point B:

$$\dot{\theta}^2 < \frac{1}{L}\left[g - \frac{mgL^2\ sin^2\left(\theta - \beta/2\right)}{\left(I_{\text{COM}} + mL^2\right)\ cos(\theta - \beta/2)}\right] \quad (for\ \theta \geq 0) \tag{3.57}$$

3.3.2 Minimum Friction Force

Not only are the equations of motion restricted to the normal force constraints of Equations 3.56 and 3.57, but the validity is also limited to the regions where the friction force F_f does not exceed the maximum friction force dictated by the coefficient of static friction μ_s. Refer to the free body diagram in Figure 3.7 for rotation about point A.

Kinematics:

$$x_{\text{COM}} = L\ sin\left(\theta + \frac{\beta}{2}\right) \tag{3.58}$$

$$\dot{x}_{\text{COM}} = L\ cos\left(\theta + \frac{\beta}{2}\right)\dot{\theta} \tag{3.59}$$

$$\ddot{x}_{\text{COM}} = -L\ sin\left(\theta + \frac{\beta}{2}\right)\dot{\theta}^2 + L\ cos\left(\theta + \frac{\beta}{2}\right)\ddot{\theta} \tag{3.60}$$

Dynamics:

$$\sum F_x = m\ddot{x}_{\text{COM}} \tag{3.61}$$

$$F_f = -mL\ sin\left(\theta + \frac{\beta}{2}\right)\dot{\theta}^2 + mL\ cos\left(\theta + \frac{\beta}{2}\right)\ddot{\theta} \tag{3.62}$$

For $F_f \geq 0$, the dynamic equations are valid when $F_f < F_n\mu_s$. Substituting in for F_f from Equation 3.62 and F_n from Equation 3.52:

$$-L\ sin\left(\theta + \frac{\beta}{2}\right)\dot{\theta}^2 + L\ cos\left(\theta + \frac{\beta}{2}\right)\ddot{\theta} < \left(g - L\dot{\theta}^2\right) cos\left(\theta + \frac{\beta}{2}\right)\mu_s - L\ sin\left(\theta + \frac{\beta}{2}\right)\ddot{\theta}\mu_s \tag{3.63}$$

$$\dot{\theta}^2 L\left[\mu_s\ cos\left(\theta + \frac{\beta}{2}\right) - sin\left(\theta + \frac{\beta}{2}\right)\right] < \left(g\mu_s - L\ddot{\theta}\right) cos\left(\theta + \frac{\beta}{2}\right) - L\ddot{\theta}\mu_s\ sin\left(\theta + \frac{\beta}{2}\right) \tag{3.64}$$

$$\dot{\theta}^2 < \frac{\left(g\mu_s - L\ddot{\theta}\right)\ cos\left(\theta + \beta/2\right) - L\ddot{\theta}\mu_s\ sin\left(\theta + \beta/2\right)}{L\left[\mu_s\ cos\left(\theta + \beta/2\right) - sin\left(\theta + \beta/2\right)\right]} \qquad (for\ F_f \geq 0) \tag{3.65}$$

Likewise, for $F_f \leq 0$, the dynamic equations are valid when $-F_f < F_n\mu_s$:

$$\dot{\theta}^2 < \frac{\left(g\mu_s + L\ddot{\theta}\right)\ cos\left(\theta + \beta/2\right) - L\ddot{\theta}\mu_s\ sin\left(\theta + \beta/2\right)}{L\left[\mu_s\ cos\left(\theta + \beta/2\right) + sin\left(\theta + \beta/2\right)\right]} \qquad (for\ F_f \leq 0) \tag{3.66}$$

Similarly, for rotation about point B:

$$\dot{\theta}^2 < \frac{\left(g\mu_s - L\ddot{\theta}\right)\ cos\left(\theta - \beta/2\right) - L\ddot{\theta}\mu_s\ sin\left(\theta - \beta/2\right)}{L\left[\mu_s\ cos\left(\theta - \beta/2\right) - sin\left(\theta - \beta/2\right)\right]} \qquad (for\ F_f \geq 0) \qquad (3.67)$$

$$\dot{\theta}^2 < \frac{\left(g\mu_s + L\ddot{\theta}\right)\ cos\left(\theta - \beta/2\right) - L\ddot{\theta}\mu_s\ sin\left(\theta - \beta/2\right)}{L\left[\mu_s\ cos\left(\theta - \beta/2\right) + sin\left(\theta - \beta/2\right)\right]} \qquad (for\ F_f \leq 0) \qquad (3.68)$$

3.3.3 The Intersecting Regions of Validity

The complete region of validity for Equation 3.14 is the intersection of the areas defined by $F_n > 0$ and $|F_f| < F_n\mu_s$. For $\theta \leq 0$, the valid region is defined by Equations 3.56, 3.65, and 3.66. For $\theta \geq 0$, the valid region is defined by Equations 3.57, 3.67, and 3.68. Note that when a discontinuity exists in one or more of the equations, the valid region is the single set where all equations are closed and bounded.

To visually illustrate the interaction of these equations, the phase portrait of the biped to be introduced in Section 4.1 is shown in Figure 3.8. Given an initial condition starting in the stable regions (the union of the stable regions is shown by the dashed line), the biped will eventually come to rest standing upright on both feet. This can only be guaranteed as long as the entire trajectory stays within the white valid region (the single closed set that is the intersection of the regions of validity). Outside of the valid region, the motion of the biped will no longer be governed by Equation 3.14.

3.4 Deriving the Foot Placement Estimator

The work in Section 3.1 addressed whether a given foot placement would be stable. The foot placement estimator (FPE) essentially rephrases the question. Given the current conditions of the biped, where does the foot need to be placed such that the biped will be stable? Assume that the simple biped shown in Figure 3.1 is still being used, except B (the swing leg) is now free to move when it is not in contact with the ground. Since the swing leg has no mass, its movement

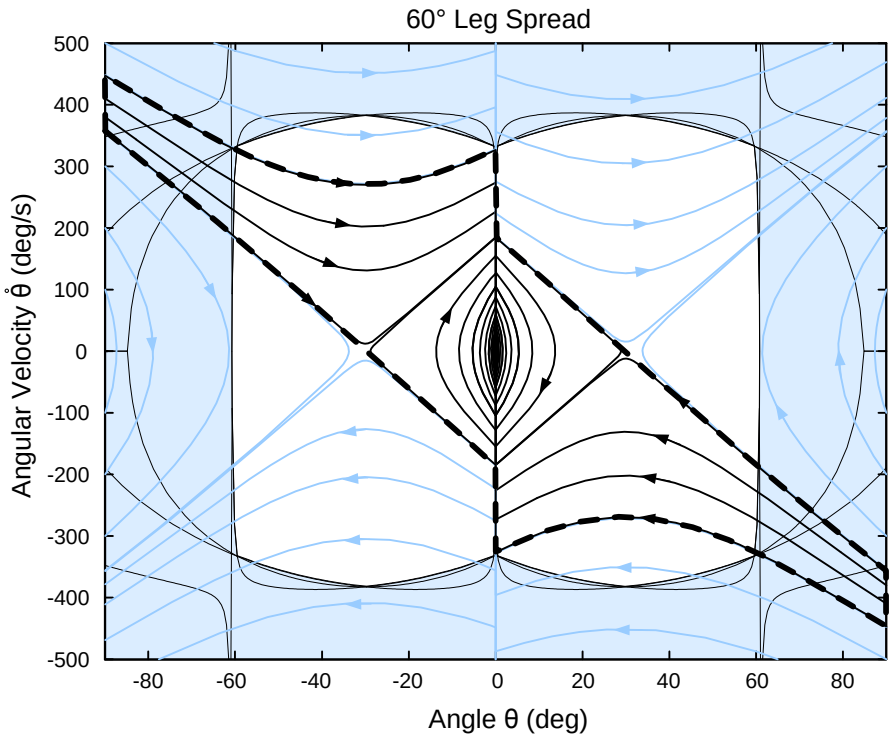


Figure 3.8: A phase portrait of a simple biped using the parameters of the biped in Section 4.1 and $\beta = 60°$. The dashed line is the union of the three stable regions. The white region is the intersection of the regions of validity.

has no effect on the dynamics of the system.

Consider the three cases shown in Figure 3.9. In Figure 3.9a, the biped takes a very short step. The kinetic energy after impact exceeds the peak potential energy so the biped keeps traveling forward and falls down. In Figure 3.9b, the biped takes a very large step. The kinetic energy after impact is less than the peak potential energy so the biped remains stable. In Figure 3.9c, the biped steps at a location in between the previous two cases and the kinetic energy after impact is exactly equal to the peak potential energy, so the biped comes to rest at a balanced (but unstable) position. This balanced step location is what will be referred to as the foot placement estimator.

Pratt's work in [44] was the source of inspiration for this approach. He used a simpler form of this approach for the frontal plane control of his biped. The approach presented here generalizes

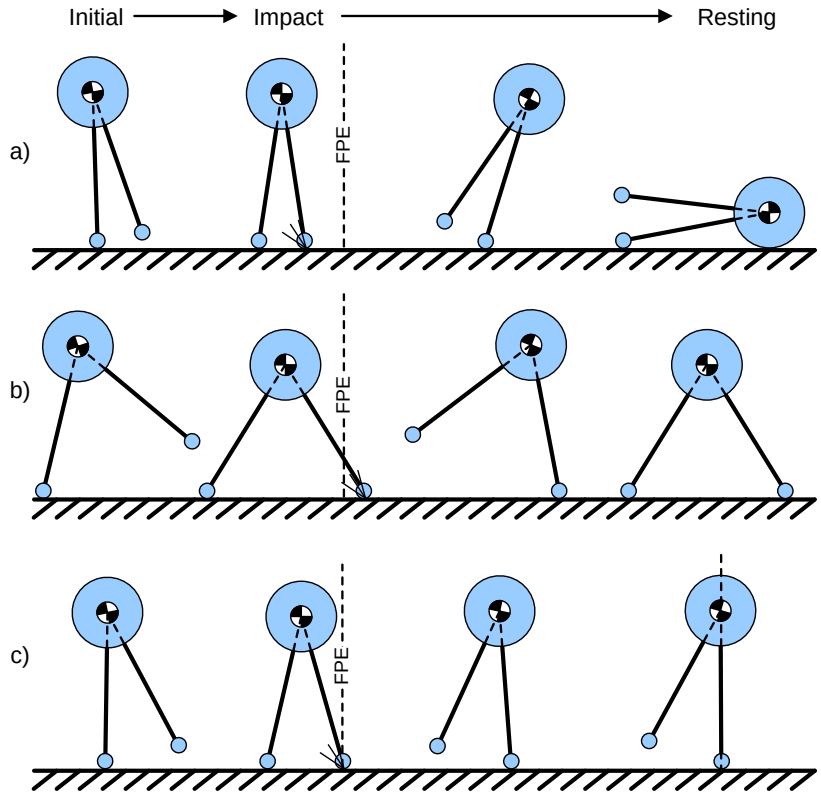


Figure 3.9: A simplified biped stepping relative to the FPE. (a) Stepping closer than the FPE results in falling forward. (b) Stepping further than the FPE causes the biped to fall back onto the swing leg. (c) Stepping precisely at the FPE will perfectly balance the COM above the standing foot.

his technique and extends it to a complete gait cycle in the sagittal plane. Recently, Pratt also developed a philosophically similar methodology [205, 206] independently of this work. Although similar to Pratt's one-step capture point, the FPE includes the rotational energy and the losses due to impact for a more accurate foot placement estimate, and the FPE does not require that the biped be in contact with the ground prior to impact.

To develop the FPE, the conservation of angular momentum equation that describes the impacts must be revisited. Referring to Figure 3.10, it was previously assumed that v_{t1} and $\dot{\theta}_1$ were coupled prior to impact, but now that restriction is removed such that v_{t1} is described in terms of the linear velocity components v_x and v_y. This allows the biped to be in free-flight

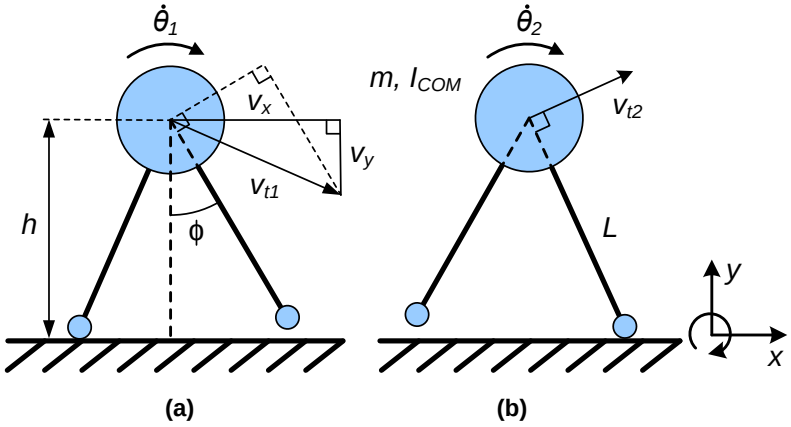


Figure 3.10: (a) Given the parameters immediately preceding impact, conservation of angular momentum is used to predict (b) the velocities immediately after impact.

prior to impact. In addition, instead of the leg separation angle β, the reference angle ϕ is used to describe the absolute angle of the swing leg relative to the vertical as shown in Figure 3.10 ($\phi = \beta/2$ when both feet are in contact with the ground). Integrating the free-flight condition and reference angle into the conservation of angular momentum equation:

$$(H_B)_1 = (H_B)_2 \tag{3.69}$$

$$mL(v_x\ cos\phi + v_y\ sin\phi) + I_{\text{COM}}\dot{\theta}_1 = (mL^2 + I_{\text{COM}})\dot{\theta}_2 \tag{3.70}$$

Let the leg length be described in terms of the current height:

$$L = \frac{h}{cos\phi} \tag{3.71}$$

This removes the necessity of knowing L explicitly, thus allowing for variable leg lengths. Substituting into Equation 3.70 and solving for $\dot{\theta}_2$:

$$\dot{\theta}_2 = \frac{mh(v_x\ cos\phi + v_y\ sin\phi)cos\ \phi + I_{\text{COM}}\dot{\theta}_1 cos^2\phi}{mh^2 + I_{\text{COM}}cos^2\phi} \tag{3.72}$$

The location of the FPE is at the angle ϕ when the total system energy after impact is equal to the peak potential energy. As before, the impact is assumed to be plastic so v_{t2} and $\dot{\theta}_2$ are coupled:

$$T_2 + V_2 = mgh_{peak} \tag{3.73}$$

$$\frac{1}{2}(I_{\text{COM}} + mL^2){\dot{\theta}_2}^2 + mgL\ cos(\phi) = mgL \tag{3.74}$$

$$(I_{\text{COM}}cos^2\phi + mh^2){\dot{\theta}_2}^2 + 2mgh\ cos\ \phi(cos\ \phi - 1) = 0 \tag{3.75}$$

Substituting Equation 3.72 to describe $\dot{\theta}_2$ in terms of pre-impact conditions results in the FPE equation:

$$\frac{\left[mh(v_x\ cos\ \phi + v_y\ sin\ \phi)cos\ \phi + I_{\text{COM}}\dot{\theta}_1 cos^2\phi\right]^2}{mh^2 + I_{\text{COM}}cos^2\phi} + 2mgh\ cos\ \phi(cos\ \phi - 1) = 0 \tag{3.76}$$

The projection from the COM to the walking surface at angle ϕ relative to the vertical indicates the location of the FPE (see Figure 3.11). Given the current linear and angular velocity of the COM $(v_x, v_y, \dot{\theta}_1)$, and the height of the COM (h), Equation 3.76 can be solved for the angle ϕ using numerical methods for nonlinear equations. The FPE location indicates where the foot

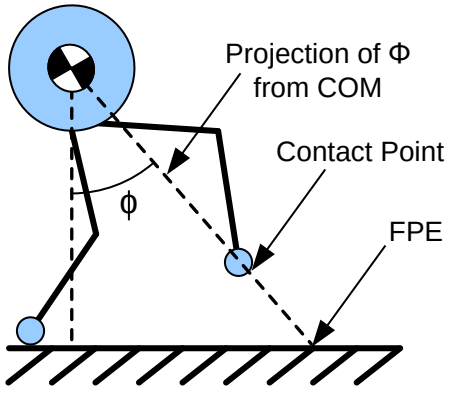


Figure 3.11: The projection of the angle ϕ from the COM to the walking surface is the location of the FPE. This projection is used as a tracking reference for the swing foot until impact occurs.

would need to be placed to balance the biped *if impact were to occur in the next instant.* However, since the legs have a finite length, the angle ϕ is calculated at a regular sample period and is used as a tracking reference so that when impact does occur, the foot is properly positioned. Note that in general, two solutions exist to Equation 3.76 in the range of $-90° < \phi < 90°$. The appropriate solution can be selected based on the signs of v_x and $\dot{\theta}_1$.

Let us examine the application of the FPE to the specific case of the simple biped with a walking gait (at least one foot is on the ground at all times). If both legs are assumed to be length L then at the point of impact $\phi = \beta/2$. Since the biped is walking, $\dot{\theta}_2$ reverts back to Equation 3.7, and the derivation of the FPE takes the same form of the derivation of stable region 2 in Section 3.2.2. Thus, the derivation of stable region 2 is a special case of the FPE equation except that $\dot{\theta}_1$ is being solved for instead of β (or ϕ).

3.4.1 Beyond the Stick Man

So far, it has been assumed that the inertial properties remain constant for the idealized biped model. This is reasonable if the legs have no mass, but in a physical biped, motion of the legs will cause changes in the inertial properties, introduce additional dynamics and, if actively moved, add or remove energy from the system. In addition, unless the horizontal velocity difference between the foot and the ground is zero, there will be a short period of slippage during which some additional energy will be lost. The combination of all these factors is why this measure of balance is called an *estimator*.

As will be shown in the coming chapters, the estimate is actually very good. As the swing leg converges with the FPE position and the joint velocities go to zero, the unmodeled dynamics introduced by the motion of the swing leg become insignificant and the inertia of the system approaches a constant. If the biped achieves a state with no internal motions just prior to impact, then the only inaccuracy of the FPE will be due to slipping immediately following impact. (This could be compensated for by some swing foot retraction prior to impact [207].)

A particular detail to note is the calculation of $\dot{\theta}_1$. In multibody dynamics, $\dot{\theta}_1$ is the average angular velocity of all the limb segments [208]. This is calculated as a weighted sum (similar to

calculating the COM) of the inertia of each body segment I_k about the total COM:

$$\dot{\theta}_1 = \dot{\theta}_{Avg} = \frac{\sum_{k=1}^{n} I_k \dot{\theta}_k}{\sum_{k=1}^{n} I_k} \tag{3.77}$$

One further issue that has not yet been addressed is the physical limits of the biped workspace. The FPE is the location where the biped must step in order to restore balance in a single step. If the physical limits of a particular biped architecture prevent it from achieving that position, then the biped can potentially reach a balanced position over multiple steps as long as the energy added to the system from taking subsequent steps is less than the energy dissipated during impact (so the net change of the system energy is negative).

3.5 Every Gait Cycle Begins with a Single Step

The derivation of the FPE in Section 3.4 addresses the question of where to place the foot in order to restore balance given initial conditions where the robot is already in motion. If the robot always steps slightly further than the FPE location, then each step will be stable as per the definition in Section 3.2. To extend this to a complete gait cycle, the trailing foot just needs to push with sufficient force that the FPE moves in front of the leading foot, causing the biped to leave the current stable region. This forces the biped to take a step at the new FPE location in order to restore its balance.

A state machine is used to break the gait cycle into smaller, manageable pieces. Similar approaches have been used in [44, 164, 185]. The state machine shown in Figure 3.12 is a simplified version of what is implemented in Chapters 4 and 5 for a simple 5-link planar biped (torso, 2 thighs, 2 shanks, point-feet, rotary joints). Since this biped only has point-feet, the system is constantly under-actuated, and therefore the most difficult class of biped to control [188], but it is a perfect example of the capability of the FPE. (A more practical bipedal implementation would utilize a foot not only for the added stability control, but also for a more efficient pushing mechanism.)

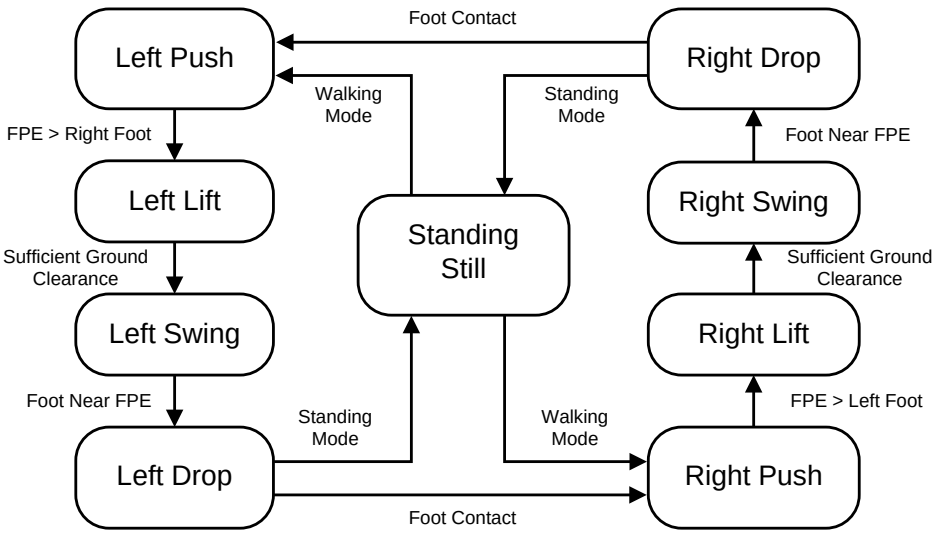


Figure 3.12: Overview of the state machine coordinating the actions of the robot.

The function of each state is as follows:

1. **Standing Still** - All joint controllers are held at the current references. The system transitions to a push state on a user command to start the gait cycle.

2. **Push Off** - The standing knee and hip are held fixed. The pushing leg is extended at a controlled rate. The state transitions when the FPE has moved in front of the standing foot.

3. **Lift Foot** - The pushing leg is retracted. The state transitions when the foot has sufficient clearance to swing through.

4. **Swing** - The standing hip is used to regulate the torso to an upright position. The swing knee adjusts the leg length to maintain the foot at a constant height above the ground. The swing hip moves the swing foot to the position that intersects the line from the COM to the FPE position on the ground (see Figure 3.11). The state transitions when the swing leg is close to the reference angle.

5. **Foot Drop** - The standing hip continues to regulate the torso upright. The swing hip continues to maintain the swing foot along the line from the COM to the FPE position on the ground. The swing knee extends the leg to its normal standing length. The state transitions when contact of the swing leg is made with the ground. If the user issues a stop gait command then the state will change to the standing state, otherwise it will transition to the push state of the opposite foot.

An elegant feature of this control system is that gait initiation and termination are already incorporated. If every step is slightly further than the FPE location, then every step is stable. Stopping the gait cycle is simply a matter of not pushing and remaining in the standing state. No further action is required by the control system other than for the individual joint controllers to maintain their current reference position. More advanced implementations of this control scheme could include specialized states to satisfy secondary objectives, but they are not needed to satisfy the primary objective of maintaining balance.

The state machine demonstrates several useful aspects of the FPE as a feedback mechanism, but the FPE could also be used in radically different control structures. For instance, it could replace the swing phase in Pratt's work [44], or it could complement a ZMP approach for the emergency strategy suggested by Vukobratović [101]. Note that although the FPE does not distinguish between different styles of gaits, the control system as described above is not compatible with free-flight phases. Gaits that include running or jumping would require some control system changes, but the measure of balance would stay the same.

Chapter 4

Simulation

The theoretical basis provided in Chapter 3 makes a number of assumptions to simplify the system. It is very difficult to provide a theoretical basis for a system with any higher level of complexity that provides a closed form solution. However, further validation of the selected approach was needed before proceeding to a physical prototype. The next level of investigation came from simulation tools that allowed a much more complex and detailed analysis than the pure theoretical treatment, while still allowing some level of simplification over a physical system.

4.1 Modeling Reality

Although simulation technically preceded implementation of the physical system, a large portion of the physical system had to be constructed in order to find the basis for the simulation models. The first prototype was designed based on the author's experience and engineering intuition. The original design was shorter, used gear ratios oriented slightly more towards torque instead of speed, and was designed around a $50ms$ sample period. From this starting point, the design evolved towards the final version presented in this thesis.

4.1.1 Mechanical Model

To keep the robot design relatively simple, it was decided to restrict the physical implementation to an approximation of a 2D system. This was achieved by connecting the torso to a central hub by means of a light-weight boom. The boom prevents lateral falls, but the biped can move freely horizontally (x) and vertically (y), and is also free to rotate about the axis of the boom (θ) (see Figures 4.1 and 5.1). This is a common method of emulating a planar biped [46, 72–74, 76, 128, 150, 164]. The mass and inertia of the boom were insignificant compared to the biped (the boom represents a 3% error in the weight), and rotational friction around the hub was assumed to be negligible. Therefore, all effects of the hub were ignored in the simulation.

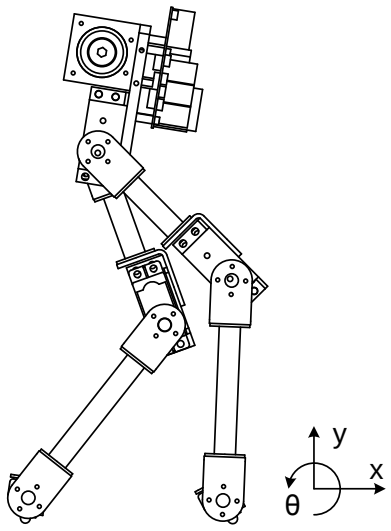


Figure 4.1: A side profile for both the simulated and physical robot.

The biped itself was modeled as a planar device constructed of rigid links with the appropriate geometric and inertial properties. These properties were derived from a CAD model of the physical system. Each rigid link segment of the CAD model was verified against the geometry and mass of the corresponding physical construct and the inertial matrix was calculated based on standard material densities [209]. The key parameters are summarized in Table 4.1. The completed 5

Limb Segment	Mass *(kg)*	Inertia (COM) $(kg\ m^2)$	Proximal (m)	Distal (m)
Torso	0.5315	0.0004971	-	0.0423
Thigh	0.0981	0.0001096	0.0826	0.0303
Shank	0.0553	0.0001212	0.0852	0.1101

Table 4.1: The mass, inertia (with respect to the COM of each segment about θ), and geometric parameters used in simulation. The proximal (closer to the torso) and distal (further from the torso) lengths are measured from the joints to the each segment COM.

segment biped has 7 degrees of freedom (2 hips, 2 knees, and 2 degrees of translation and 1 degree of rotation in the world space).

It is important to note that the interaction between the foot and the contact surface only occurs at a single point. As a result, many of the control methodologies that depend on a foot with multiple points of contact, including zero moment point, are not compatible with this mechanical configuration. Although this might not be a practical design for general purpose bipedal robots, it is an excellent example to demonstrate the fundamental application of the FPE in a difficult problem that some other approaches cannot even attempt.

4.1.2 Contact Model

The walking surface was constructed of a $3.8mm$ layer of neoprene mounted on $12.5mm$ of plywood. The contact points on the robot legs were a rounded-profile, hard nylon. A nonlinear spring-damper based on the work of Hunt and Crossley [210, 211] was used for the contact model. The function F_n describes the normal force where y is the positive penetration depth into the surface in meters, k is the spring constant, b is the damping constant, and n, p and q are used to fit nonlinear effects:

$$F_n = ky^n + by^p\dot{y}^q \tag{4.1}$$

To find the parameters of the first term of Equation 4.1, a sample of the walking surface and a duplicate foot were used in an Instron 5547 material testing machine (see Figure 4.2) to generate force versus penetration profiles. A total of three static compression tests were performed on the

Figure 4.2: An Instron materials testing machine. Photo courtesy of Instron® Corporation.

sample, each in slightly different locations, but all near the center of the sample. The results of the compression test are shown in Figure 4.3. An average of the three test results was calculated and the model parameters of the first term were fit to the resulting average by finding the minimum of the mean square error. Not shown in these results is a small amount of hysteresis that indicates the material is either slow to recover, or permanent deformation is occurring at this depth of penetration. However, since each step takes place in a new location, these hysteretic effects were ignored.

To identify the second term of Equation 4.1, a dynamic test was needed. To mimic the impact of the completed robot, the foot contact point was mounted to a mass comparable to the final robot. The mass was connected to two vertical guidelines to maintain its orientation (see Figure 4.4), and its motion was tracked with an NDI Certus position sensor[1] paired with an infrared LED marker operating at a sample rate of $875Hz$. A series of 14 drops from heights ranging from $0.04m$ to $0.16m$ were used. The parameters of the contact model were manually tuned to match

[1]Manufactured by NDI, 103 Randall Dr, Waterloo, Ontario, www.ndigital.com

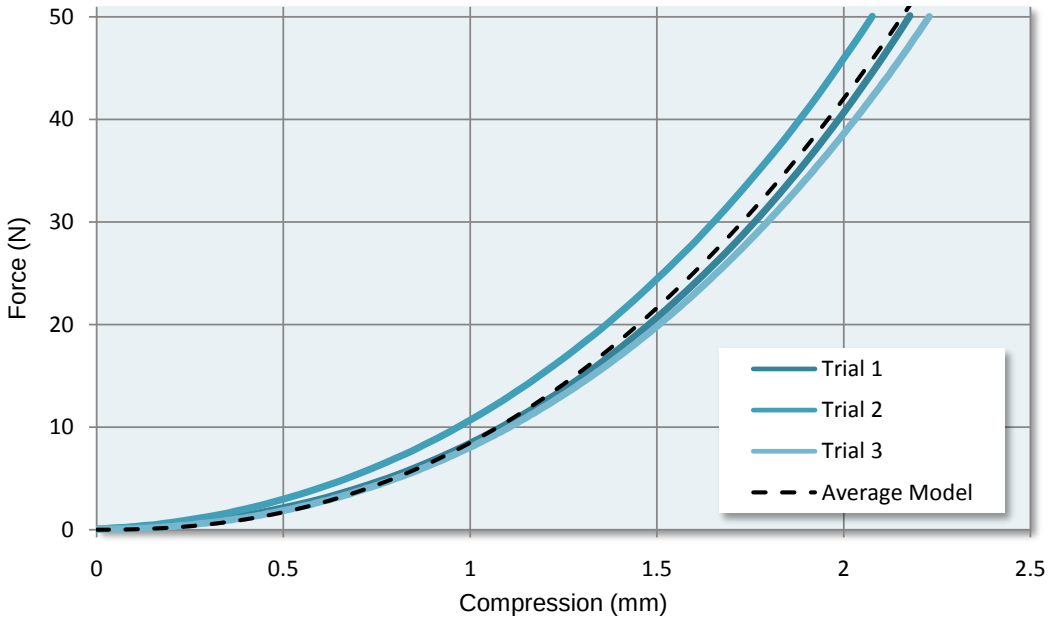


Figure 4.3: Compression results of the neoprene and contact point.

the measured data. A sample of the results is shown in Figure 4.5.

The model did not perfectly capture the impact conditions. There was approximately a 10% error at the peak amplitudes, but only a 1% error at the lower amplitudes. The error in amplitude was allowed in favour of matching the bounce frequency and total settling time. The final parameters selected for the contact model in Equation 4.1 were $k = 7.21 \times 10^7$, $n = 2.31$, $b = 3.8 \times 10^4$, $p = 1.1$, and $q = 1.0$ where the penetration y is in meters, and the force F_n is in Newtons.

The Hunt and Crossley model was selected above others because there are no discontinuities at the point of contact. The solver used for the simulations was well suited to solving stiff problems, but abrupt discontinuities resulted in a substantial increase in simulation time. (Stiff problems have solutions that change in a very short time period relative to the total time span of the solution of interest.)

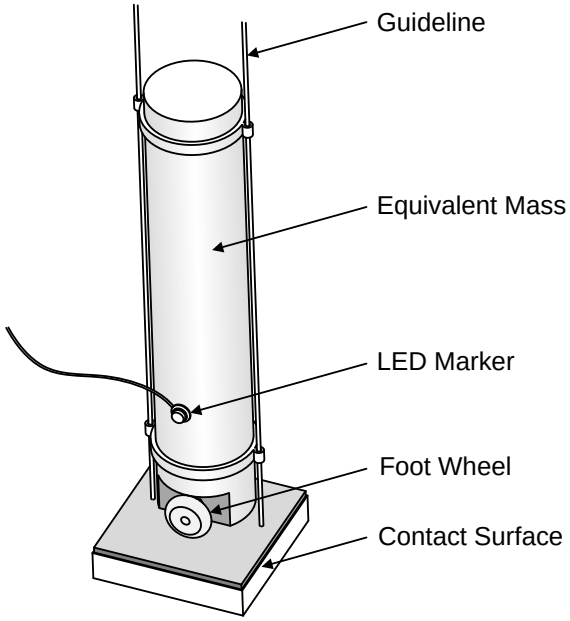


Figure 4.4: The drop experiment used to identify the dynamic component of the impact equation. The foot wheel used as the contact point is the same wheel used on the physical experiment.

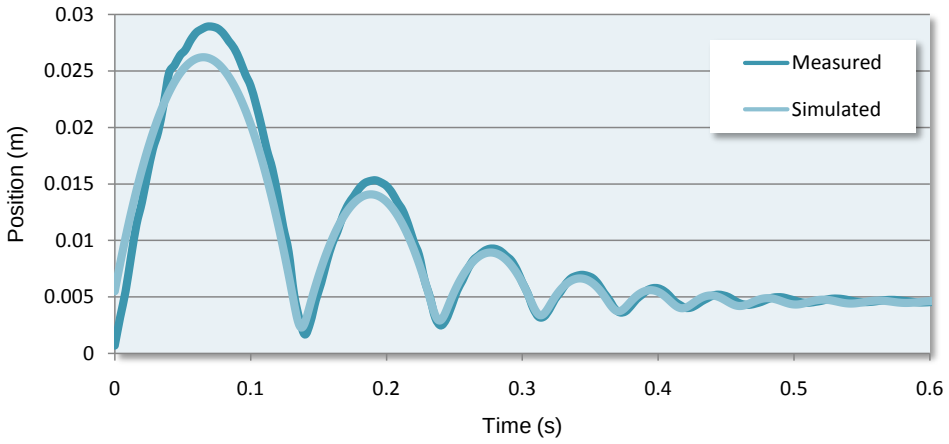


Figure 4.5: A comparison of measured data from the drop test to the simulated model. The initial drop height was $0.1m$.

4.1.3 Friction Model

In simulation, the force due to friction F_f was represented by a first order model based on the work of Bliman and Sorine [212]. The model was adapted to represent the coefficient of friction as shown in Equation 4.2 rather than the friction itself, so that it could be further modulated by the normal force F_n as shown in Equation 4.3.

$$\dot{u} = \frac{-3|\dot{x}|}{s_p}u + \frac{3f_k}{s_p}\dot{x} \tag{4.2}$$

$$F_f = F_n u \tag{4.3}$$

The variable $\dot{x}$ is the horizontal velocity of the contact point relative to the surface, f_k is the coefficient of kinetic friction, s_p is roughly equivalent to the displacement before saturation of the friction occurs, and u is an internal state of the friction model. This model results in a hysteretic curve as shown in Figure 4.6. The model was initialized at $u_0 = 0$ and was re-initialized to $u_0 = 0$

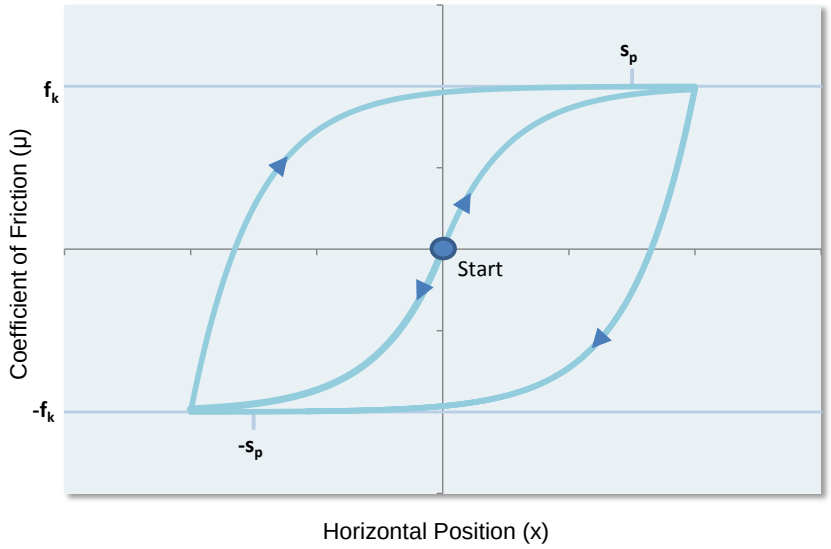


Figure 4.6: A visualization of the hysteretic curve created by Equation 4.2. The variable f_k is the saturation level of the coefficient of friction, and s_p is roughly the displacement before saturation occurs.

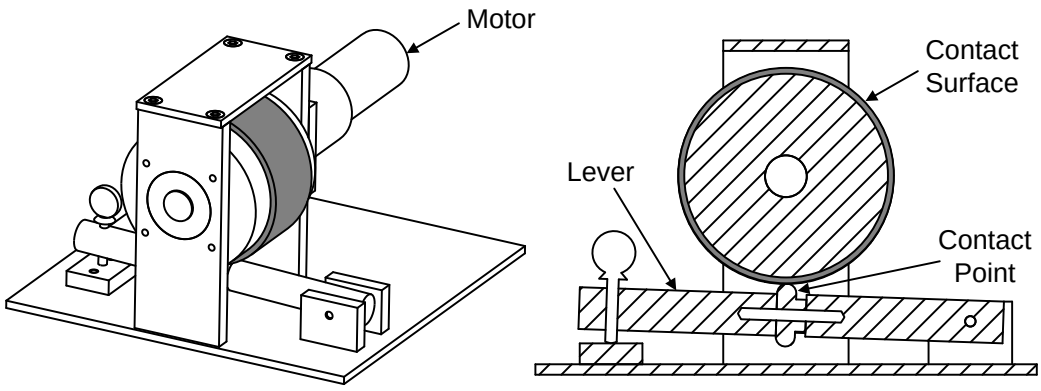


Figure 4.7: The experimental apparatus constructed to determine the coefficient of friction. The cross-section shows the lever and the integrated contact point.

whenever the foot was not in contact with the walking surface. This effectively allowed the origin of the hysteretic friction function to move to each new landing location.

The parameters for the friction model were identified using the experimental apparatus shown in Figure 4.7. A layer of neoprene was affixed to a wooden dowel with cyanoacrylate, and the foot contact point was mounted to the lower lever. As the lever was raised to compress the neoprene, the friction force was calculated from the applied motor torque and physical geometry. (The motor current, and indirectly the motor torque, was calibrated against a known mass hanging from the radius.) This data was combined with the results in Section 4.1.2 to derive the coefficient of friction. The final parameters used for the friction model were optimized around the penetration depth when the robot was statically balanced on one foot. The parameters selected were $s_p = 0.001$ and $f_k = 0.6$ where $\dot{x}$ is in meters per second, u is unitless, and F_f and F_n are in Newtons.

Like the contact model, this friction model was selected to avoid discontinuities while balancing necessary detail with computational efficiency. The hysteretic properties of this function nicely represented the elastic nature of the neoprene surface.

4.1.4 Servo Motor Model

The biped's actuators were standard hobby servo motors (HiTec HS-475HB) with custom electronics. The servos had an integrated spur gear train with an input to output ratio of 300 to 1, and were specified to have a peak output torque of $0.43Nm$ and a maximum angular velocity of $235°/s$ at the output shaft. The custom electronics improved the response time and torque of the servo, provided feedback to the main processor (with an effective resolution of 0.225° per bit), and allowed the main processor to vary the individual motor controller gains (see Section 5.1.4 for specific design details). Backlash in the gear train was minimal and ignored in simulation.

Starting from a standard direct current motor model, the servo model was constructed as shown in Figure 4.8. It was assumed that the dynamics would be dominated by the mechanical component, thus the electrical constant was composed of the electrical resistance and torque constant, but the dynamics due to the inductance were ignored. Since the motors would not operate in isolation, the mechanical component of the motor was amalgamated into the rest of the mechanical model.

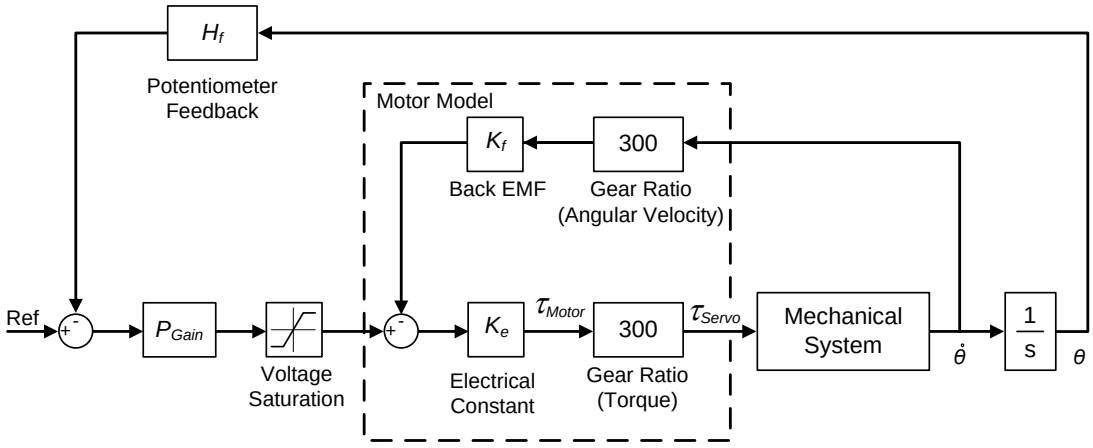


Figure 4.8: The servo control loop and motor model.

The position control loop was based on a simple proportional controller using the built-in potentiometer in the servo for feedback. The viscous friction in the system generally provided sufficient damping for a stable step response over the the range of utilized gains. The only addition

to the typical proportional controller was the saturation function which limited the controller output to $\pm 5V$ as per the actual control circuit.

Identification of the motor parameters was done by feeding an identical control pattern into the simulation and physical servos, and then adjusting the simulation parameters to match the position profile of the physical robot. The robot was suspended in the air and the motion patterns were selected to represent typical swing motions. This was not repeated for standing loads because the motion of the stance leg is relatively limited by comparison. The results of the simulation compared to the actual response are shown in Figures 4.9 and 4.10. Some slight differences can be observed at steady state. In the physical system, a change of friction as a function of the load was observed. This results in approximately 4% error when the leg is held horizontal against gravity, which is not reflected in the simulated model. The final parameters used in the model were $K_f = 4 \times 10^{-3} Vs/^\circ$, $K_e = 3.3 \times 10^{-4} Nm/V$, and $H_f = 4.44\ counts/^\circ$.

4.1.5 Sensor and Communication Models

The foot placement estimator requires the position and velocity of the COM, the average angular velocity, and foot contact information. To determine the absolute position of the torso, the central hub was instrumented with optical encoders. Given the encoder resolution and boom length, this gave an effective resolution of approximately $0.12mm$ in the x and y, and approximately 0.088° about θ (see Figure 4.1). The simulated sensors included this quantization.

The process of calculating the total COM also required the position of the leg segments. Once every $10ms$, the main processor would send out an information request to each servo over the local I^2C network to determine its current position. Each servo in turn would respond with a 10-bit number with a resolution of 0.225° per bit (the full range of motion for the servo was slightly over 200°). Combined with the hub encoders, this provided all the information needed to calculate the position, and indirectly the velocity, of the COM.

The full set of COM, FPE, and kinematic calculations, plus the communication delays in the I^2C channel required almost the full $10ms$ sample period before the new joint position references reached the servos. Therefore, this calculation period was represented as a fixed $10ms$ sample delay

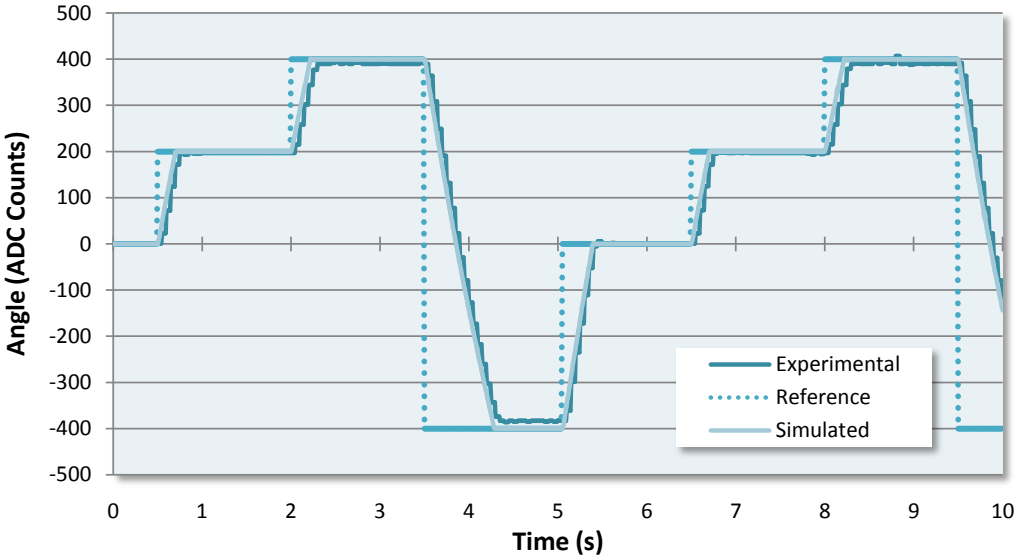


Figure 4.9: The simulation results of the motor model compared to the actual motor's response at the hip joint.

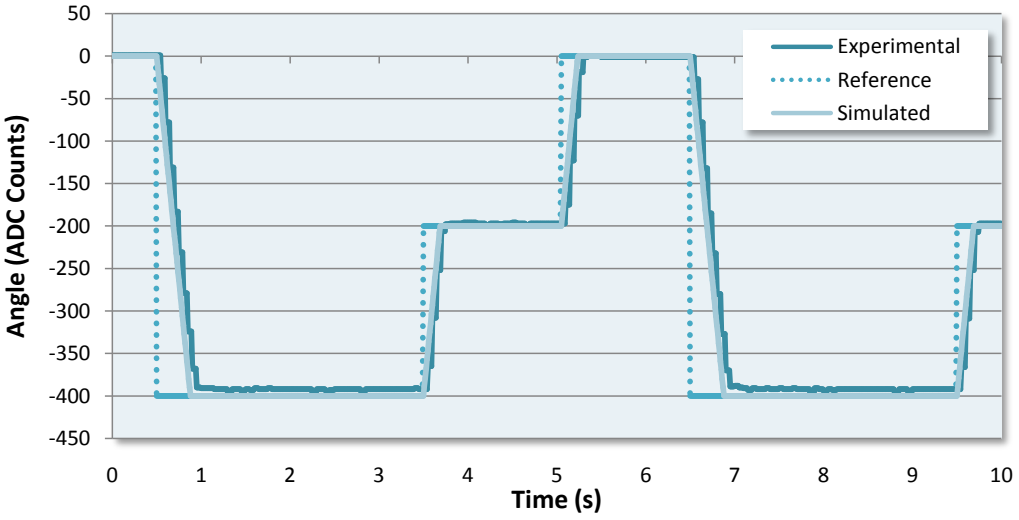


Figure 4.10: The simulation results of the motor model compared to the actual motor's response at the knee joint.

in simulation. One difference between the simulated and physical systems was that the sensors were read simultaneously in simulation, and sequentially over a $3ms$ period in the physical robot. Considering the bandwidth of the motors, this difference is negligible.

Foot contact in the physical hardware was achieved with a simple contact switch that was wired directly to a general input-output (I/O) port on the main processor, so there was no delay and it could be read asynchronously. In simulation, the foot contact switch was modeled as a digital input based on a threshold tied into the contact model.

4.1.6 Control Routines

The control routines generally followed the approach outlined in Section 3.5. The code was segregated into functions that were specific to the simulation environment, and functions that applied cross-platform, such that a single set of core code could be used for both the simulation and physical robot. The core code included all the calculations for the physical properties of the system (COM, inertia, etc.), the FPE solver, the state machine, and all of the kinematic calculations. Further details of the specific control functions will be deferred to Section 5.2.

The control gains of the biped were selected to ensure the validity of the FPE solution where necessary. For example, the control gain of the swing hip was set sufficiently high such that the foot could swing through and be tracking the reference FPE position prior to impact occurring. Once the foot was properly tracking the reference position, the internal motions of the biped became minimal, thus validating the assumptions of the stick man model and maximizing the accuracy of the FPE as discussed in Section 3.4.1.

During the drop state, the control gains of the swing leg were selected to provide a damped response to the impact, so the landing foot would not bounce off the surface and violate the assumption of a plastic impact. It is interesting to note that a number of other bipedal robots try to provide a damped response to improve stability [8, 113]. Given the parameters of the simulation, it was found that only a small amount of damping was necessary to meet the plastic assumption. More damping means more energy loss that is not accounted for by the FPE solution, thus generally making the FPE more conservative than necessary. After impact, the gain of the

landing leg knee controller was increased to provide the rigid standing leg that was assumed in the derivation of the FPE.

The remainder of the control references and gains affected the style, stride length, and average gait speed of the FPE. It was found to be possible to achieve a range of gait speeds simply by altering the velocity of the leg extension during the push state, however, the transients and consistency of that gait speed were found to be a complex, nonlinear function of the other control parameters. The parameters selected for a complete range of gait speeds will be dependant on the secondary objectives. This is left to future work.

4.2 Simulating the System

Matlab[2], and more specifically the Simulink component with the SimMechanics Toolbox, was the primary simulation environment. These same tools have been used by other researchers in this field [22]. The equations of motion for the mechanical system were automatically derived [213] at the beginning of each simulation using a Lagrangian approach. To make use of a general purpose ordinary differential equation (ODE) solver for the entire system, the resulting constraint equations of the differential algebraic equations (DAE's) were explicitly differentiated to produce index-1 DAE's which could be solved by treating the system as an ODE on a manifold.

The complete set of differential equations were solved using an ODE45 (the Dormand-Prince pair for a Runge-Kutta formula [214]) variable step solver. The use of a variable step solver allowed large time steps (up to the smallest discrete controller sample period) during periods of slow dynamics to improve simulation time and small time steps during periods of fast dynamics to improve accuracy. This was aided by zero-crossing detection to prompt the solver to reduce the time step as the system passed through a discontinuity or encountered a sudden change in system stiffness.

The Dormand-Prince variable rate solver performed well for this problem by reducing the overall simulation time without sacrificing accuracy during the impacts. Other solvers, such as

[2]MathWorks Inc, www.mathworks.com

those based on numerical differentiation formulas (ODE15), specialize in stiff problems, but these can have lower accuracy than the ODE45. During the initial implementation of the contact and friction models, it was found that the ODE45 solver would stall under certain circumstances. The ODE15 was capable of finding a solution, but it would occasionally produce results that were obviously erroneous. Thus, for greater confidence in the simulation results, the ODE45 was used so that if a problem arose, it would stall rather than produce a potentially erroneous result that could be overlooked.

The majority of the simulation results were output as various graphs. However, at the end of each simulation, a data log was produced similar to one that would be recorded from the physical robot. This could be loaded by a custom PC interface (see Section 5.1.5) to animate a visualization of the robot. The visualizations proved incredibly useful as they often gave more information about the complex nonlinear relationships between the multiple parameters, where such details were easily lost in a two-dimensional graph.

The simulation can be found on the disc included in the hardcopy version of this thesis. Given the large number of modules and visual interface to the Simulink components, it was not feasible to include the code directly in this thesis. However, the core algorithms are identical to the physical hardware and the source code is included in Appendix C.

4.2.1 Simulation Results

The simulation results successfully demonstrate that the FPE can be used to provide closed-loop balance control for a walking bipedal robot. A gait diagram along with the position of the feet and the FPE relative to the COM is shown in Figure 4.11. Although the simulated biped was capable of walking indefinitely, this simulation demonstrates 12 steps of the gait cycle including gait initiation and termination. The reference position used for the swing foot was the location of the FPE plus a small constant, so that when contact occurred the foot was slightly ahead of the FPE which ensured each step was stable. In Figure 4.11 where the dashed lines transition to the solid lines, this indicates the landing point of the swing leg. Since all landing points were in front of the FPE, then if all the references were held fixed after a given step was completed, then

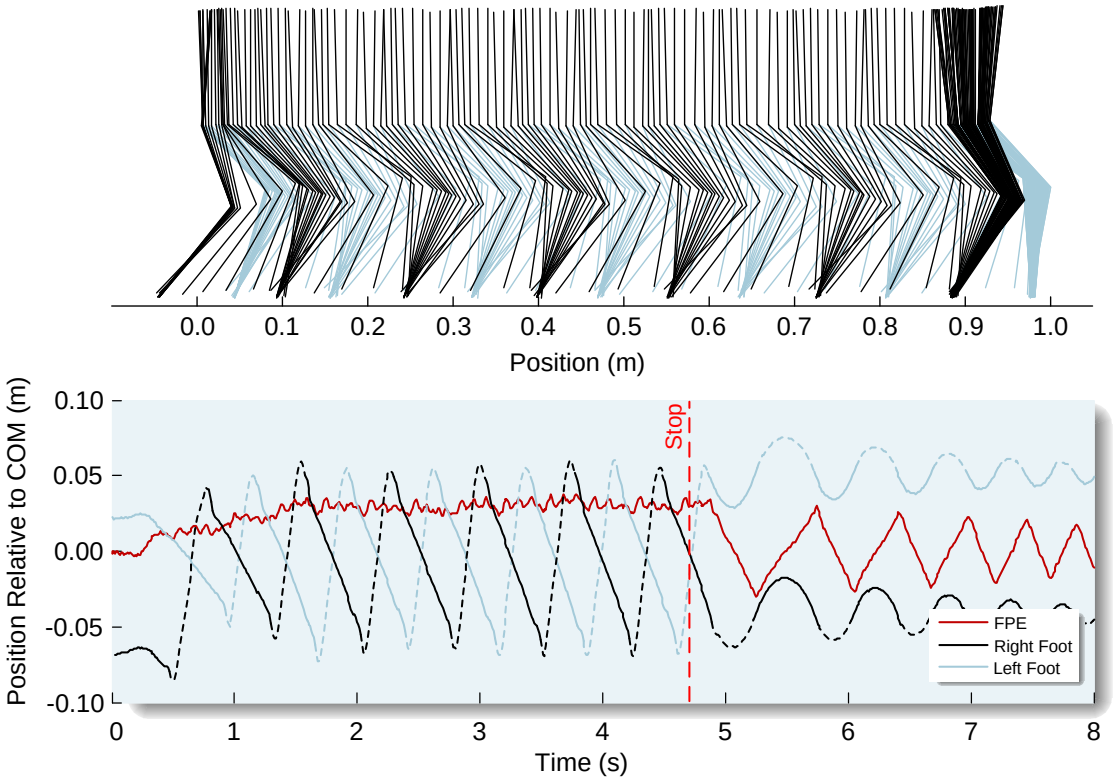


Figure 4.11: Simulation results from gait initiation through 12 steps to gait termination. The dashed portions of the foot position indicate when the foot was off the ground, and the solid portions indicate when the foot was on the ground.

based on the theory from Chapter 3, the step would be stable.

To achieve gait termination, a stop command was issued which caused the state machine to enter the standing still state after the drop state. The result is observed on the right side of Figure 4.11. All references were held fixed, causing the biped to rock between the leading and trailing feet until the impacts dampened out the motion, bringing the biped to rest. Note that the FPE is contained between the two contact positions. This indicates that the biped is in stable region 1, and is therefore guaranteed to approach a stable standing position in the absence of any geometric changes or external disturbances.

The simulations indicated that there were a wide set of parameters suitable for achieving stable

gaits (most of which could also be used for gait initiation), but only a narrow range of parameters were suitable for single-step gait termination. If gait termination must be consistently achieved in a single step regardless of the gait parameters, or a more damped response is desired, then a dedicated gait termination state could be used.

The consistency of the steps can be analyzed by examining the joint angles. Using the same parameters as those used in Figure 4.11, the simulation ran for 50 steps, or 25 complete gait cycles. The joint angles for each gait cycle were overlaid on top of one another to produce Figure 4.12. In a physical system, the variation could be attributed to noise or friction, but no artificial noise was included in the simulations. It could also be dismissed as the chaotic nature of the system [215], but the patterns appear widest for the standing leg. It is possible that small variations in the dynamic loading became more exaggerated as they propagated through the serial chain to the supporting point. The exact cause is unknown, but the variations were small enough that further investigation was not warranted.

4.2.2 Robustness to External Disturbances

To quote the definition of robustness as it applies to control systems from [216]:

Suppose a plant of function P belongs to a set $\bar{P}$. Consider some characteristic of the feedback system ... a controller C is robust with respect to this characteristic if this characteristic holds for every plant in $\bar{P}$.

The set, $\bar{P}$, refers to the possible variations of the plant. These variations can include control parameters, structural parameters, or external disturbances. While all of these variations are possible, the primary interest of this work is robustness to external disturbances. All other goals, such as gait speed and step size can be altered at the expense of maintaining balance, even to the point of not walking at all, or walking opposite to the intended direction, as long as balance is maintained. The use of the FPE as a feedback mechanism makes the system inherently capable of compensating for external disturbances.

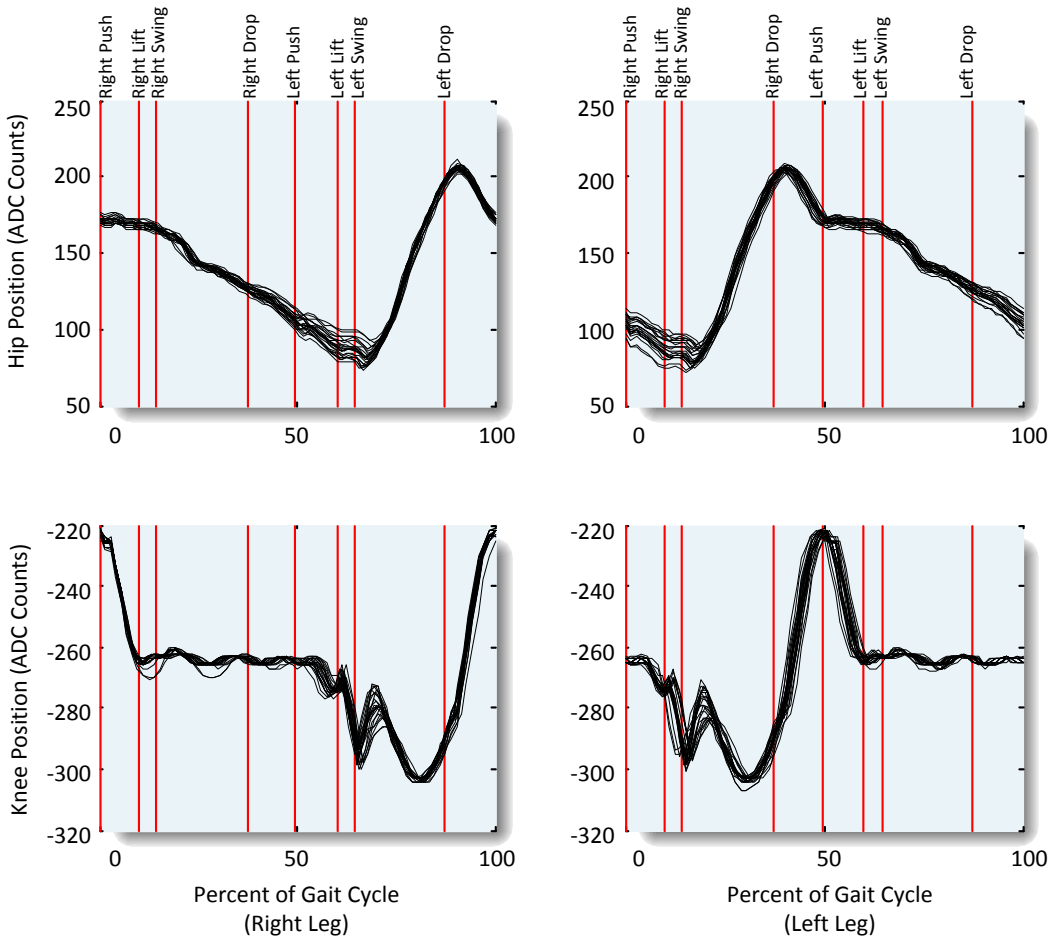


Figure 4.12: The joint angles of the biped in simulation over 50 steps.

The resilience of the system to external disturbances was demonstrated with single step-pulses applied horizontally in the world space to the hip joint. By applying the pulse at the hip location, the intent was to test the response of the FPE control algorithms to reject external disturbances, rather than test the abilities of the individual joint controllers. However, as noted in Figure 4.12, even under nominal conditions there was some minor variance from step to step, so it was impossible to completely isolate the FPE algorithm from the joint performance.

To deal with the minor step variance, a simulation without any disturbances was first computed and the start time of each step was recorded. The simulation was then repeated multiple times with single pulses applied at 10 points in each stride with a pulse width of 10% of the undisturbed stride time. This was repeated for 20 unique steps to account for the minor variations in strides. Finally, the pulse amplitude was varied from 10% to 200% of the biped's weight. All the pulses were applied from behind the biped to the torso at the location of the hip joint. The biped was deemed to have successfully rejected the disturbance if the vertical position of the COM remained higher than 40% of its nominal height during the 5 seconds following the external disturbance. If the biped fell as a result of the disturbance, this usually occurred in less than 1 second following the pulse.

The effort expended to keep the simulations as fast as possible proved worthwhile for the disturbance testing regime. The results shown in Figure 4.13 are the culmination of over 4000 simulations, which in total were completed in approximately 16 hours. Each vertex in Figure 4.13 is the average success of 20 simulations. The simulation results indicate that the biped can reject disturbances up to 20% of its weight during any state of the gait cycle (as indicated by the plateau at the back). This is a result of the limited bandwidth of the actuators. The probability of success quickly drops off for all states except the swing state. The extended robustness during the swing state is attributed to the swing leg already being in motion. Since time is not expended accelerating the leg to respond to the disturbance, there is a higher probability the biped will recover. There is a marginal robustness improvement during the drop state compared to the push and lift states since the swing foot is already out in front, but since the leg still needs to be accelerated, this benefit is limited. These results suggest that if the robot has the ability to

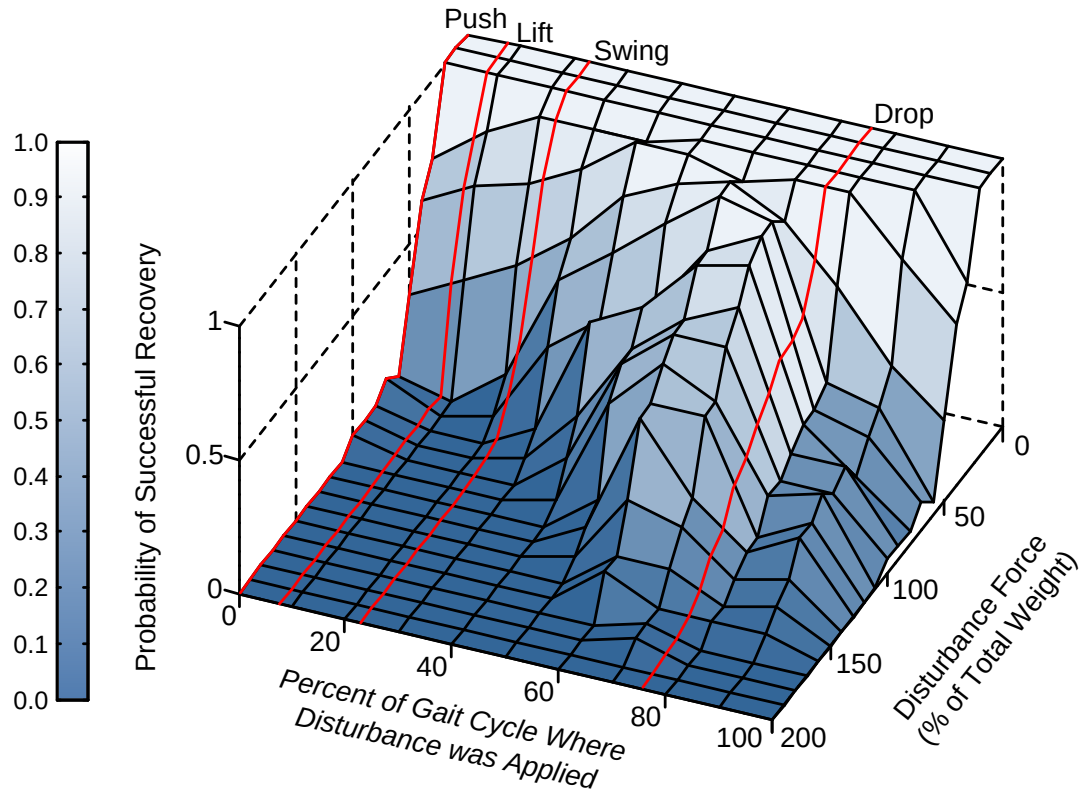


Figure 4.13: Simulation results of single pulse disturbance rejection. Each vertex is the average rate of successful recovery from the disturbance pulse over 20 simulations. The red lines indicate the average start time of each state during a normalized stride.

anticipate external disturbances, then it could preemptively modify its gait cycle to allow the disturbance to occur during its swing state to maximize its chance to recover.

Note that successful disturbance rejection does not necessarily mean that the biped was walking after the disturbance. Particularly for the larger disturbances, the pulse sometimes resulted in a step that was so large that the state machine became stuck in the push state. The pushing leg was fully extended, yet the FPE had not moved in front of the lead foot. Further experimentation determined that the addition of a recovery state to draw the feet together and retry the step could be used to get out of this position. Humans in a similar circumstance would also need to

use a recovery strategy to return to a position more conducive to resuming gait. In a more general biped architecture, rather than endlessly adding states to handle special conditions, it would make more sense to have a high-level controller select different control gains, and even different control approaches to suit the current environment and control objectives.

Although various simulations were performed with multiple disturbances, the results are largely anecdotal. Figure 4.14 shows a series of random pulses applied to the biped which demonstrates the behavior in various circumstances. The biped is capable of withstanding this particular

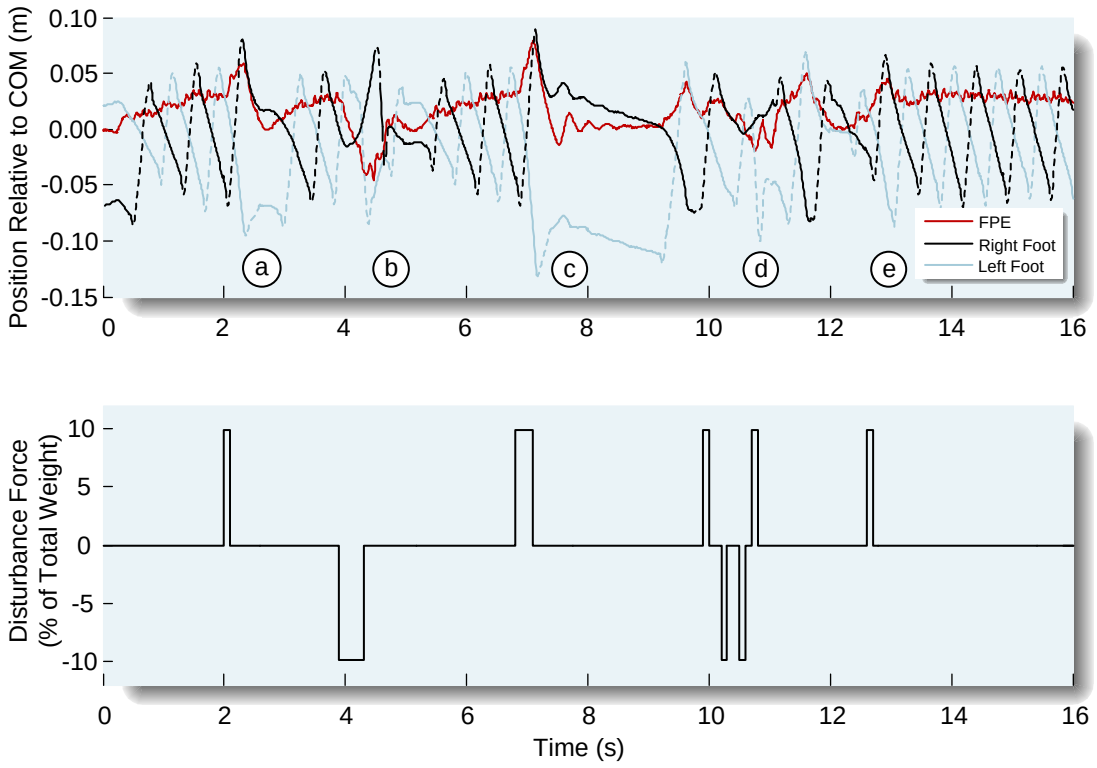


Figure 4.14: The gait response to a random disturbance. (a) A small positive push results in a small step forward. (b) A large negative push requires a step backward. (c) A large positive push results in a large step forward. The pushing leg extends to move the FPE in front of the lead foot to resume walking. (d) After landing at 10.5s, the robot must use the same swing leg to step backward in response to the negative pulse. (e) In the absence of disturbances, the biped returns to a normal regular gait pattern.

disturbance sequence, but another random series of pulses might cause it to fall. It is difficult to draw any general conclusions in multi-disturbance cases. For instance, a random sequence that causes the biped to fall might not be a problem with a lower disturbance force, but a sequence with the same lower force could be crafted to create a resonant effect that leads to a fall by pushing the robot at just the right times. A complete multi-disturbance analysis of the biped becomes a hyper-dimensional problem. The results are difficult to present in a complete manner and the simulation time needed to complete an exhaustive analysis becomes very large. One of the difficulties in the bipedal robotics literature is making comparisons of different control approaches without implementing every one on a common platform. Thus, it is the opinion of this author that the only fair way to map the disturbance rejection abilities of a biped is using single pulses. Hopefully other researchers will adopt a similar non-dimensional approach to allow for better comparisons between control approaches across different platforms.

4.3 Discussion

There were very few problems in the process of creating the simulation. The most difficult part was developing accurate contact and friction models that were computationally efficient. The models presented in this chapter made a reasonable compromise.

As noted before, the design began with some guesses based on engineering intuition. Although actuator selection will be discussed in detail in Section 5.1.4, the actuators are clearly a key determinant of a biped's performance. The selected motors were utilized well within their normal safe operating region as one would expect with any good engineering design. However, there are many applications where motors regularly operate outside of their safe operating parameters, and the bipedal robot is clearly an application that would benefit from this. By the time this was determined, there was already a significant investment in the hardware in order to determine the parameters needed for the simulations. Further investigation of this possible design refinement is left to future work.

One aspect that the simulations do not directly address from Chapter 3 is the region of validity.

The gaits implemented in the simulations stay within the region of validity under disturbance-free conditions through the appropriate selection of offsets and control gains. Given the relatively low number of degrees of freedom, this biped was limited in where its foot could be placed. However, future designs of higher complexity may have a wider range of options. In this case, dynamic bounds on the workspace of each leg to respect the region of validity might be worth exploring.

The physical robot design evolved with the simulation results, but eventually further refinements could only be found by putting the theory to practice. Simulations make it easy and fast to modify the mechanical system and control parameters, but it is important to validate the concepts on a physical prototype because a simulation can never capture all the intricacies of the physical reality.

Chapter 5

Physical Robot

The theory behind the foot placement estimator (FPE) was mathematically justified for a simple biped in Chapter 3. The simulation results from Chapter 4 suggested that the proposed control approach should work for a non-trivial bipedal robot in a realistic environment. However, the FPE theory could never be absolutely proven to work for a non-trivial system until it was implemented on a physical robot.

Designing the ideal bipedal robot is the epitome of multi-disciplinary engineering. The structure should be stiff enough to minimize vibration and flexibility, yet trim enough to avoid excess mass. The motors must be properly geared to make the appropriate tradeoff between speed and torque, but typical engineering margins of safety can be detrimental in terms of added weight and power consumption. The motor selection in turn feeds back into the structural design where the motor mass and maximum torque must be taken into account. Fortunately, the power source in this design was external which eliminated the need to take the size, weight, and peak current output of a local power source into account for the structural design and motor selection. Ultimately, a bipedal design must find a balance between performance and cost. For this research, cost governed many of the decisions.

5.1 Designing the Bipedal Robot

The principle goal of creating a physical robot was to provide proof that the FPE theory would work in practice. A secondary goal was to demonstrate that this could be accomplished without a multi-million dollar budget, or more specifically, for about $1000. To help minimize the cost, an off-the-shelf approach was taken to the design. In retrospect, this may have added more complication than was worth the money it saved. Given the vast number of bipedal prototypes that have been built by various institutions, this new bipedal robot was named Yet Another Bipedal Robot, or YABR for short.

5.1.1 The Central Hub

The design of YABR is shown in Figure 5.1. YABR used a spherical surface (the surface traced by the end of the boom) to approximate a 2D planar environment. The central hub provided lateral stability to the biped while still allowing free motion in the x and y axis with free rotation about the z axis.

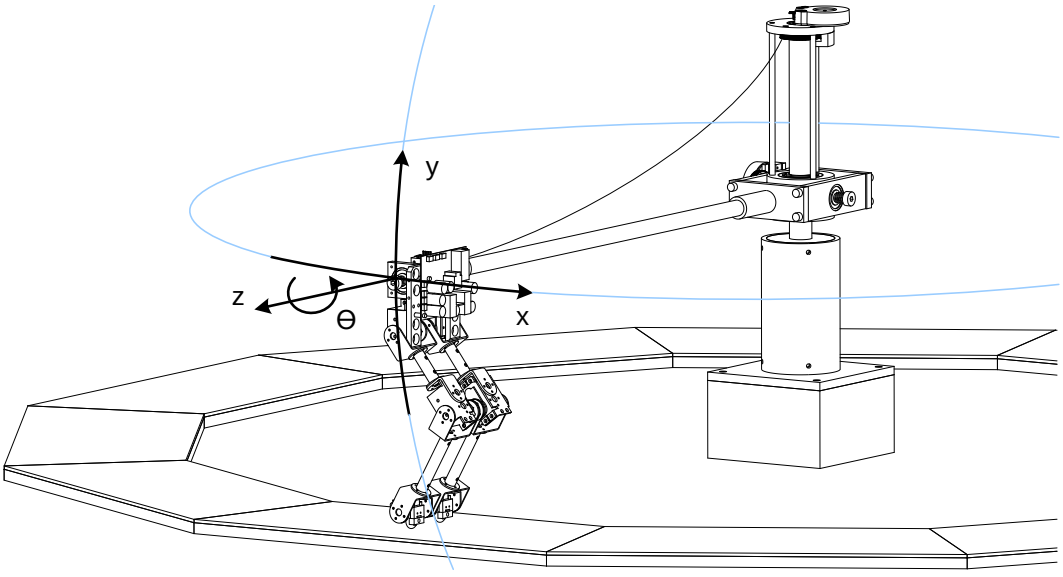


Figure 5.1: YABR approximated a planar system by using a boom attached to a central hub for lateral support.

The central hub was custom machined using a combination of aluminum and PVC plastic. The two rotary joints at the hub (x, y) were constructed with ball bearings to minimize friction. The boom itself was constructed of carbon fiber tubing to minimize the weight. The force due to gravity of the boom at the connection point on the torso was approximately 3% of the total robot weight, so it was relatively insignificant. No counterweight was used to avoid increasing the rotary inertia of the boom. Finally, a safety wire was connected between the upper structure of the hub and the outer edge of the boom to protect YABR's electronics from falls. A complete set of the mechanical drawings can be found in Appendix A.

The two axes of rotation were instrumented with optical encoders with a disc resolution of 2048 counts. Using mechanical gearing and quadrature decoding, this gave an effective resolution of approximately 0.009° (or $0.12mm$ as a linear approximation around the circumference) in the x and y directions.

5.1.2 The Walking Surface

Based on the impact assumptions made in Chapter 3, the goal of the material selected for the walking surface was to provide a high-friction surface that was stiff, but not solid. When used in combination with the appropriate joint controllers during impact, the material would ideally implement the assumption of a plastic impact as well as reduce the physical shock to the mechanical structure of YABR.

After studying various rubbers and foams, it was decided that the material that best represented these objectives was a $3.8mm$ (nominally $1/8in$), 20A[1] neoprene rubber (part number 9109K43 from McMaster-Carr[2]). The rubber was mounted on trapezoidal, $12.5mm$ thick plywood panels using cyanoacrylate. These panels were then bolted to the experiment table in a circular shape. The walking path can be seen in the mechanical drawing in Figure 5.1.

[1] Type A Shore durometer hardness scale

[2] www.mcmaster.com

5.1.3 The Robot Design

Many of the components that made up the leg frame of YABR were off-the-shelf parts from Lynxmotion[3]. The torso plate was custom machined to provide a solid base to mount the legs and electronics, as well as the bearings and corresponding optical encoder that provided the θ axis of rotation (see Figure 5.1). The optical encoder used a disk resolution of 1024 counts. Using quadrature decoding, this corresponded to an angular resolution of 0.088°.

The boom design necessitated the use of a wheel on the foot with the rotational axis of the wheel perpendicular to the boom axis. As seen in Figure 5.2, the motion of the legs resulted in a changing radius from the central hub. In a worst case scenario without the wheel, this could result in a lateral foot deflection of $23.4mm$ which would create undesirable flex in the frame or unpredictable slipping. Therefore, an uncontrolled degree of freedom was added to the foot in the form of a rounded-profile, nylon wheel that allowed free movement along the radial axis, while still providing friction along the tangent. A similar approach was used by Chevallereau et al. [150] and Pratt [44].

Finally, to determine when the feet were in contact with the ground, the nylon wheels were mounted on a small slider blocks which in turn triggered contact switches. A photo of the completed experiment is shown in Figure 5.3.

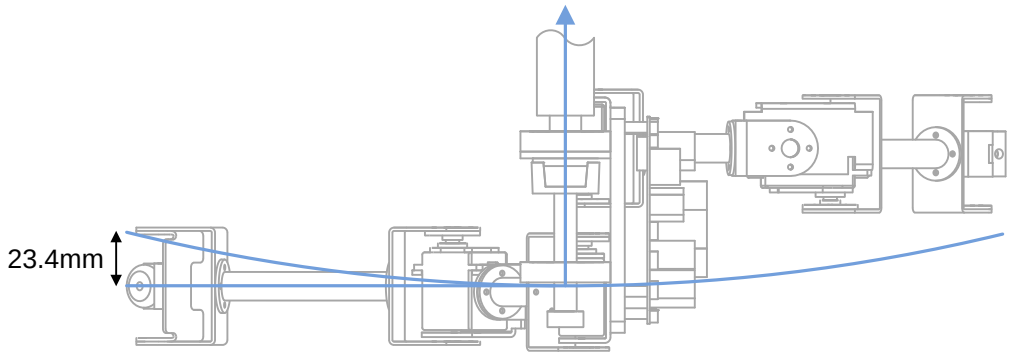


Figure 5.2: A top view of YABR. The wheels on the feet were needed to compensate for the rounded path.

[3]www.lynxmotion.com

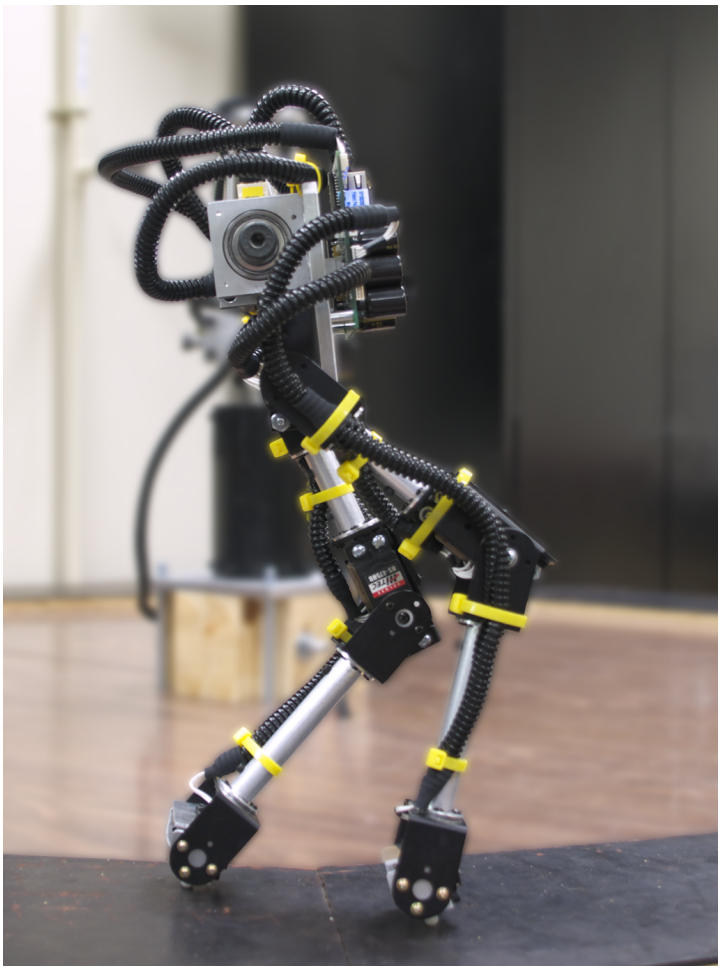


Figure 5.3: The completed YABR robot.

Note, unlike many other designs, there was no attempt to minimize the mass and inertia of the legs relative to the torso. A significant leg mass increased the complexity of the problem, but also added further validation to the FPE approach. A more general purpose design would certainly benefit from moving the mass out of the legs to minimize the dynamic forces resulting from the leg motions. This would also move the COM higher to reduce the frequency of the inverted pendulum.

5.1.4 The Servo Motors

Arguably, the most critical component in the bipedal robot was the actuators. The most economical and practical category of motors for this project were standard, rotary hobby servo motors. A decent selection of hobby servos are available that cover a range of torques and speeds where one is usually a tradeoff for the other. The selected motor was a mid-range HiTec[4] HS-475HB ball bearing servo. Although higher quality hobby servos were available at the time, most had brushless motors which are considerably more complex to drive than the commutated DC motors, and are significantly more expensive. The use of hobby servo motors has become relatively common for "desktop" bipedal robots [18, 33, 72].

The control distribution of the physical robot mirrored the distribution used in the simulations. Rather than using a single processor that controlled the entire system, the control of each individual servo was delegated to a dedicated motor controller which received instructions from the main processor. This approach has also been used by other research groups [18, 19, 33, 217, 218].

Electronics

Preliminary experiments aimed at assessing the capabilities of the servos indicated that the built-in controller was over-damped. Most hobby servos use a low-gain proportional controller to avoid chatter, but at the expense of performance. For this application, the maximum performance of the motor was needed, so a custom motor controller was designed to replace the existing circuit.

Space inside the servos was limited, so the new controller was designed using a Microchip[5] 16F819 microcontroller, a discrete H-bridge made of four MOSFET's, a small 7-pin connector, and as much capacitance as would fit in the remaining space to buffer the power regulators on the main board. The microcontroller featured a 10-bit analog to digital converter (ADC) to read the value of the servo's position potentiometer, an internal oscillator to eliminate the bulky external crystal, a pulse-width modulation (PWM) output to drive the MOSFET's, and an I^2C controller for communication. The schematic of the controller circuit is shown in Appendix B.

[4] www.hitecrcd.com

[5] www.microchip.com

Firmware

The motor controller firmware ran a proportional control loop at a $1ms$ sample rate. When a timer interrupt triggered a control calculation, the ADC was sampled. This value was used as the index for a calibration lookup table (see Section 5.1.6) which returned the current joint position on a scale consistent across all four servos. The joint position was then used to calculate the error from the position reference. The result was multiplied by an integer gain to produce the control output.

Since the custom controller generally ran with higher gains than the original controller, a dead band was incorporated to eliminate chatter due to noise from the ADC samples. Any remaining noise that resulted in an erroneous control output was filtered out by the mechanical dynamics of the system. Although the noise did not impact the servo performance, it presented a substantial challenge for the FPE calculations. Simply reporting the last ADC sample to the main board during a communication request frequently resulted in one of the noise spikes being reported rather than the true position. Filtering the servo positions on the main board resulted in excessive lag, but since the motor controller sampled the ADC ten times faster than the communication requests, rudimentary filtering could be done on the motor controller with minimal increase to the signal lag. Every time the ADC was sampled, the value was copied into a 20 entry FIFO buffer. When a communication request was received, an integer average of the FIFO buffer was reported.

The communication routines were all processed through the I^2C network. During initialization, each motor controller loaded its I^2C controller with a unique numerical identifier. When this identifier was seen on the network, the motor controller's I^2C controller automatically replied to the main board. This was followed by 3 bytes from the main board to indicate the new position reference and gain. The motor controller then returned 2 bytes to indicate the current servo position.

To simplify code maintenance, a single set of core code was used for all the servos. The variations between the code, such as the I^2C identifier and the ADC calibration table, were stored as constants in separate header files for each servo. When the project was compiled, four binary files were produced, one for each servo.

5.1.5 The Main Board

Every electrical connection and byte of data eventually flowed through the main board. This complex circuit was responsible for performing the FPE and kinematic calculations, controlling all the internal and external communications, reading every sensor directly or indirectly, and regulating all the power. An overview of the subsystems is shown in Figure 5.4.

Electronics

The main processor was a 30 MIPS, integer-based, Microchip dsPIC 30F6012A. This is a relatively low-end digital signal processor (DSP), but unlike most DSP's it has built-in flash memory and I/O ports, which saved considerable development time. In particular, the dsPIC supported a master I^2C interface to communicate with the servos, digital I/O to read the foot contact switches, and multiple RS-232 interfaces capable of operating at a baud rate of $921,600bps$ to communicate with the ethernet module. The only peripheral feature that was not included was quadrature decoding, but this was achieved using external US Digital[6] LFLS7166 encoder-to-microprocessor chips.

The primary means of communicating with the outside world was via ethernet. Lantronix[7]

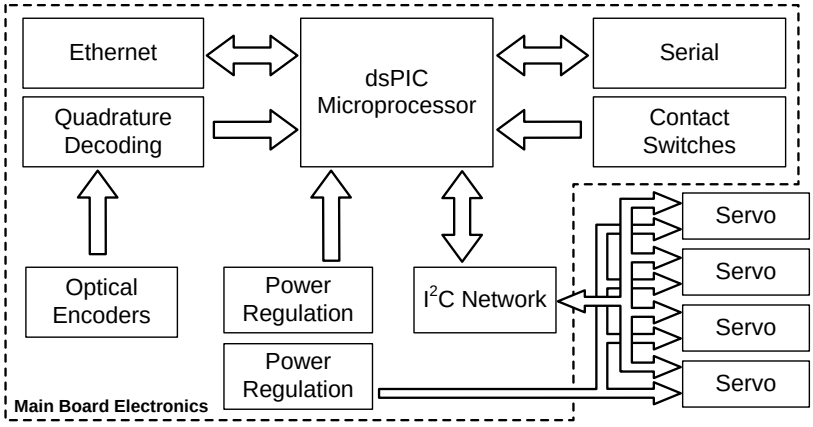


Figure 5.4: An overview of the main board subsystems.

[6] www.usdigital.com

[7] www.lantronix.com

offered a single module solution called the XPort that provided serial to ethernet conversion at a bandwidth of $921,600bps$. Direct serial communication between the PC and dsPIC was included as a backup, but serial ports on PC's are becoming obsolete, and ethernet can support much higher data rates anyway.

The final key section of the main board was the power regulation. Although the power could have been regulated to 5V by an external desktop power supply, motors cause current transients, which can feed noise back into the power rails or cause periodic drops in the voltage. To deal with this, two independent on-board voltage regulators reduced the source voltage of 7.5V down to 5V. With the addition of filtering and careful circuit layout, the noise generated by the motors was kept isolated from the more sensitive electronics. The schematic of the main board is included in Appendix B.

Firmware

On power-up, the firmware first put the hardware in a safe state. The gains of all the servos were set to zero which effectively disabled them, and the XPort module was reset to force it to reacquire the network. The robot was moved by hand such that all three encoders on the central hub passed through an index pulse that indicated where the zero position was located. Once all three indices had been registered, the robot entered the main processing loop.

The main processing loop was responsible for sending data to the PC, parsing any commands or parameter changes received from the PC, and shutting down the servos if the biped appeared to have fallen. The transmit function essentially tried to send information as fast as it could to the PC interface, but never at the expense of the control algorithms which were called by a timer interrupt every $10ms$. Each transmitted packet included a time stamp so it could be determined in post-processing if a time sample had been skipped. The details of the control routines are covered in Section 5.2, but an overview is shown in Figure 5.5. The complete code can be found on the CD accompanying the hardcopy of this thesis, and the implementation of the core algorithms is shown in Appendix C.

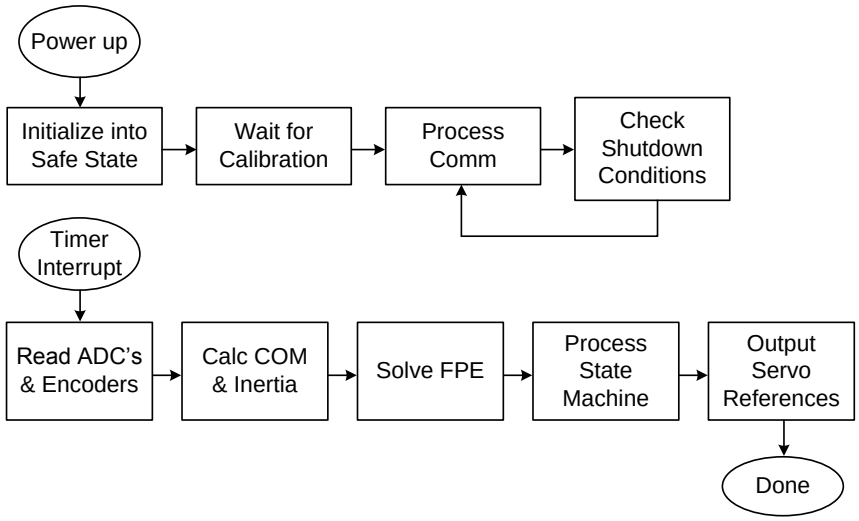


Figure 5.5: A flow chart of the main board firmware.

PC Interface Software

The PC interface for the biped served several purposes. Its primary purpose was to provide basic start and stop commands to the biped. (In hindsight, a physical button on the main board for this function would have been useful during development and testing.) In addition, the interface provided the ability to modify many of YABR's internal parameters. A complete recompile and upload of the code to the dsPIC would take several minutes, so the ability to modify parameters on the fly proved useful during development.

The secondary purpose of the interface was to continuously log the current state of the biped such that data could be analyzed in a less-than-realtime speed for clues on how to modify the control parameters. The analysis tools also proved useful for quickly visualizing simulation results. A screen shot of the interface is shown in Figure 5.6.

The interface was constructed using code shared with the simulations and physical hardware. Therefore, if any modifications were made to the structure of the communication packet or the geometric or physical parameters, then these changes would be immediately reflected in the interface upon recompiling.

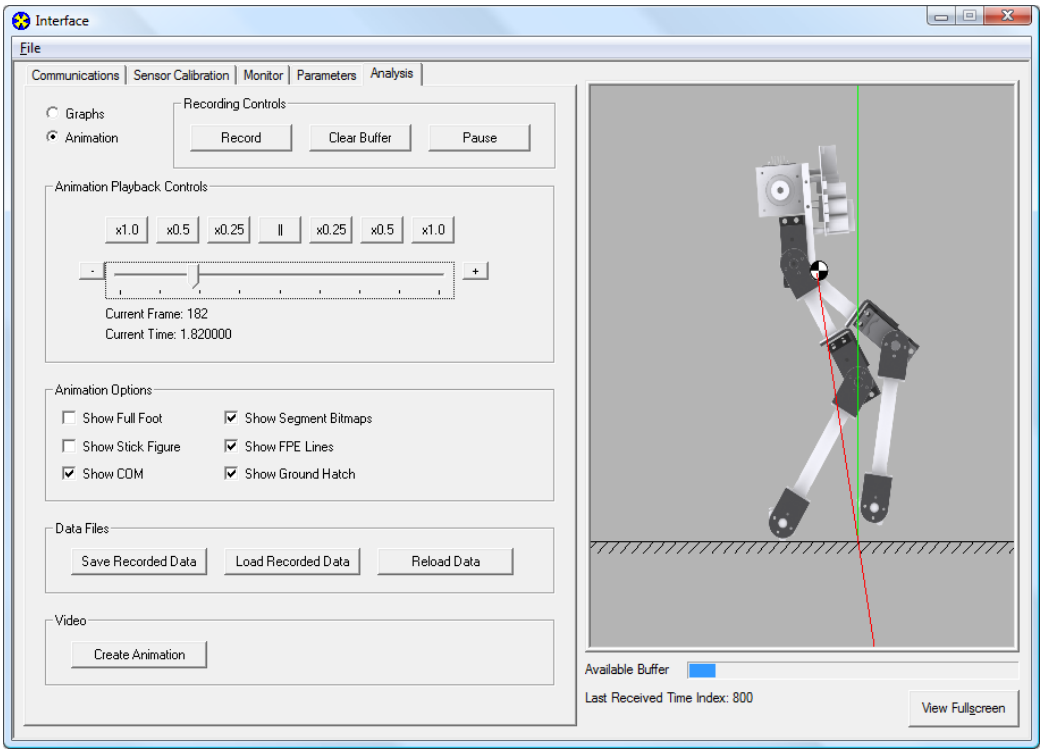


Figure 5.6: A screen shot of the custom interface used to command the biped, alter parameters, record experimental data, and display previous results.

5.1.6 System Calibration

Once all the mechanical and electrical systems were complete, the sensors needed to be calibrated. Using the detailed CAD model of YABR, a series of wooden blocks were designed to fit around the various joints of the biped in order to calibrate the relative joint angles to known values. The servos were manipulated by hand to fit around these blocks and the positions reported by the ADC's were recorded. Calibration points were taken in 45° increments through each servo's range of motion. In post-processing, the recorded values were interpolated using cubic splines to produce lookup tables such a given angle would be reported as the same ADC value for all four servos. The calibration data was compiled into the servo code and uploaded to each of the four motor controllers.

Calibration of the optical encoders for y and θ was much simpler by comparison (x did not

need to be calibrated). The torso plate was aligned to a calibration block that corresponded to a known height and orientation. The resulting encoder offset for y and θ was hard-coded as a constant to be subtracted from the position reported from the quadrature decoder chips.

5.2 Realtime Control

Outside of the hardware specific coding described in Section 5.1.5, the core algorithms for the main board are identical to their simulation counterparts (they are actually the same files). This section describes the specifics of implementing the general outline described in Section 3.5.

5.2.1 Calculating the Center of Mass and Rotational Inertia

The first step to calculating the COM was finding the location of the boom. The x position was calculated using $Encoder_x$ on the hub (rotating about the y axis) divided by the number of encoder counts (8192) times the total circumference:

$$Origin_x = \frac{2\pi(Boom\ Length)}{8192(Encoder_x\ Gear\ Ratio)}(Encoder_x\ Position) \tag{5.1}$$

The y position was calculated in a similar manner using $Encoder_y$ on the hub with the addition of the offset to account for the elevation of the boom above the experiment table:

$$Origin_y = \frac{2(Boom\ Length)\pi}{8192(Encoder_y\ Gear\ Ratio)}(Encoder_y\ Position) + (Origin\ Height) \tag{5.2}$$

Since only a limited range of motion was used for the y axis, Equation 5.2 was actually implemented as lookup table to avoid the floating point operations. The memory requirements made this approach impractical for Equation 5.1.

Next, all the angles from the servos were converted from servo space (1600 counts per revolution) to the $Encoder_\theta$ space (8192 counts per revolution) and then from relative to absolute angles. The absolute angle of each joint is simply the sum of the angles that precede it in the serial chain beginning with $Encoder_\theta$. The sine and cosine of each angle were determined using a lookup table to avoid the computationally expensive sine and cosine floating point operations (in addition to the expense of the operations to convert all the encoder counts into radians). Furthermore, the lookup table minimized the amount of memory used by only storing the first quarter of the sine wave pattern corresponding to 0 through 2047 counts. The GetSin function used the following logic to flip and mirror the values as needed to determine the entire sine pattern:

$$Index < 2048 \qquad return\ SineTable[Index] \tag{5.3}$$

$$2048 \leq Index < 4096 \qquad return\ SineTable[4095 - Index] \tag{5.4}$$

$$4096 \leq Index < 6144 \qquad return\ -SineTable[Index - 4096] \tag{5.5}$$

$$6144 \leq Index \qquad return\ -SineTable[8191 - Index] \tag{5.6}$$

The GetCos function made use of the same table by adding 90° (2048 counts) to the current angle and then calling the GetSin function. The GetSin and GetCos functions returned floating point values.

The next step was to calculate the absolute position of each limb segment COM. Beginning from the torso, the CAD model was used determine the vector from each joint to the corresponding limb segment COM, and to the distal joint. The vectors were stored as constants in the code. To determine the absolute position of segment COM's, each vector was multiplied by a two dimensional rotation matrix:

$$R_{\theta i} = \begin{bmatrix} cos\theta_i & sin\theta_i \\ -sin\theta_i & cos\theta_i \end{bmatrix} \tag{5.7}$$

Which was composed of the sines and cosines that were just determined. The result was then added to the location of the previous joint. The calculations proceeded as follows (the overscore denotes a column vector):

$$\overline{Current\ Joint} = \overline{Origin} \tag{5.8}$$

$$\overline{TorsoCOM} = \left[R_{\theta\ Encoder\theta}\right]\overline{TorsoCOM} + \overline{Current\ Joint} \tag{5.9}$$

$$\overline{Current\ Joint} = \left[R_{\theta\ Encoder\theta}\right]\overline{Torso\ To\ Hip} + \overline{Current\ Joint} \tag{5.10}$$

$$\overline{ThighCOM} = \left[R_{\theta\ Hip}\right]\overline{Hip\ To\ ThighCOM} + \overline{Current\ Joint} \tag{5.11}$$

$$\overline{Current\ Joint} = \left[R_{\theta\ Hip}\right]\overline{Hip\ To\ Knee} + \overline{Current\ Joint} \tag{5.12}$$

$$\overline{ShankCOM} = \left[R_{\theta\ Knee}\right]\overline{Knee\ To\ ShankCOM} + \overline{Current\ Joint} \tag{5.13}$$

$$\overline{Foot\ Position} = \left[R_{\theta\ Knee}\right]\overline{Knee\ To\ Foot} + \overline{Current\ Joint} \tag{5.14}$$

The total COM could now be calculated using the segment COM positions:

$$\begin{aligned} COM_x &= \Big[(TorsoCOM_x)(Mass_{Torso}) + (Right\ ThighCOM_x)(Mass_{Thigh}) \\ &+ (Left\ ThighCOM_x)(Mass_{Thigh}) + (Right\ ShankCOM_x)(Mass_{Shank}) \\ &+ (Left\ ShankCOM_x)(Mass_{Shank})\Big] \Big/ Mass_{Total} \end{aligned} \tag{5.15}$$

$$\begin{aligned} COM_y &= \Big[(TorsoCOM_y)(Mass_{Torso}) + (Right\ ThighCOM_y)(Mass_{Thigh}) \\ &+ (Left\ ThighCOM_y)(Mass_{Thigh}) + (Right\ ShankCOM_y)(Mass_{Shank}) \\ &+ (Left\ ShankCOM_y)(Mass_{Shank})\Big] \Big/ Mass_{Total} \end{aligned} \tag{5.16}$$

The velocity v_i of the COM was calculated as a difference of the current COM position from the COM position from two samples prior, divided by the two-sample period:

$$v_x = \frac{COM_{x(t)} - COM_{x(t-2)}}{2(SamplePeriod)} \tag{5.17}$$

$$v_y = \frac{COM_{y(t)} - COM_{y(t-2)}}{2(SamplePeriod)} \tag{5.18}$$

Derivatives are notoriously "noisy" in digital systems when using a high sample rate relative to the rate of change of the encoder position count. Taking the derivative over a two sample period

reduces the sample rate and helps to filter the result without introducing significant signal lag.

Now that the position of the total COM was known, the total angular inertia could be calculated. The segment inertia $I_{seg\ COM}$ about each segment COM $SegCOMi$ was hard coded as a constant. Using the parallel axis theorem, the inertia of each segment I_{seg} about the total COM was found using:

$$I_{seg} = I_{seg\ COM} + Mass_{seg}\left[(COM_x - SegCOM_x)^2 + (COM_y - SegCOM_y)^2\right] \tag{5.19}$$

Then total angular inertia I_{Total} was calculated as the sum of the individual weighted segments.

$$I_{Total} = I_{Torso} + I_{Right\ Thigh} + I_{Left\ Thigh} + I_{Right\ Shank} + I_{Left\ Shank} \tag{5.20}$$

Finally, the average angular velocity was calculated using a weighted sum of the angular velocity of each segment proportional to the fraction of the segment's angular inertia to the total angular inertia (Equation 3.77) [208]. The angular velocity for each segment $\dot{\theta}_i$ was calculated as a difference from the current servo position $\theta_{Joint(t)}$ to the position two samples prior $\theta_{Joint(t-2)}$, divided by the two-sample period, and converted to radians. (Recall that all the angles have been converted from relative to absolute.) Again, a two-sample period was used to help filter the results.

$$\dot{\theta}_{seg} = \frac{I_{seg}\left(\theta_{Joint(t)} - \theta_{Joint(t-2)}\right)(2\pi)}{8192(2)(SamplePeriod)I_{Total}} \tag{5.21}$$

The average angular velocity $\dot{\theta}_{avg}$ was calculated as a sum of the segments.

$$\dot{\theta}_{avg} = \dot{\theta}_{Torso} + \dot{\theta}_{Right\ Thigh} + \dot{\theta}_{Left\ Thigh} + \dot{\theta}_{Right\ Shank} + \dot{\theta}_{Left\ Shank} \tag{5.22}$$

The order of operations was altered slightly in the code to improve the computational efficiency, but the result was the same. With the COM position, COM velocity, total inertia, and average angular velocity calculated, all of the necessary components were now available to evaluate the FPE equation.

5.2.2 Solving the FPE Equation

The FPE calculation shown in Equation 3.76 was solved using numerical methods for nonlinear equations because a closed form solution for ϕ (Figure 3.11) does not appear to exist. For YABR, a basic line search solver was coded using a bisection approach.

Referring to Figure 5.7, the bisection search began by setting one boundary at zero, and the other boundary at $\pm 89.9°$ (in the code, it was actually ± 1.57 radians) depending on the sign of v_x. Recall that the solution of $|\phi| < 90°$. Technically, $\dot{\theta}_{avg}$ should be taken into account when selecting $\pm 89.9°$, but since it is coupled to v_x during walking, this is only necessary for free-flight.

Both initial bounds for ϕ were evaluated to determine which was the high position (greater than zero) and which was the low position (less than zero). The next guess was the bisection of the upper and lower points. This guess was evaluated to determine if it was greater or less than zero at which point it became the new high or low position respectively.

The bisection continued until 20 iterations had occurred or the absolute value of the solution was less than a tolerance of 10^{-7}. Beyond this tolerance, there was no discernable difference in subsequent kinematic calculations. The solver typically found a solution within 17 iterations. It occasionally required 19 iterations, and rarely required 20. The consequence of requiring more than 20 iterations would be a lower than desired precision, but this was preferable to a solution

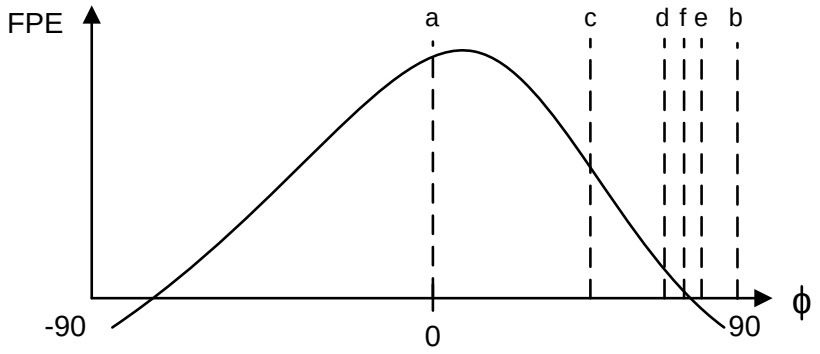


Figure 5.7: The search is initialized at a, b. The line search iteratively bisects the current upper and lower limits to find the zero crossing. ab ⇒ c, cb ⇒ d, db ⇒ e, de ⇒ f

that took so long that the robot did not have time to utilize it.

The final step was to evaluate the ground position of the FPE relative to the COM. This was done using a trigonometric identity:

$$FPE = COM_y \ tan(\phi) \tag{5.23}$$

5.2.3 Evaluating the FPE on the dsPIC

The dsPIC could not perform floating-point operations natively, but floating point operations were needed to evaluate the FPE. However, they are computationally expensive and their use needed to be minimized. Some gains were found through the ordering and type of operations performed. The number of cycles per floating point operation are summarized in Table 5.1. From this table it can been seen that multiplication should be used in place of division when possible, and the number of sines and cosines should be kept to an absolute minimum.

Instruction	Cycles	Time (μs)
Addition	122	4.07
Subtraction	124	4.13
Multiplication	109	3.63
Division	361	12.03
Sine	2238	74.60
Cosine	3249	108.30

Table 5.1: The number of cycles per floating point operation in the dsPIC. The time per instruction is based on a 30 MIPS processor speed.

Consider FPE equation again (Equation 3.76):

$$\frac{\left[mh(v_x \ cos \ \phi + v_y \ sin \ \phi)cos \ \phi + I_{\text{COM}}\dot{\theta}_{avg}cos^2\phi\right]^2}{mh^2 + I_{\text{COM}}cos^2\phi} + 2mgh \ cos \ \phi(cos \ \phi - 1) = 0 \tag{5.24}$$

During each iteration in the solver, the only variable that changes is ϕ. If all of the other variables

are precalculated and stored as constants, the FPE equation would take the form:

$$\frac{\left[A(v_x\ cos\ \phi + v_y\ sin\ \phi)cos\ \phi + Bcos^2\phi\right]^2}{C + I_{\text{COM}}cos^2\phi} + D\ cos\ \phi(cos\ \phi - 1) = 0 \tag{5.25}$$

Where A, B, C, and D are the constants. Assuming that $sin\ \phi$ and $cos\ \phi$ are only calculated once each iteration, the function in this form would require at least 7552 cycles or $252\mu s$. By trading the division for multiplication and rearranging terms:

$$cos\ \phi\left[mhv_y\ sin\ \phi + \left(mhv_x + I_{\text{COM}}\dot{\theta}_{avg}\right)cos\ \phi\right]^2 + \left(2m^2gh^3 + 2I_{\text{COM}}mgh\ cos^2\phi\right)(cos\ \phi - 1) = 0 \tag{5.26}$$

Then substituting single variables for the constants in Equation 5.26 again:

$$cos\ \phi\left[A\ sin\ \phi + B\ cos\ \phi\right]^2 + (C + D\ cos^2\phi)(cos\ \phi - 1) = 0 \tag{5.27}$$

This would require at least 6740 cycles or $224\mu s$, which is an 11% improvement over the original form. At the maximum of 20 iterations, the solution would take approximately $4.5ms$ to find. This was the form that was implemented in code. The variables A through D were calculated once at the start of each search as follows:

$$a = mh \tag{5.28}$$

$$B = a\ v_x + I_{\text{COM}}\dot{\theta}_{avg} \tag{5.29}$$

$$c = 2ag \tag{5.30}$$

$$D = c\ I_{\text{COM}} \tag{5.31}$$

$$C = ach \tag{5.32}$$

$$A = a\ v_y \tag{5.33}$$

5.2.4 The State Machine

Once the COM and FPE had been calculated, the next step was to evaluate the state machine and calculate the appropriate kinematics. The state machine shown in Figure 5.8 is a slightly more advanced version of the one introduced in Section 3.5 in order to deal with external disturbances. In particular, more conditions allow the biped to enter the gait cycle so that if a disturbance moves the FPE outside of the base of support, the biped will take a step. The other key modification is the added paths from the push state to the lift state of the opposite foot. This allowed YABR to deal with disturbances during the push state that required a step backwards. In the state machine, all of the joint angles were relative, and $Encoder_\theta$ is in servo space.

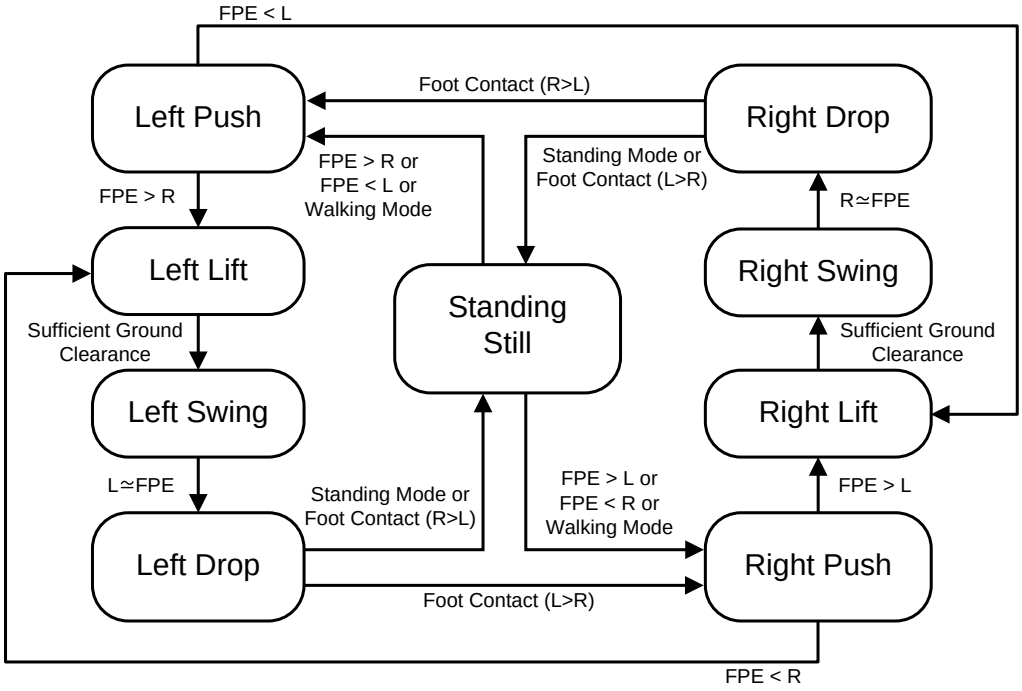


Figure 5.8: A more robust state machine coordinating the actions of the robot. Forward motion corresponds to a positive displacement for the FPE and feet positions.

Standing Still

When the system was first initialized, the joint references were set to the angles shown in Figure 5.9. Note that the biped was designed to walk with bent knees so that it could extend the length of its leg in order to push. This was necessary due to the lack of an actuated foot.

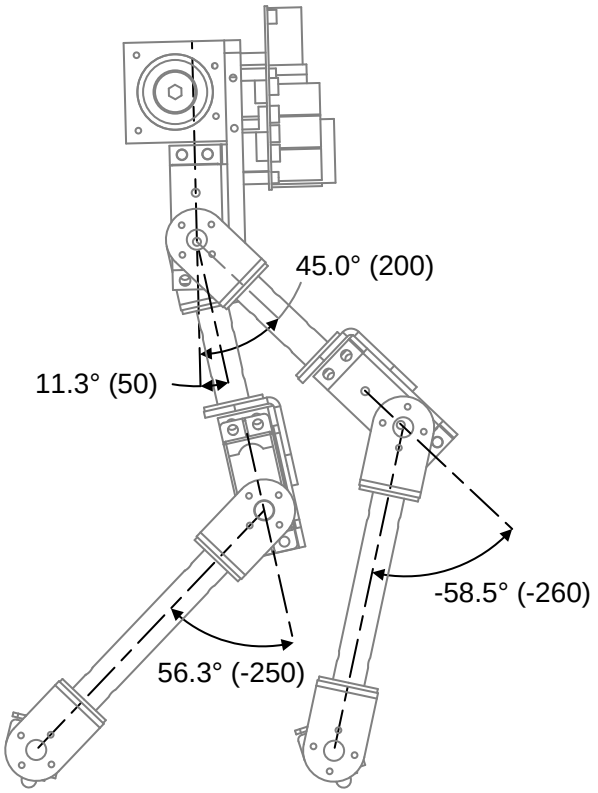


Figure 5.9: The angles used to initialize YABR. The angles are in degrees (ADC counts).

Outside of initialization, anytime the biped entered the standing still state, the previous references were held constant. The biped exited the standing still state if it received a command from the user interface to walk, or if the FPE moved outside of its base of support.

Push Off

The push off is arguably the most complex portion of the kinematics due to the over-actuated closed-chain structure when both feet are on the ground. The easiest solution would have been to set the gain of the pushing hip to zero and just use the pushing knee servo to extend the leg, thus no longer over-constraining the problem. However, it was determined experimentally that the torque of both motors was required in some configurations, and therefore the kinematic problem had to be dealt with directly.

Referring to Figure 5.10, when the biped entered into the push state, a triangle was formed by L_{Base}, L_{Stand}, and L_{Push}. As the push leg was extended, L_{Stand} and L_{Base} should remain constant if no slipping occurs. The angles at the corners of L_{Base} were modeled as pin joints, so the only angle of interest in the triangle was β, from which the proper hip angles could be calculated.

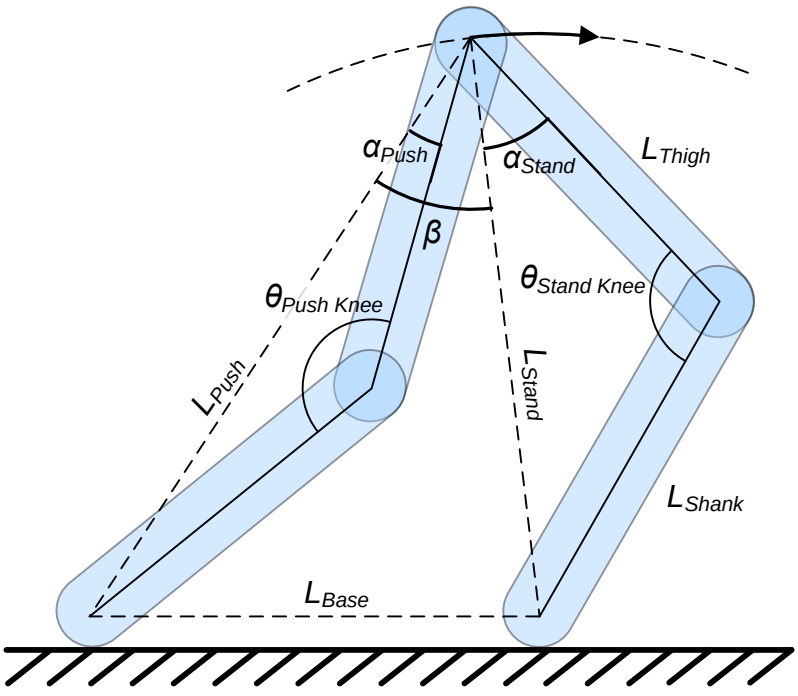


Figure 5.10: The parameters used during the push state.

During each sample of the push state, the reference angle of the standing hip $\theta_{Ref\ Standing\ Hip}$ was calculated to regulate the torso to an upright position using the current standing hip position $\theta_{Pos\ Standing\ Hip}$ and $Encoder_\theta$:

$$\theta_{Ref\ Standing\ Hip} = \theta_{Pos\ Standing\ Hip} + Encoder_\theta \tag{5.34}$$

And the reference angle of the pushing knee was incremented to a maximum of zero (a straight leg):

$$\theta_{Push\ Knee} = \begin{cases} \theta_{Push\ Knee} + PushRate & \theta_{Push\ Knee} < 0 \\ 0 & \theta_{Push\ Knee} \geq 0 \end{cases} \tag{5.35}$$

Next, β was solved using the cosine law. L_{Base} was simply the difference of the push foot position from the standing foot position, both of which were calculated during the COM calculations. L_{Stand} and L_{Push} were precalculated using a vector rotation and stored as a lookup table:

$$L_{Leg} = \left| \overline{L_{Thigh}} + \left[R_{\theta\ Knee}\right]\overline{L_{Shank}} \right| \tag{5.36}$$

$$= \Big[\big[(L_{Thigh})_x + (L_{Shank})_x cos(\theta_{Knee}) - (L_{Shank})_y sin(\theta_{Knee})\big]^2 + \big[(L_{Thigh})_y + (L_{Shank})_x sin(\theta_{Knee}) + (L_{Shank})_y cos(\theta_{Knee})\big]^2\Big]^{\frac{1}{2}} \tag{5.37}$$

$$= \text{LegLengthLookup}(\theta_{Knee}) \tag{5.38}$$

Therefore:

$$L_{Stand} = \text{LegLengthLookup}(\theta_{Stand\ Knee}) \tag{5.39}$$

$$L_{Push} = \text{LegLengthLookup}(\theta_{Push\ Knee}) \tag{5.40}$$

$$L_{Base} = (Stand\ Foot)_x - (Push\ Foot)_x \tag{5.41}$$

$$\beta = cos^{-1}\left[\frac{L_{Base}^2 - L_{Stand}^2 - L_{Push}^2}{-2L_{Stand}L_{Push}}\right]\frac{\pi}{800} \tag{5.42}$$

Note that Equation 5.42 includes the conversion from radians to servo space. The hip offsets α_{Push}

and α_{Stand} from the corresponding L_{Push} and L_{Stand} to the thigh position were also calculated using the cosine law. Unlike β though, the possible positions were quantized by the knee angle so they could be precalculated and stored as a lookup table:

$$\alpha_i = cos^{-1}\left[\frac{L_{Shank}^2 - L_{Thigh}^2 - \text{LegLengthLookup}^2(\theta_i)}{-2L_{Thigh}\text{LegLengthLookup}(\theta_i)}\right]\frac{\pi}{800} \tag{5.43}$$

$$\alpha_{Stand} = \text{HipOffsetLookup}(\theta_{Stand\ Knee}) \tag{5.44}$$

$$\alpha_{Push} = \text{HipOffsetLookup}(\theta_{Push\ Knee}) \tag{5.45}$$

Finally, the hip angle of the pushing leg could be calculated:

$$\theta_{Push\ Hip} = \theta_{Standing\ Hip} - \alpha_{Stand} - \beta + \alpha_{Push} \tag{5.46}$$

Under normal conditions, the state transitioned to the lift state of the pushing foot when the FPE moved in front of the standing foot. Under some external disturbance conditions though, the FPE could move behind the push foot. When this occurred, the pushing leg joint references were frozen at the current joint position, the FPE offset was inverted (see the swing state) and the state transitioned to the lift state of the opposite foot so that the robot could step backwards to compensate for the disturbance.

Lift Foot

During the lift state, the standing leg regulated the torso to maintain an upright position using Equation 5.34, and the swing leg attempted to maintain the foot at a constant distance from the walking surface by changing the swing knee angle. The gain of the swing hip was set to zero to relieve any pressure applied to the swing leg and to help release the foot from the surface friction. Referring to Figure 5.11, the knee angle θ_{Knee} was calculated to move the foot to the target $Lift$ height. The target vertical distance from the hip joint to the contact point on the foot was:

$$h_{Foot} = Hip_y - h_{Lift} \tag{5.47}$$

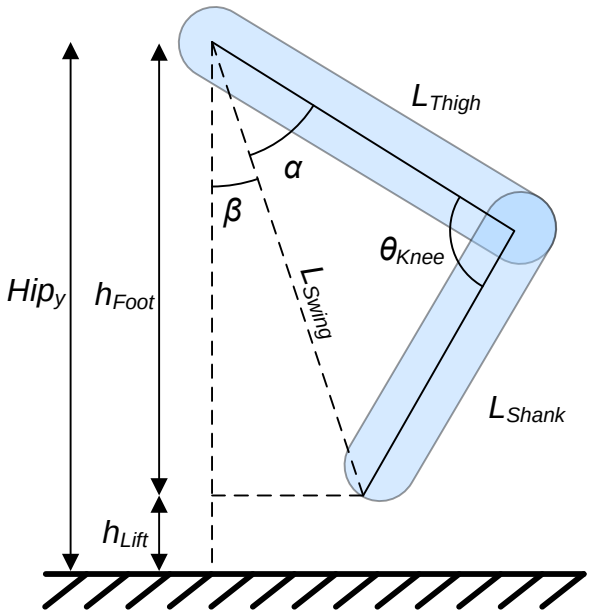


Figure 5.11: Calculating the joint angles to maintain a constant foot height above the ground.

And the angles α and β were calculated as follows:

$$\alpha = \text{HipOffsetLookup}(\theta_{Knee}) \tag{5.48}$$

$$\beta = \theta_{Hip} + Encoder_{\theta} - \alpha \tag{5.49}$$

From which the cosine identity was used to calculate the nominal L_{Swing}:

$$L_{Swing} = \frac{h_{Foot}}{\text{GetSin}(\beta)} \tag{5.50}$$

The value L_{Swing} was quantized and used as the input to another lookup table that was the inverse of Equation 5.36 in order to find the appropriate θ_{Knee} for the target leg length. The state transitioned to the swing state once the swing foot was above a given height threshold.

Swing

The kinematics of the swing state used the same algorithm from the lift state to regulate the foot height using the knee, and the swing hip was used to track the FPE. Referring to Figure 5.12, recall that the position of the FPE and Hip_x are relative to the COM.

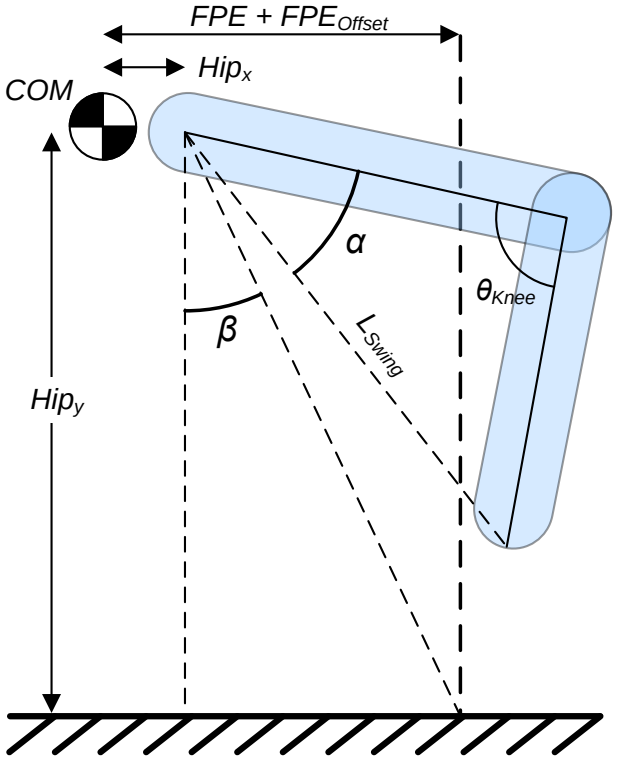


Figure 5.12: Calculating the swing hip angle to track the FPE position.

The goal was to orient the swing foot such that L_{Swing} was aligned with the line from the hip to the FPE position on the ground. First, the angle β was calculated using the FPE position plus a small offset to ensure each step was stable:

$$\beta = tan^{-1}\left[\frac{FPE + FPE_{\text{Offset}} - Hip_x}{Hip_y}\right]\frac{800}{\pi} \tag{5.51}$$

Note the result in Equation 5.51 is in servo space. The angle α was found using the lookup table for the hip offset:

$$\alpha = \text{HipOffsetLookup}(\theta_{Knee}) \tag{5.52}$$

And finally the reference angle for the hip was calculated as follows:

$$\theta_{Hip} = \alpha + \beta - Encoder_{\theta} \tag{5.53}$$

Note that there is no inherent bias in the swing state to a particular direction of walking. The kinematics were the same for forwards and backwards walking aside from the FPE offset constant, which was inverted if the standing or push state determined a backwards step was required.

The state transitioned when the swing hip angle was within a threshold distance from the reference angle.

Foot Drop

During the drop state, the standing hip continued to regulate the torso upright using Equation 5.34. The swing hip continued to track the FPE position as shown in the swing state. The swing knee extended the leg slightly beyond its normal standing length and switched to a low gain to help dampen the impact.

The state transitioned when the swing foot contacted the ground. Upon contact, the hip reference angles were set to the current hip positions to stop any further tracking, and the gain of the swing knee was increased. If the user issued a stop command or the swing foot landed on the wrong side of the standing foot (due to an external disturbance), then the state changed to the standing state. Otherwise, the state transitioned to the push state of the opposite foot.

5.3 Experimental Results

The experiment successfully demonstrated the feasibility of using the FPE as a measure of balance for a bipedal robot. A number of modifications were made to the physical robot in order to compensate for the differences between the simulated model and reality. For instance, servo gains in almost every state were higher than in the simulation to deal with unmodeled friction. Another key adjustment was the knee reference angle during standing. The angle was decreased by 50 ADC counts, or approximately 11° to make the standing leg length slightly longer. This moved the COM higher and reduced the pendulum frequency, but at the expense of the maximum pushing velocity. With almost every system at its limit, this slight reduction in the pendulum frequency made a significant improvement in reliability.

5.3.1 Walking

YABR was generally capable of walking the entire circumference of the track. When falls did occur, it was apparent that it was the result of the preceding step. Sometimes YABR would take a larger step than normal and as a result, YABR required a longer extension of the pushing leg in order to move the FPE in front of the standing foot. Once this occurred, the state machine would continue with the lift and swing states, but due to the extra extension, the motors simply could not move fast enough to get the leg in front of the FPE before the impact occurred. The kinetic energy of the system would continue to increase, and a fall was inevitable.

The other problem that was quite challenging was the noise in the ADC in the servo controllers. The noise gave the appearance that large, high-velocity motions were being performed by the legs. Since the legs were purposely designed to include a significant portion of the total mass to challenge the FPE approach, the noise had a significant effect on the apparent FPE position. The noise can be seen in the FPE and the position of the feet in Figure 5.13.

The noise also affected the tracking reference during the swing and drop states, but it was filtered out by the mechanical dynamics. However, the noise did present a problem for the threshold that triggered the transition from the push to lift states. Premature triggering of a state transition could require a step backwards, but excessively delayed triggering might not leave

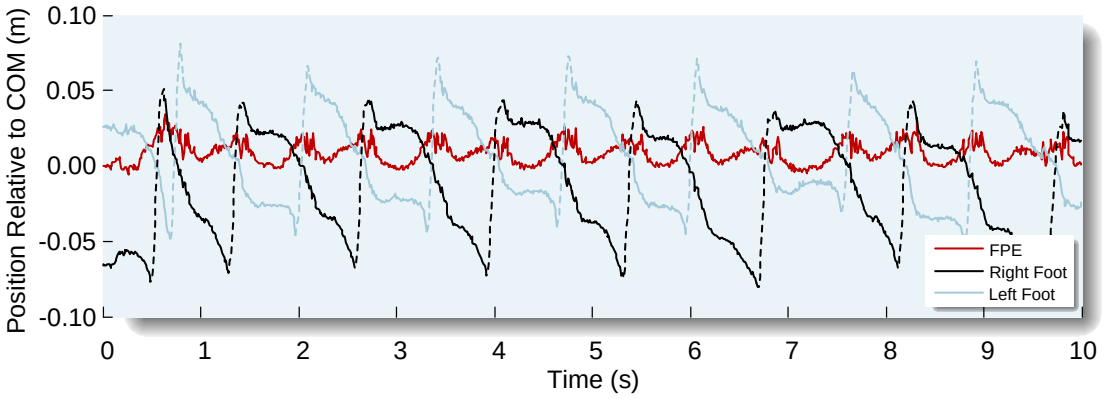


Figure 5.13: The position of the FPE and the feet from the experimental results of a slow gait. The dashed portions of the foot position indicate when the foot was off the ground, and the solid portions indicate when the foot was on the ground.

enough time to get the foot in front of the FPE. By using a larger FPE offset, a relatively slow, but consistent gait could be achieved as shown in Figure 5.13. The extra FPE offset dissipated a lot of energy each step which needed to be replaced by the push. Since there was no regulator to vary the push velocity, a constant, slow push was used to ensure that the subsequent motor velocity requirements during the swing phase did not exceed the available capacity. The asymmetry of the left and right foot is a result of the slow gait and lack of regulator for the average gait speed.

Figure 5.14 examines the consistency of the gait by plotting the joint angles against the proportion of the gait cycle. Again, the asymmetry is apparent. The consistency is not as good as in simulation, but a definite pattern is present. Note that the indicated state positions are the average points in the gait cycle, so the widespread features, particularly on the left leg, do not directly correlate to the indicated states.

By reducing the safety margin of the large FPE offset, a truly dynamic gait similar to the results from the simulation could be achieved. In Figure 5.15, the biped initially started with a slower gait similar to the results from Figure 5.13, but eventually it transitioned into a purely dynamic gait. From the gait diagram also shown in Figure 5.15, it can be seen that this dynamic gait covered a very large distance compared to the slower gait. This switching between fast

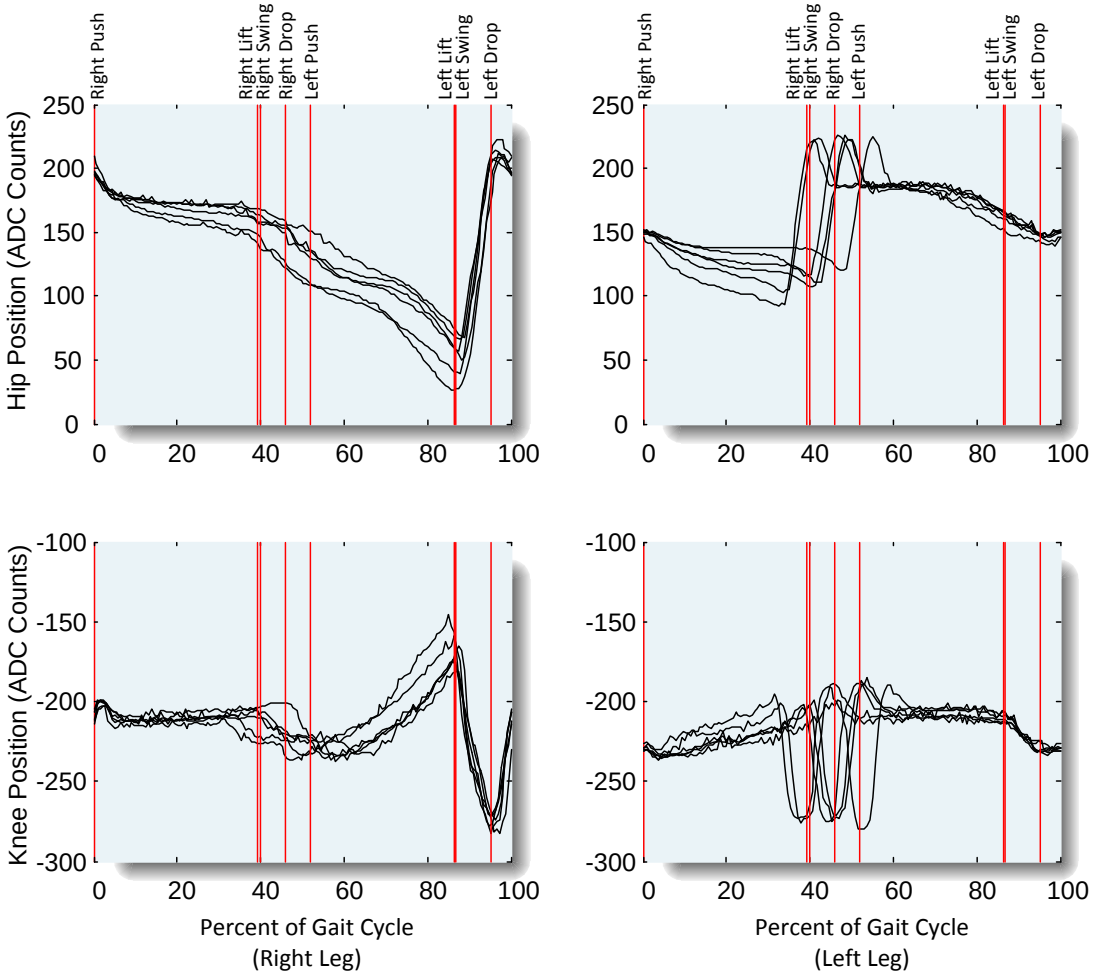


Figure 5.14: The joint angles of the biped over 12 steps.

and slow gaits was frequently observed when using the reduced FPE offset. Since there was no mechanism in place to explicitly control the gait velocity, it is suspected that this may be similar to the gait speed bifurcation observed in passive biped models [219, 220].

As previously described, noise in the FPE presented the problem of occasionally triggering a step prematurely. This occurred in the results shown in Figure 5.15 shortly after gait initiation. The step with the left foot was completed just after $1s$, but the FPE was now behind the standing foot. In response, the left foot was again lifted in order to step behind and restore balance.

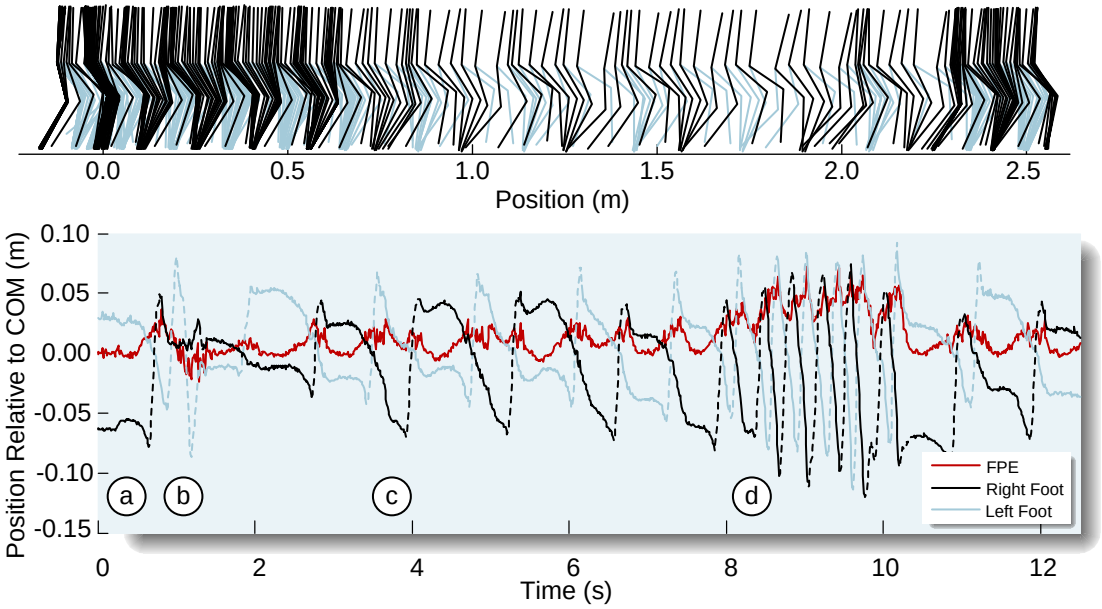


Figure 5.15: Experimental results demonstrating a more dynamic gait. (a) The biped begins with gait initiation. (b) Shortly after, noise in the system triggers a premature step which requires a step backwards to correct. (c) After 10 steps, (d) the biped enters into a dynamic gait similar to the simulation results.

YABR's gait proceeded normally after this point.

One of the difficulties of comparing algorithms in the field of bipedal robotics is the lack of a common platform. Despite this, papers frequently report the absolute speed of their bipeds. A more useful metric proposed by Geng [62] is the normalized walking speed based on leg-lengths per second. Using a nominal leg length of $220mm$ (the distance from the hip to the foot with a bent knee), the results from Figure 5.15 indicate that the slow gait had a speed of 0.56 *Leg Lengths/s*, and the fast gait has a speed of 3.7 *Leg Lengths/s*. The slow gait was particularly slow compared to other published results, but the majority of the time was spent in the push state due to the slow constant pushing rate. A more intelligent rate controller could significantly improve this. On the other hand, to the knowledge of the author, the fully dynamic gait is the fastest robotic walking gait currently published. This includes the bipeds based on McGeer's passive walkers (0.75 *Leg Lengths/s*), Pratt's Spring Flamingo based on his Virtual Model Control

(1.39 *Leg Lengths/s*), Honda's Asimo (1.03 *Leg Lengths/s*), and even Geng's RunBot, which achieved remarkable walking speeds (3.48 *Leg Lengths/s*). During the course of the experiments, YABR was never observed to enter a ballistic state, thus it might be possible to achieve even higher walking speeds given faster actuators.

5.3.2 Starting and Stopping

The results from gait initiation and termination generally mirrored the trends seen in simulation. Gait initiation could be achieved with a wide variety of parameter configurations, but only a small subset were also suitable for termination. Gait initiation can be seen in Figures 5.13, 5.15, and 5.16. Gait termination is demonstrated in Figure 5.16. This experiment used the same parameters as used for the fast walking from Figure 5.15, but these parameters were not one-step compatible. After the stop command was issued, an extra step was required before a stable stance was achieved. Note the slightly wider stance at gait termination was due to reduced hip controller gains in the standing still state, which allowed the legs to be pried apart by the dynamic forces.

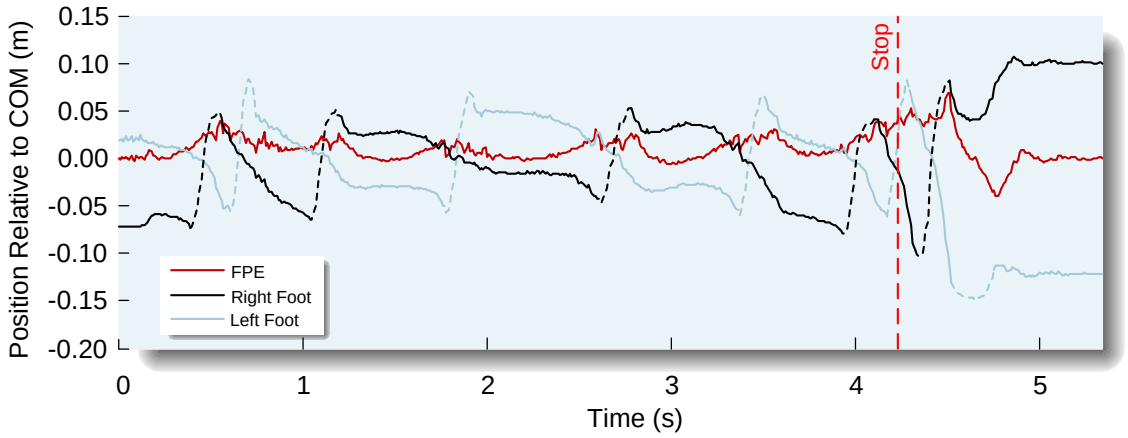


Figure 5.16: Gait initiation and termination. After the stop command was issued, an extra step was required before a stable stance was achieved.

5.3.3 Disturbance Rejection

The walking results validated the use of the FPE as a feedback mechanism to achieve a balanced gait, but as with simulation, the real power of this approach is the ability to deal with external disturbances. Disturbance testing was performed by manually pushing the biped with a single impact to the torso, horizontal to the ground. This testing was performed more for anecdotal curiosity than scientific rigor. Since the purpose of the prototype was simply to validate the use of the FPE, a quantified examination of the range of disturbances that the biped could deal with is beyond the scope of this work.

Despite the lack of quantification, YABR demonstrated an impressive resilience to external disturbances. As with the simulation, light impacts during walking had little effect, but slightly harder impacts would cause falls depending on the current state of the gait cycle. Given the variability present in the experimental walking results, it was difficult to distinguish the results of minor impacts from the undisturbed gait.

The results were much more clear when the biped started in the standing still state. Results of forward and backward pushing are shown in Figures 5.17 and 5.18 respectively. The forward push was of moderate force. After a single step, YABR rocked forward slightly, then came to rest in a stable position. The backward push by comparison was very hard. The first step did not quite catch the FPE so a second step was required. The relative force of these impacts is apparent from the angle of the torso following the impacts.

Given that procedural testing of the robustness of bipedal robots to external disturbances is rarely performed, let alone measured in a quantifiable way, it is difficult to know how YABR's performance compares to other implementations. However, based on the works seen in the literature and various supplemental videos, it is the author's belief that this disturbance rejection would rank quite highly.

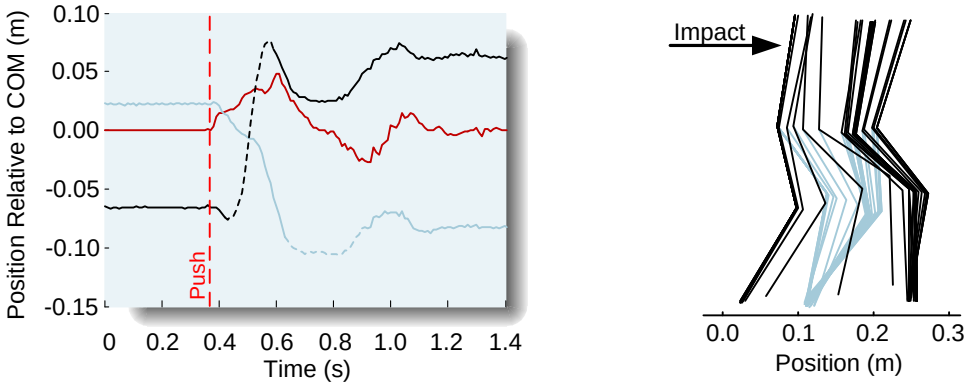


Figure 5.17: Experimental results in response to an external disturbance from behind.

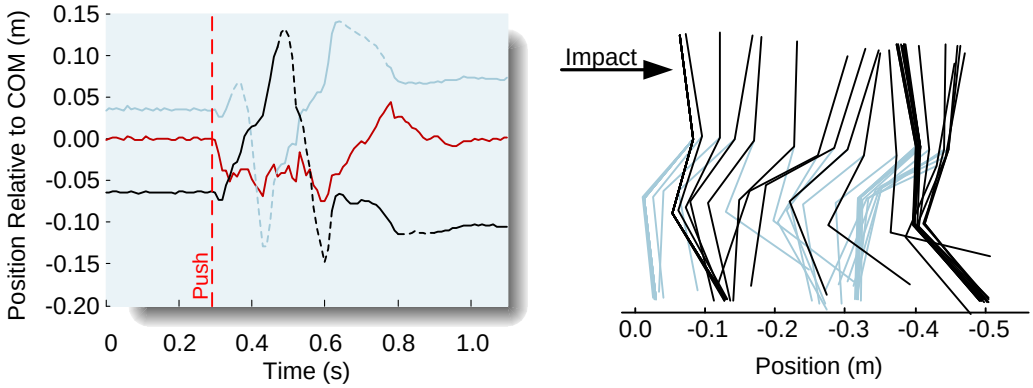


Figure 5.18: Experimental results in response to an external disturbance from in front.

5.4 Discussion

The design of YABR was dictated more by economic consideration than optimal performance. YABR cost approximately $1200 in materials, but the author spent well over 150 hours in construction time in addition to the hundreds of hours of design time. The experiment succeeded in validating the use of the FPE as a feedback mechanism, but it fared poorly as a research platform. The motors, sensors, and processor were utilized to their limits. A redesign of the electronics could eventually alleviate the noise issues, but there would still be limited expansion possibilities. Replacing the servo potentiometers with optical encoders and upgrading the reso-

lution of the existing encoders would cost approximately $2000. Replacing the servos with high quality motors and planetary geartrains would cost at least another $1000. Upgrading the processor would actually be the most expensive improvement. The dsPIC was selected because at the design time, it was the most powerful DSP with the necessary capabilities and free development tools. DSP's with floating-point capabilities are available for approximately $50, but the real cost is in the development kits and compilers which are close to $5000 per seat. An ideal version of the robotic platform would be close to $10,000 just in material costs. Given the financial resources and long-term research goals of this author, it would be hard to justify the cost of the ideal platform solely for this experiment.

There are several specific recommendations that can be made for future researchers attempting to pursue a similar low cost approach. Despite having limited computational resources, the dsPIC processor was an excellent processor that required minimal external components compared to a traditional DSP. At the time of design, only the 30 MIPS processor was available, but Microchip has since introduced a 40 MIPS version into production which would help alleviate some of the computational limitations.

Sensors, unfortunately, do not advance at the same rate as microprocessors. It is unlikely that a low cost solution to the resolution limitation of the hub encoders will be viable in the near future. The use of gearing to increase the encoder resolution was definitely not the best solution. Many weeks were lost in order to eliminate backlash and minimize friction. One possible alternative would be to use two optical encoders stacked on the same shaft out of phase with each other. A dedicated processor or Complex Programmable Logic Device (CPLD) could decode the four pulse streams in a manner similar to quadrature decoding, but increasing the resolution to 16 times the encoder disk resolution rather than just 4.

The boom was originally designed with a $6.35mm$ diameter carbon fiber tube under the assumption that since it only provided lateral support, the forces applied to it would be minimal. However, the vibrations induced in the boom were underestimated, which resulted in oscillations that appeared on the hub encoder outputs and in turn wreaked havoc on the FPE position. Beam flexibility calculations determined that increasing the boom diameter to $18.0mm$ would decrease

the vibration by a factor of 10 while only doubling the weight of the carbon fiber tube (most of the total boom weight was in the metal components). After this diameter change was made, the vibration problem was eliminated (or at least reduced below what could be perceived by the sensors).

It would definitely be preferable to use optical encoders on the leg joints to eliminate the overhead related to correcting the nonlinearity of potentiometers, as well as the risk of sensor noise affecting the FPE calculations. If external optical encoders were used for the leg joints, then there would little reason to use hobby servos for the actuators. Although there are hobby servos with integrated optical encoders, they are very expensive. These servos also typically use brushless DC motors which are integrated into the PCB design making it difficult to replace the position controller. The motors are central to the performance of the biped, and typically the motors found in hobby servos have a very low efficiency. It would also be advisable to use a metal gear train instead of plastic. During one particularly violent external disturbance testing session, one of YABR's plastic knee gears was stripped and needed to be replaced.

A much better alternative to hobby servos are motors from Maxon Motors[8], or the Faulhaber Group [9] with integrated gearheads. Although they are typically very expensive directly from the manufacturers, small gearhead motors frequently appear on online auction sites such as eBay. As discussed in Section 4.3, this application could benefit from exceeding the normal safe operating voltage range of the motors to boost the performance. Maxon and Faulhaber motors are well documented and include specifications into this range. A higher operating voltage increases the risk of damaging motors though, so additional precautions would need to be taken. The robot might need to incorporate some form cooling system or occasionally be allowed to rest so its motors can cool down.

The first iteration of the design sought to utilize as many off-the-shelf components as possible, but as the design progressed it was evident that this was not feasible without compromising the performance of the biped. The purchased brackets that made up the leg structure were part

[8] www.maxonmotorusa.com

[9] www.micromo.com

of the off-the-shelf approach. These parts were manufactured with generous clearances, which meant that the impact of each step caused these components to slowly shift. A small shift at the top of the leg created a noticeable discrepancy at the foot from its nominal position. As a result, the leg assemblies required constant adjustment. Considering the total time spent doing all the machining for YABR, a custom structure for the legs would have provided more rigidity without adding a substantial amount of time to the machining. This would have also provided the opportunity to include a cable channel to protect against wire abrasion, as opposed to the cable sheath that was used which hindered the joint motion.

The few last suggestions are with regards to the communications. Although the Lantronix XPort worked very well to provide ethernet connectivity to the PC, it was challenging to minimize the effect of cable drag on the system. Future designers might consider using the Lantronix WiPort instead to provide wireless connectivity. It would also be prudent to include a manual interface option such as a push-buttons connected to a general I/O port of the processor so that interfacing would not need to be done solely through the PC.

The I^2C network between the main processor and the motors never achieved its full design capability for several reasons. The cable length undoubtedly contributed inductance and/or capacitance effects that resulted in a rounding of the pulse train. Therefore, the I^2C network had to be operated at less than its rated speed. This might be solved with the use of buffers, different gauge wiring, or a protocol more suited to longer transmission lines such as RS-232 or RS-422. The original aim of daisy-chaining the motors together failed due to space limitations, but this idea might be revived again if new motors and a controller were used that were not bound by the space limitations of a hobby servo. At the very least, future designs should use multiple I^2C channels to communicate to multiple motors in parallel because the communication rate between the processor core and the I^2C buffer was much faster than the I^2C communication channel itself.

Chapter 6

Human Application

The foot placement estimator (FPE) derived in Chapter 3 provided the ability for bipedal robots to respond to a broad range of unstable conditions. After demonstrating how this control approach could be applied in simulation to a bipedal robot in Chapter 4, it was successfully applied to a physical robot in Chapter 5. The analysis of human motion in this chapter demonstrates that the FPE is also a remarkable predictor of human foot placement. A human study was conducted in collaboration with PhD candidate Matthew Millard at the University of Waterloo. This study received clearance through the Office of Research Ethics (ORE #13827).

6.1 The Human Parallels of the FPE

The FPE and its application to bipedal robots shares a number of similarities to our present understanding of the mechanisms behind human motion. Traditionally, the human gait cycle is divided into eight phases per leg [221]. However, Davis and Vaughan [221, 222] used a statistical analysis to show the existence of four distinct phases in Winter's EMG gait data [223]. This is an interesting parallel to the four state per leg state machine implemented in Chapters 4 and 5.

It has been suggested that humans maintain internal variables of velocity by integrating the accelerations measured by our vestibular system as well as through direct measurements from our

proprioceptive receptors [224]. While this does not directly endorse the FPE itself, it does suggest that velocity could be a key variable in selecting our gait.

With respect to disturbances, Patla's review of human strategies for dynamic locomotion stability [225] notes that our selection of foot placement dictates our stability. This is more akin to the foot placement approach adopted in this thesis than to the robot torso adjustment utilized by other approaches to deal with disturbances. Furthermore, Hemani et al. discuss the mechanisms involved in a human's step response to an external disturbance [224]. They note that humans utilize limb position and velocity as a feedback mechanism to maintain their balance, and that a step can be triggered when the COM moves in front of the toes or "the velocity exceeds a threshold". The FPE could be considered a direct embodiment of this idea. At low velocities, the dynamics are minimal and the COM dominates the position of the FPE. Conversely, at high velocities, the FPE is dominated by the dynamics. A high COM velocity would move the FPE in front of the toes before the COM arrives there.

6.2 Experimental Setup

The goal of this experiment was to test where humans step in relation to the FPE. By tracking the motion of each limb segment using infrared LED markers, the position and velocity of the subject's COM was calculated in order to determine the location of the FPE. The human motion was captured using three Optotrak 3020[1] position sensors as shown in Figure 6.1. The cameras were positioned to provide the maximum capture volume for a subject walking in a straight line towards the cameras. The resulting three-camera configuration was calibrated using a 20 marker cube with a $45s$ capture period. The average multi-camera discrepancy error was less than $0.5mm$.

The markers were placed on each subject's body as shown in Figure 6.2. The skeletal landmarks for positioning the markers were the 5th distal metatarsal head (near the pinky toe), the trochlear process of the calcaneus (heel), the lateral malleolus (ankle), the fibular head (knee), the greater trochanter (hip), and the acromion process (shoulder). These locations correspond to

[1]Manufactured by NDI, www.ndigital.com

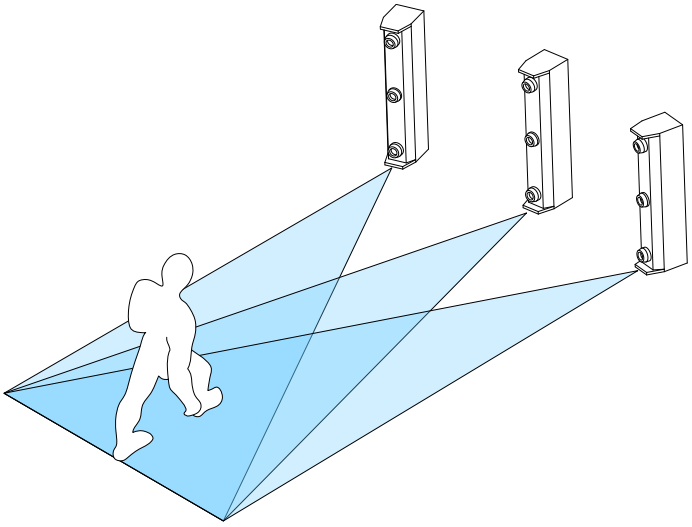


Figure 6.1: Three Optotrak cameras were positioned to maximize the capture volume when walking in a straight line towards the cameras.

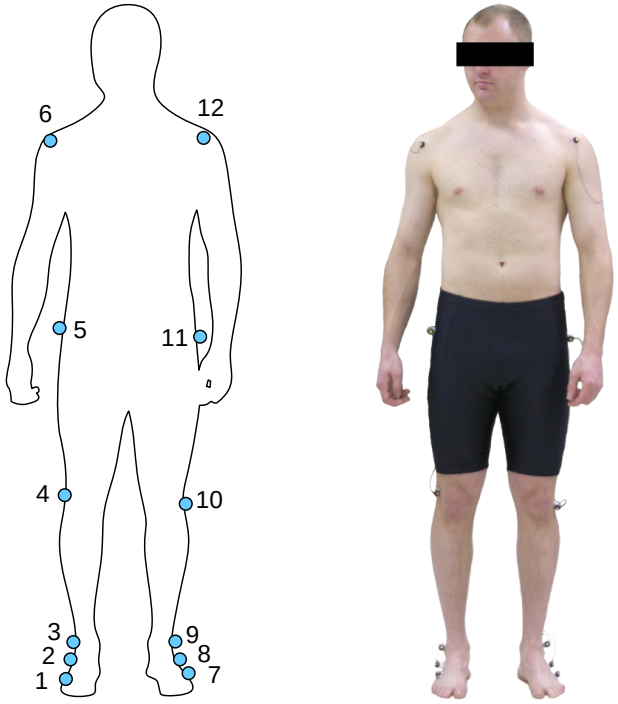


Figure 6.2: The position of the infrared markers on each subject's body.

the landmarks used in Winter's anthropometric data [223].

6.3 Experimental Procedure

All of the participants in the study were healthy individuals with no medical conditions affecting their gait. This was determined by a medical questionnaire included with the participant consent form. A total of 14 subjects were recruited for this study (7 males, 7 females). Each subject's height and weight were measured and recorded. The height of participants ranged from $1.42m$ to $1.92m$, their mass ranged from $60.8kg$ to $114.3kg$, and their age ranged from 19 to 46.

The participants were asked to walk around the room at a natural pace while a silent metronome was set to their pace by one of the observers. The metronome sound was then turned on, and the participants were given the metronome to fine tune it to their natural pace. The participants were then asked to perform at least 10 sets each of:

1. Walking at a natural pace without a metronome through the entire volume.
2. Walking at a natural pace without a metronome including start and stop inside the volume.
3. Walking at the metronome pace through the entire volume.
4. Walking at the metronome pace including start and stop inside the volume.
5. Walking at the metronome pace plus 20% through the entire volume.
6. Walking at the metronome pace plus 20% including start and stop inside the volume.
7. Walking at the metronome pace minus 20% through the entire volume.
8. Walking at the metronome pace minus 20% including start and stop inside the volume.
9. A two-foot jump approximately $1m$ in length.
10. A single step forward in response to a push on the participant's back between thoracic vertebrae T3-T5.

Since no markers were placed on the arms, the participants were instructed to keep their arms crossed across their chest during data collections. This eliminated the possibility of any effect on the FPE and it also minimized potential marker occlusion. Pilot work for this study indicated that arm motion only had a significant effect on the FPE when the arms were moving quickly. The effect of the arms on the FPE during normal walking was negligible.

6.4 Data Processing

The study generated over 1500 data sets, therefore the data processing needed to be as automated as possible. Trials with excessive marker dropout were discarded, but it was impossible to prevent the markers from occasionally being briefly occluded. Since the FPE calculation relied on the entire data set being valid, the gaps needed to be patched with spline interpolations.

First, each data set was cropped to the first and last point where every marker was visible. Next, any sequences in the data set where less than three sequential data points existed were removed. Finally, each patch was created using a spline to match the end points as well as the first derivative. Although the patching process was automatic, the results were presented for visual verification of each set to ensure there were no anomalies resulting from the patching process.

The calculation of the FPE from the human data followed the same procedure used in Chapters 4 and 5. The mass and inertia for each limb segment were determined using each subject's height and weight in combination with Winter's anthropometric data [223]. The marker data was used to determine the position and orientation of each limb segment, from which the total COM and angular inertia were calculated. Finally, the FPE position was calculated using the same algorithm described in Section 5.2.2.

The patched data sets and FPE results were then passed on to Matthew Millard for foot contact extraction and the subsequent statistical analysis. The details of this portion of the analysis will be covered in our joint conference publication submitted to the 2008 North American Congress on Biomechanics entitled "Evaluation of a Human Foot Placement Model" [226], and our joint journal publication entitled "A Model for Sagittal Plane Human Foot Placement".

6.5 Experimental Results

The statistical results calculated by Matthew Millard indicate that there is a 98% correlation between the actual foot placement and the FPE, and a 0.0% chance that they are uncorrelated [226]. In general, the foot placement for each individual was very consistent with respect to the FPE, but the different trial types had different offsets for each individual. Figure 6.3 plots the average step position for each individual with respect to the FPE. Some interesting trends can been seen in the data. As the walking speed increased from slow through to a medium pace, the FPE offset increased, but at the fast pace, the offset suddenly decreased again. This trend was repeated for both the cases where subjects walked through the volume, and the start-stop cases.

Determining when foot contact occurred using an automated script proved to be challenging. Some thresholds and techniques that worked well for one individual would fail for another. Many of the outliers in Figure 6.3 are a result of this difficulty. For instance, Figure 6.4 shows all of the

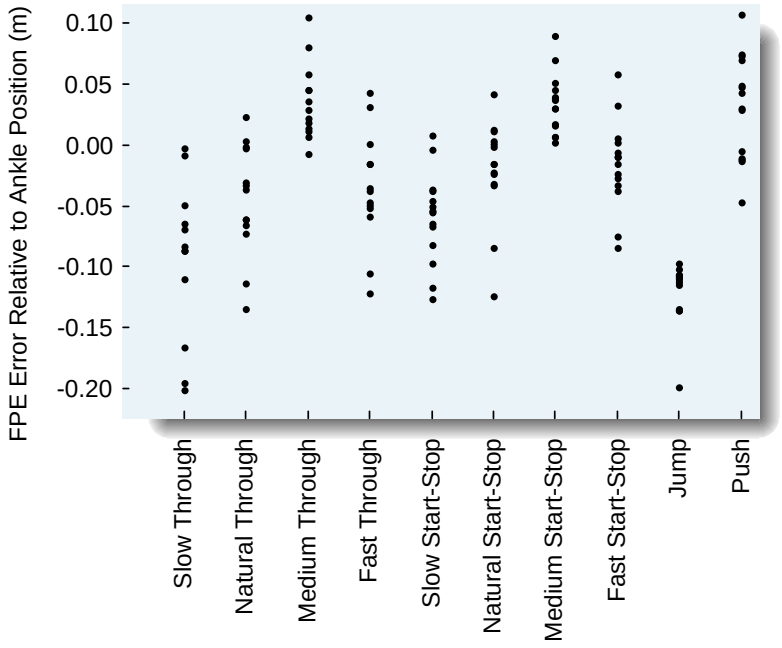


Figure 6.3: The average error of FPE position relative to the ankle position for each subject for each trial. Negative error indicates the subject stepped behind the FPE.

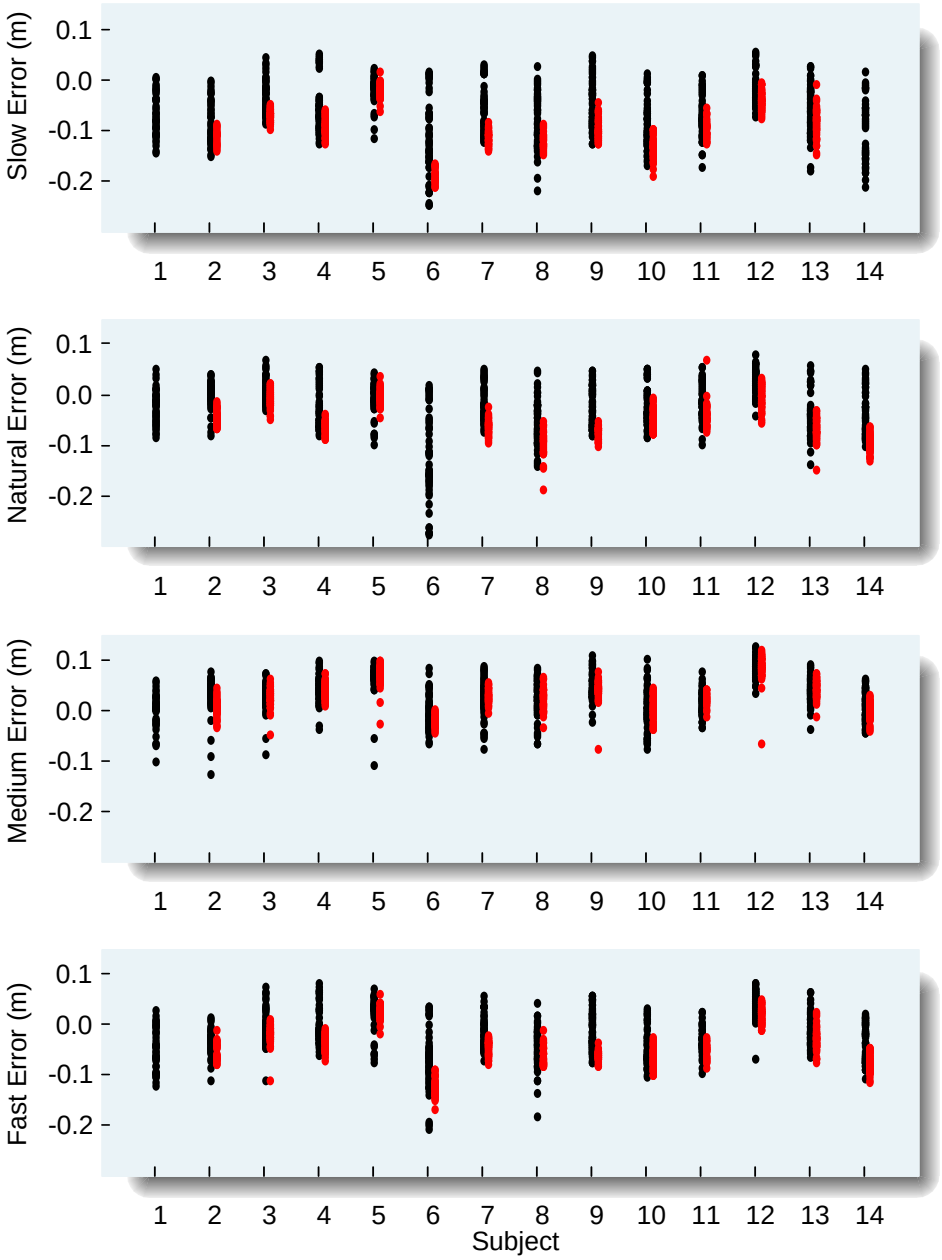


Figure 6.4: The error of FPE position relative to the ankle position for each step. The start-stop cases are shown in black, and walking through the volume is shown in red.

foot placements with respect to the FPE for each subject. The subjects generally demonstrated excellent repeatability, but subjects 6 and 8 consistently presented problems. A manual analysis of the data determined that these subjects rolled their ankles upon landing which masked the true landing time. Future studies should include some form of foot contact sensor to eliminate this problem. As expected, Figure 6.4 also shows that there was less offset variability for walking through the volume, compared to when starting and stopping were included.

Referring back to Figure 6.3 again, the push typically resulted in a step further than the FPE, which was expected. The participants were instructed to only take a single step forward, so stepping further than the FPE ensured stability. The jump on the other hand was consistently behind the FPE. This may seem counter-intuitive, but most people's feet span at least $10cm$ from their ankle to their toes. The results in Figure 6.3 place the FPE just behind the ball of the foot and near the middle of the base of support. Again, this helped to ensure the subject's stability.

Given all the approximations and potential for error, the FPE was remarkably accurate for predicting when a subject would fall. There were just a few cases where a subject required a second step after the jump, but in all these cases, the need for an additional step was predicted by the FPE. One particular example is shown in Figure 6.5. Once the subject left the ground, only conservation of energy and momentum dictated the path of the COM, so the FPE snapped to a location near the point of impact. As the subject approached the impact, the FPE drifted forward slightly because the virtual leg from the COM to the walking surface was shortening. At impact, the FPE was slightly in front of the subject's toes. The subject applied ankle and hip strategies, but once those motions were exhausted, the FPE continued moving forward and a step (not shown for clarity) was required to restore balance.

This is in contrast to another jump by the same subject shown in Figure 6.6. This time, the subject landed with his feet on top of the FPE. Some ankle flexion was observed which maintained the FPE within the base of support, and the subject remained balanced.

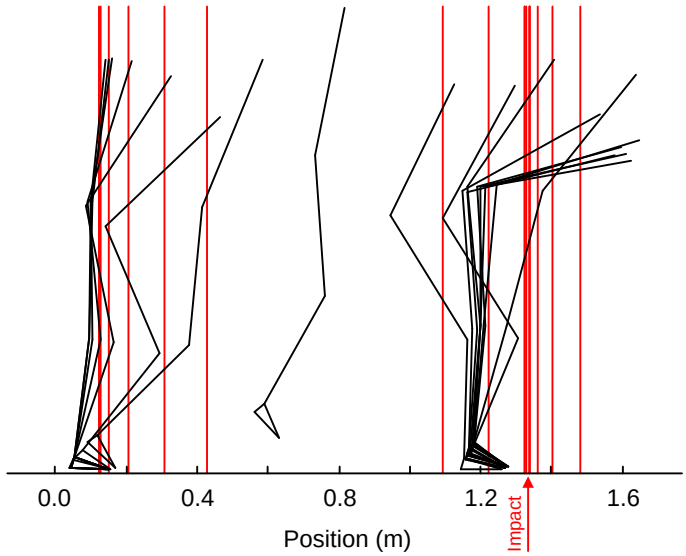


Figure 6.5: The progression of the FPE (in red) as a subject jumped. The landing location was behind the FPE and eventually the subject fell forward.

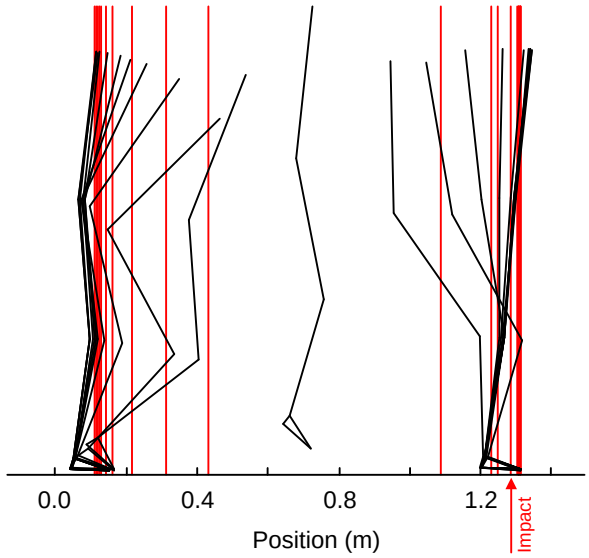


Figure 6.6: The progression of the FPE (in red) as a subject jumped. The landing location was on top of the FPE and the subject maintained his balance.

6.6 Discussion

The direct application of the FPE to human gait was an unintended benefit of this research. It is envisioned that the FPE may find a role in quantifying the stability margin of individuals' gaits. This may allow a physician to preemptively prescribe an assistive device to an aging patient to prevent a fall that results in serious injury. The FPE may also have a role in providing a dynamic evaluation of lower limb prosthetics.

Another application proposed by Dr. John McPhee at the University of Waterloo in relation to the research performed by Dr. Quincy Almeida at Wilfred Laurier University aims to help improve the gaits of Parkinson's patients. Almeida used a laser line projected on the ground from a belt-mounted device (developed by David Wang at the University of Waterloo) to help cue patients to take a step. Rather than projecting a line at a fixed position, a device incorporating the FPE algorithms could provide a continuous projection of where the patient's swing foot should be placed in order to remain balanced.

The FPE has been shown to be a remarkably accurate estimate for human motion, and hopefully further study will refine this estimate. The application of the FPE to human motion has a number of unique challenges that need to be studied in more detail. For instance, Winter's anthropometric data [223] is based on an average of young, healthy males. Although the FPE study included both men and women, the subjects were still a representative sample of height and weight for the typical North American population. It is unknown how the anthropometric data will scale to more extremes of the population.

Another issue to consider is that the FPE was designed for a planar device and therefore it only considers motion in the sagittal plane. It is possible that motion in the frontal plane accounts for the differences between subjects and trial types. Hopefully, the development of a full 3D FPE will bring further insight to the reasons for the variance in FPE offsets between subjects.

The FPE provides a new means of dynamic evaluation and quantification of human gaits. It will be interesting to see what insights it can provide to further our understanding of human motion.

Chapter 7

Future Research

This research has created a number of potential avenues to further the development of the foot placement estimator (FPE). With a functional physical experiment, the first task might be to advance the simulation to match the physical experiment, which would allow further theoretical development with higher confidence in the simulation results. However, due to the apparent limitations of the physical experiment, the first course of action should be to develop a refined, research-grade version that has the physical capability to make use of new theoretical developments as well as the ability to advance to free-flight gaits such as running.

7.1 Advancement in Two Dimensions

Although advancing to 3D is the next natural progression, there are many benefits that can still be drawn from a 2D biped platform. The nonlinearities of the bipedal model are incredibly complex and often nonintuitive. A clear understanding of one plane of operation would undoubtedly help to understand the effects of adding a second.

Now that the FPE has been proven to work for the most dynamic of bipedal architectures, the next area to investigate should be the addition of a foot. If the ankle joint on the standing foot were to function as a free pin joint, then the biped would behave similarly to the point-foot

architecture presented in this thesis. However, it is expected that the ankle torque could be used to modify the position of the FPE such that the biped could perform dynamic walking in a stone-stepping or obstacle strewn environment. This could be used in conjunction with torso motions to further influence the FPE.

The current simulation and physical experiment resulted in a self-selected gait speed and step length as a function of all the control gains and thresholds. It would be useful to find a means of controlling the speed and step length such that they could be regulated as secondary objectives to balance. Preliminary work indicated that the push velocity could be used to control the speed, and the FPE offset could be used to control step size, but only within a limited range. After adjusting other gains and thresholds, a new range of speeds and step sizes could be achieved. The details of these parameter relationships need to be extracted and mapped out.

The controllers of the various states could also use some advancement. High accelerations invariably create disturbances. Smoothing out some of the accelerations with nonlinear controllers could help reduce the effect. The addition of gravity compensation or even a computed torque approach for the limb control might also help the controllers achieve their objectives in a more timely manner and under a wider range of operating conditions. A more intelligent push controller could be developed to reduce the time spent pushing without increasing the COM velocity during the other states. Other secondary objectives could also be investigated such as maintaining a more constant COM velocity, or minimizing power consumption.

To advance to even more dynamic motions such as running and jumping, the state machine would need to be modified to take into account free-flight phases of the gait. Separate state machines would likely be needed for each activity. To deal with slipping and tripping, extra state transitions could be used in combination with a high-level controller that could modify control parameters in response to varying environmental conditions. Highly dynamic motions also require determining which of the two FPE solutions should be used when there are high rates of angular velocity independent of the linear velocity.

Another important extension would be the advancement to variable terrain. Assuming the terrain profile was known or could be measured, the foot placement might be solved through a

series of FPE calculations assuming planes at different altitudes. It may also require a multi-step planner to help determine where to step in the event that there are multiple solutions, or even no solution. The balance of a single step may need to be sacrificed to acquire a limited foothold on the assumption that the balance could be restored in a future step.

One of the key components needed to advance to 3D is determining the absolute position and orientation of a biped without the use of ground-anchored sensors. The stability of the biped presented in this thesis was dependent on a reliable FPE measurement to control the transition from the push state to the lift state. A biped that is fully equipped with optical encoders could produce a relatively noise-free FPE when both feet are on the ground, and then utilize analog range finding or inertial measurements during the single stance states. While sensor fusion is a research topic in itself, Lebastard et al. have demonstrated how Kalman filters for sensor fusion combined with observers can provide an internal estimation of the absolute position and orientation for a bipedal robot [227, 228].

7.2 Advancing to the Third Dimension

To be useful, a bipedal robot must be able to operate in the same three-dimensional world that we live in. The two-dimensional model is a valuable learning tool, but the FPE theory must be extended to three dimensions before it can become a serious competitor to the other well established methods of bipedal robot gait control.

7.2.1 The Foot Placement Estimator in 2.5 and 3.0 Dimensions

The simplest way to advance the FPE theory might be to use two decoupled copies of the FPE algorithm. One copy of the FPE algorithm would be used for the sagittal plane as was done for the 2D experiments, and the other copy would be used for the frontal plane. Just as the FPE represented the dividing point in 2D between falling forward, and rocking back, the FPE would now represent the dividing point of four quadrants which would dictate the subsequent sagittal and frontal motion as shown in Figure 7.1. This is more of a 2.5D solution than a full 3D solution

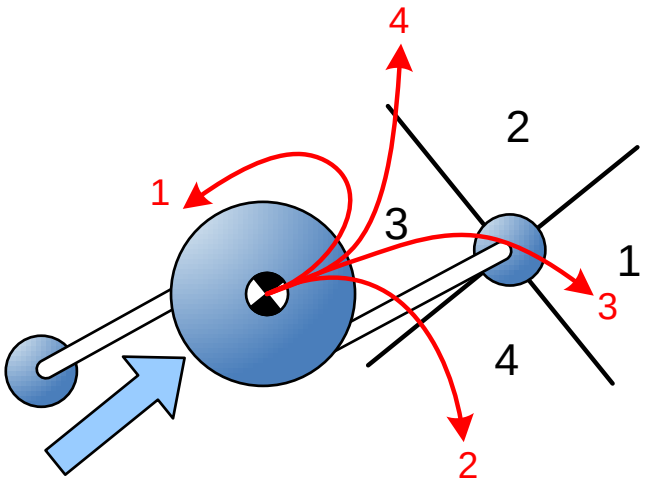


Figure 7.1: A top view of a simple biped in 3D using two decoupled FPE calculations. Assuming no rotation about the vertical axis, stepping in one of the four black quadrants will cause the COM to follow the corresponding path in red.

because it does not consider rotation about the vertical axis. With a full foot of sufficient surface area, it might be possible to enforce this assumption under normal forward walking, but eventually the robot will need to turn. If turning is done as a very dynamic motion, then the two decoupled FPE calculations may not provide a sufficiently accurate estimate of where the feet should be placed.

Consider a 3D body with mass and inertia spinning about the vertical axis on a massless leg. According to the definitions from Chapter 3, this is a balanced position. To stop the rotation, the second massless leg can be placed anywhere on the plane *except* the present standing location. Once linear motions of the COM are added to the problem, the solution of where to place the foot becomes much more complex. A complete 3D FPE solution will need to resolve how to deal with the extra rotational axis.

7.2.2 The Control System in the Third Dimension

Advancing the remainder of the control system to 3D will also present a number of challenges. The fundamental state machine would probably be very similar, but many of the underlying controllers would change. For instance, the push controller would need to account for motion in both the sagittal and frontal planes. The pushing profile must drive the robot forward while keeping the FPE from moving too far laterally.

During the lift and swing states, limb collision would need to be taken into account. In a worst case scenario where an external disturbance forces a lateral step in front of the standing foot, the trajectories of both the swing and standing legs would need to be planned in concert to avoid collision. Alternatively, the stability of one step could be proactively relinquished such that the opposite foot could immediately be used to maximize the chance of restoring balance to the system. This may require that extra conditions be added to the state machine.

Perhaps the most challenging addition to the control system would be the inclusion of turns about the vertical axis in the robots's repertoire of actions. Slow rotations could probably be treated as disturbances using two decoupled FPE calculations. However, a truly dynamic solution will revolve around finding a complete 3D version of the FPE that takes rotation about the vertical axis into account.

7.3 In Closing

This research offers many possible avenues for future research. In its present state of development, the FPE can be used to augment the abilities of existing control strategies in the literature to improve the range of disturbances that can be endured. In addition to the future uses in bipedal robotics and biomechanics, the FPE might also be useful in highly dynamic robots with more than two legs. Although the FPE does not share the same breadth of development as other methods with years of research behind them, the FPE has great potential for future bipedal applications.

Chapter 8

Contributions

The primary contribution of this thesis is the development of the foot placement estimator (FPE) and its associated theoretical proof in Chapter 3. A paper based on the contents of this chapter has been accepted for publication [203]. The theoretical work was applied to a point-foot bipedal robot using a state machine and simple joint controllers in both simulated and physical forms. The simulation work in Chapter 4 uses a non-dimensional, comprehensive mapping of the stability margin for single disturbances. This exhaustive approach is a novel contribution as far as the author is aware. The experimental results from the physical robot in Chapter 5 not only validate the application of the FPE, but also show that the biped produced a dynamic gait that is the fastest normalized walking speed currently in publication that the author is aware of. A final significant contribution is the application of the FPE to estimate human foot placement in Chapter 6. The results that were jointly developed with Matthew Millard show that the FPE is highly correlated with the human selection of foot placement, and a paper entitled "Evaluation of a Human Foot Placement Model" by Millard et al. [226] has been submitted to the 2008 North American Congress on Biomechanics for publication. It is hoped that the development of the FPE will have a lasting impact on both the fields of bipedal robotics and kinesiology.

References

[1] D. A. R. Michael D. Sockol and H. Pontzer. "Chimpanzee locomotor energetics and the origin of human bipedalism." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 30 pp. 12265–12269, July 2007.

[2] D. L. Wight. *Control of a Pneumatically Powered Orthosis*. Master's thesis, University of Waterloo, 2003.

[3] Anon. "Self-powered exoskeleton assists human locomotion." In *Jane's International Defense Review*, 2004, April, p. 2.

[4] P. Kennedy, R. Bakay, M. Moore, K. Adams, and J. Goldwaithe. "Direct control of a computer from the human central nervous system." *IEEE Transactions on Rehabilitation Engineering*, vol. 8 pp. 198–202, 2000.

[5] J. Chestnutt, P. Michel, K. Nishiwaki, J. Kuffner, and S. Kagami. "An intelligent joystick for biped control.", 2006, vol. 2006, pp. 860–865.

[6] K. Hashimoto, T. Hosobata, Y. Sugahara, Y. Mikuriya, H. Sunazuka, M. Kawase, H. Lim, and A. Takanishi. "Development of foot system of biped walking robot capable of maintaining four-point contact." In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 1361–1366.

[7] Y. Sugahara, K. Hashimoto, M. Kawase, T. Sawato, A. Hayashi, N. Endo, A. Ohta, C. Tanaka, H. Lim, and A. Takanishi. "Walking pattern generation of a biped walking

vehicle using a dynamic human model." In *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 2497–2502.

[8] Y. Sugahara, K. Hashimoto, H. Sunazuka, M. Kawase, A. Ohta, C. Tanaka, H. Lim, and A. Takanishi. "Towards the biped walking wheelchair." In *2006 1st IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2006, p. 6.

[9] Y. Takahashi, H. Nakayama, and T. Nagasawa. "Biped robot to assist walking and moving up-and-down stairs." In *IECON Proceedings (Industrial Electronics Conference)*, 1998, vol. 2, pp. 1140–1145.

[10] A. Bejczy. "Towards development of robotic aid for rehabilitation of locomotion-impaired subjects." In *Proceedings of the First Workshop on Robot Motion and Control. RoMoCo'99 (Cat. No.99EX353)*, 1999, pp. 9–16.

[11] H. Hirukawa. "Walking biped humanoids that perform manual labour." *Philosophical Transactions of the Royal Society London, Series A (Mathematical, Physical and Engineering Sciences)*, vol. 365, no. 1850 pp. 65–77, 2007.

[12] T. Nishiyama, H. Hoshino, K. Sawada, A. Baba, T. Sekine, W. Yamada, A. Terasawa, R. Nakajima, Y. Tokunaga, and M. Yoneda. "Communication agent embedded in humanoid robot." In *SICE 2003 Annual Conference (IEEE Cat. No.03TH8734)*, 2003, vol. 2, pp. 1514–1519.

[13] K. Sakata, K. Inoue, T. Takubo, T. Arai, and Y. Mae. "Wheelchair user support system using humanoid robots - System concept and experiments on pushing wheelchair." In *Proceedings of the SICE Annual Conference*, 2004, pp. 1955–1960.

[14] F. Kanehiro, H. Hirukawa, K. Kaneko, S. Kajita, K. Fujiwara, K. Harada, and K. Yokoi. "Locomotion planning of humanoid robots to pass through narrow spaces." In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, vol. 1, pp. 604–609.

[15] D. Wright and W. Chun. "Advanced rover for planetary exploration." In *Proc. SPIE - Int. Soc. Opt. Eng. (USA)*, 1989, vol. 1007, pp. 120–127.

[16] Z. Wang and H. Gu. "A review of locomotion mechanisms of urban search and rescue robot." *Industrial Robot*, vol. 34, no. 5 pp. 400–411, 2007.

[17] T. Zhe, Z. Changjiu, and S. Zenqi. "Trajectory planning for smooth transition of a biped robot." In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003, vol. 2, pp. 2455–2460.

[18] S. Behnke. "Online trajectory generation for omnidirectional biped walking." In *Proceedings. 2006 Conference on International Robotics and Automation (IEEE Cat. No. 06CH37729D)*, 2006, pp. 1597–1603.

[19] D. Gouaillier and P. Blazevic. "A mechatronic platform, the aldebaran robotics humanoid robot.", 2006, pp. 4049–4053.

[20] C. Azevedo, B. Espiau, B. Amblard, and C. Assaiante. "Bipedal locomotion: Toward unified concepts in robotics and neuroscience." *Biological Cybernetics*, vol. 96, no. 2 pp. 209–228, 2007.

[21] A. Aceves-Lopez and A. Melendez-Calderon. "Human-inspired walking-style for a low-cost biped prototype." In *2006 IEEE 3rd Latin American Robotics Symposium*, 2006, p. 8.

[22] T. Ha and C. Choi. "An effective trajectory generation method for bipedal walking." *Robotics and Autonomous Systems*, vol. 55, no. 10 pp. 795–810, 2007.

[23] H. Choi and Y. Park. "Development of a biped walking robot actuated by a closed-chain mechanism." *Robotica*, vol. 24, no. 1 pp. 31–37, 2006.

[24] A. Dasgupta and Y. Nakamura. "Making feasible walking motion of humanoid robots from human motion capture data." In *Proceedings - IEEE International Conference on Robotics and Automation*, 1999, vol. 2, pp. 1044–1049.

[25] H. Hemami and R. Farnsworth. "Postural and gait stability of a planar five link biped by simulation." *IEEE Transactions on Automatic Control*, vol. AC-22, no. 3 pp. 450–456, Jan. 1977.

[26] H. Lim, Y. Yamamoto, and A. Takanishi. "Control to realize human-like walking of a biped humanoid robot." In *SMC 2000 Conference Proceedings. 2000 IEEE International Conference on Systems, Man and Cybernetics. 'Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions' (Cat. No.00CH37166)*, 2000, vol. 5, pp. 3271–3276.

[27] Y. Ogura, K. Shimomura, H. Kondo, A. Morishima, T. Okubo, S. Momoki, H. Lim, and A. Takanishi. "Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot." In *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 3976–3981.

[28] I. Park, J. Kim, J. Lee, and J. Oh. "Mechanical design of humanoid robot platform KHR-3 (KAIST Humanoid Robot 3: HUBO)." In *2005 5th IEEE-RAS International Conference on Humanoid Robots (IEEE Cat. No.05EX1139C)*, 2005, pp. 321–326.

[29] W. Yang, N. Y. Chong, C. Kim, and B. J. You. "Locomotion imitation of humanoid using goal-directed self-adjusting adaptor." In *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 5650–5656.

[30] X. Zhao, Q. Huang, Z. Peng, and K. Li. "Kinematics mapping and similarity evaluation of humanoid motion based on human motion capture." In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004, vol. 1, pp. 840–845.

[31] A. Kuo. "A simple model of bipedal walking predicts the preferred speed-step length relationship." *Journal of Biomechanical Engineering*, vol. 123, no. 3 pp. 264–269, 2001.

[32] T. Hirai, N. Issiki, Y. Kawabe, T. Terada, and K. Watanabe. "Effective development of biped robots by CAE." In *2006 SICE-ICASE International Joint Conference (IEEE Cat. No.06TH8879)*, 2007, p. 4.

[33] Y. Takahashi, T. Suzuki, S. Hisamitsu, Y. Matsuo, S. Yamawaki, and S. Isonuma. "Simple humanoid robot for university education." In *IECON Proceedings (Industrial Electronics Conference)*, 2004, vol. 1, pp. 146–151.

[34] A. Yonemura, Y. Nakajima, A. Hirakawa, and A. Kawamura. "Experimental approach for the biped walking robot MARI-1." In *6th International Workshop on Advanced Motion Control. Proceedings (Cat. No.00TH8494)*, 2000, pp. 548–553.

[35] T. McGeer. "Powered flight, child's play, silly wheels and walking machines." In *IEEE International Conference on Robotics and Automation*, 1989, pp. 1592–1597.

[36] T. McGeer. "Passive dynamic walking." *International Journal of Robotics Research*, vol. 9, no. 2 pp. 62–82, 1990.

[37] M. Wisse, A. Schwab, and F. Van Der Helm. "Passive dynamic walking model with upper body." *Robotica*, vol. 22, no. 6 pp. 681–688, 2004.

[38] T. McGeer. "Passive walking with knees." In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, 1990, pp. 1640–1645.

[39] Y. Ikemata, A. Sano, and H. Fujimoto. "A physical principle of gait generation and its stabilization derived from mechanism of fixed point." In *IEEE International Conference on Robotics and Automation*, 2006, vol. 2006, pp. 836–841.

[40] E. Borzova and Y. Hurmuzlu. "Passively walking five-link robot." *Automatica*, vol. 40, no. 4 pp. 621–629, 2004.

[41] S. H. Collins, M. Wisse, and A. Ruina. "A three-dimensional passive-dynamic walking robot with two legs and knees." *International Journal of Robotics Research*, vol. 20, no. 7 pp. 607–615, 2001.

[42] F. Asano, M. Yamakita, and K. Furuta. "Virtual passive dynamic walking and energy-based control laws." In *IEEE International Conference on Intelligent Robots and Systems*, 2000, vol. 2, pp. 1149–1154.

[43] K. Koganezawa and O. Matsumoto. "Active/passive hybrid walking by the biped robot TOKAI ROBO-HABILIS 1." In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (Cat. No.02CH37332C)*, 2002, vol. 3, pp. 2461–2466.

[44] J. Pratt. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. Ph.D. thesis, Massachusetts Institute of Technology, Computer Science Department, Cambridge, Massachusetts, 2000.

[45] M. Wisse, A. L. Schwab, R. Q. van der Linde, and F. C. 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, 2005.

[46] G. Tao, B. Porr, and F. Worgotter. "A reflexive neural network for dynamic biped walking control." *Neural Computation*, vol. 18, no. 5 pp. 1156–1196, May 2006.

[47] S. Anderson, M. Wisse, C. Atkeson, J. Hodgins, G. Zeglin, and B. Moyer. "Powered bipeds based on passive dynamic principles." In *2005 5th IEEE-RAS International Conference on Humanoid Robots (IEEE Cat. No.05EX1139C)*, 2005, pp. 110–116.

[48] A. D. Kuo. "Energetics of actively powered locomotion using the simplest walking model." *Journal of Biomechanical Engineering*, vol. 124, no. 1 pp. 113–120, 2002.

[49] S. Suzuki, K. Furuta, and S. Hatakeyama. "Lazy control strategy for active passive walking." In *IECON Proceedings (Industrial Electronics Conference)*, 2001, vol. 3, pp. 305–310.

[50] K. Ono, T. Furuichi, and R. Takahashi. "Self-excited walking of a biped mechanism with feet." *International Journal of Robotics Research*, vol. 23, no. 1 pp. 55–68, 2004.

[51] M. Wisse. "Three additions to passive dynamic walking; actuation, an upper body, and 3D stability." In *2004 4th IEEE/RAS International Conference on Humanoid Robots (IEEE Cat. No. 04EX989)*, 2004, vol. 1, pp. 113–132.

[52] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. "Efficient bipedal robots based on passive-dynamic walkers." *Science*, vol. 307, no. 5712 pp. 1082–1085, 2005.

[53] J. Hass, J. Herrmann, and T. Geisel. "Optimal mass distribution for passivity-based bipedal robots." *International Journal of Robotics Research*, vol. 25, no. 11 pp. 1087–1098, 2006.

[54] A. Schwab and M. Wisse. "Basin of attraction of the simplest walking model." In *Proceedings of the ASME Design Engineering Technical Conference*, 2001, vol. 6 A, pp. 531–539.

[55] R. Ringrose. "Self-stabilizing running." In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, 1997, vol. 1, pp. 487–493.

[56] K. D. Mombaur, R. W. Longman, H. G. Bock, and J. P. Schloder. "Open-loop stable running." *Robotica*, vol. 23, no. 1 pp. 21–33, 2005.

[57] K. D. Mombaur, H. G. Bock, J. P. Schloder, and R. W. Longman. "Self-stabilizing somersaults." *IEEE Transactions on Robotics*, vol. 21, no. 6 pp. 1148–1157, 2005.

[58] S. Aoi and K. Tsuchiya. "Self-stability of a simple walking model driven by a rhythmic signal." *Nonlinear Dynamics*, vol. 48, no. 1-2 pp. 1–16, 2007.

[59] S. Kitamura, Y. Kurematsu, and M. Iwata. "Motion generation of a biped locomotive robot using an inverted pendulum model and neural networks." In *Proceedings of the IEEE Conference on Decision and Control*, 1990, vol. 6, pp. 3308–3312.

[60] W. Miller. "Real-time neural network control of a biped walking robot." *IEEE Control Systems Magazine*, vol. 14, no. 1 pp. 41–48, 1994.

[61] D. Lee and W. ElMaraghy. "A neural network solution for bipedal gait synthesis." In *IJCNN International Joint Conference on Neural Networks*, 1992, pp. 763–768.

[62] T. Geng, B. Porr, and F. Worgotter. "Fast biped walking with a sensor-driven neuronal controller and real-time online learning." *International Journal of Robotics Research*, vol. 25, no. 3 pp. 243–259, 2006.

[63] J. G. Juang. "Intelligent trajectory control using recurrent averaging learning." *Applied Artificial Intelligence*, vol. 15, no. 3 pp. 277–296, 2001.

[64] C. Sabourin and O. Bruneau. "Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks." *Robotics and Autonomous Systems*, vol. 51, no. 2-3 pp. 81–99, May 2005.

[65] A. Kun and I. Miller, W.T. "Control of variable speed gaits for a biped robot." *IEEE Robotics & Automation Magazine*, vol. 6, no. 3 pp. 19–29, Sep. 1999.

[66] C. Sabourin, O. Bruneau, and G. Buche. "Control strategy for the robust dynamic walk of a biped robot." *International Journal of Robotics Research*, vol. 25, no. 9 pp. 843–860, Sep. 2006.

[67] D. Kee, G. Wyeth, and J. Roberts. "Biologically inspired joint control for a humanoid robot." In *2004 4th IEEE/RAS International Conference on Humanoid Robots (IEEE Cat. No. 04EX989)*, 2004, vol. 1, pp. 385–401.

[68] J. Jih-Gau. "Fuzzy neural network approaches for robotic gait synthesis." *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 4 pp. 594–601, Aug. 2000.

[69] Z. Liu and Y. Zhang. "Fuzzy logic-based neural modeling and robust control for robot." In *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 2006, vol. 2, pp. 8976–8980.

[70] C. Zhou and K. Iagannathan. "Adaptive network based fuzzy control of a dynamic biped walking robot." In *IEEE International Joint Symposia on Intelligence and Systems*, 1996, pp. 109–116.

[71] L. Jalics, H. Hemami, and Y. F. Zheng. "Pattern generation using coupled oscillators for robotic and biorobotic adaptive periodic movement." In *IEEE International Conference on Robotics and Automation*, 1997, vol. 1, pp. 179–184.

[72] S. Jiang, J. Cheng, and J. Chen. "Design of central pattern generator for humanoid robot walking based on multi-objective GA." In *IEEE International Conference on Intelligent Robots and Systems*, 2000, vol. 3, pp. 1930–1935.

[73] T. Matsubara, J. Morimoto, J. Nakanishi, M. Sato, and K. Doya. "Learning CPG-based biped locomotion with a policy gradient method." In *2005 5th IEEE-RAS International Conference on Humanoid Robots (IEEE Cat. No.05EX1139C)*, 2005, pp. 208–213.

[74] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato. "Learning from demonstration and adaptation of biped locomotion." *Robotics and Autonomous Systems*, vol. 47, no. 2-3 pp. 79–91, Jun. 2004.

[75] L. Righetti and A. Ijspeert. "Programmable central pattern generators: An application to biped locomotion control." In *Proceedings. 2006 Conference on International Robotics and Automation (IEEE Cat. No. 06CH37729D)*, 2006, pp. 1585–1590.

[76] G. Endo, J. Morimoto, J. Nakanishi, and G. Cheng. "An empirical exploration of a neural oscillator for biped locomotion control." In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, vol. 3, pp. 3036–3042.

[77] M. Cao and A. Kawamura. "A design method of neural oscillatory networks for generation of humanoid biped walking patterns." In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, 1998, vol. 3, pp. 2357–2362.

[78] S. Aoi and K. Tsuchiya. "Bifurcation and chaos of a simple walking model driven by a rhythmic signal." *International Journal of Non-Linear Mechanics*, vol. 41, no. 3 pp. 438–446, 2006.

[79] S. Aoi and K. Tsuchiya. "Locomotion control of a biped robot using nonlinear oscillators." *Autonomous Robots*, vol. 19, no. 3 pp. 219–232, Dec. 2005.

[80] J. Morimoto, G. Endo, J. Nakanishi, S. Hyon, G. Cheng, D. Bentivegna, and C. Atkeson. "Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking." In *Proceedings. 2006 Conference on International Robotics and Automation (IEEE Cat. No. 06CH37729D)*, 2006, pp. 1579–1584.

[81] K. Feng, C. M. Chew, G. S. Hong, and T. Zielinska. "Bipedal locomotion control using

a four-compartmental central pattern generator." In *IEEE International Conference on Mechatronics and Automation, ICMA 2005*, 2005, pp. 1515–1520.

[82] H. Benbrahim and J. A. Franklin. "Biped dynamic walking using reinforcement learning." *Robotics and Autonomous Systems*, vol. 22, no. 3-4 pp. 283–302, 1997.

[83] Y. Zheng. "A neural gait synthesizer for autonomous biped robots." In *Proceedings. IROS '90. IEEE International Workshop on Intelligent Robots and Systems '90. Towards a New Frontier of Applications (Cat. No.90TH0332-7)*, 1990, pp. 601–608.

[84] D. Katic and M. Vukobratovic. "Survey of intelligent control techniques for humanoid robots." *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 37, no. 2 pp. 117–141, Jun. 2003.

[85] M. Vukobratovic and D. Juricic. "Contribution to the synthesis of biped gait." In *Proc. IFAC Symp. Technical and Biological Problem on Control,*, 1968.

[86] M. Vukobratovic, B. Borovac, and V. Potkonjak. "ZMP: A review of some basic misunderstandings." *International Journal of Humanoid Robotics*, vol. 3, no. 2 pp. 153–175, 2006.

[87] H. Lim and A. Takanishi. "Biped walking robots created at waseda university: WL and WABIAN family." *Philosophical Transactions of the Royal Society London, Series A (Mathematical, Physical and Engineering Sciences)*, vol. 365, no. 1850 pp. 49–64, 2007.

[88] K. Hirai. "Current and future perspective of honda humamoid robot." In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97 (Cat. No.97CH36108)*, 1997, vol. 2, pp. 500–508.

[89] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. "The development of Honda humanoid robot." In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 1998, vol. 2, pp. 1321–1326.

[90] M. Hirose and K. Ogawa. "Honda humanoid robots development." *Philosophical Transactions of the Royal Society London, Series A (Mathematical, Physical and Engineering Sciences)*, vol. 365, no. 1850 pp. 11–19, 2007.

[91] Y. Kuroki. "A small biped entertainment robot." In *MHS2001. Proceedings of 2001 International Symposium on Micromechatronics and Human Science (Cat. No.01TH8583)*, 2001, pp. 3–4.

[92] Y. Kuroki, M. Fujita, T. Ishida, K. Nagasaka, and J. Yamaguchi. "A small biped entertainment robot exploring attractive applications." In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003, vol. 1, pp. 471–476.

[93] Y. Kuroki, T. Fukushima, K. Nagasaka, T. Moridaira, T. Doi, and J. Yamaguchi. "A small biped entertainment robot exploring human-robot interactive applications." In *Proceedings. RO-Man 2003. The 12th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Cat. No. 03TH8711)*, 2003, pp. 303–308.

[94] T. Ishida. "A small biped entertainment robot SDR-4X II." In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No.03EX694)*, 2003, vol. 3, pp. 1046–1051.

[95] L. Geppert. "QRIO the robot that could." *IEEE Spectrum*, vol. 41, no. 5 pp. 26–28, 2004.

[96] T. Ishida. "Development of a small biped entertainment robot QRIO." In *Proceedings of the 2004 International Symposium on Micro-NanoMechatronics and Human Science, MHS2004; The Fourth Symposium 'Micro-NanoMechatronics for and Information-Based Society' The 21st Century*, 2004, pp. 23–28.

[97] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh, and J. Yamaguchi. "Integrated motion control for walking, jumping and running on a small bipedal entertainment robot." In *Proceedings - IEEE International Conference on Robotics and Automation*, 2004, vol. 2004, pp. 3189–3194.

[98] Y. Ayaz, B. Afzal, M. Saeed, and U. R. Saeed. "Design, fabrication and control of a two-legged walking robot." In *Proceedings of the Fourth International Workshop on Robot Motion and Control, RoMoCo'04*, 2004, pp. 73–78.

[99] C.-L. Shih and C. Chien-Jung. "The motion control of a statically stable biped robot on an uneven floor." *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 2 pp. 244–249, Apr. 1998.

[100] K. Takahashi, M. Noda, D. Nenchev, Y. Tsumaki, and A. Sekiguchi. "Static walk of a humanoid robot based on the singularity-consistent method." In *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 5484–5489.

[101] M. Vukobratovic and B. Borovac. "Zero-moment point - Thirty five years of its life." *International Journal of Humanoid Robotics*, vol. 1, no. 1 pp. 157–173, 2004.

[102] M. B. Popovic, A. Goswami, and H. Herr. "Ground reference points in legged locomotion: Definitions, biological trajectories and control implications." *International Journal of Robotics Research*, vol. 24, no. 12 pp. 1013–1032, 2005.

[103] A. Siqueira and M. Terra. "Nonlinear $H_{Infinity}$ control applied to biped robots." In *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, 2006, p. 6.

[104] Q. Huang and Y. Nakamura. "Sensory reflex control for humanoid walking." *IEEE Transactions on Robotics*, vol. 21, no. 5 pp. 977–984, 2005.

[105] A. Goswami. "Postural stability of biped robots and the Foot-Rotation Indicator (FRI) point." *International Journal of Robotics Research*, vol. 18, no. 6 pp. 523–533, Jun. 1999.

[106] A. Goswami. "Foot Rotation Indicator (FRI) point: A new gait planning tool to evaluate postural stability of biped robots." In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 1999, vol. 1, pp. 47–52.

[107] P. N. Mousavi and A. Bagheri. "Mathematical simulation of a seven link biped robot on various surfaces and ZMP considerations." *Applied Mathematical Modelling*, vol. 31, no. 1 pp. 18–37, 2007.

[108] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. "Biped walking pattern generation by using preview control of zero-moment point." In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003, vol. 2, pp. 1620–1626.

[109] C. Zhu, Y. Tomizawa, X. Luo, and A. Kawamura. "Biped walking with variable ZMP, frictional constraint, and inverted pendulum model." In *Proceedings. 2004 IEEE International Conference on Robotics and Biomimetics (IEEE Cat. No. 04EX911)*, 2004, pp. 425–430.

[110] K. Loffler, M. Gienger, and F. Pfeiffer. "Model based control of a biped robot." In *International Workshop on Advanced Motion Control, AMC*, 2002, pp. 443–448.

[111] K. Erbatur and O. Kurt. "Humanoid walking robot control with natural ZMP references." In *IECON 2006. 32nd Annual Conference on IEEE Industrial Electronics*, 2006, pp. 4100–4106.

[112] H. P. Jong. "Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots." *Fuzzy Sets and Systems*, vol. 134, no. 1 pp. 189–203, Feb. 2003.

[113] H. ok Lim, S. Setiawan, and A. Takanishi. "Balance and impedance control for biped humanoid robot locomotion." In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, 2001, vol. 1, pp. 494–499.

[114] R. Kurazume, T. Hasegawa, and K. Yoneda. "The sway compensation trajectory for a biped robot." In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003, vol. 1, pp. 925–931.

[115] J. Yamaguchi, A. Takanishi, and I. Kato. "Development of a biped walking robot compensating for three-axis moment by trunk motion." In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Jul 26-30 1993*, 1993, pp. 561–566.

[116] Z. Chi, M. Okamura, A. Kawamura, and Y. Tomizawa. "Experimental approach for high speed walking of biped robot MARI-1." In *8th IEEE International Workshop on Advanced Motion Control (IEEE Cat. No.04TH8725)*, 2004, pp. 427–432.

[117] H. Lim and A. Takanishi. "Compensatory motion control for a biped walking robot." *Robotica*, vol. 23 pp. 1–11, Jan. 2005.

[118] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, and H. Inoue. "A fast dynamically equilibrated walking trajectory generation method of humanoid robot." *Autonomous Robots*, vol. 12, no. 1 pp. 71–82, 2002.

[119] Q. Li, A. Takanishi, and I. Kato. "Learning control for a biped walking robot with a trunk." In *IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems. Intelligent Robots for Flexibility (Cat. No.93CH3213-6)*, 1993, pp. 1771–1777.

[120] A. Takanishi, H. Lim, M. Tsuda, and I. Kato. "Realization of dynamic biped walking stabilized by trunk motion on a sagittally uneven surface." In *Proceedings. IROS '90. IEEE International Workshop on Intelligent Robots and Systems '90. Towards a New Frontier of Applications (Cat. No.90TH0332-7)*, 1990, pp. 323–330.

[121] I. Park, J. Kim, and J. Oh. "Online biped walking pattern generation for humanoid robot KHR-3 (KAIST Humanoid Robot-3: HUBO)." In *2006 6th IEEE-RAS International Conference on Humanoid Robots (IEEE Cat No. 06EX1315C)*, 2006, p. 6.

[122] C. Fu, M. Shuai, K. Xu, J. Zhao, J. Wang, Y. Huang, and K. Chen. "Planning and control for THBIP-I humanoid robot." In *2006 IEEE International Conference on Mechatronics and Automation (IEEE Cat. No. 06EX1416C)*, 2006, pp. 1066–1071.

[123] D. Djoudi, C. Chevallereau, and J. Grizzle. "A path-following approach to stable bipedal walking and zero moment point regulation." In *2007 IEEE International Conference on Robotics and Automation (IEEE Cat No. 07CH37836D)*, 2007, p. 6.

[124] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, and H. Hirukawa. "Biped walking pattern generator allowing auxiliary ZMP control." In *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 2993–2999.

[125] Y. Okumura, T. Tawara, K. Endo, T. Furuta, and M. Shimizu. "Realtime ZMP compensation for biped walking robot using adaptive inertia force control." In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, 2003, vol. 1, pp. 335–339.

[126] T. Fukuda, Y. Komata, and T. Arakawa. "Recurrent neural network with self-adaptive GA's for biped locomotion robot." In *1997 IEEE International Conference on Neural Networks. Proceedings (Cat. No.97CH36109)*, 1997, vol. 3, pp. 1710–1715.

[127] L. Hu and Z. Sun. "Reinforcement learning method-based stable gait synthesis for biped robot." In *2004 8th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2004, vol. 2, pp. 1017–1022.

[128] D. Kim, S. Seo, and G. Park. "Zero-moment point trajectory modelling of a biped walking robot using an adaptive neuro-fuzzy system." *IEE Proceedings: Control Theory and Applications*, vol. 152, no. 4 pp. 411–426, 2005.

[129] C. Zhou. "Neuro-fuzzy gait synthesis with reinforcement learning for a biped walking robot." *Soft Computing*, vol. 4, no. 4 pp. 238–250, Dec. 2000.

[130] Y. Zhou and M. J. Er. "Dynamic fuzzy Q-learning control of uncertain systems with applications to humanoids." In *2005 IEEE International Conference on Control Applications (CCA) (IEEE Cat. No.05CH37713)*, 2005, vol. 1, pp. 459–464.

[131] K. Nishiwaki and S. Kagami. "High frequency walking pattern generation based on preview control of ZMP." In *Proceedings - IEEE International Conference on Robotics and Automation*, 2006, vol. 2006, pp. 2667–2672.

[132] M. Morisawa, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, S. Nakaoka, and H. Hirukawa. "A biped pattern generation allowing immediate modification of foot placement in real-time." In *2006 6th IEEE-RAS International Conference on Humanoid Robots (IEEE Cat No. 06EX1315C)*, 2006, p. 6.

[133] J. Vermeulen, B. Verrelst, B. Vanderborght, D. Lefeber, and P. Guillaume. "Trajectory planning for the walking biped "Lucy"." *International Journal of Robotics Research*, vol. 25, no. 9 pp. 867–887, Sep. 2006.

[134] B. Vanderborght, B. Verrelst, M. Van Damme, R. Van Ham, P. Beyl, and D. Lefeber. "Locomotion control architecture for the pneumatic biped lucy consisting of a trajectory generator and joint trajectory tracking controller." In *2006 6th IEEE-RAS International Conference on Humanoid Robots (IEEE Cat No. 06EX1315C)*, 2006, p. 6.

[135] P. Zhaoqin, H. Qiang, Z. Xiaojun, X. Tao, and L. Kejie. "Online trajectory generation based on off-line trajectory for biped humanoid." In *Proceedings. 2004 IEEE International Conference on Robotics and Biomimetics (IEEE Cat. No. 04EX911)*, 2004, pp. 752–756.

[136] A. Albert and W. Gerth. "Analytic path planning algorithms for bipedal robots without a trunk." *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 36, no. 2 pp. 109–127, Feb. 2003.

[137] I. Park, J. Kim, J. Lee, and J. Oh. "Online free walking trajectory generation for biped humanoid robot KHR-3 (HUBO)." In *Proceedings - IEEE International Conference on Robotics and Automation*, 2006, vol. 2006, pp. 1231–1236.

[138] A. Sano and J. Furusho. "Realization of natural dynamic walking using the angular momentum information." In *Proceedings 1990 IEEE International Conference on Robotics and Automation (Cat. No.90CH2876-1)*, 1990, pp. 1476–1481.

[139] S. Kajita, K. Yokoi, M. Saigo, and K. Tanie. "Balancing a humanoid robot using backdrive concerned torque control and direct angular momentum feedback.", 2001, vol. 4, pp. 3376–3382.

[140] A. Goswami and V. Kallem. "Rate of change of angular momentum and balance maintenance of biped robots." In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, vol. 4, pp. 3785–3790.

[141] M. Popovic, A. Hofmann, and H. Herr. "Zero spin angular momentum control: Definition and applicability." In *2004 4th IEEE-RAS International Conference on Humanoid Robots*, 2004, vol. 1, pp. 478–493.

[142] J. Rojas-Estrada, J. Marot, P. Sardain, and G. Bessonnet. "Control of a biped robot by total rate of angular momentum using the task function approach." *Applied Bionics and Biomechanics*, vol. 2, no. 1-4 pp. 111–116, 2005.

[143] H. Minakata and Y. Hori. "Realtime speed-changeable biped walking by controlling the parameter of virtual inverted pendulum." In *IECON Proceedings (Industrial Electronics Conference)*, 1994, vol. 2, pp. 1009–1014.

[144] M. Popovic, A. Hofmann, and H. Herr. "Angular momentum regulation during human walking: Biomechanics and control." In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, vol. 3, pp. 2405–2411.

[145] W. S. Levine, editor. *Control Handbook*. CRC Press, 1995.

[146] X. Wang. "On the generalized zero dynamics." In *Decision and Control, Proceedings of the 37th IEEE Conference on*, 1998, vol. 4, pp. 4109–4110.

[147] E. Westervelt, J. Grizzle, and D. Koditschek. "Hybrid zero dynamics of planar biped walkers." *IEEE Transactions on Automatic Control*, vol. 48, no. 1 pp. 42–56, 2003.

[148] E. Westervelt, J. Grizzle, and C. Canudas de Wit. "Switching and PI control of walking

motions of planar biped walkers." *IEEE Transactions on Automatic Control*, vol. 48, no. 2 pp. 308–312, Feb. 2003.

[149] E. Westervelt, G. Buche, and J. Grizzle. "Inducing dynamically stable walking in an underactuated prototype planar biped." In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, vol. 4, pp. 4234–4239.

[150] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, C. Canudas-De-Wit, E. Westervelt, and J. Grizzle. "RABBIT: A testbed for advanced control theory." *IEEE Control Systems Magazine*, vol. 23, no. 5 pp. 57–79, 2003.

[151] A. Chemori and M. Alamir. "Multi-step limit cycle generation for Rabbit's walking based on a nonlinear low dimensional predictive control scheme." *Mechatronics*, vol. 16, no. 5 pp. 259–277, 2006.

[152] B. Morris and J. Grizzle. "Hybrid invariance in bipedal robots with series compliant actuators." In *Proceedings of the 45th IEEE Conference on Decision and Control (IEEE Cat. No. 06CH37770)*, 2006, p. 8.

[153] C. Chevallereau, E. Westervelt, and J. Grizzle. "Asymptotically stable running for a five-link, four-actuator, planar bipedal robot." *International Journal of Robotics Research*, vol. 24, no. 6 pp. 431–464, 2005.

[154] J. E. Pratt. *Virtual Model Control of a Biped Walking Robot*. Master's thesis, Massachusetts Intstitute of Technology, 1995.

[155] J. Pratt and G. Pratt. "Intuitive control of a planar bipedal walking robot." In *Proceedings - IEEE International Conference on Robotics and Automation*, 1998, vol. 3, pp. 2014–2021.

[156] J. Pratt, C. M. Chew, A. Torres, P. Dilworth, and G. Pratt. "Virtual model control: An intuitive approach for bipedal locomotion." *International Journal of Robotics Research*, vol. 20, no. 2 pp. 129–143, 2001.

[157] J. Hu, J. Pratt, C. Chee-Meng, H. Herr, and G. Pratt. "Adaptive virtual model control of a bipedal walking robot." In *Proceedings. IEEE International Joint Symposia on Intelligence and Systems (Cat. No.98EX174)*, 1998, pp. 245–251.

[158] C. M. Chew and G. A. Pratt. "Minimum model adaptive control approach for a planar biped." In *IEEE International Conference on Intelligent Robots and Systems*, 1999, vol. 3, pp. 1469–1474.

[159] C. M. Chew and G. A. Pratt. "Dynamic bipedal walking assisted by learning." *Robotica*, vol. 20, no. 5 pp. 477–491, 2002.

[160] J. Hu, J. Pratt, and G. Pratt. "Stable adaptive control of a bipedal walking robot with CMAC neural networks." *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2 pp. 1050–1056, 1999.

[161] C. Chew and G. Pratt. "A general control architecture for dynamic bipedal walking." In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2000, vol. 4, pp. 3989–3995.

[162] F. B. Horak and L. M. Nashner. "Central programming of postural movements: Adaptation to altered support-surface configurations." *J Neurophysiol*, vol. 55, no. 6 pp. 1369–1381, Jun 1986.

[163] M. Guihard and P. Gorce. "Hip strategy applied to biped dynamic control." In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2002, vol. 1, pp. 159–164.

[164] M. H. Raibert. *Legged robots that balance*. MIT Press, Cambridge, Mass., 1986.

[165] M. Abdallah and K. Waldron. "A physical model and control strategy for biped running." In *2007 IEEE International Conference on Robotics and Automation*, 2007, p. 7.

[166] E. R. Dunn and R. D. Howe. "Foot placement and velocity control in smooth bipedal

walking." In *Proceedings - IEEE International Conference on Robotics and Automation*, 1996, vol. 1, pp. 578–583.

[167] J. Pratt and S. Drakunov. "Derivation and application of a conserved orbital energy for the inverted pendulum bipedal walking model." In *2007 IEEE International Conference on Robotics and Automation (IEEE Cat No. 07CH37836D)*, 2007, p. 8.

[168] J. Morimoto, G. Cheng, C. Atkeson, and G. Zeglin. "A simple reinforcement learning algorithm for biped walking." In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, vol. 3, pp. 3030–3035.

[169] J. Morimoto, J. Nakanishi, G. Endo, and G. Cheng. "Acquisition of a biped walking pattern using a poincare map." In *2004 4th IEEE/RAS International Conference on Humanoid Robots (IEEE Cat. No. 04EX989)*, 2004, vol. 2, pp. 912–924.

[170] M. A. Townsend. "Biped gait stabilization via foot placement." *Journal of Biomechanics*, vol. 18, no. 1 pp. 21–38, 1985.

[171] C. M. Chew and G. A. Pratt. "Frontal plane algorithms for dynamic bipedal walking." *Robotica*, vol. 22, no. 1 pp. 29–39, 2004.

[172] G. van Oort and S. Stramigioli. "Using time-reversal symmetry for stabilizing a simple 3D walker model." In *2007 IEEE International Conference on Robotics and Automation (IEEE Cat No. 07CH37836D)*, 2007, p. 6.

[173] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation." In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, 2001, vol. 1, pp. 239–246.

[174] S. Kajita and K. Tani. "Experimental study of biped dynamic walking in the linear inverted pendulum mode." In *Proceedings of 1995 IEEE International Conference on Robotics and Automation (Cat. No.95CH3461-1)*, 1995, vol. 3, pp. 2885–2891.

[175] S. Kajita, T. Yamaura, and A. Kobayashi. "Dynamic walking control of a biped robot along a potential energy conserving orbit." *IEEE Transactions on Robotics and Automation*, vol. 8, no. 4 pp. 431–438, Aug. 1992.

[176] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa. "A realtime pattern generator for biped walking." In *Proceedings - IEEE International Conference on Robotics and Automation*, 2002, vol. 1, pp. 31–37.

[177] S. Kudoh and T. Komura. "C^2 continuous gait-pattern generation for biped robots." In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, 2003, vol. 2, pp. 1135–1140.

[178] J. Kuffner Jr., K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. "Footstep planning among obstacles for biped robots." In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, 2001, vol. 1, pp. 500–505.

[179] J. K. Hodgins and M. H. Raibert. "Adjusting step length for rough terrain locomotion." *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3 pp. 289–298, 1991.

[180] K. Loffler, M. Gienger, and F. Pfeiffer. "Control of a biped jogging robot." In *6th International Workshop on Advanced Motion Control. Proceedings (Cat. No.00TH8494)*, 2000, pp. 601–605.

[181] A. Chemori and A. Loria. "Walking control strategy for a planar under-actuated biped robot based on optimal reference trajectories and partial feedback linearization." In *Proceedings of the Fourth International Workshop on Robot Motion and Control (IEEE Cat. No.04EX891)*, 2004, pp. 61–66.

[182] H. Lum, M. Zribi, and Y. Soh. "Planning and control of a biped robot." *International Journal of Engineering Science*, vol. 37, no. 10 pp. 1319–1349, 1999.

[183] K. Mitobe, N. Mori, K. Aida, and Y. Nasu. "Nonlinear feedback control of a biped walking

robot." In *Proceedings - IEEE International Conference on Robotics and Automation*, 1995, vol. 3, pp. 2865–2870.

[184] M. Nikkhah, H. Ashrafiuon, and F. Fahimi. "Robust control of underactuated bipeds using sliding modes." *Robotica*, vol. 25, no. 3 pp. 367–374, 2007.

[185] F. Pfeiffer, K. Loffler, and M. Gienger. "The concept of jogging JOHNNIE." In *Proceedings - IEEE International Conference on Robotics and Automation*, 2002, vol. 3, pp. 3129–3135.

[186] G. Song and M. Zefran. "Underactuated dynamic three-dimensional bipedal walking." In *Proceedings - IEEE International Conference on Robotics and Automation*, 2006, vol. 2006, pp. 854–859.

[187] M. Spong, R. Lozano, and R. Mahony. "An almost linear biped." In *Proceedings of the IEEE Conference on Decision and Control*, 2000, vol. 5, pp. 4803–4808.

[188] C. Chevallereau. "Time-scaling control for an underactuated biped robot." *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2 pp. 362–368, 2003.

[189] C. Chevallereau, A. Formal'sky, and D. Djoudi. "Tracking a joint path for the walk of an underactuated biped." *Robotica*, vol. 22 pp. 15–28, Jan. 2004.

[190] D. Djoudi and C. Chevallereau. "Feet can improve the stability property of a control law for a walking robot." In *Proceedings. 2006 Conference on International Robotics and Automation (IEEE Cat. No. 06CH37729D)*, 2006, pp. 1206–1212.

[191] M. Doi, Y. Hasegawa, and T. Fukuda. "Passive trajectory control of the lateral motion in bipedal walking." In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, vol. 3, pp. 3049–3054.

[192] M. Doi, Y. Hasegawa, and T. Fukuda. "Passive dynamic autonomous control of bipedal walking." In *Proceedings of the 2004 International Symposium on Micro-NanoMechatronics and Human Science, MHS2004; The Fourth Symposium 'Micro-NanoMechatronics for and Information-Based Society' The 21st Century*, 2004, pp. 151–156.

[193] C. Fu, M. Shuai, Y. Huang, J. Wang, and K. Chen. "Parametric walking patterns and optimum atlases for underactuated biped robots." In *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 342–347.

[194] S. Pannu, H. Kazerooni, G. Becker, and A. Packard. "μ-synthesis control for a walking robot." *IEEE Control Systems Magazine*, vol. 16, no. 1 pp. 20–25, 1996.

[195] J. H. Park. "Impedance control for biped robot locomotion." *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6 pp. 870–882, 2001.

[196] M. Raibert, S. Tzafestas, and C. Tzafestas. "Comparative simulation study of three control techniques applied to a biped robot." In *Conference Proceedings. 1993 International Conference on Systems, Man and Cybernetics. Systems Engineering in the Service of Humans (Cat. No.93CH3242-5)*, 1993, vol. 1, pp. 494–502.

[197] V. Sangwan and S. Agrawal. "Differentially flat design of bipeds ensuring limit-cycles." In *2007 IEEE International Conference on Robotics and Automation (IEEE Cat No. 07CH37836D)*, 2007, p. 6.

[198] T. Tsuji and K. Ohnishi. "A control of biped robot which applies inverted pendulum mode with virtual supporting point." In *International Workshop on Advanced Motion Control, AMC*, 2002, pp. 478–483.

[199] H. Van der Kooij, R. Jacobs, B. Koopman, and F. Van der Helm. "An alternative approach to synthesizing bipedal walking." *Biological Cybernetics*, vol. 88, no. 1 pp. 46–59, 2003.

[200] M. Xiuping and W. Qiong. "Development of a complete dynamic model of a planar five-link biped and sliding mode control of its locomotion during the double support phase." *International Journal of Control*, vol. 77, no. 8 pp. 789–799, May 2004.

[201] N. Neville, M. Buehler, and I. Sharf. "A bipedal running robot with one actuator per leg." In *Proceedings - IEEE International Conference on Robotics and Automation*, 2006, vol. 2006, pp. 848–853.

[202] G. Figliolini and M. Ceccarelli. "Walking programming for an electropneumatic biped robot." *Mechatronics*, vol. 9, no. 8 pp. 941–964, Dec. 1999.

[203] D. L. Wight, E. G. Kubica, and D. W. Wang. "Introduction of the foot placement estimator: A dynamic measure of balance for bipedal robotics." In *ASME Journal of Computational and Nonlinear Dynamics*, Accepted for publication, January 2008.

[204] H. K. Khallil. *Nonlinear Systems*. Prentice Hall Inc, Upper Saddle River, New Jersey, 3rd edn., 2002.

[205] J. Pratt and R. Tedrake. "Velocity-based stability margins for fast bipedal walking." In *First Ruperto Carola Symposium in the International Science Forum of the University of Heidelberg entitled "Fast Motions in Biomechanics and Robots"*, September 7-9 2006, pp. 299–324.

[206] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. "Capture point: A step toward humanoid push recovery." In *Proceedings of the 2006 IEEE-RAS International Conference on Humanoid Robots, Genoa, Italy*, December 4-6 2006.

[207] M. Wisse, C. Atkeson, and D. Kloimwieder. "Swing leg retraction helps biped walking stability." In *2005 5th IEEE-RAS International Conference on Humanoid Robots (IEEE Cat. No.05EX1139C)*, 2005, pp. 295–300.

[208] H. Essen. "Average angular velocity." *European Journal of Physics*, vol. 14, no. 5 pp. 201–205, 1993.

[209] C. T. Lynch, editor. *Practical Handbook of Material Science*. CRC Press, 2000 Corporate Blvd NW, Boca Raton, Florida, 1989.

[210] K. H. Hunt and F. R. E. Crossley. "Coefficient of restitution interpreted as damping in vibroimpact." *Journal of Applied Mechanics, Transactions ASME*, vol. 42 Ser E, no. 2 pp. 440–445, 1975.

[211] G. Gilardi and I. Sharf. "Literature survey of contact dynamics modelling." *Mechanism and Machine Theory*, vol. 37, no. 10 pp. 1213–1239, 2002.

[212] P. Bliman and M. Sorine. "Easy-to-use realistic dry friction models for automatic control." *Proceedings of the Third European Control Conference. ECC 95*, vol. 4 pp. 3788–3794, 1995.

[213] G. Wood and D. Kennedy. "Simulating mechanical systems in simulink with SimMechanics." In *The MathWorks, Article 91124v00*, 2003.

[214] J. R. Dormand and P. J. Prince. "Family of embedded Runge-Kutta formulae." *Journal of Computational and Applied Mathematics*, vol. 6, no. 1 pp. 19–26, 1980.

[215] M. J. Kurz, N. Stergiou, J. Heidel, and E. Foster. "A template for the exploration of chaotic locomotive patterns." *Chaos, Solitons and Fractals*, vol. 23, no. 2 pp. 485–493, 2005.

[216] J. C. Doyle, B. A. Francis, and A. R. Tannebaum. *Feedback Control Theory*. MacMillan Publishing Co., Toronto, 1992.

[217] M. Albero, F. Blanes, G. Benet, P. Perez, J. Simo, and J. Coronel. "Distributed real time architecture for small biped robot YABIRO." In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA*, 2005, pp. 653–658.

[218] D. Wollherr, M. Hardt, M. Buss, and O. Von Stryk. "Actuator selection and hardware realization of a small and fast-moving, autonomous humanoid robot." In *IEEE International Conference on Intelligent Robots and Systems*, 2002, vol. 3, pp. 2491–2496.

[219] S. Suzuki, K. Furuta, and S. Hatakeyama. "Passive walking towards running." *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11, no. 4 pp. 371–395, Dec. 2005.

[220] G. Howell and J. Baillieul. "Simple controllable walking mechanisms which exhibit bifurcations." In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, 1998, vol. 3, pp. 3027–3032.

[221] J. C. O. Christopher L. Vaughan, Brian L. Davis. *Dynamics of Human Gait*. Kiboho Publishers, 2nd edn., 1999.

[222] B. L. Davis and C. L. Vaughan. "Phasic behavior of emg signals during gait: Use of multivariate statistics." *Journal of Electromyography and Kinesiology*, vol. 3, no. 1 pp. 51–60, 1993.

[223] D. Winter. *Biomechanics and Motor Control of Human Movement*. John Wiley & Sons Inc., 2nd edn., 1990, New York.

[224] H. Hemami, K. Barin, L. Jalics, and D. Heiss. "Dynamics, stability, and control of stepping." *Annals of Biomedical Engineering*, vol. 32, no. 8 pp. 1153–1160, 2004.

[225] A. Patla. "Strategies for dynamic stability during adaptive human locomotion." *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, no. 2 pp. 48–52, Mar. 2003.

[226] M. Millard, D. Wight, J. McPhee, E. Kubica, and D. Wang. "Evaluation of a human foot placement model." In *Proceedings of the North American Congress on Biomechanics*, 2008.

[227] V. Lebastard, Y. Aoustin, and F. Plestan. "Finite time observer for absolute orientation estimation of a five-link walking biped robot." In *2006 American Control Conference (IEEE Cat. No. 06CH37776C)*, 2006, p. 6.

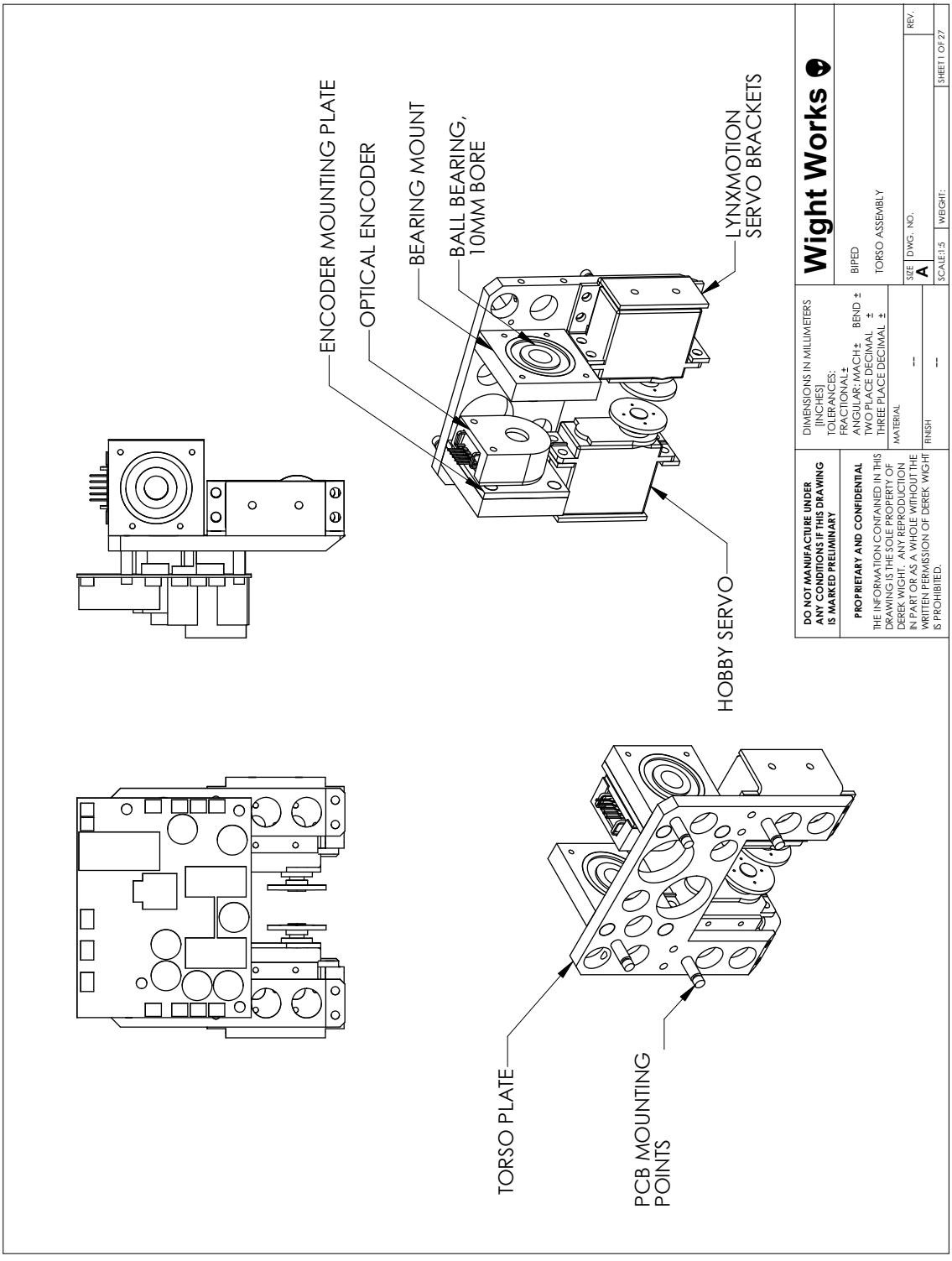
[228] V. Lebastard, Y. Aoustin, F. Plestan, and L. Fridman. "Absolute orientation estimation based on high order sliding mode observer for a five link walking biped robot." In *2006 International Workshop on Variable Structure Systems (IEEE Cat. No. 06EX1316C)*, 2006, pp. 373–378.

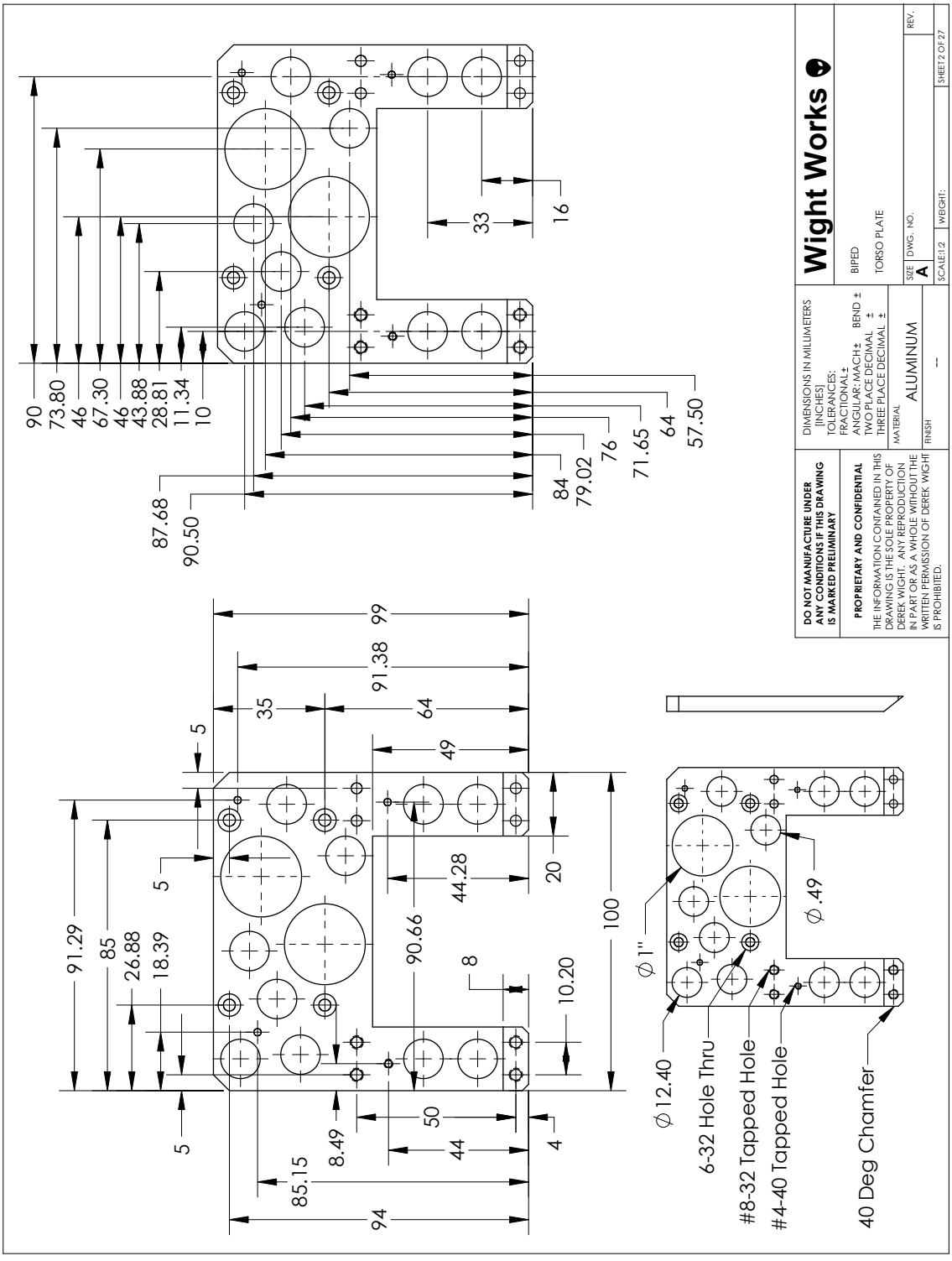
Appendix A

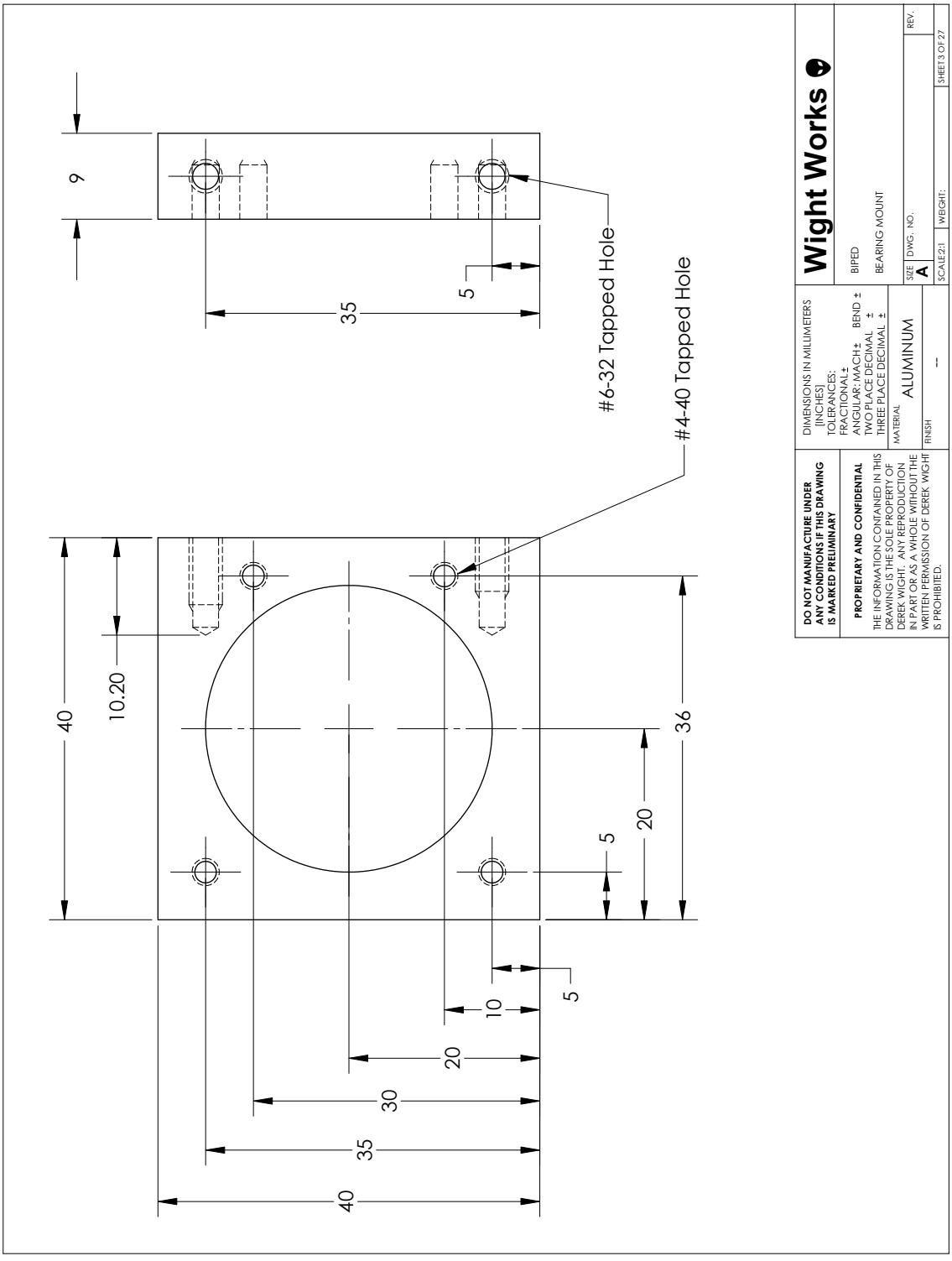
Mechanical Drawings

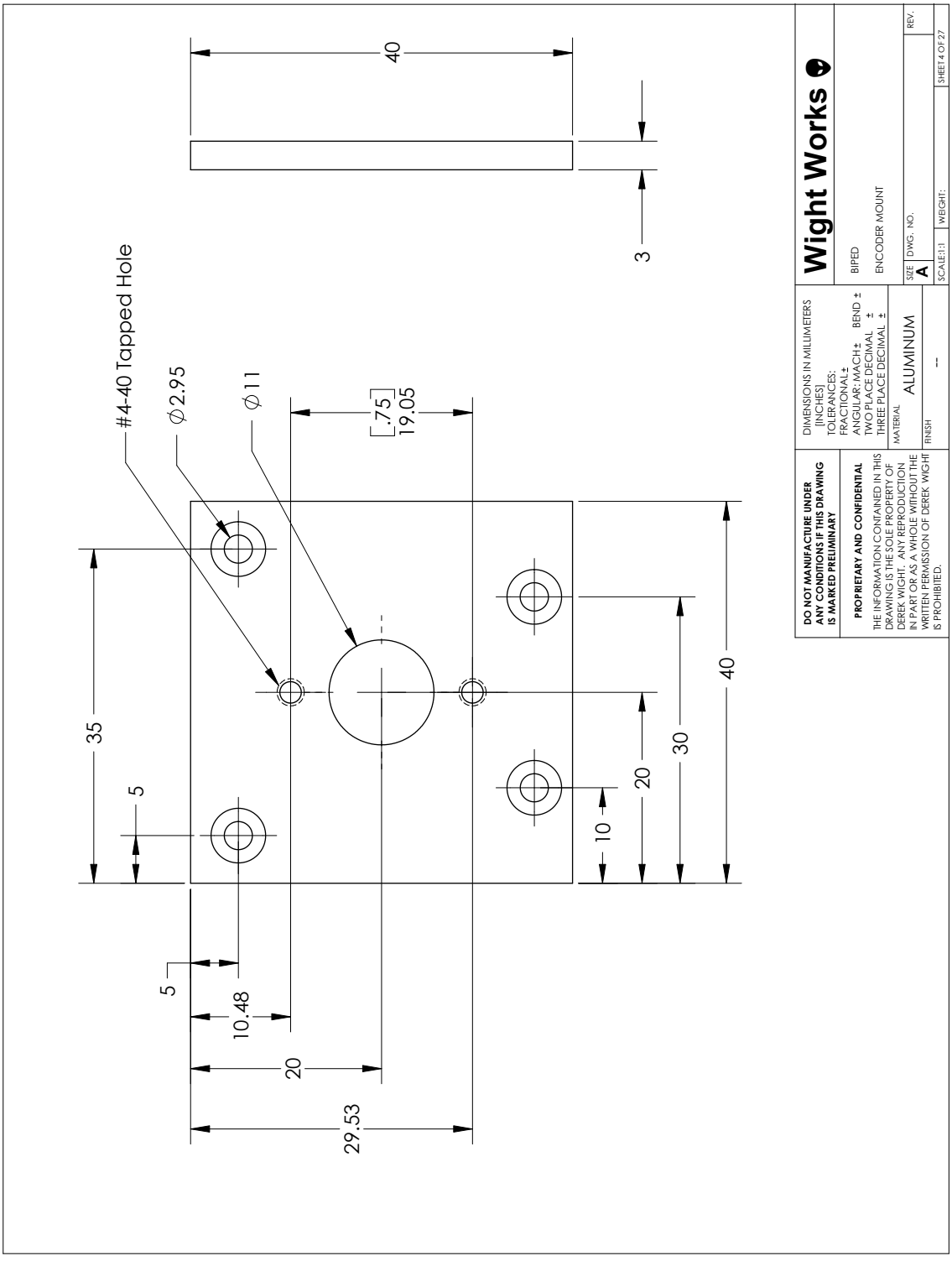
The following drawings detail the mechanical components that were manufactured to construct YABR. The majority of the leg structure was composed of brackets that were purchased from Lynxmotion Inc[1].

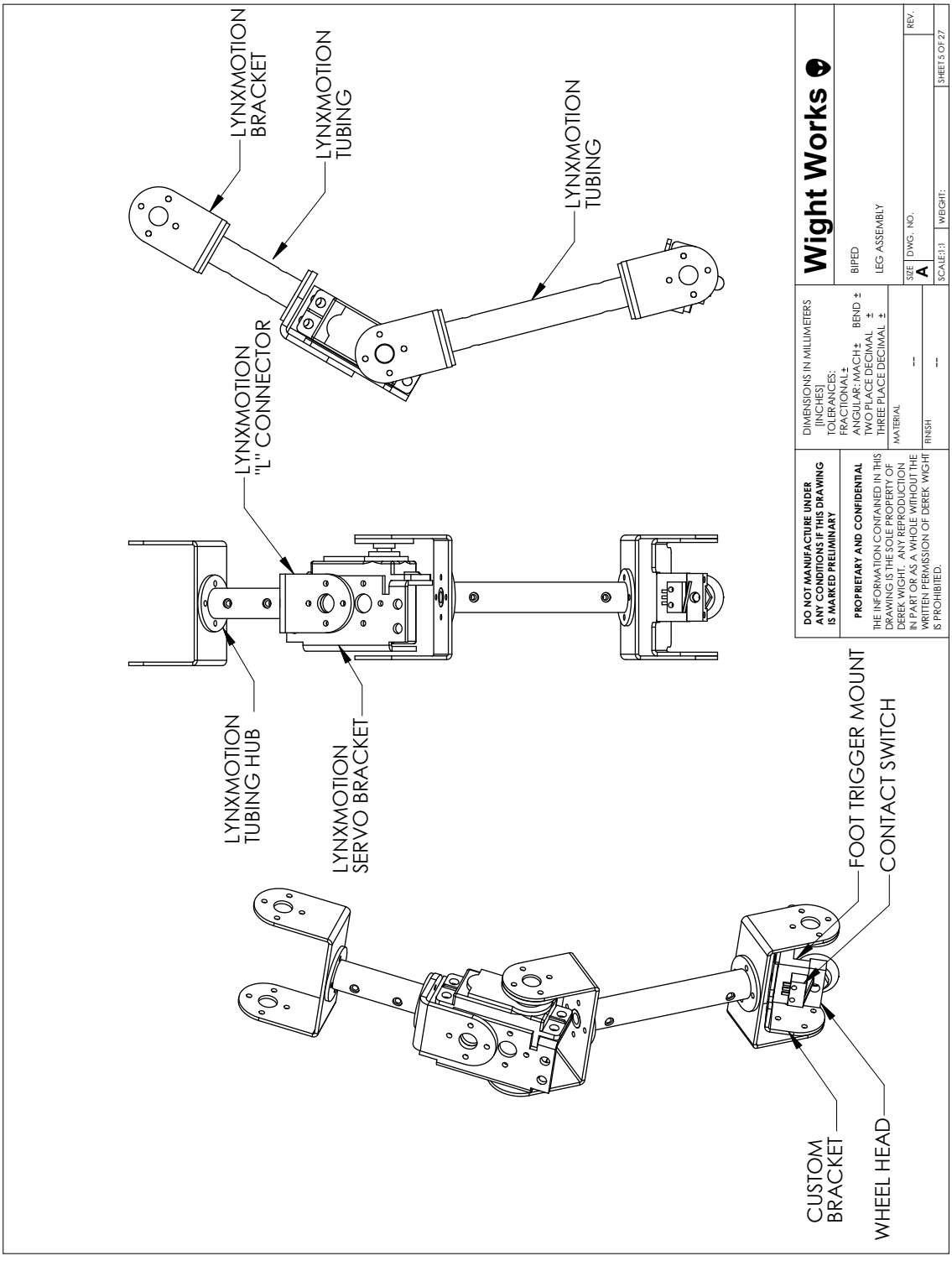
[1] www.lynxmotion.com

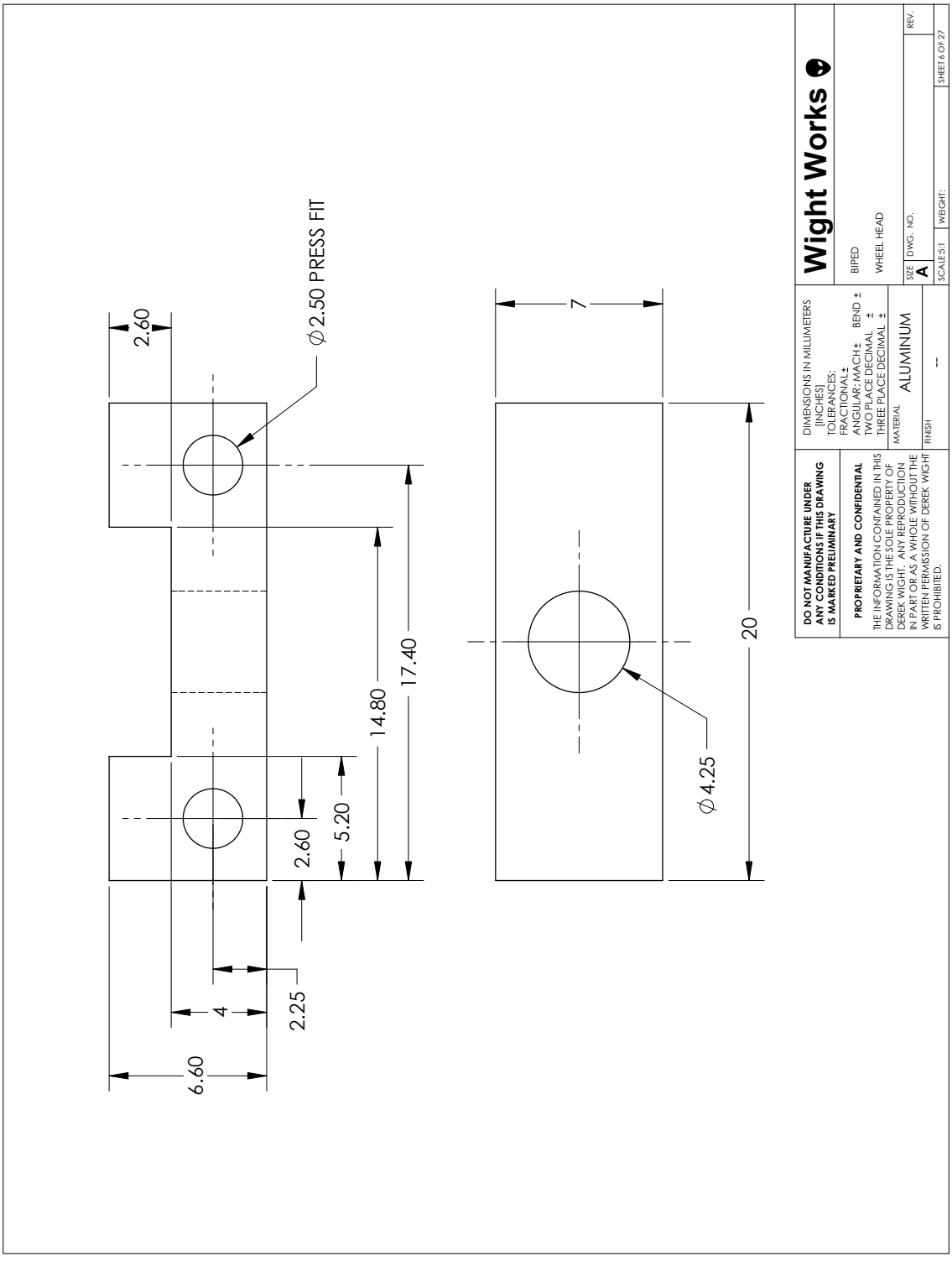

ENCODER MOUNTING PLATE
OPTICAL ENCODER
BEARING MOUNT
BALL BEARING, 10MM BORE
LYNXMOTION SERVO BRACKETS
HOBBY SERVO
TORSO PLATE
PCB MOUNTING POINTS
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL --
FINISH --
Wight Works
BIPED
TORSO ASSEMBLY
SIZE A
DWG. NO.
REV.
SCALE: 1:5
WEIGHT:
SHEET 1 OF 27

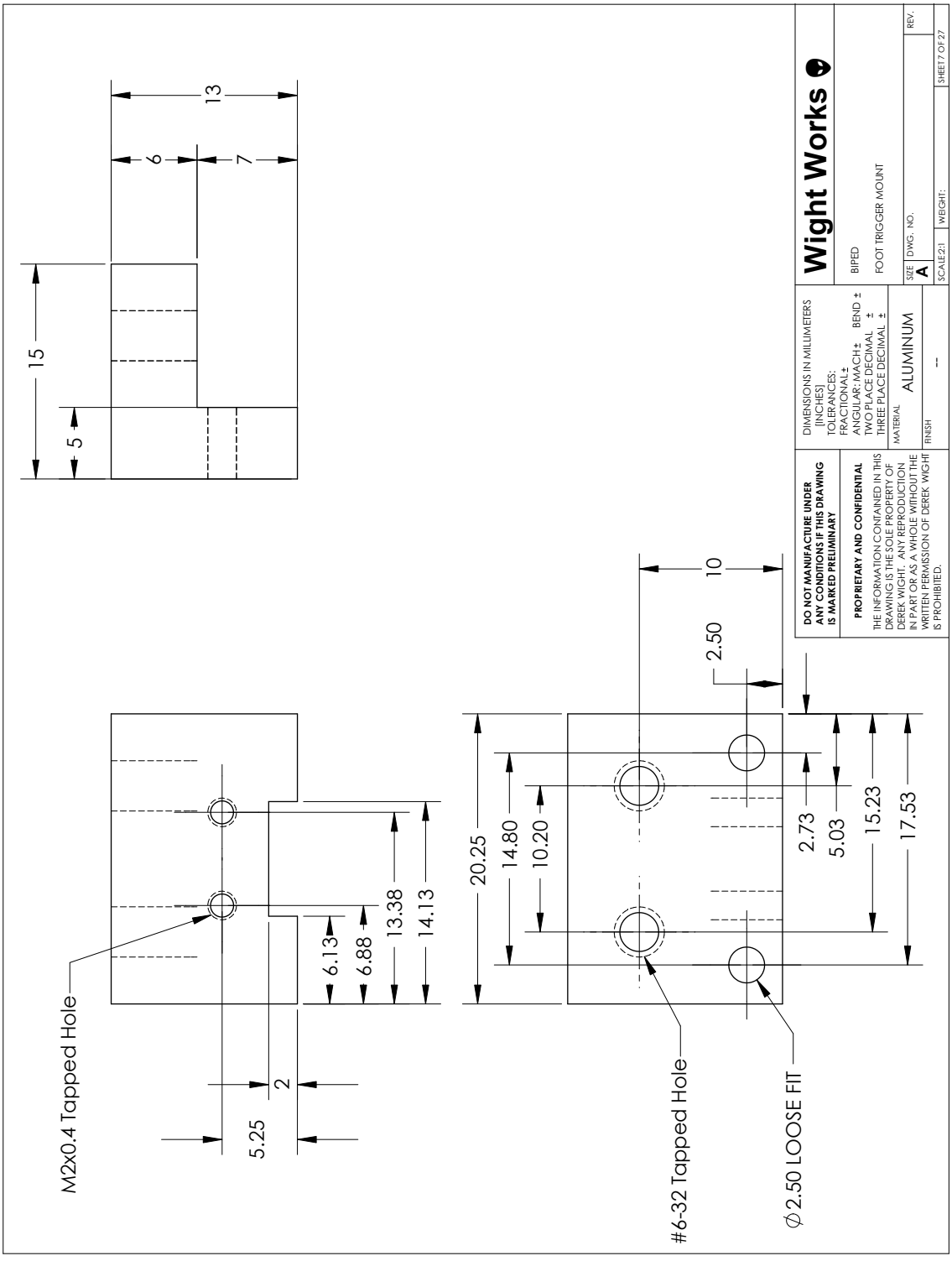

90
73.80
46
67.30
46
43.88
28.81
11.34
10
87.68
90.50
84
79.02
76
71.65
64
57.50
33
16
99
91.38
35
64
49
5
5
44.28
20
90.66
8
10.20
100
91.29
85
26.88
18.39
5
5
8.49
50
44
4
85.15
94
Ø1"
Ø.49
Ø12.40
6-32 Hole Thru
#8-32 Tapped Hole
#4-40 Tapped Hole
40 Deg Chamfer
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ALUMINUM
FINISH
--
Wight Works
BIPED
TORSO PLATE
SIZE A
DWG. NO.
REV.
SCALE: 1:2
WEIGHT:
SHEET 2 OF 27

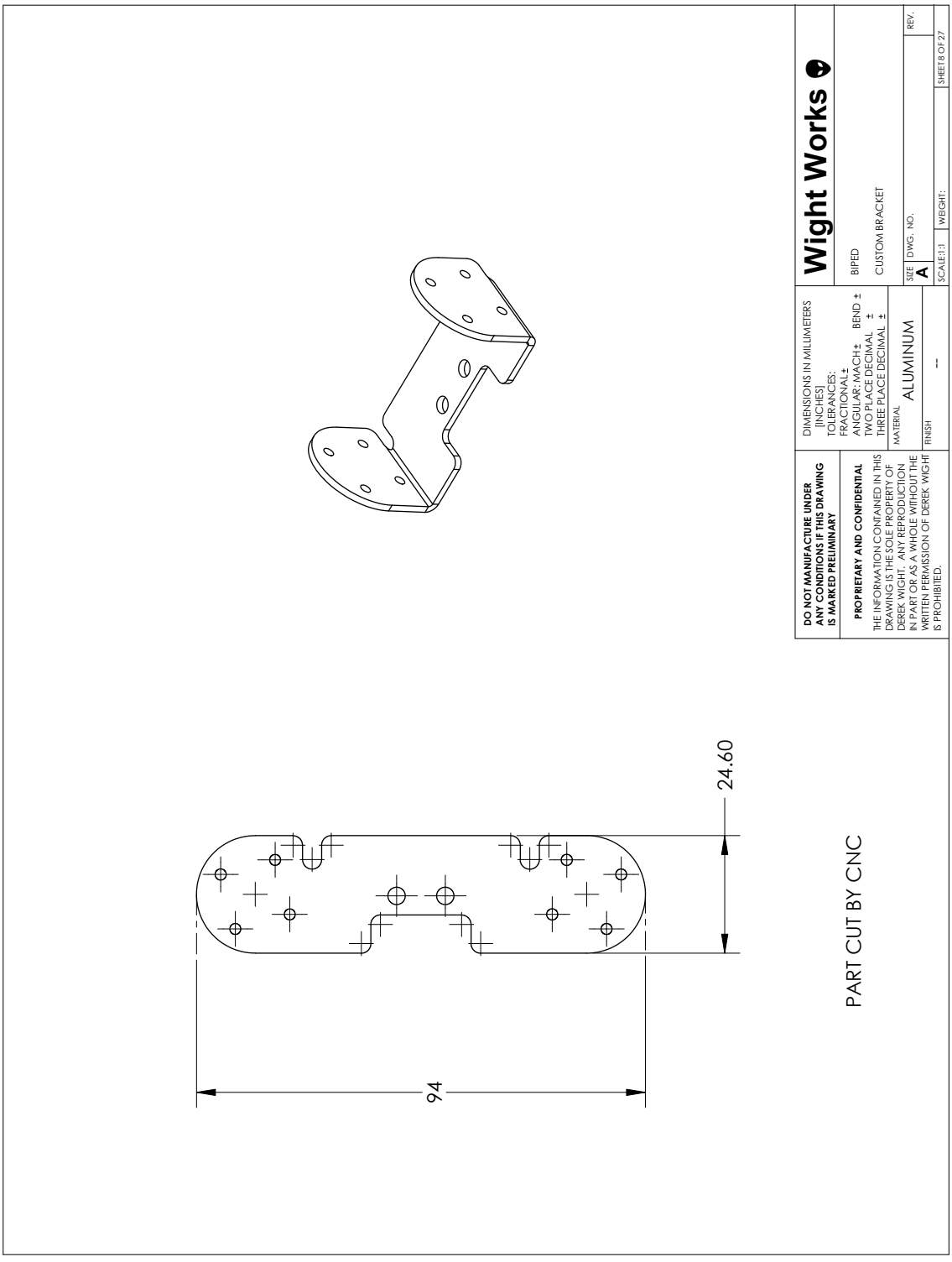

9
35
5
#6-32 Tapped Hole
#4-40 Tapped Hole
40
10.20
36
20
5
10
5
20
30
35
40
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
BEARING MOUNT
SIZE A
DWG. NO.
REV.
SCALE:2:1
WEIGHT:
SHEET 3 OF 27

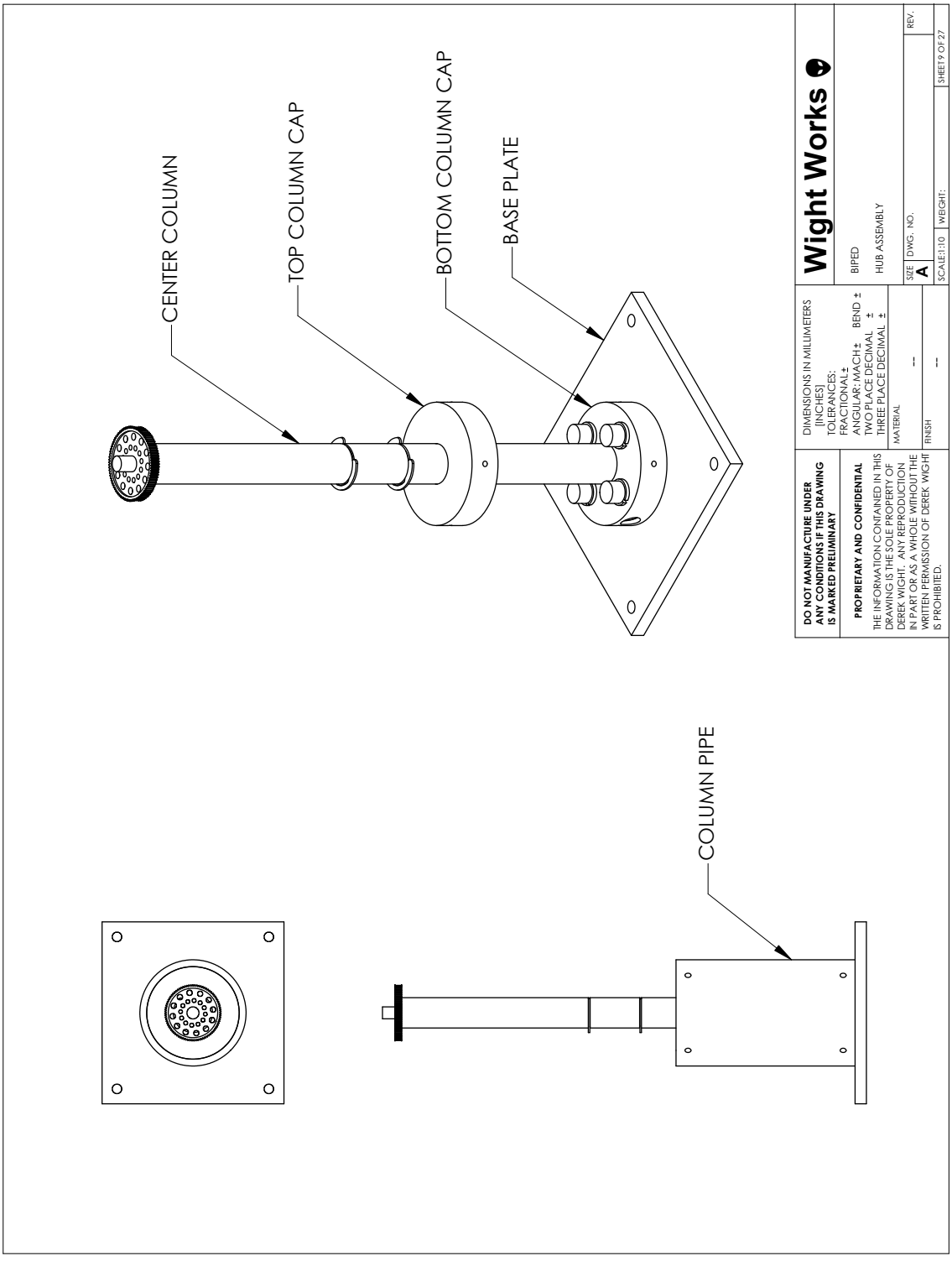

40
3
#4-40 Tapped Hole
ϕ2.95
ϕ11
[.75]
19.05
35
5
5
10.48
20
29.53
10
20
30
40
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ALUMINUM
FINISH
--
Wight Works
BIPED
ENCODER MOUNT
SIZE
A
DWG. NO.
REV.
SCALE:1:1
WEIGHT:
SHEET 4 OF 27

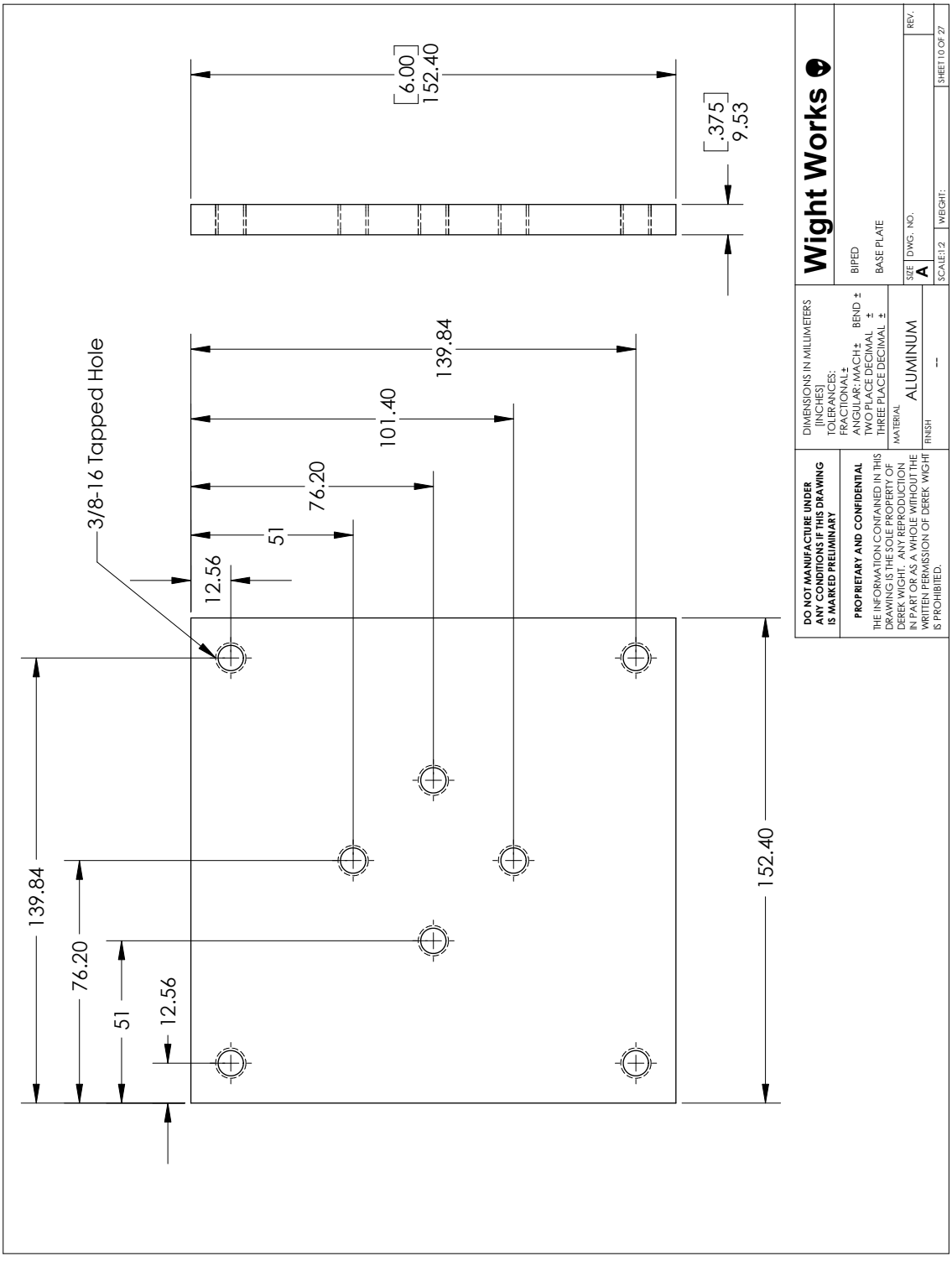

LYNXMOTION BRACKET
LYNXMOTION TUBING
LYNXMOTION TUBING
LYNXMOTION "L" CONNECTOR
LYNXMOTION TUBING HUB
LYNXMOTION SERVO BRACKET
FOOT TRIGGER MOUNT
CONTACT SWITCH
CUSTOM BRACKET
WHEEL HEAD
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL --
FINISH --
Wight Works
BIPED
LEG ASSEMBLY
SIZE A
DWG. NO.
REV.
SCALE:1:1
WEIGHT:
SHEET 5 OF 27

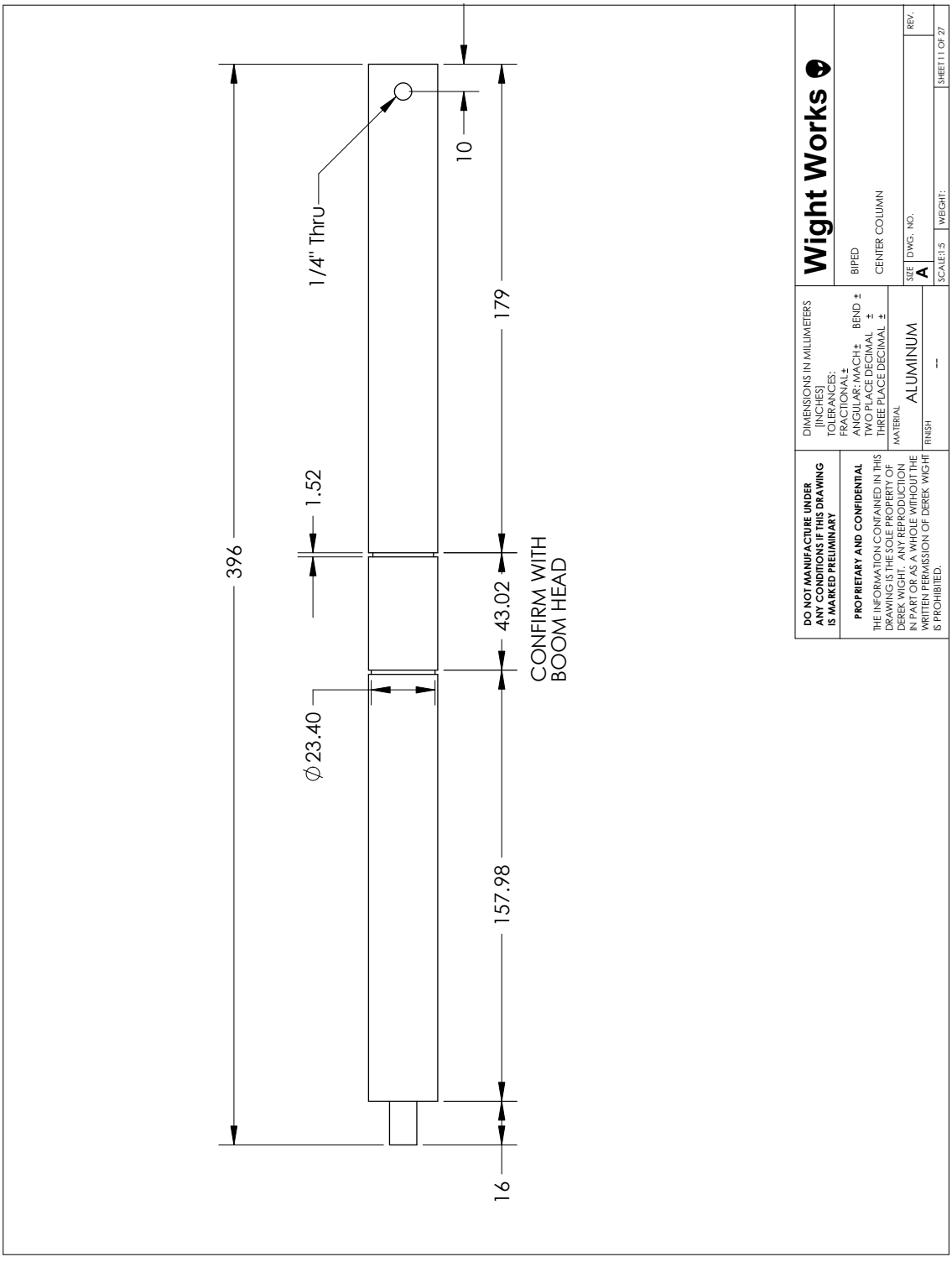

2.60
Ø2.50 PRESS FIT
17.40
14.80
5.20
2.60
2.25
4
6.60
7
20
Ø4.25
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
WHEEL HEAD
SIZE A
DWG. NO.
REV.
SCALE:5:1
WEIGHT:
SHEET 6 OF 27


13
6
7
15
5
M2x0.4 Tapped Hole
5.25
2
6.13
6.88
13.38
14.13
10
2.50
20.25
14.80
10.20
2.73
5.03
15.23
17.53
#6-32 Tapped Hole
Ø 2.50 LOOSE FIT
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ALUMINUM
FINISH
--
Wight Works
BIPED
FOOT TRIGGER MOUNT
SIZE A
DWG. NO.
REV.
SCALE:2:1
WEIGHT:
SHEET 7 OF 27

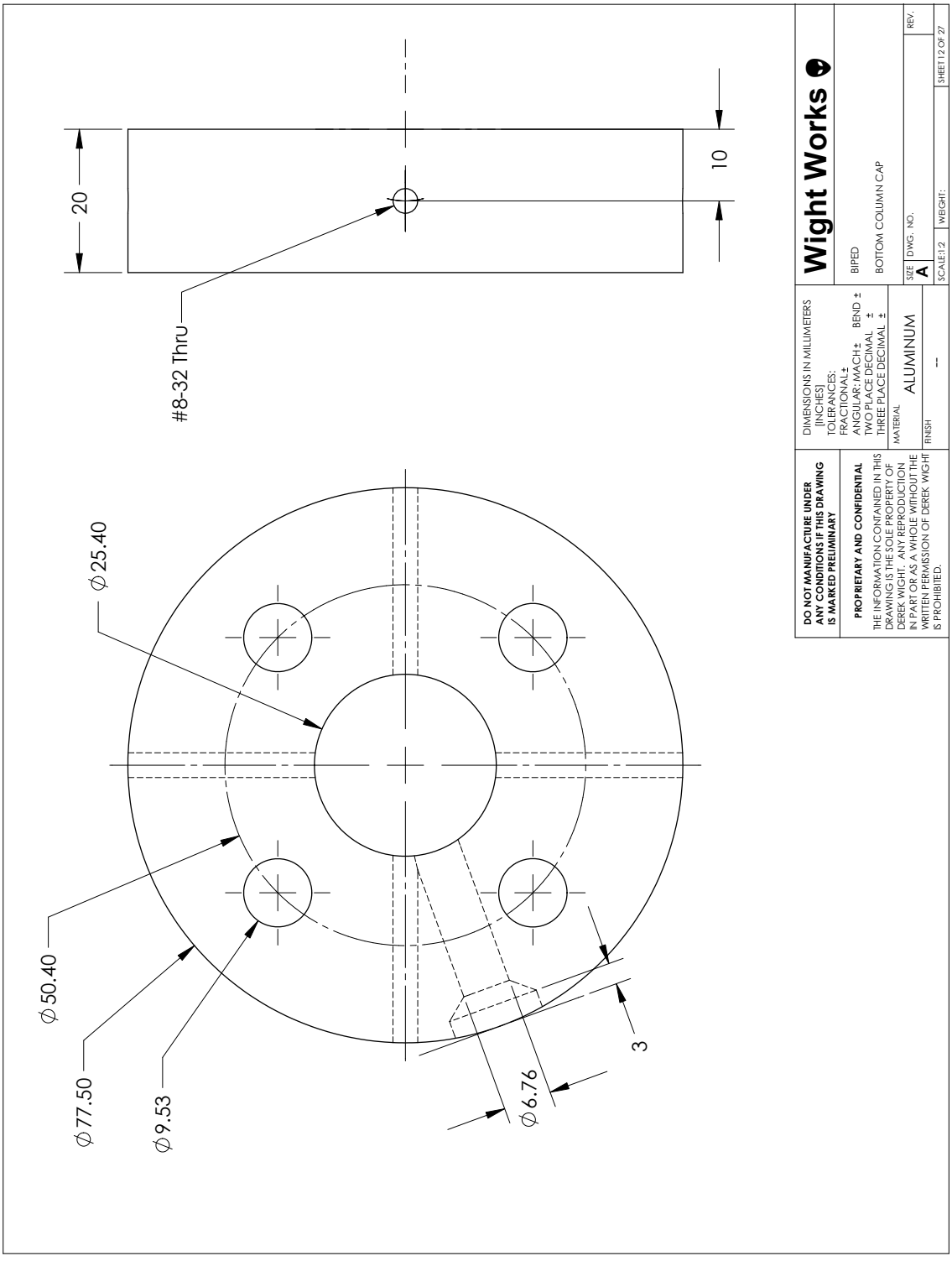

24.60
94
PART CUT BY CNC
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
CUSTOM BRACKET
SIZE A
DWG. NO.
REV.
SCALE:1:1
WEIGHT:
SHEET 8 OF 27

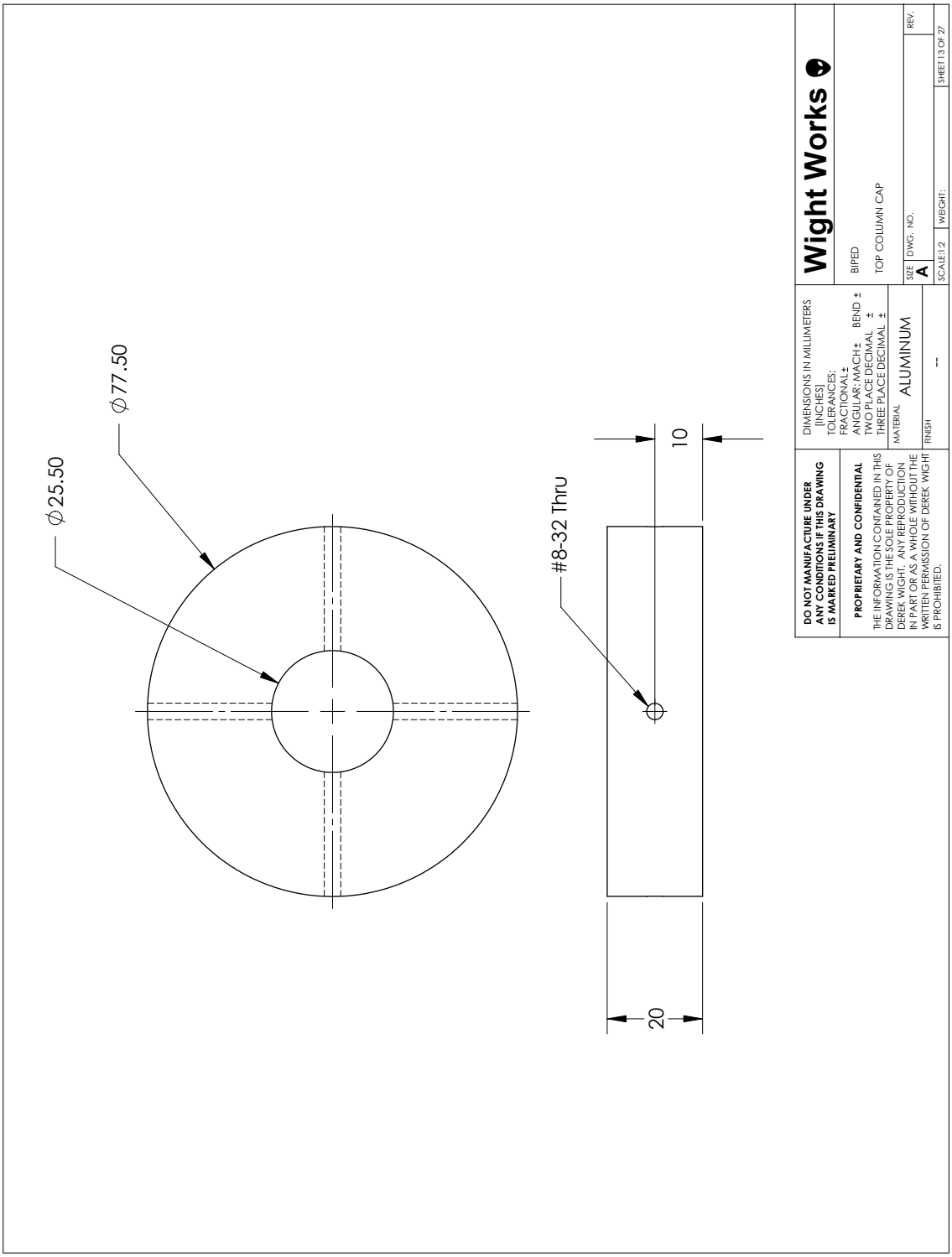

CENTER COLUMN
TOP COLUMN CAP
BOTTOM COLUMN CAP
BASE PLATE
COLUMN PIPE
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL --
FINISH --
Wight Works
BIPED
HUB ASSEMBLY
SIZE A
DWG. NO.
REV.
SCALE:1:10
WEIGHT:
SHEET 9 OF 27

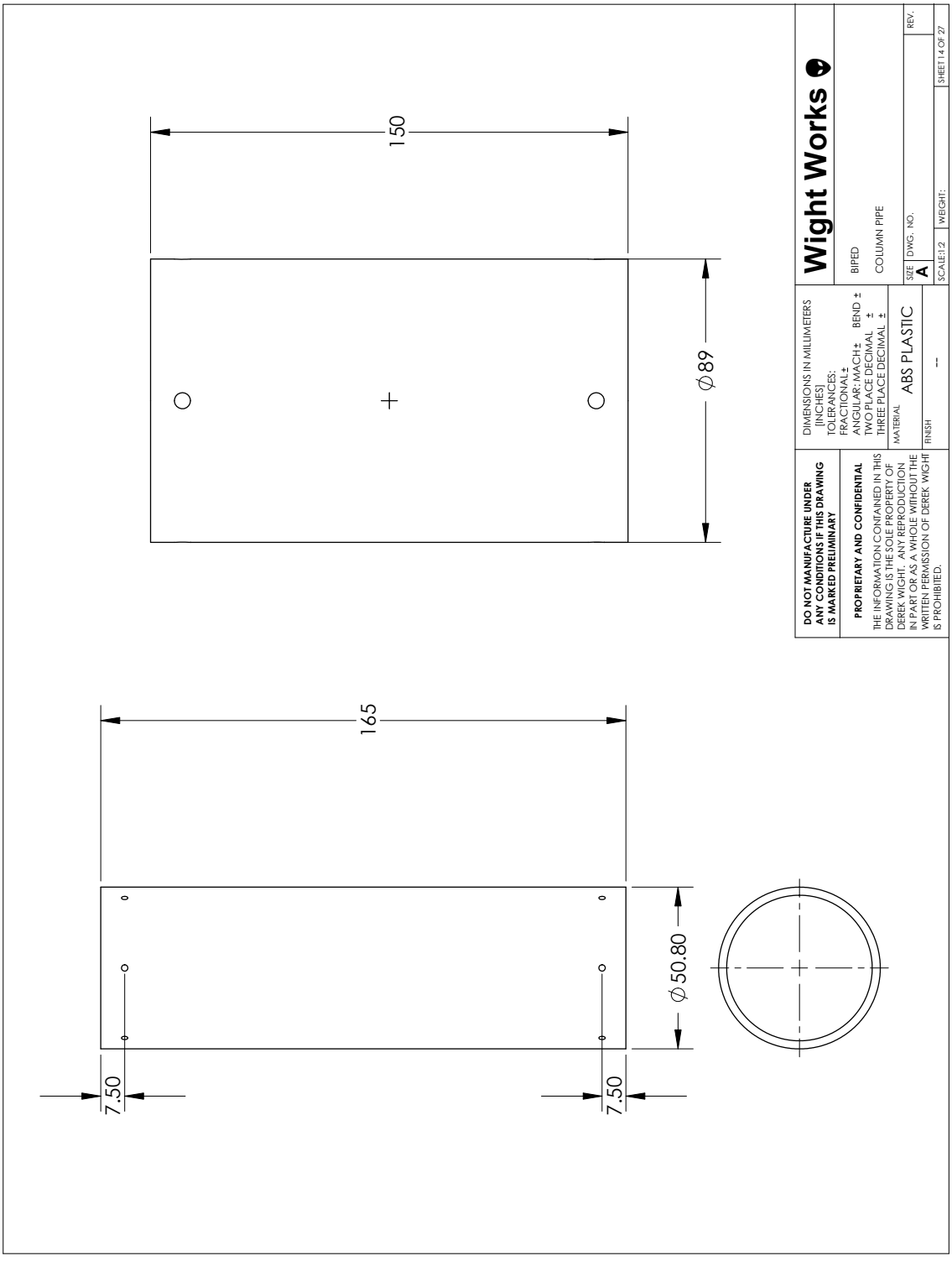

[6.00]
152.40
[.375]
9.53
3/8-16 Tapped Hole
139.84
101.40
76.20
51
12.56
139.84
76.20
51
12.56
152.40
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ALUMINUM
FINISH
--
Wight Works
BIPED
BASE PLATE
SIZE
A
DWG. NO.
REV.
SCALE:1:2
WEIGHT:
SHEET 10 OF 27

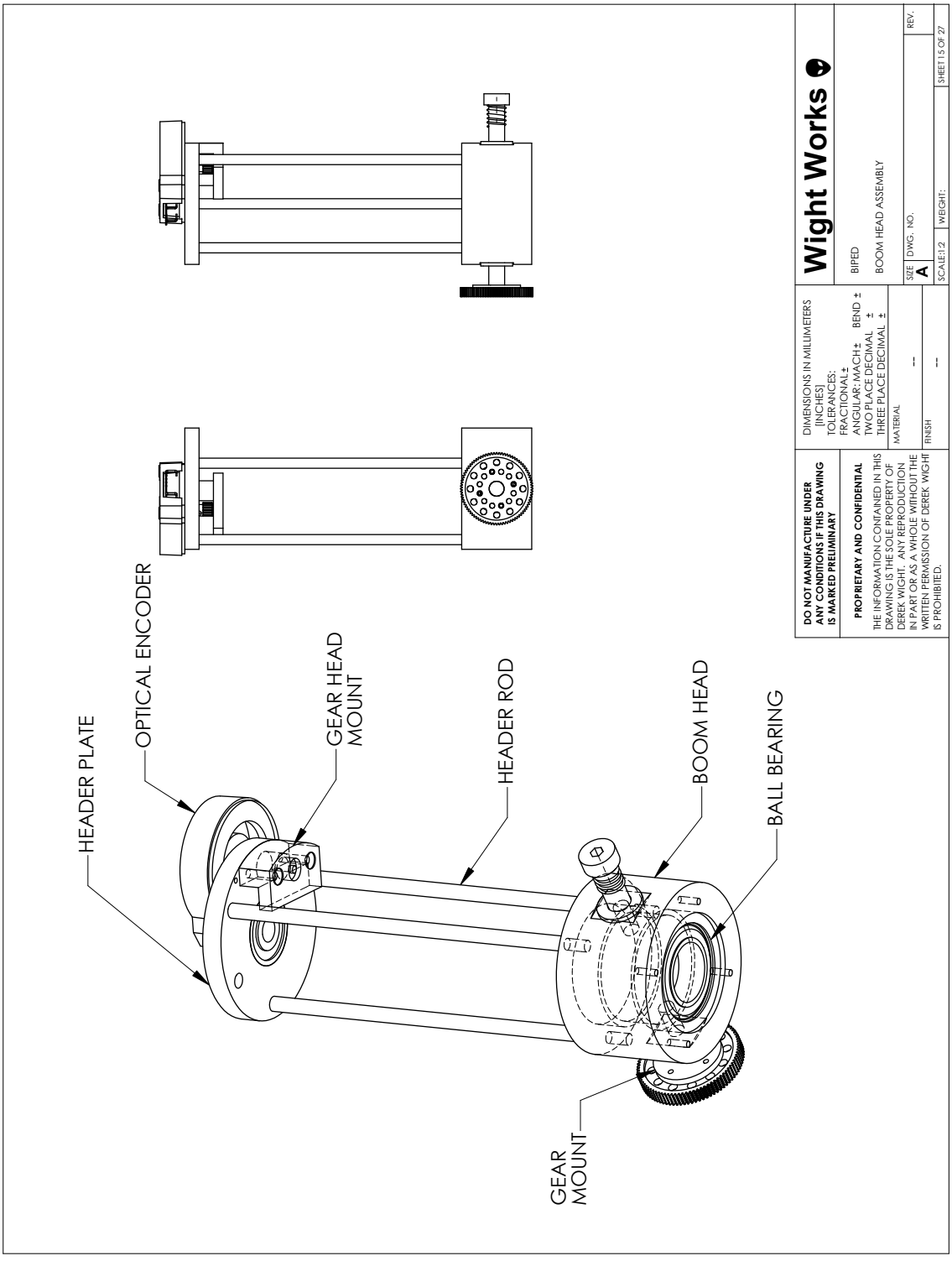

396
1.52
1/4" Thru
10
179
43.02
CONFIRM WITH
BOOM HEAD
ϕ23.40
157.98
16
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ALUMINUM
FINISH

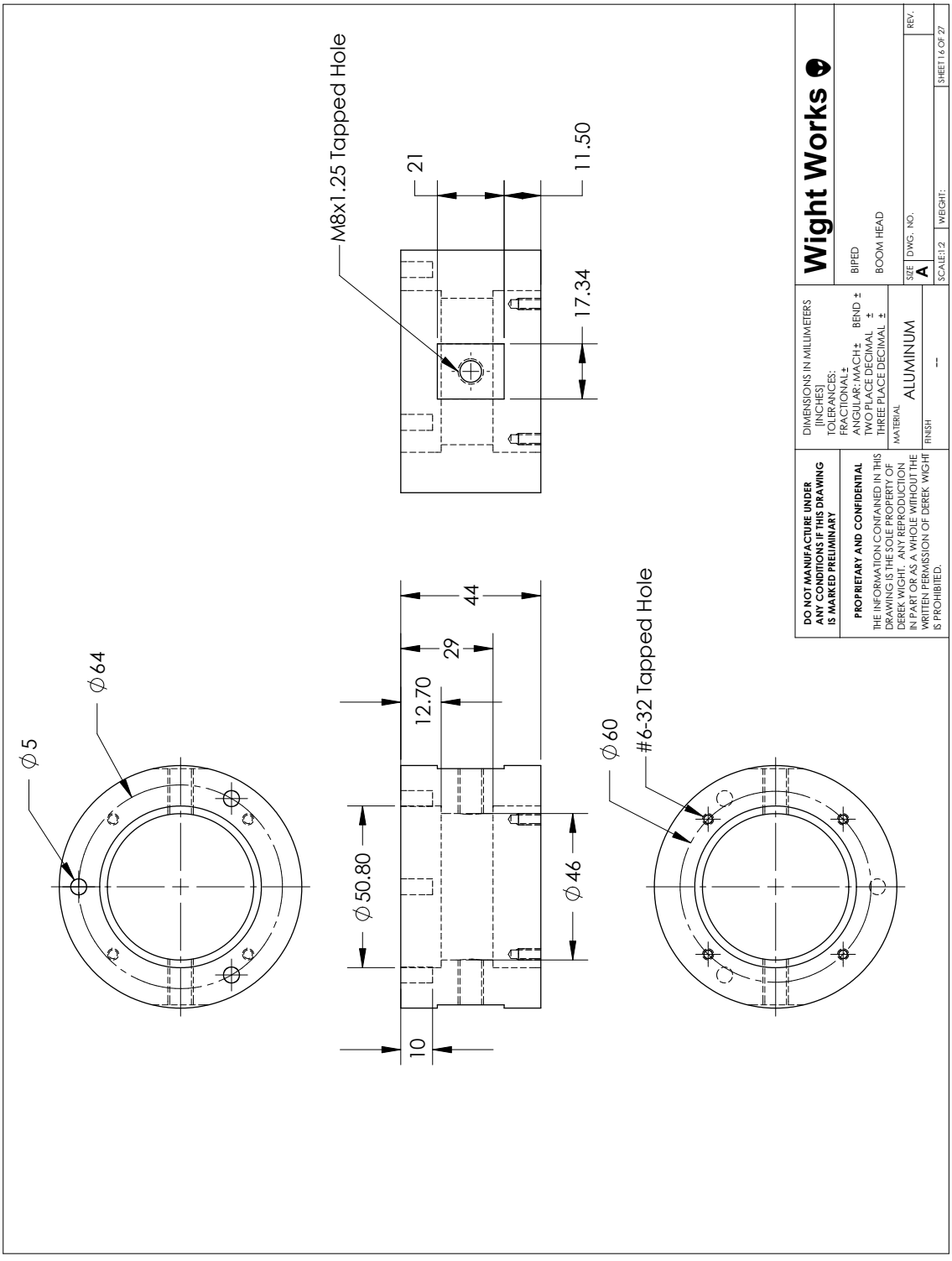
Wight Works
BIPED
CENTER COLUMN
SIZE
A
DWG. NO.
REV.
SCALE:1:5
WEIGHT:
SHEET 11 OF 27

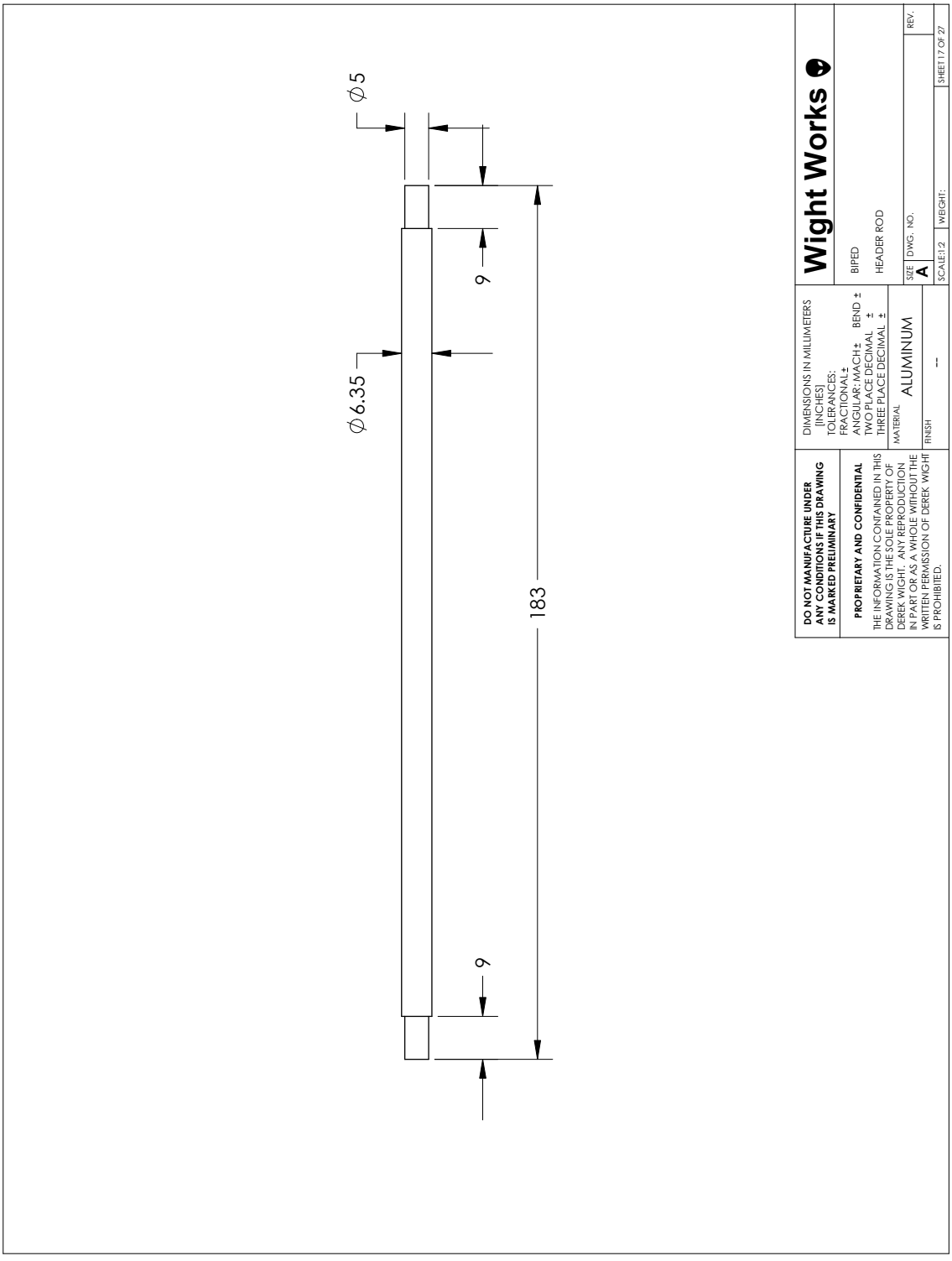

20
10
#8-32 Thru
ϕ25.40
ϕ50.40
ϕ77.50
ϕ9.53
ϕ6.76
3
Wight Works
BIPED
BOTTOM COLUMN CAP
DIMENSIONS IN MILLIMETERS
[INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ALUMINUM
FINISH
--
SIZE
A
DWG. NO.
REV.
SCALE: 1:2
WEIGHT:
SHEET 12 OF 27
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.

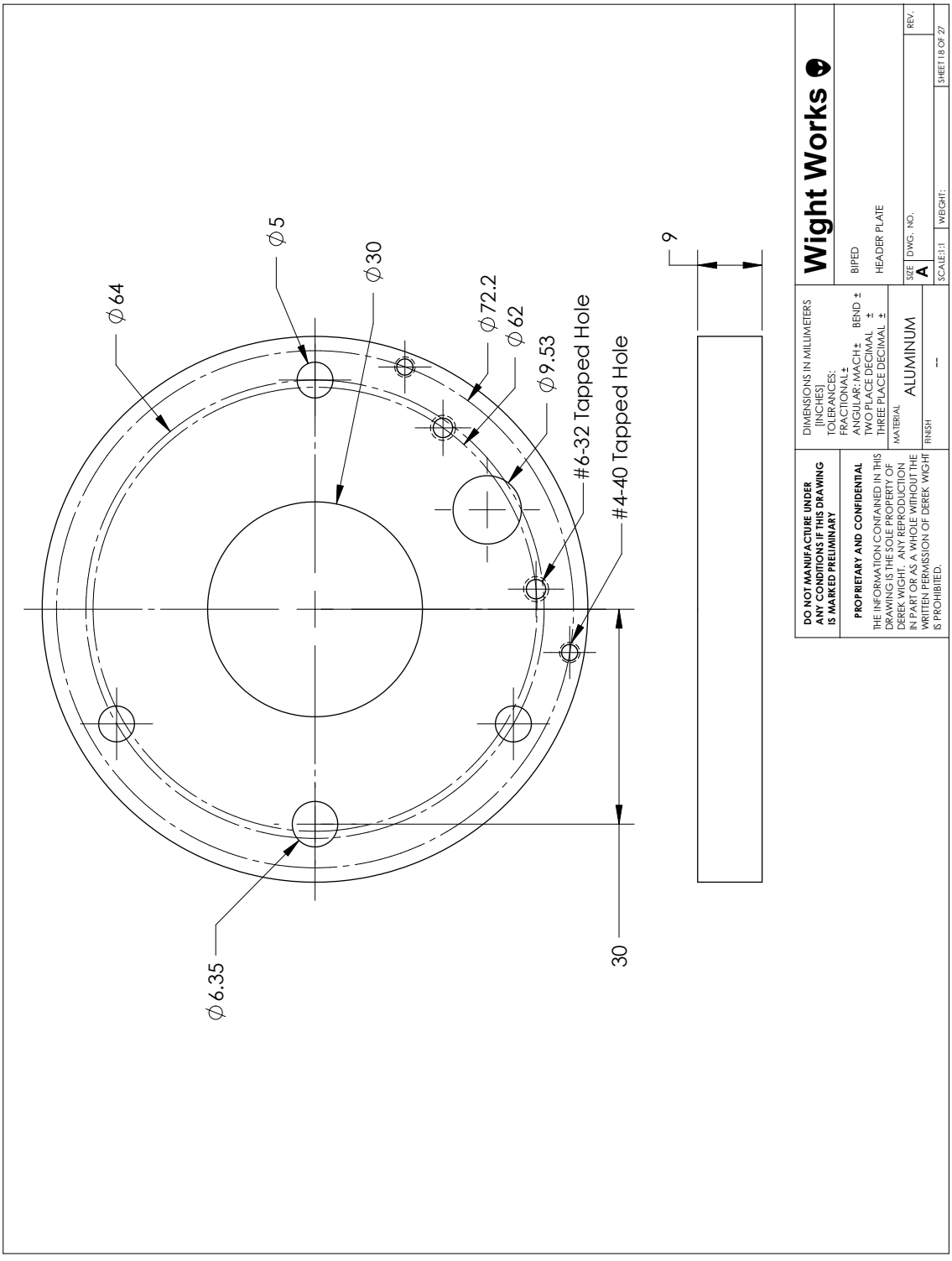

ϕ77.50
ϕ25.50
#8-32 Thru
10
20
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
TOP COLUMN CAP
SIZE A
DWG. NO.
REV:
SCALE:1:2
WEIGHT:
SHEET 13 OF 27

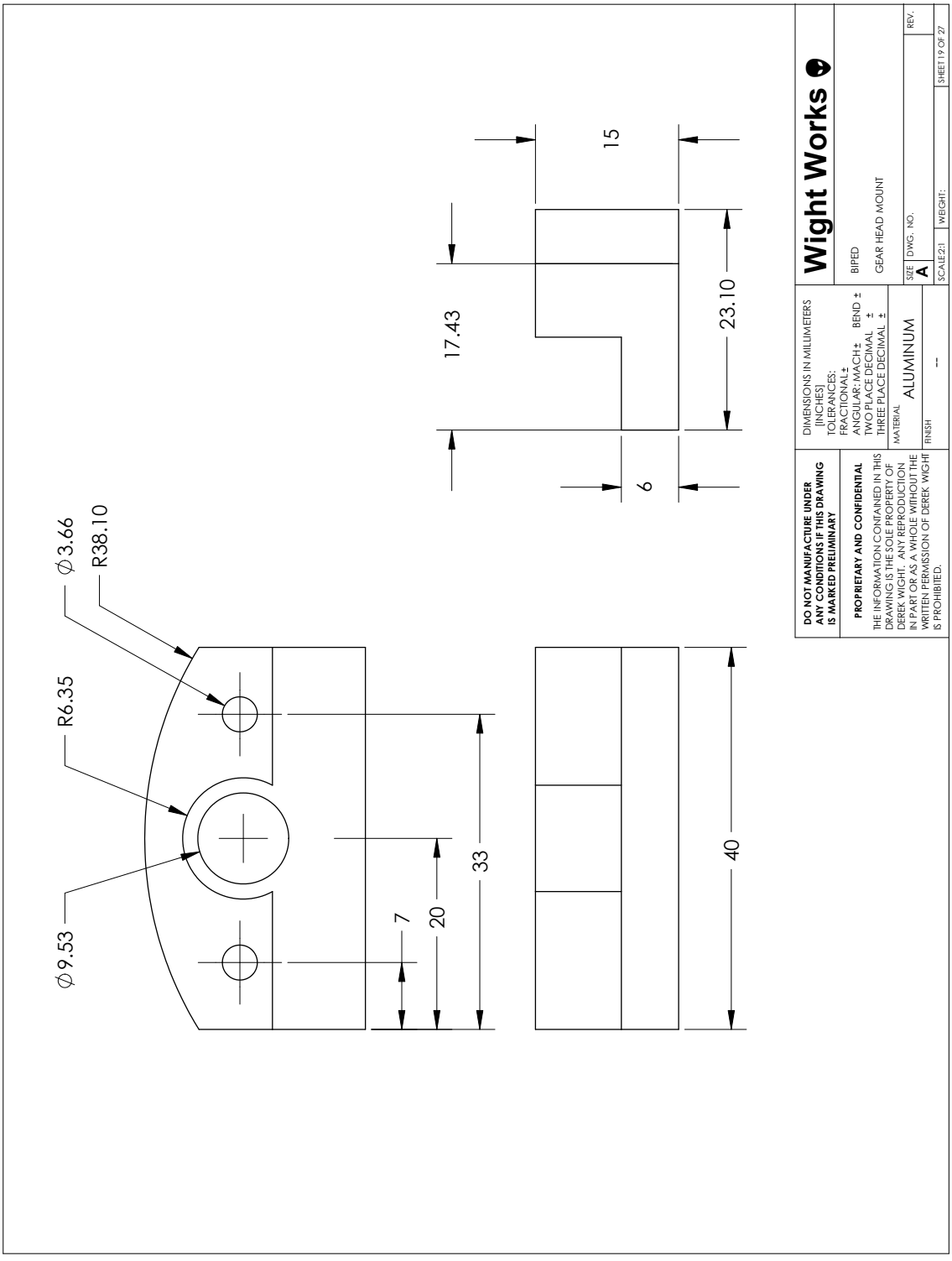

150
Ø89
165
Ø50.80
7.50
7.50
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ABS PLASTIC
FINISH
--
Wight Works
BIPED
COLUMN PIPE
SIZE
A
DWG. NO.
REV:
SCALE:1:2
WEIGHT:
SHEET 14 OF 27

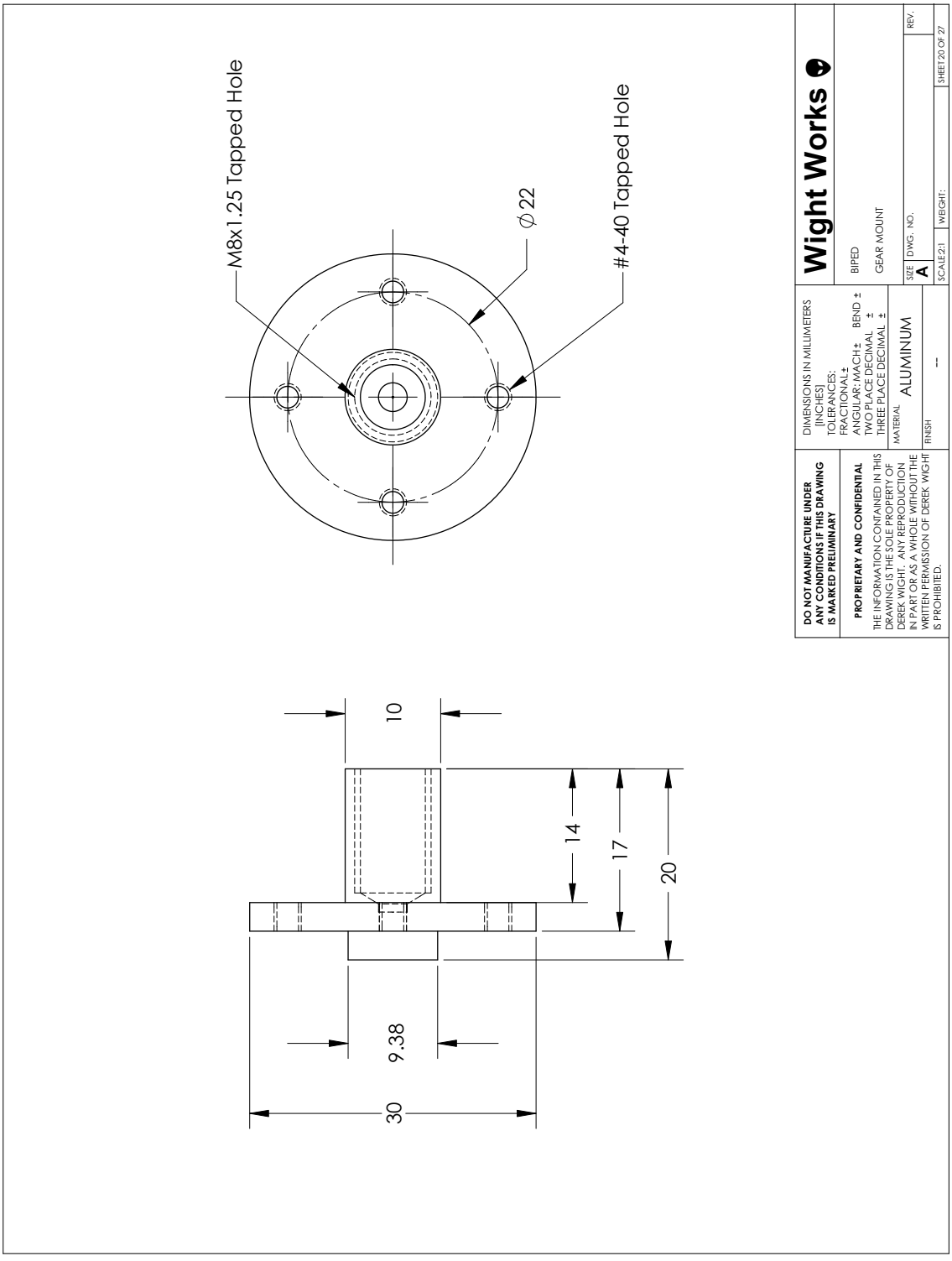

HEADER PLATE
OPTICAL ENCODER
GEAR HEAD MOUNT
HEADER ROD
BOOM HEAD
BALL BEARING
GEAR MOUNT
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL --
FINISH --
Wight Works
BIPED
BOOM HEAD ASSEMBLY
SIZE A
DWG. NO.
REV.
SCALE:1:2
WEIGHT:
SHEET 15 OF 27

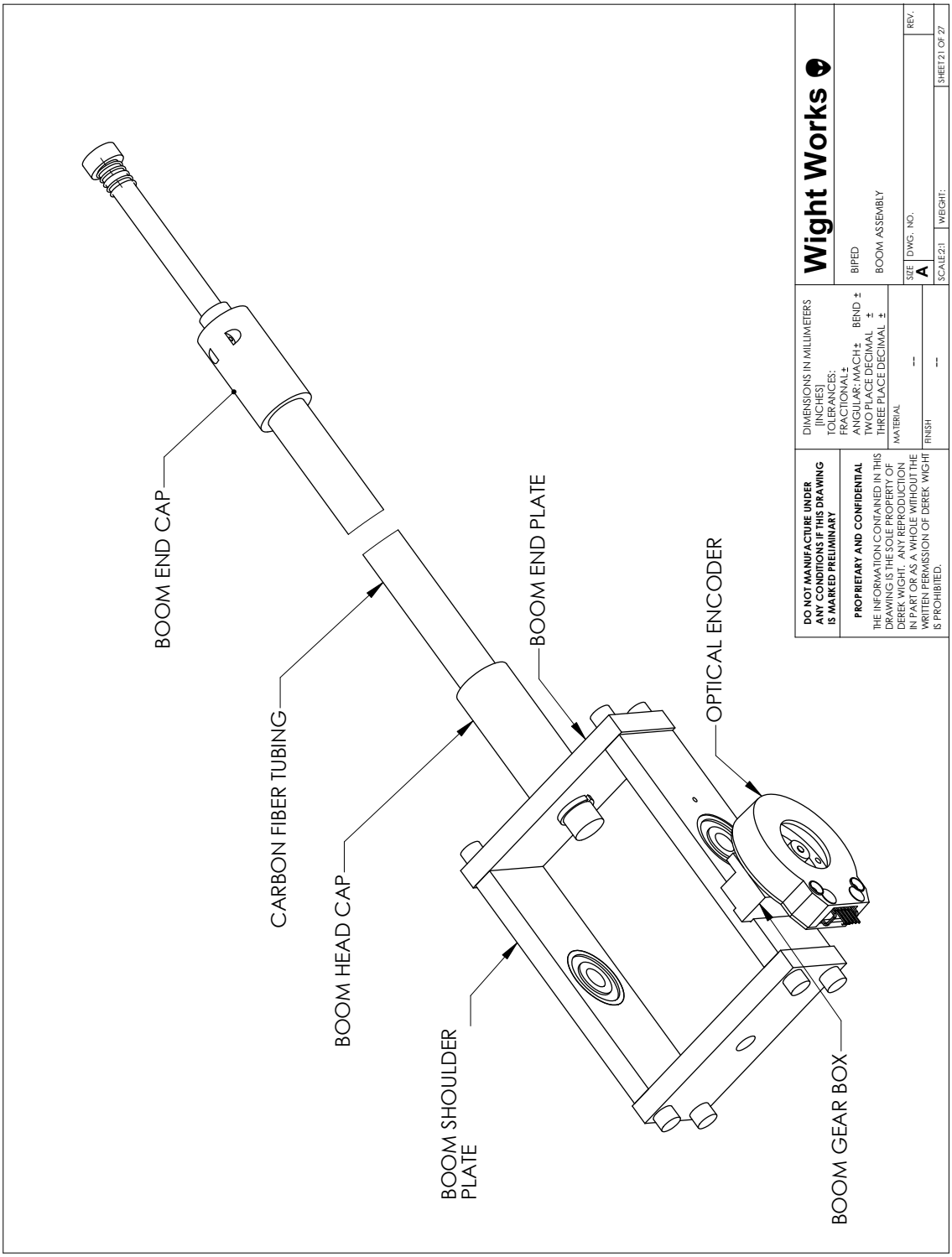

M8x1.25 Tapped Hole
21
11.50
17.34
ϕ64
ϕ5
44
29
12.70
ϕ50.80
ϕ46
10
ϕ60
#6-32 Tapped Hole
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
BOOM HEAD
SIZE A
DWG. NO.
REV.
SCALE: 1:2
WEIGHT:
SHEET 16 OF 27

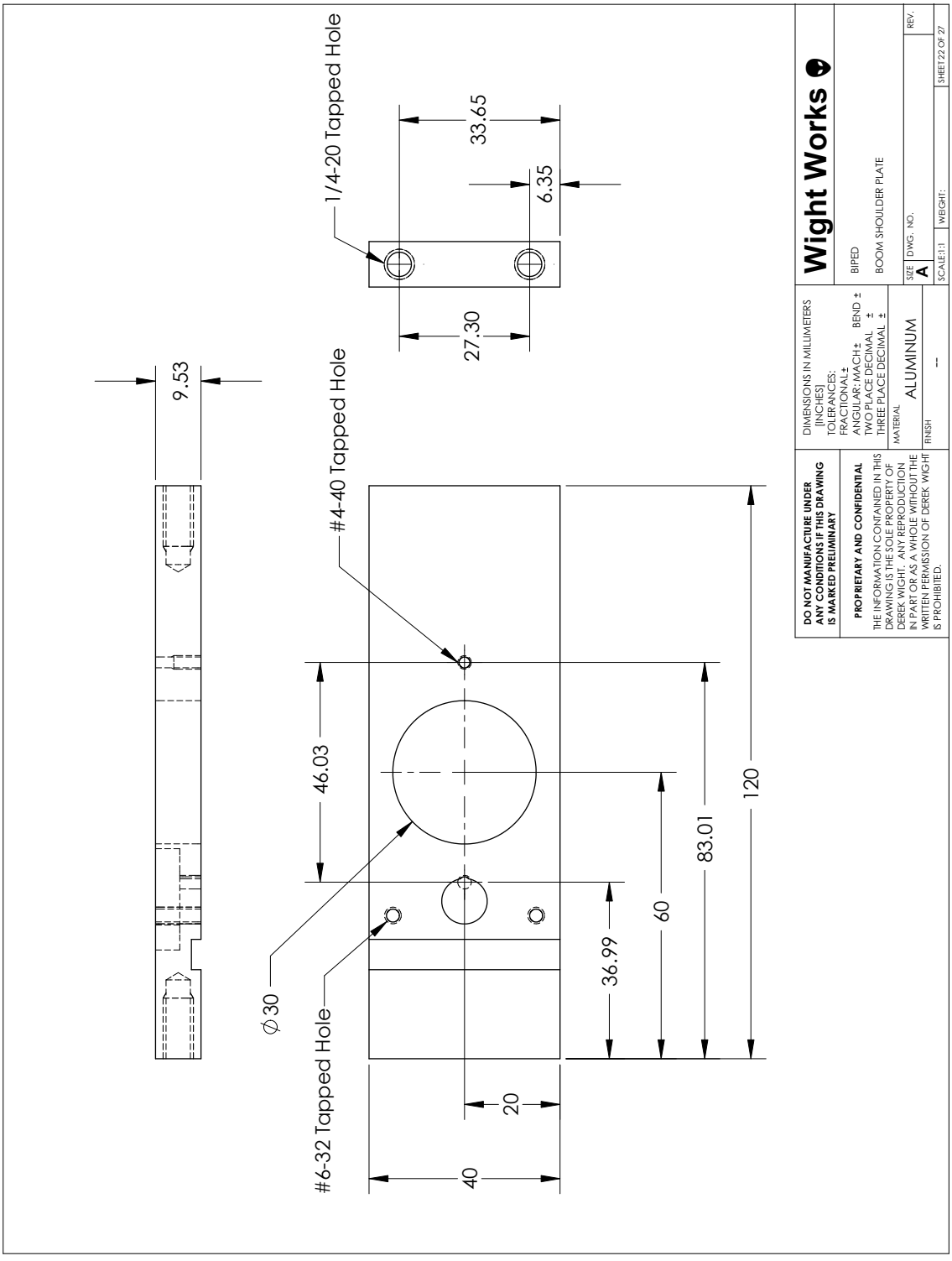

∅5
∅6.35
9
183
9
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
HEADER ROD
SIZE A
DWG. NO.
REV:
SCALE:1:2
WEIGHT:
SHEET 17 OF 27

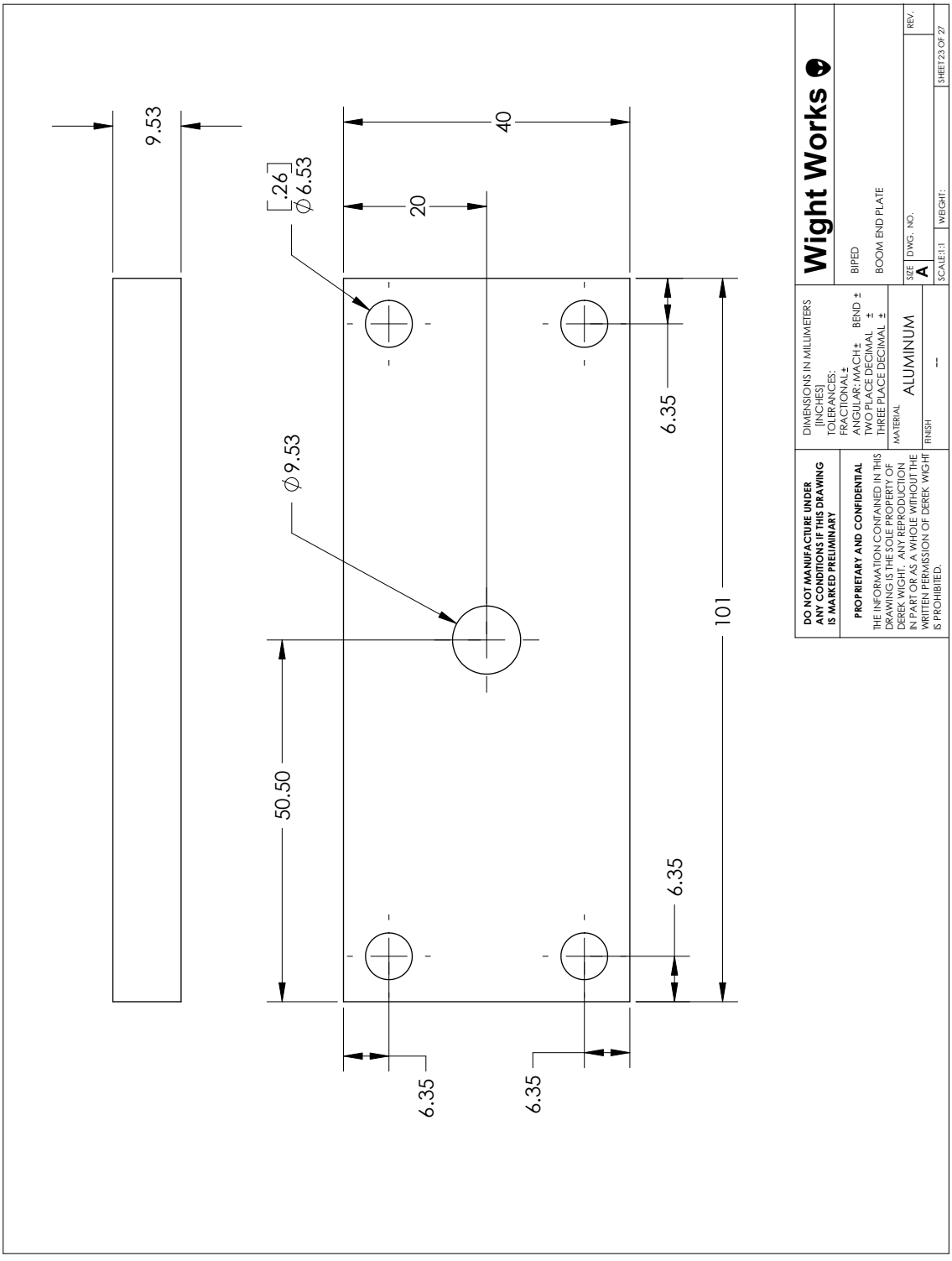

ϕ64
ϕ5
ϕ30
ϕ72.2
ϕ62
ϕ9.53
#6-32 Tapped Hole
#4-40 Tapped Hole
ϕ6.35
30
9
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
HEADER PLATE
SIZE A
DWG. NO.
REV.
SCALE: 1:1
WEIGHT:
SHEET 18 OF 27


Ø3.66
R38.10
R6.35
Ø9.53
7
20
33
40
6
17.43
15
23.10
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
GEAR HEAD MOUNT
SIZE A
DWG. NO.
REV.
SCALE:2:1
WEIGHT:
SHEET 19 OF 27

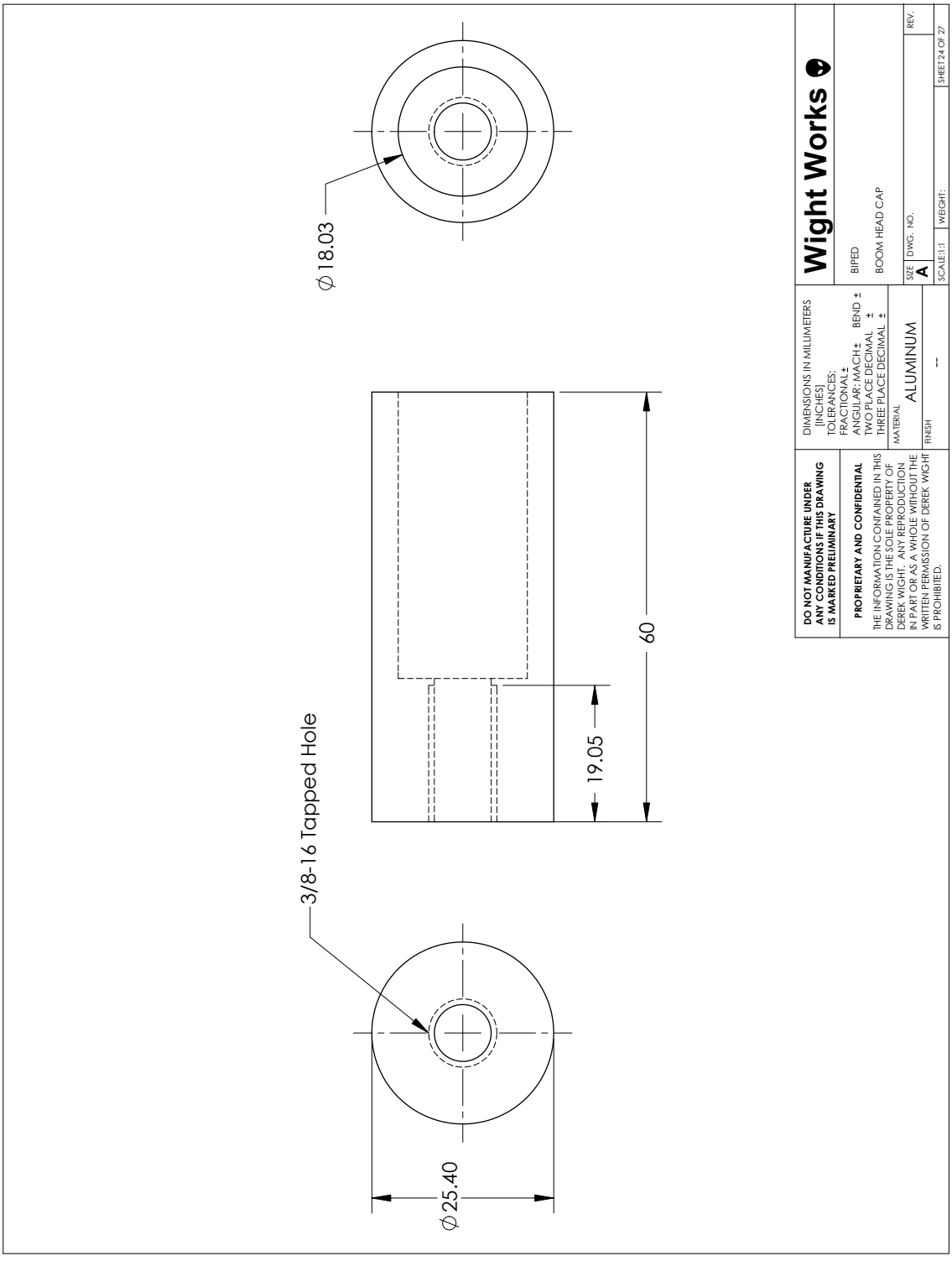

M8x1.25 Tapped Hole
∅22
#4-40 Tapped Hole
10
14
17
20
9.38
30
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
GEAR MOUNT
SIZE A
DWG. NO.
REV.
SCALE:2:1
WEIGHT:
SHEET 20 OF 27

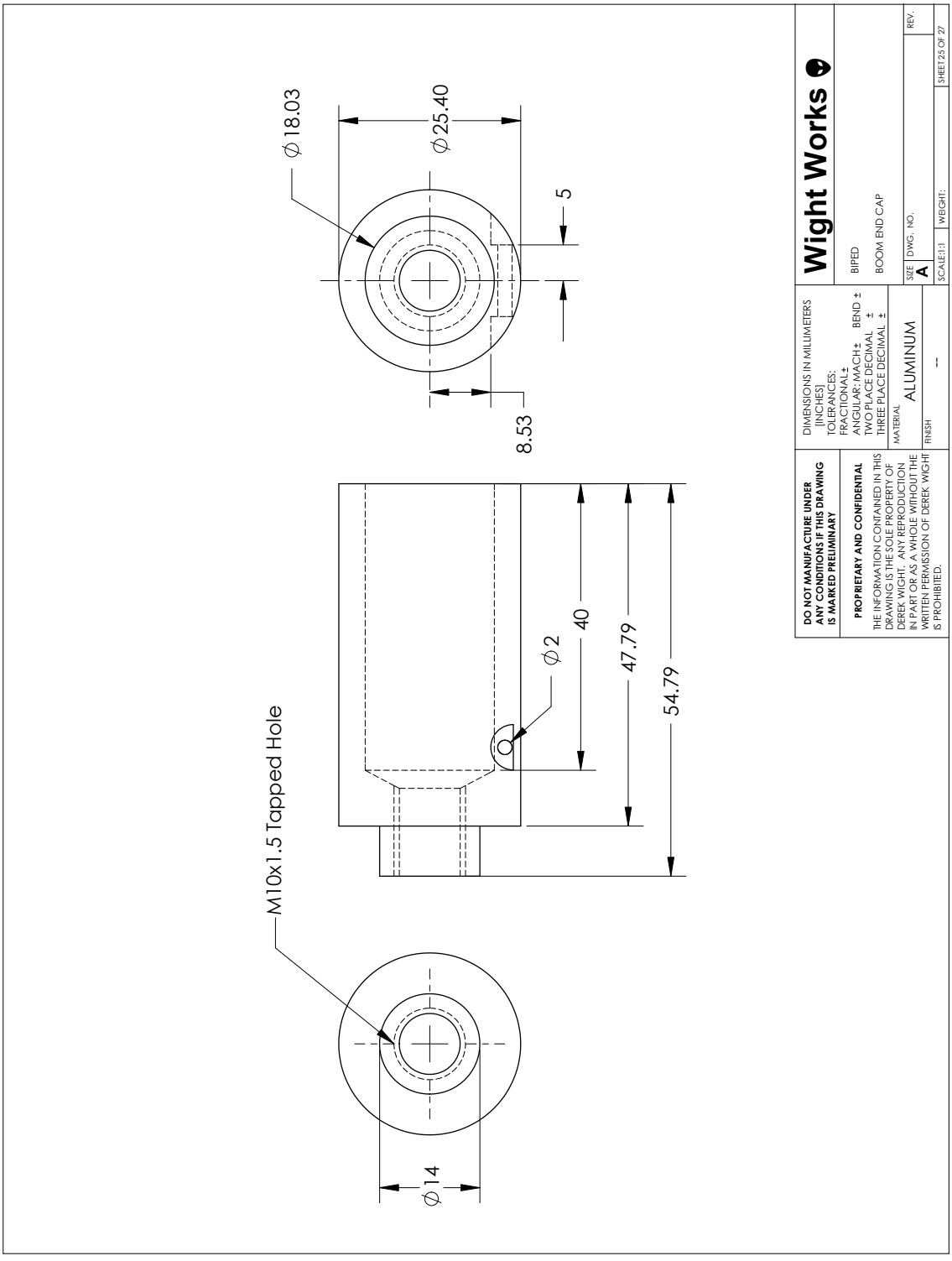

BOOM END CAP
CARBON FIBER TUBING
BOOM HEAD CAP
BOOM SHOULDER PLATE
BOOM END PLATE
OPTICAL ENCODER
BOOM GEAR BOX
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL --
FINISH --
Wight Works
BIPED
BOOM ASSEMBLY
SIZE A
DWG. NO.
REV.
SCALE:2:1
WEIGHT:
SHEET 21 OF 27

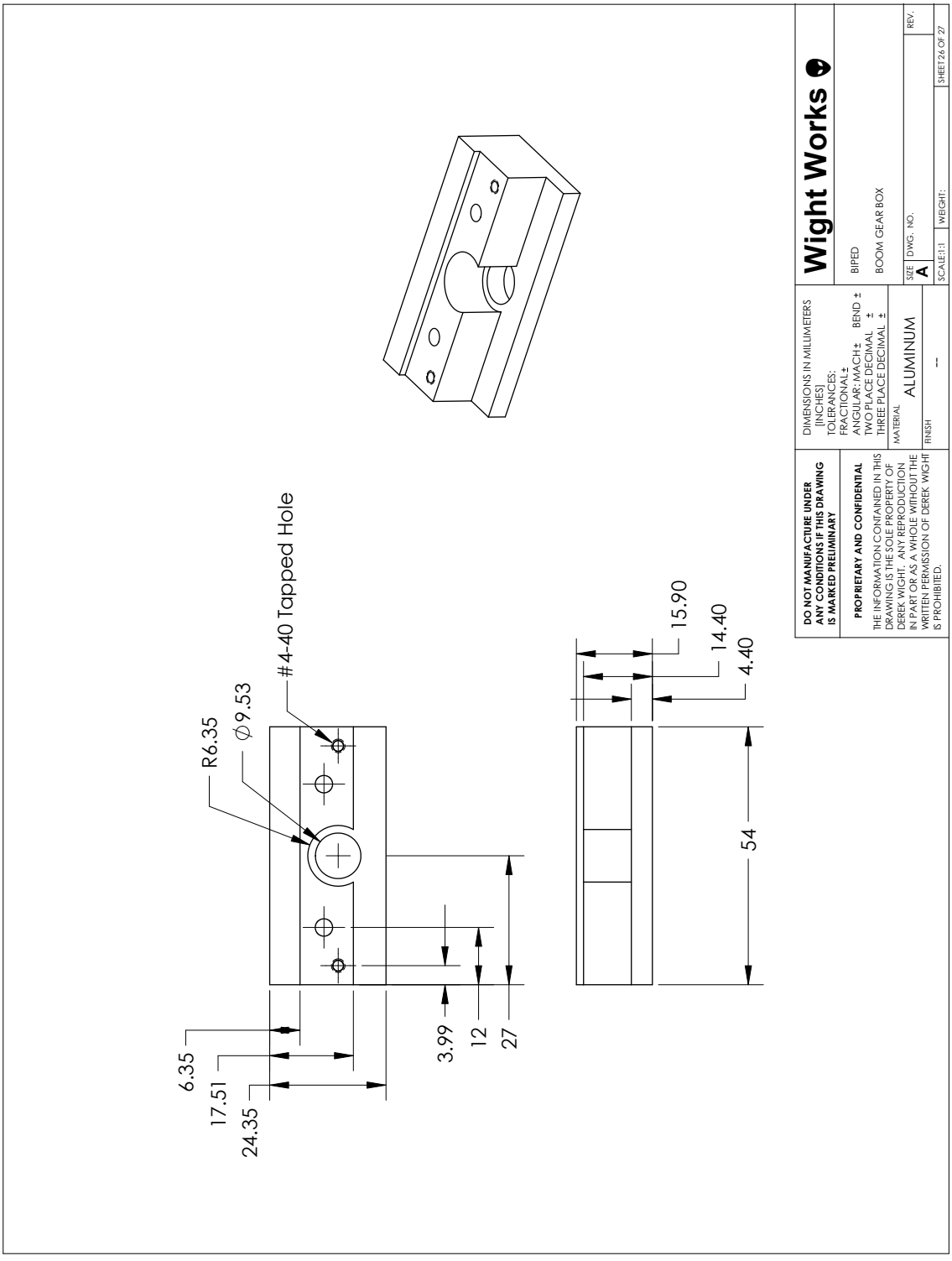

1/4-20 Tapped Hole
33.65
6.35
27.30
9.53
#4-40 Tapped Hole
46.03
83.01
120
60
36.99
Ø30
#6-32 Tapped Hole
20
40
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
BOOM SHOULDER PLATE
SIZE A
DWG. NO.
REV:
SCALE:1:1
WEIGHT:
SHEET 22 OF 27

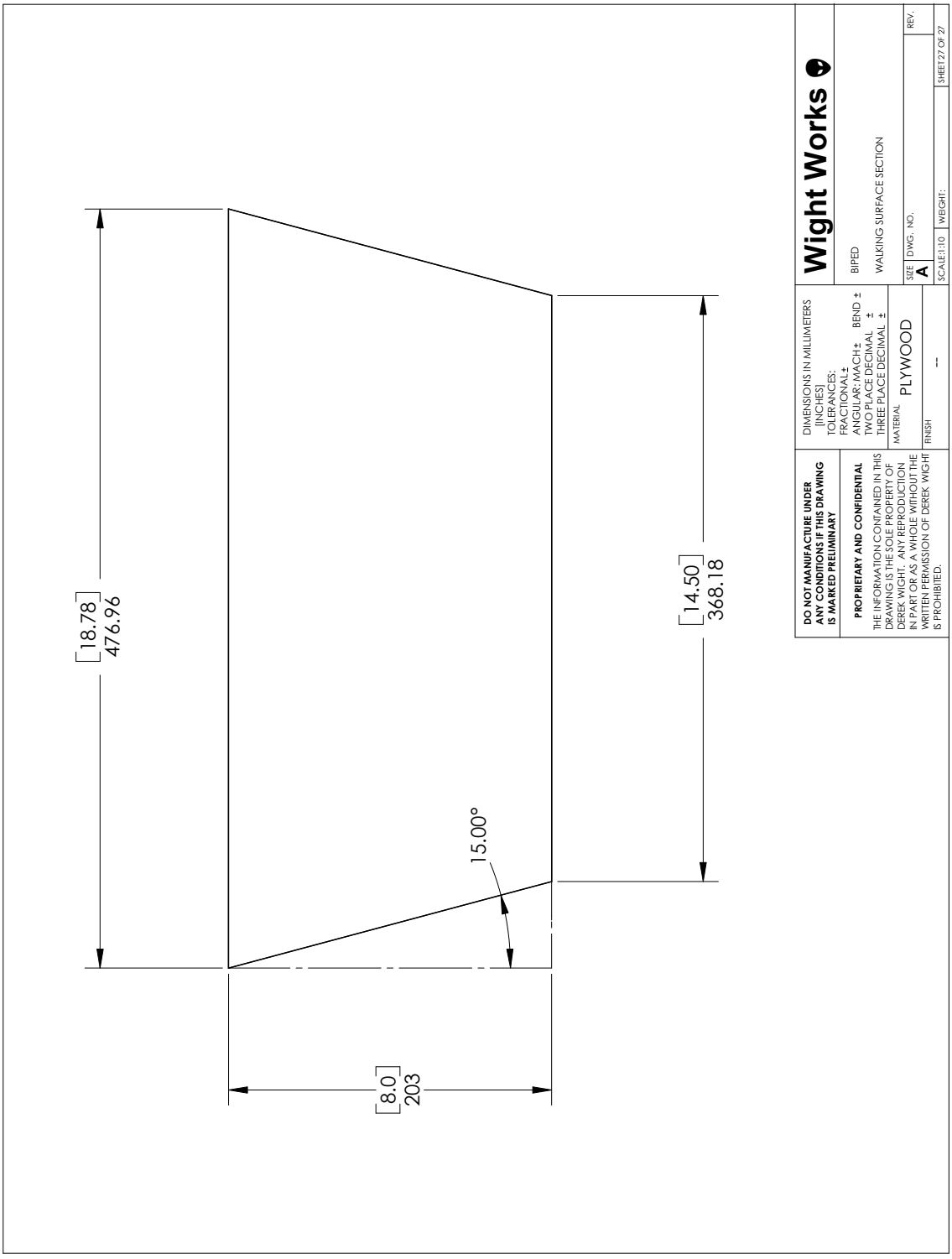

9.53
[.26]
ϕ6.53
40
20
ϕ9.53
6.35
50.50
101
6.35
6.35
6.35
Wight Works
DIMENSIONS IN MILLIMETERS
[INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ALUMINUM
FINISH

BIPED
BOOM END PLATE
SIZE
A
DWG. NO.
REV.
SCALE: 1:1
WEIGHT:
SHEET 23 OF 27
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.


Ø18.03
3/8-16 Tapped Hole
19.05
60
Ø25.40
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ALUMINUM
FINISH
--
Wight Works
BIPED
BOOM HEAD CAP
SIZE
A
DWG. NO.
REV:
SCALE:1:1
WEIGHT:
SHEET 24 OF 27

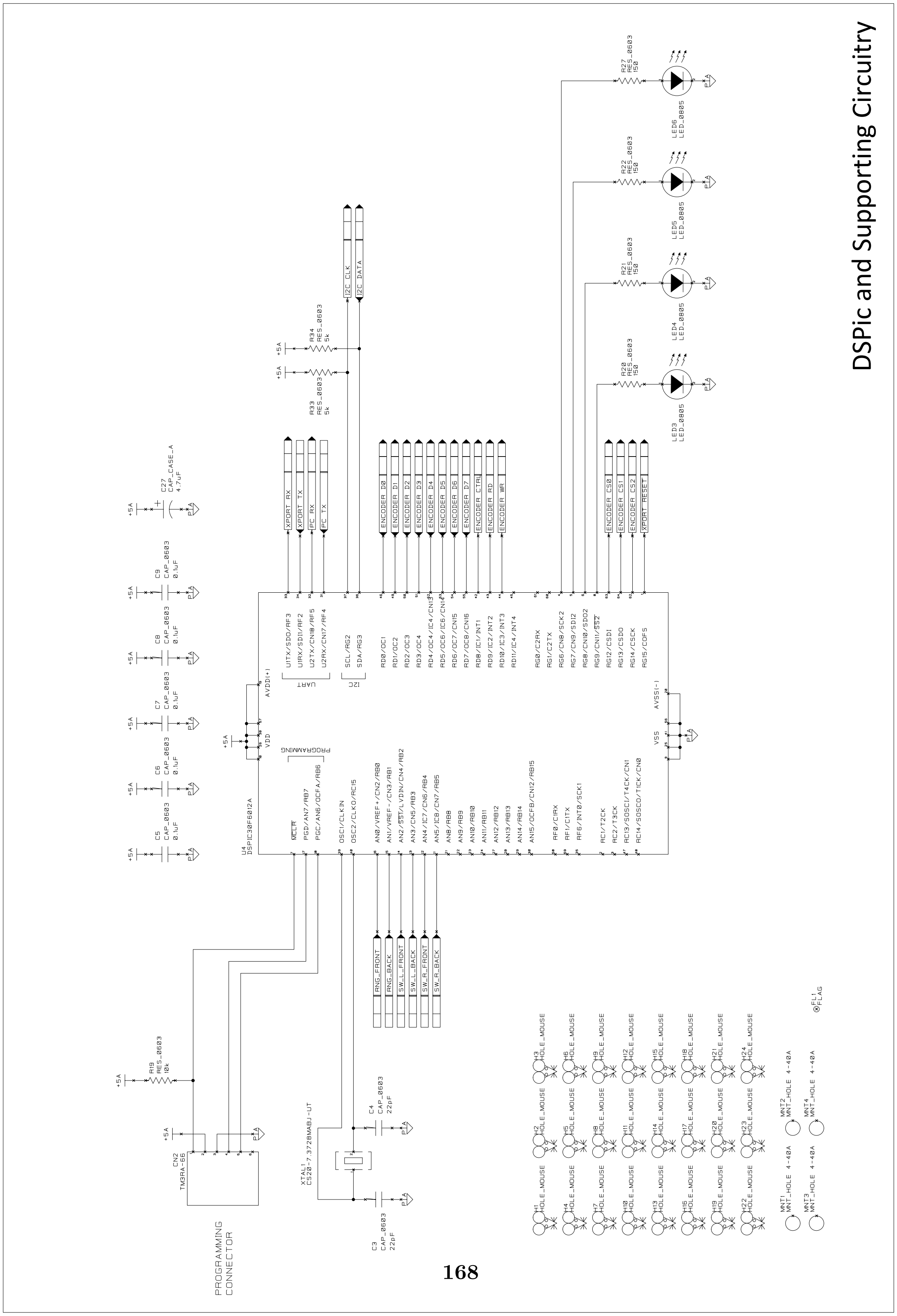

ϕ18.03
ϕ25.40
5
8.53
ϕ2
40
47.79
54.79
M10x1.5 Tapped Hole
ϕ14
Wight Works
DIMENSIONS IN MILLIMETERS
[INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL
ALUMINUM
FINISH
--
BIPED
BOOM END CAP
SIZE
A
DWG. NO.
REV.
SCALE: 1:1
WEIGHT:
SHEET 25 OF 27
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.

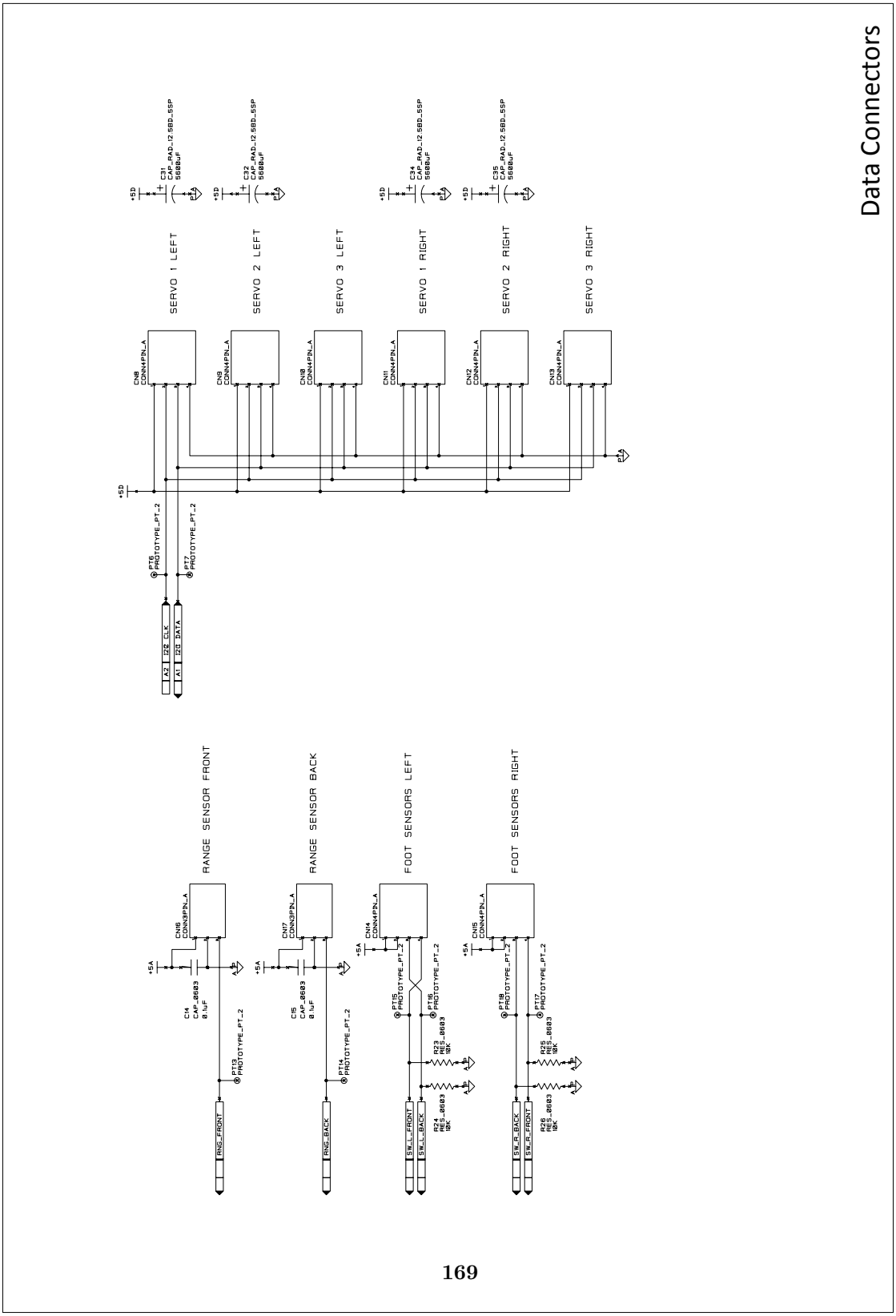

#4-40 Tapped Hole
R6.35
⌀9.53
6.35
17.51
24.35
3.99
12
27
15.90
14.40
4.40
54
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.
DIMENSIONS IN MILLIMETERS [INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL ALUMINUM
FINISH --
Wight Works
BIPED
BOOM GEAR BOX
SIZE A
DWG. NO.
REV:
SCALE:1:1
WEIGHT:
SHEET 26 OF 27

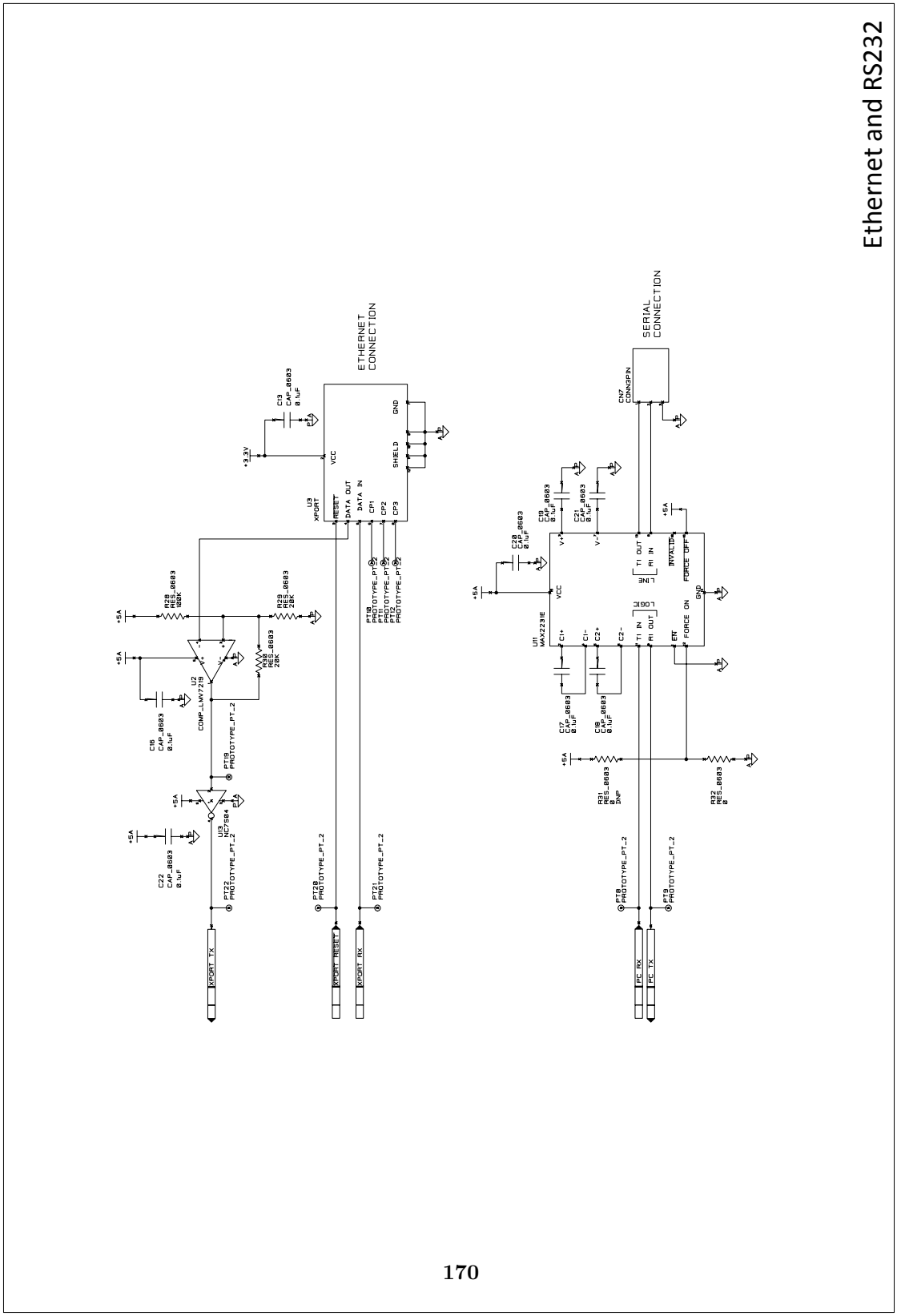

[18.78]
476.96
[14.50]
368.18
15.00°
[8.0]
203
Wight Works
DIMENSIONS IN MILLIMETERS
[INCHES]
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
MATERIAL PLYWOOD
FINISH --
BIPED
WALKING SURFACE SECTION
SIZE A
DWG. NO.
REV:
SCALE:1:10
WEIGHT:
SHEET 27 OF 27
DO NOT MANUFACTURE UNDER ANY CONDITIONS IF THIS DRAWING IS MARKED PRELIMINARY
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF DEREK WIGHT. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF DEREK WIGHT IS PROHIBITED.

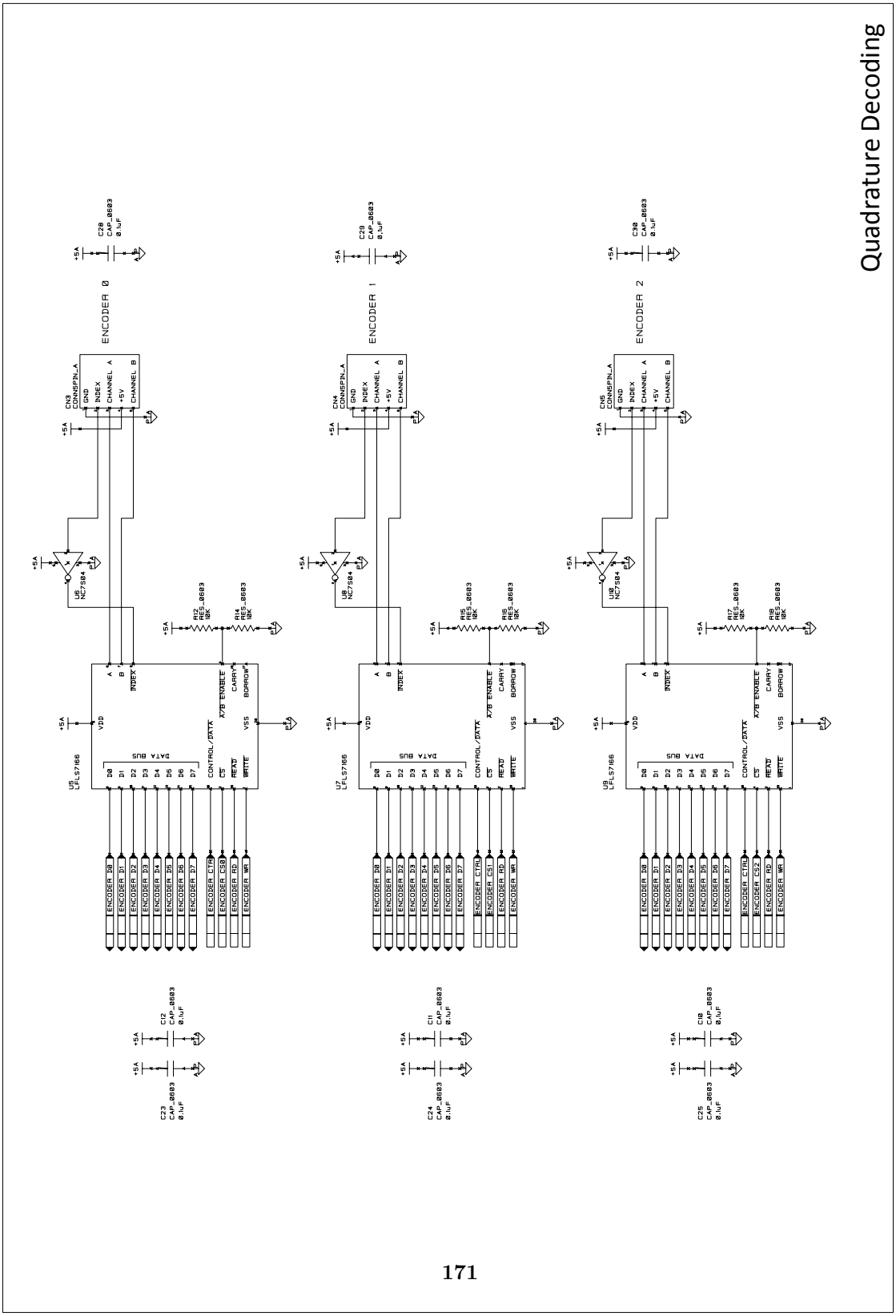
Appendix B

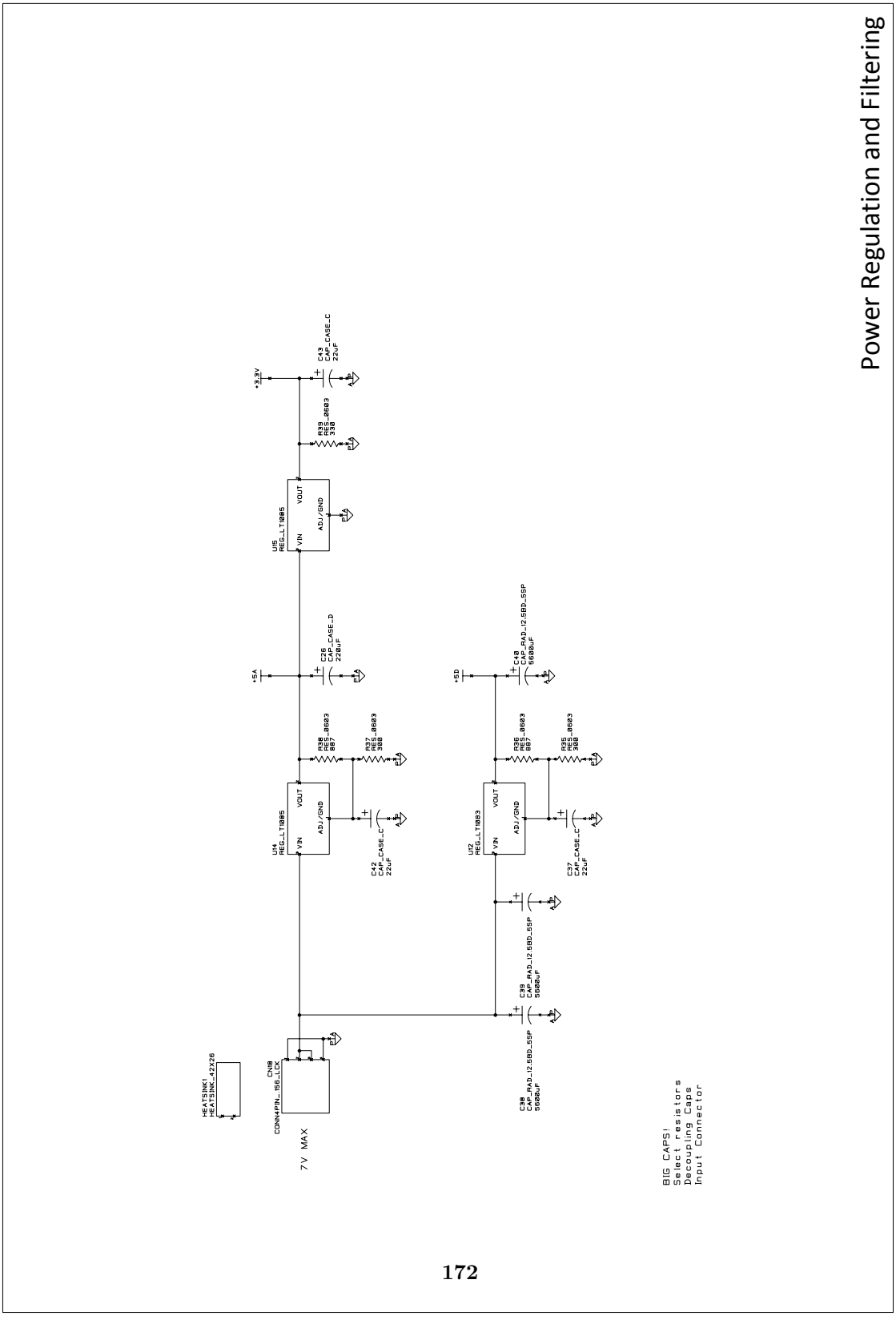
Electrical Schematics

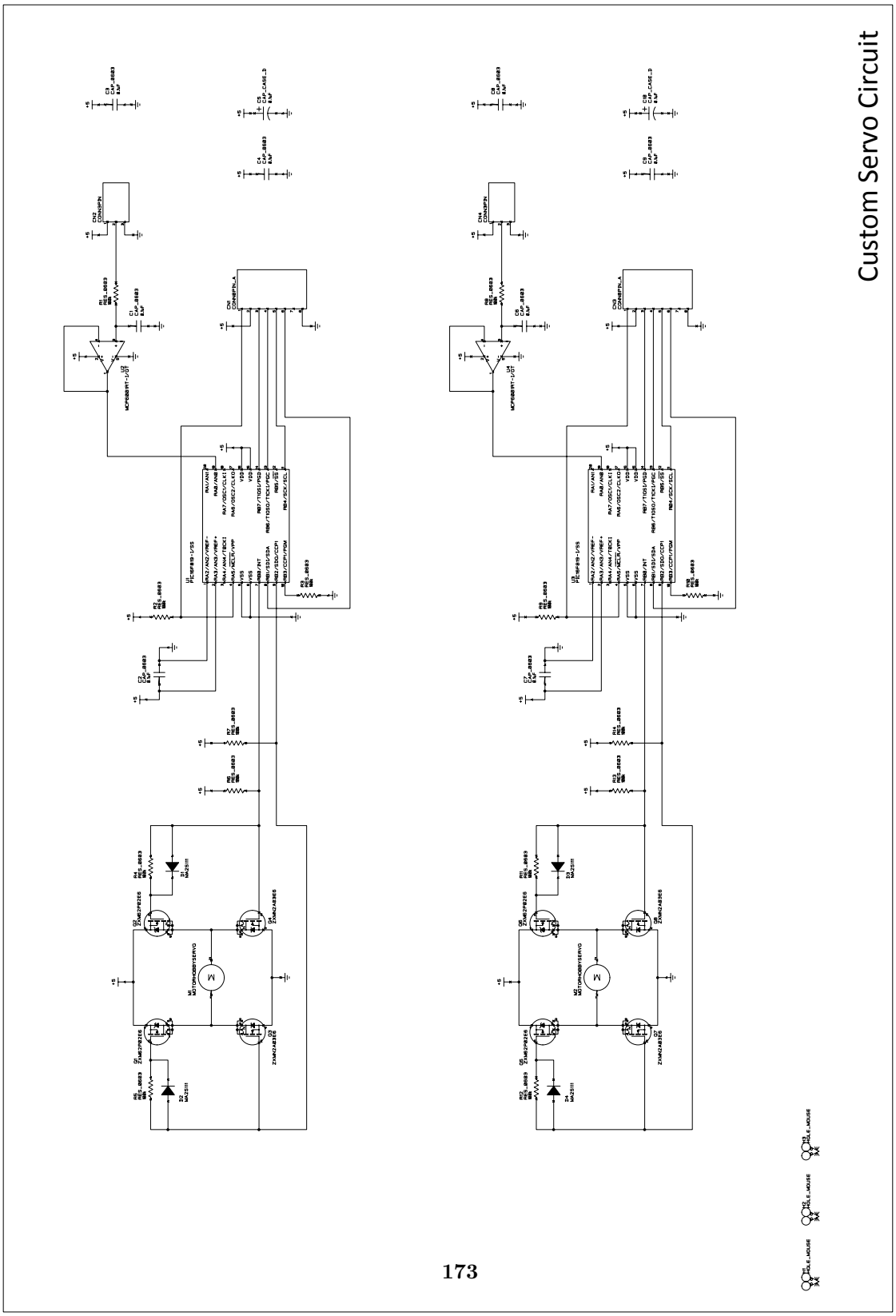

DSPic and Supporting Circuitry


Data Connectors
SERVO 1 LEFT
SERVO 2 LEFT
SERVO 3 LEFT
SERVO 1 RIGHT
SERVO 2 RIGHT
SERVO 3 RIGHT
RANGE SENSOR FRONT
RANGE SENSOR BACK
FOOT SENSORS LEFT
FOOT SENSORS RIGHT
+5D
+5A
I2C CLK
I2C DATA
RNG_FRONT
RNG_BACK
SW_L_FRONT
SW_L_BACK
SW_R_BACK
SW_R_FRONT


Ethernet and RS232
ETHERNET CONNECTION
SERIAL CONNECTION
XPORT_TX
XPORT_RESET
XPORT_RX
PC_RX
PC_TX


Quadrature Decoding
ENCODER 0
ENCODER 1
ENCODER 2
CONNSPN_A
GND
INDEX
CHANNEL A
+5V
CHANNEL B
NC7S04
LFLS7166
VDD
VSS
DATA BUS
CONTROL/DATA
X/Y ENABLE
CARRY
BORROW
CS
READ
WRITE
RES_0603
10K
CAP_0603
0.1uF


Power Regulation and Filtering
+3.3V
C19
CAP_CASE_C
22uF
R25
RES_0603
330
U6
REG_LT1085
VIN
VOUT
ADJ/GND
+5A
C26
CAP_CASE_D
220uF
R38
RES_0603
887
R37
RES_0603
383
U4
REG_LT1085
C42
CAP_CASE_C
22uF
+5D
C40
CAP_RAD_12.5BD_5SP
5600uF
R36
RES_0603
887
R35
RES_0603
388
U12
REG_LT1083
C37
CAP_CASE_C
22uF
C39
CAP_RAD_12.5BD_5SP
5600uF
C38
CAP_RAD_12.5BD_5SP
5600uF
HEATSINK1
HEATSNK_42X26
CN8
CONN4PIN_156_LCX
7V MAX
BIG CAPS!
Select resistors
Decoupling Caps
Input Connector


Custom Servo Circuit

Appendix C

Core Code

This code includes the implementation of all of the core algorithms performed on the main board for calculating the FPE and evaluating the state machine. The hardware-specific code for device initialization, and communications is not included.

The following is from the header file PhysicalDefs.h. It was used to define all the physical parameters used in the model.

```
1  #define TORSO_MASS 0.53153f              // kg
2  #define THIGH_MASS 0.098155f             // kg
3  #define SHANK_MASS 0.055305f             // kg
4  #define TOTAL_MASS (TORSO_MASS+2*THIGH_MASS+2*SHANK_MASS)
5
6  #define TORSO_MASS_RATIO (TORSO_MASS/TOTAL_MASS)
7  #define THIGH_MASS_RATIO (THIGH_MASS/TOTAL_MASS)
8  #define SHANK_MASS_RATIO (SHANK_MASS/TOTAL_MASS)
9
10 #define TORSO_INERTIA 0.00049709f        // kg * m^2
11 #define THIGH_INERTIA 0.00010959f        // kg * m^2
12 #define SHANK_INERTIA 0.00012121f        // kg * m^2
13
14 #define g 9.81f                          // m/s^2 (gravity)
15
16 #define ORIGIN_TORSO_COMx   0.012939f    // m
17 #define ORIGIN_TORSO_COMy   -0.017028f   // m
18 #define ORIGIN_HIPx         0.0071754f   // m
19 #define ORIGIN_HIPy         -0.0589f     // m
```

```
20 #define HIP_THIGH_COMx       0.0036006f  // m
21 #define HIP_THIGH_COMy       -0.08254f   // m
22 #define HIP_KNEEx            0.0025056f  // m
23 #define HIP_KNEEy            -0.11085f   // m
24
25 #define KNEE_SHANK_COMx      -0.0000915f // m
26 #define KNEE_SHANK_COMy      -0.08521f   // m
27 #define KNEE_ANKLEx          0.00770f    // m
28 #define KNEE_ANKLEy          -0.14484f   // m
29
30 #define ENCODERx_GEAR        4.83f       // Gear ratio 87/18
31 #define ENCODERy_GEAR        4.67f       // Gear ratio 84/18
32
33 #define ARM_LENGTH           0.788f      // m
34 #define CIRCUMFERENCE_CALC   1.2513e-4f  // 2*ARM_LENGTH*Pi/(8192*ENCODERx)
35 #define CIRCUMFERENCE_TOTAL  1.02506496  //(CIRCUMFERENCE_CALC*8192)
36                                          // This isn't the circumference
37                                          // since the previous result is
38                                          // now divided by the ENCODERx_GEAR
39                                          // ratio.
40
41 #define ENCODERy_OFFSET          -370    // Encoder calibration offsets
42 #define ENCODERtheta_OFFSET      -181    // Enc3 is gained by -2, so the
43                                          // error offset is divided by -2.
44
45 #define SERVO_RIGHT_HIP_OFFSET   400     // Servo calibration offsets
46 #define SERVO_RIGHT_KNEE_OFFSET  400
47 #define SERVO_LEFT_HIP_OFFSET    400
48 #define SERVO_LEFT_KNEE_OFFSET   400
49
50 #define SERVO_RIGHT_HIP_MAX      400     // Servo position limits.
51 #define SERVO_RIGHT_KNEE_MAX     0
52 #define SERVO_LEFT_HIP_MAX       400
53 #define SERVO_LEFT_KNEE_MAX      0
54
55 #define SERVO_RIGHT_HIP_MIN      -400
56 #define SERVO_RIGHT_KNEE_MIN     -400
57 #define SERVO_LEFT_HIP_MIN       -400
58 #define SERVO_LEFT_KNEE_MIN      -400
59
60 #define ENCODER_SCALE        0.1953125f  // Encoder to servo pot scaling
61 #define SERVO_TO_ENC_SCALE   5.12f       // Servo pot to encoder scaling
```

The following is from the header file CommDefines.h. It was used to define the communication packet. In addition to its namesake, tt was a used to efficiently pass a large number of variables to many functions using a pointer to the structure. This file also contains an the *UpdatePacket* structure which holds all of the control parameters. Again, this was used to pass a large number of variables between functions, as well as a means to update the control parameters remotely from the PC interface.

```
1  #define COMM_VERSION_CODE       1.6 // Used to ensure the PC interface
2                                      // is working on the same comm protocol.
3
4  #define ENCODER_CALIBRATION     22  // Operating mode
5  #define DEAD_MAN_HIT            23
6  #define START_DATA_STREAM       24
7  #define STOP_DATA_STREAM        25
8  #define UPLOAD_CONTROL_PARAMS   26
9  #define BURN_CONTROL_PARAMS     27
10 #define DOWNLOAD_CONTROL_PARAMS 28
11 #define WALKING_MODE            29
12 #define BALANCE_MODE            30
13 #define MANUAL_MODE             31
14 #define RIGOR_MORTIS_MODE       32
15
16 #define STATE_INACTIVE      0       // State machine states
17 #define STATE_STANDING      1
18 #define STATE_LEFT_PUSH     2
19 #define STATE_LEFT_LIFT     3
20 #define STATE_LEFT_SWING    4
21 #define STATE_LEFT_DROP     5
22 #define STATE_RIGHT_PUSH    6
23 #define STATE_RIGHT_LIFT    7
24 #define STATE_RIGHT_SWING   8
25 #define STATE_RIGHT_DROP    9
26 #define STATE_FALL          10
27
28 #define SAMPLE_TIME         0.010f
29 #define INV_SAMPLE_TIME     100.0f  // The inverse of SAMPLE_TIME to
30                                     // eliminate a floating point op.
31 #define VEL_SAMPLE_TIME     0.0767f // 2*pi/8192/0.01 - Used for
32                                     // the segment angular velocity
33 #define bitLeftContact      0x0001  // For FootContactSensors member
34 #define bitRightContact     0x0002  // For FootContactSensors member
```

```
35 // Note: Set code generation struct alignment to 2 bytes
36
37 struct DataStreamPacket {
38     WORD PacketHeader;
39     WORD TimeIndex;             // Packet time stamp
40     short EncoderXPos;
41     short EncoderYPos;
42     short EncoderThPos;
43     WORD EncoderValid;          // (bits 0-2, x y theta)
44     short ServoRightHipPos;
45     short ServoRightKneePos;
46     short ServoLeftHipPos;
47     short ServoLeftKneePos;
48     short ServoRightHipRef;
49     short ServoRightKneeRef;
50     short ServoLeftHipRef;
51     short ServoLeftKneeRef;
52     WORD ServoRightHipGain;
53     WORD ServoRightKneeGain;
54     WORD ServoLeftHipGain;
55     WORD ServoLeftKneeGain;
56     WORD FootContactSensors;    // (bits 1-0, RF LF)
57     float COMx;                 // Current COM
58     float COMy;                 // Current COM
59     float COMdx;                // Linear velocity of current COM
60     float COMdy;                // Linear velocity of current COM
61     float w;                    // Angular velocity of current COM
62     float Inertia;              // Inertia of current geometry
63     float FPEAngle;             // Angle phi from the FPE solver
64     float FPEGndPos;            // Projection relative to COM
65     float LeftFootPosx;
66     float LeftFootPosy;
67     float RightFootPosx;
68     float RightFootPosy;
69     float HipPosx;
70     float HipPosy;
71     WORD WalkingState;          // State machine state
72     WORD Mode;                  // (Walking, standing, manual, etc)
73     float Debug1;               // Debugging
74     float Debug2;               // Debugging
75     WORD CheckSum;              // 16 bit sum of packet members
76 };
```

```
77  struct UpdatePacket {
78      WORD  PacketHeader;
79
80      short Standing_R_Hip_Pos;
81      short Standing_R_Knee_Pos;
82      short Standing_L_Hip_Pos;
83      short Standing_L_Knee_Pos;
84      short Standing_All_Gain;
85      short Standing_FPE_Trans_Thres;
86
87      short Push_StandingLeg_Knee_Pos;
88      short Push_StandingLeg_Hip_Gain;
89      short Push_StandingLeg_Knee_Gain;
90      short Push_SwingLeg_Hip_Gain;
91      short Push_SwingLeg_Knee_Gain;
92      short Push_RateConstant;
93      short Push_FPE_Trans_Thres;
94      short Push_HipOffset;
95
96      short Lift_SwingLeg_Knee_Gain;
97      short Lift_SwingLeg_Hip_Gain;
98      short Lift_Height;
99      short Lift_Trans_Thres;
100
101     short Swing_StandingLeg_Hip_Gain;
102     short Swing_SwingLeg_Hip_Gain;
103     short Swing_FPE_Offset;
104     short Swing_Trans_Thres;
105
106     short Drop_SwingLeg_Knee_Gain;
107     short Drop_SwingLeg_Knee_Pos;
108     short Drop_StandingLeg_Knee_Gain;
109
110     WORD  CheckSum;
111 };
```

The following is the core code for the COM, inertia, and velocity calculations, the FPE solver, and the state machine.

```
1  #include <p30f6012A.h>
2  #include <math.h>
3  #include "../MainBoard/HardwareDefines.h"
4  #include "PhysicalDefs.h"
5  #include "CommDefines.h"
6  #include "SinTable.h"
7  #include "SwingKinematics.h"
8  #include "HeightTable.h"
9
10
11 ////////////////////////////////// Globals ///////////////////////////////////
12
13 int gLastServoRightHipPos;          // Servo position at t-2
14 int gLastServoRightKneePos;
15 int gLastServoLeftHipPos;
16 int gLastServoLeftKneePos;
17 int gLastTorsoPos;
18
19 int gLastServoRightHipPosTM1;       // Servo position at t-1
20 int gLastServoRightKneePosTM1;
21 int gLastServoLeftHipPosTM1;
22 int gLastServoLeftKneePosTM1;
23 int gLastTorsoPosTM1;
24
25 float gLastCOMx, gLastCOMy;         // COM position at t-2
26 float gLastCOMxTM1, gLastCOMyTM1;   // COM position at t-1
27
28 float gWorkingVarA;                 // State machine temporary variables
29 float gWorkingVarB;
30 float gWorkingVarC;
31
32 short gCurrentFPEOffset;            // Offset signs modified based on
33 short gCurrentHipOffset;            // direction of travel
34 short gCurrentLiftOffset;
```

```
35 ////////////////////////////// Functions //////////////////////////////
36
37 // Using the lookup table to return the sine function for a given
38 // encoder count
39
40 float GetSin(int Index) {
41
42     float ReturnValue;          // Note there is a bug in the MPLAB
43                                 // compiler which requires that arrays
44     while (Index < 0) {         // located on a separate memory page
45         Index = Index + 8192;   // must be passed to a local variable
46     }                           // before being returned or else the
47                                 // processor will throw an exception.
48     Index = Index % 8192;
49
50     if (Index < 2048) {
51         ReturnValue = SinTbl[Index];
52     } else if (Index < 4096) {
53         ReturnValue = SinTbl[4095 - Index];
54     } else if (Index < 6144) {
55         ReturnValue = SinTbl[Index - 4096];
56         ReturnValue = -ReturnValue;
57     } else {
58         ReturnValue = SinTbl[8191 - Index];
59         ReturnValue = -ReturnValue;
60     }
61
62     return ReturnValue;
63 }
64
65
66
67
68 // The cosine function adds a 90 deg phase shift and calls
69 // the sine function
70
71 float GetCos(int Index) {
72     return GetSin(Index + 2048);
73 }
```

```
74  // Get the value for the hip offset based on the knee angle.
75  int HipOffsetLookup(int KneeAngle) {
76
77      if (KneeAngle < -400) {
78          KneeAngle = -400;
79      }
80
81      if (KneeAngle > 0) {
82          KneeAngle = 0;
83      }
84
85      return HipOffsetTbl[-KneeAngle];
86  }
87
88  // Lookup the knee angle that would match the quantized leg
89  // length.
90  int KneeAngleLookup(int LegLength) {
91
92      if (LegLength < 1856) {
93          LegLength = 1856;
94      }
95
96      if (LegLength > 2559) {
97          LegLength = 2559;
98      }
99
100     return KneeAngleTbl[LegLength-1856];
101 }
102
103 // Lookup the leg length for the current knee angle.
104 float LegLengthLookup(int KneeAngle) {
105     KneeAngle = -KneeAngle;
106
107     if (KneeAngle < 0) {
108         KneeAngle = 0;
109     }
110
111     if (KneeAngle > 399) {
112         KneeAngle = 399;
113     }
114
115     return LegLengthTbl[KneeAngle];
116 }
```

```
117 // Calculate the COM, velocity of the COM, angular velocity, inertia, and
118 // feet positions
119
120 float CalcPhysicalParams(struct DataStreamPacket *Packet) {
121
122     float dx, dy, I, Diff;
123
124     float sinTorso, sinServoRightHip, sinServoRightKnee;
125     float sinServoLeftHip, sinServoLeftKnee;
126     float cosTorso, cosServoRightHip, cosServoRightKnee;
127     float cosServoLeftHip, cosServoLeftKnee;
128
129     float CurrentJointx, Originx, TorsoCOMx, LeftThighCOMx, LeftShankCOMx;
130     float RightThighCOMx, RightShankCOMx;
131     float CurrentJointy, Originy, TorsoCOMy, LeftThighCOMy, LeftShankCOMy;
132     float RightThighCOMy, RightShankCOMy;
133
134     int Servo1AbsPos, Servo2AbsPos, Servo4AbsPos, Servo5AbsPos;
135
136
137     // Calculate Absolute Angles
138     Servo1AbsPos = (int) (((float) (Packet->ServoRightHipPos))
139                           * SERVO_TO_ENC_SCALE + Packet->EncoderThPos);
140     Servo4AbsPos = (int) (((float) (Packet->ServoLeftHipPos))
141                           * SERVO_TO_ENC_SCALE + Packet->EncoderThPos);
142
143     Servo2AbsPos = (int) (((float) (Packet->ServoRightKneePos))
144                           * SERVO_TO_ENC_SCALE + Servo1AbsPos);
145     Servo5AbsPos = (int) (((float) (Packet->ServoLeftKneePos))
146                           * SERVO_TO_ENC_SCALE + Servo4AbsPos);
147
148
149     // Get sin and cos for each joint
150     sinTorso = GetSin(Packet->EncoderThPos);
151     cosTorso = GetCos(Packet->EncoderThPos);
152
153     sinServoRightHip = GetSin(Servo1AbsPos);
154     cosServoRightHip = GetCos(Servo1AbsPos);
155
156     sinServoRightKnee = GetSin(Servo2AbsPos);
157     cosServoRightKnee = GetCos(Servo2AbsPos);
158
159     sinServoLeftHip = GetSin(Servo4AbsPos);
160     cosServoLeftHip = GetCos(Servo4AbsPos);
161
162     sinServoLeftKnee = GetSin(Servo5AbsPos);
163     cosServoLeftKnee = GetCos(Servo5AbsPos);
```

```
164     // Calculate origin position
165
166     Originx = (float) CIRCUMFERENCE_CALC*(float) Packet->EncoderXPos;
167
168     if (Packet->EncoderYPos > -2000) {
169         Originy = HeightTbl[Packet->EncoderYPos+2000];  // Table ranges from
170     } else {                                            // -2000 to +1999.
171         Originy = HeightTbl[0];                         // Index starts at 0,
172     }                                                   // so offset by 2000.
173
174
175     // Calculate COM positions
176
177     CurrentJointx = Originx;
178     CurrentJointy = Originy;
179
180     TorsoCOMx = cosTorso*ORIGIN_TORSO_COMx - sinTorso*ORIGIN_TORSO_COMy
181                 + CurrentJointx;
182     TorsoCOMy = sinTorso*ORIGIN_TORSO_COMx + cosTorso*ORIGIN_TORSO_COMy
183                 + CurrentJointy;
184
185
186     CurrentJointx = cosTorso*ORIGIN_HIPx - sinTorso
187                     * ORIGIN_HIPy + CurrentJointx;
188     CurrentJointy = sinTorso*ORIGIN_HIPx + cosTorso
189                     * ORIGIN_HIPy + CurrentJointy;
190
191     // Store hip joint so we don't have to recalculate later
192     Packet->HipPosx = CurrentJointx;
193     Packet->HipPosy = CurrentJointy;
194
195     // And continue with the right leg.
196     RightThighCOMx = cosServoRightHip*HIP_THIGH_COMx - sinServoRightHip
197                      * HIP_THIGH_COMy + CurrentJointx;
198     RightThighCOMy = sinServoRightHip*HIP_THIGH_COMx + cosServoRightHip
199                      * HIP_THIGH_COMy + CurrentJointy;
200
201     CurrentJointx = cosServoRightHip*HIP_KNEEx - sinServoRightHip*HIP_KNEEy
202                     + CurrentJointx;
203     CurrentJointy = sinServoRightHip*HIP_KNEEx + cosServoRightHip*HIP_KNEEy
204                     + CurrentJointy;
205
206     RightShankCOMx = cosServoRightKnee*KNEE_SHANK_COMx - sinServoRightKnee
207                      * KNEE_SHANK_COMy + CurrentJointx;
208     RightShankCOMy = sinServoRightKnee*KNEE_SHANK_COMx + cosServoRightKnee
209                      * KNEE_SHANK_COMy + CurrentJointy;
210
211     Packet->RightFootPosx = cosServoRightKnee*KNEE_ANKLEx - sinServoRightKnee
212                             * KNEE_ANKLEy + CurrentJointx;
213     Packet->RightFootPosy = sinServoRightKnee*KNEE_ANKLEx + cosServoRightKnee
214                             * KNEE_ANKLEy + CurrentJointy;
```

```
215     // Now the left leg
216     CurrentJointx = Packet->HipPosx;
217     CurrentJointy = Packet->HipPosy;
218
219     LeftThighCOMx = cosServoLeftHip*HIP_THIGH_COMx - sinServoLeftHip
220                     * HIP_THIGH_COMy + CurrentJointx;
221     LeftThighCOMy = sinServoLeftHip*HIP_THIGH_COMx + cosServoLeftHip
222                     * HIP_THIGH_COMy + CurrentJointy;
223
224     CurrentJointx = cosServoLeftHip*HIP_KNEEx - sinServoLeftHip*HIP_KNEEy
225                     + CurrentJointx;
226     CurrentJointy = sinServoLeftHip*HIP_KNEEx + cosServoLeftHip*HIP_KNEEy
227                     + CurrentJointy;
228
229     LeftShankCOMx = cosServoLeftKnee*KNEE_SHANK_COMx - sinServoLeftKnee
230                     * KNEE_SHANK_COMy + CurrentJointx;
231     LeftShankCOMy = sinServoLeftKnee*KNEE_SHANK_COMx + cosServoLeftKnee
232                     * KNEE_SHANK_COMy + CurrentJointy;
233
234     Packet->LeftFootPosx = cosServoLeftKnee*KNEE_ANKLEx - sinServoLeftKnee
235                            * KNEE_ANKLEy + CurrentJointx;
236     Packet->LeftFootPosy = sinServoLeftKnee*KNEE_ANKLEx + cosServoLeftKnee
237                            * KNEE_ANKLEy + CurrentJointy;
238
239
240     // Finally, calculate COM
241     Packet->COMx = TorsoCOMx*TORSO_MASS_RATIO + RightThighCOMx*THIGH_MASS_RATIO
242             + RightShankCOMx*SHANK_MASS_RATIO + LeftThighCOMx*THIGH_MASS_RATIO
243             + LeftShankCOMx*SHANK_MASS_RATIO;
244
245     Packet->COMy = TorsoCOMy*TORSO_MASS_RATIO + RightThighCOMy*THIGH_MASS_RATIO
246             + RightShankCOMy*SHANK_MASS_RATIO + LeftThighCOMy*THIGH_MASS_RATIO
247             + LeftShankCOMy*SHANK_MASS_RATIO;
248
249
250     // Calculate the x distance relative to COM.  Y is absolute.
251     Packet->RightFootPosx = Packet->RightFootPosx - Packet->COMx;
252     Packet->LeftFootPosx = Packet->LeftFootPosx - Packet->COMx;
253     Packet->HipPosx = Packet->HipPosx - Packet->COMx;
254
255
256     // Now calculate inertia and average angular velocity
257
258     Diff = (float) (Packet->EncoderThPos - gLastTorsoPos);
259     if (Diff > 2000) {
260         gLastTorsoPos = gLastTorsoPos + 8192;
261     } else if (Diff < -2000) {
262         gLastTorsoPos = gLastTorsoPos - 8192;
263     }
```

```
264     dx = Packet->COMx - TorsoCOMx;
265     dy = Packet->COMy - TorsoCOMy;
266     I = TORSO_INERTIA + TORSO_MASS*(dx*dx + dy*dy);
267     Packet->w = I*(Packet->EncoderThPos - gLastTorsoPos)*VEL_SAMPLE_TIME*0.5f;
268     Packet->Inertia = I;
269
270
271     dx = Packet->COMx - RightThighCOMx;
272     dy = Packet->COMy - RightThighCOMy;
273     I = THIGH_INERTIA + THIGH_MASS*(dx*dx + dy*dy);
274     Packet->w = Packet->w + I*(Packet->ServoRightHipPos - gLastServoRightHipPos)
275                 * VEL_SAMPLE_TIME*0.5f;
276     Packet->Inertia = I + Packet->Inertia;
277
278     dx = Packet->COMx - LeftThighCOMx;
279     dy = Packet->COMy - LeftThighCOMy;
280     I = THIGH_INERTIA + THIGH_MASS*(dx*dx + dy*dy);
281     Packet->w = Packet->w + I*(Packet->ServoLeftHipPos - gLastServoLeftHipPos)
282                 * VEL_SAMPLE_TIME*0.5f;
283     Packet->Inertia = I + Packet->Inertia;
284
285
286     dx = Packet->COMx - RightShankCOMx;
287     dy = Packet->COMy - RightShankCOMy;
288     I = SHANK_INERTIA + SHANK_MASS*(dx*dx + dy*dy);
289     Packet->w = Packet->w + I*(Packet->ServoRightKneePos - gLastServoRightKneePos)
290                 * VEL_SAMPLE_TIME*0.5f;
291     Packet->Inertia = I + Packet->Inertia;
292
293     dx = Packet->COMx - LeftShankCOMx;
294     dy = Packet->COMy - LeftShankCOMy;
295     I = SHANK_INERTIA + SHANK_MASS*(dx*dx + dy*dy);
296     Packet->w = Packet->w + I*(Packet->ServoLeftKneePos - gLastServoLeftKneePos)
297                 * VEL_SAMPLE_TIME*0.5f;
298     Packet->Inertia = I + Packet->Inertia;
299
300     Packet->w = Packet->w/Packet->Inertia;
301
302
303
304     // Calculate COM Derivatives
305     Diff = Packet->COMx - gLastCOMx;
306
307     if (Diff > 0.5f) {
308         gLastCOMx = (float) (gLastCOMx + CIRCUMFERENCE_TOTAL);
309     } else if (Diff < -0.5f) {
310         gLastCOMx = (float) (gLastCOMx - CIRCUMFERENCE_TOTAL);
311     }
312
313     Packet->COMdx = (Packet->COMx - gLastCOMx)*INV_SAMPLE_TIME*0.5f;
314     Packet->COMdy = (Packet->COMy - gLastCOMy)*INV_SAMPLE_TIME*0.5f;
```

```
315     // Save current angles for next time
316     gLastServoRightHipPos     = gLastServoRightHipPosTM1;
317     gLastServoRightKneePos    = gLastServoRightKneePosTM1;
318     gLastServoLeftHipPos      = gLastServoLeftHipPosTM1;
319     gLastServoLeftKneePos     = gLastServoLeftKneePosTM1;
320     gLastTorsoPos             = gLastTorsoPosTM1;
321     gLastCOMx                 = gLastCOMxTM1;
322     gLastCOMy                 = gLastCOMyTM1;
323
324     gLastServoRightHipPosTM1  = Packet->ServoRightHipPos;
325     gLastServoRightKneePosTM1 = Packet->ServoRightKneePos;
326     gLastServoLeftHipPosTM1   = Packet->ServoLeftHipPos;
327     gLastServoLeftKneePosTM1  = Packet->ServoLeftKneePos;
328     gLastTorsoPosTM1          = Packet->EncoderThPos;
329     gLastCOMxTM1              = Packet->COMx;
330     gLastCOMyTM1              = Packet->COMy;
331
332 }
333
334
335 // Now that all the physical parameters have been determined,
336 // solve the nonlinear FPE equation.
337
338 void SolveFPE(struct DataStreamPacket *Packet) {
339
340     float A, B, C, D;
341     float HighPos, LowPos, HighValue, LowValue;
342     float CurrentPos;
343     float s1, c1;
344
345     int StepCount = 0;
346     const int MaxSteps = 20;
347     const float Tolerance = 1e-7f;
348     float CurrentValue = 1.0f;      // must be greater than Tolerance
349
350     if (Packet->COMdx > 0.0) {
351         LowPos = 0.0;
352         HighPos = 1.570f;
353     } else {
354         LowPos = 0.0;
355         HighPos = -1.570f;
356     }
357
358     A = TOTAL_MASS*Packet->COMy;
359     B = A*Packet->COMdx + Packet->Inertia*Packet->w;
360     C = 2.0f*A*g;
361     D = C*Packet->Inertia;
362     C = A*C*Packet->COMy;
363     A = A*Packet->COMdy;
```

```
364     s1 = (float) sin(LowPos);
365     c1 = (float) cos(LowPos);
366 
367     LowValue = A*s1 + B*c1;
368     LowValue = c1*LowValue*LowValue +(C+D*c1*c1)*(c1-1.0f);
369 
370     s1 = (float) sin(HighPos);
371     c1 = (float) cos(HighPos);
372 
373     HighValue = A*s1 + B*c1;
374     HighValue = c1*HighValue*HighValue +(C+D*c1*c1)*(c1-1.0f);
375 
376     while ((fabs(CurrentValue) > Tolerance) & (StepCount < MaxSteps)) {
377         CurrentPos = (HighPos + LowPos)*0.5f;
378 
379         s1 = (float) sin(CurrentPos);
380         c1 = (float) cos(CurrentPos);
381 
382         CurrentValue = A*s1 + B*c1;
383         CurrentValue = c1*CurrentValue*CurrentValue +(C+D*c1*c1)*(c1-1.0f);
384 
385         if (CurrentValue < 0.0) {
386             HighValue = CurrentValue;
387             HighPos = CurrentPos;
388         } else {
389             LowValue = CurrentValue;
390             LowPos = CurrentPos;
391         }
392 
393         StepCount++;
394 
395     }
396 
397     Packet->FPEAngle = CurrentPos;
398     Packet->FPEGndPos = Packet->COMy*(float) tan(CurrentPos);
399 }
```

```
400 // Now that the FPE has been evaluated, calculate the
401 // appropriate references for the current state.
402
403 void EvaluateStateMachine(struct DataStreamPacket *Packet,
404 struct UpdatePacket *Param) {
405
406     float a, b;
407     const short DisturbanceFPEOffset = 20;
408     const short DisturbanceLiftOffset = 50;
409     const float StepBehindThreshold = 10;
410
411     short Encoder3Scaled;
412     Encoder3Scaled = (short) ((float) Packet->EncoderThPos*ENCODER_SCALE);
413
414
415     if (Packet->WalkingState == STATE_STANDING) {
416
417         Packet->ServoRightHipGain = Param->Standing_All_Gain;
418         Packet->ServoRightKneeGain = Param->Standing_All_Gain;
419         Packet->ServoLeftHipGain = Param->Standing_All_Gain;
420         Packet->ServoLeftKneeGain = Param->Standing_All_Gain;
421
422         // Just hold all previous references.
423
424         if (Packet->Mode != RIGOR_MORTIS_MODE) {
425             if (Packet->RightFootPosx > Packet->LeftFootPosx) {
426                 if (Packet->Mode == WALKING_MODE)  {
427
428                     gCurrentFPEOffset = Param->Swing_FPE_Offset;
429                     gCurrentHipOffset = Param->Push_HipOffset;
430                     gCurrentLiftOffset = 0;
431
432                     // Right leg leading, so push with left
433                     Packet->WalkingState = STATE_LEFT_PUSH;
434
435                     gWorkingVarA = Packet->ServoLeftKneeRef;
436                     gWorkingVarB = Packet->RightFootPosx
437                                    - Packet->LeftFootPosx;
438
439                 } else if ( (Packet->FPEGndPos - Packet->RightFootPosx) >
440                     (((float) Param->Standing_FPE_Trans_Thres)*0.001f)) {
441
442                     gCurrentFPEOffset = Param->Swing_FPE_Offset
443                                         + DisturbanceFPEOffset;
444                     gCurrentHipOffset = Param->Push_HipOffset;
445                     gCurrentLiftOffset = 0;
446
447                     Packet->WalkingState = STATE_LEFT_LIFT;
```

```
448                 } else if ( (Packet->LeftFootPosx - Packet->FPEGndPos) >
449                     (((float) StepBehindThreshold)*0.001f) ) {
450
451                     gCurrentFPEOffset = -Param->Swing_FPE_Offset
452                                         -3*DisturbanceFPEOffset;
453                     gCurrentHipOffset = -Param->Push_HipOffset;
454                     gCurrentLiftOffset = DisturbanceLiftOffset;
455
456                     Packet->WalkingState = STATE_RIGHT_LIFT;
457                 }
458             } else {
459                 if (Packet->Mode == WALKING_MODE) {
460
461                     gCurrentFPEOffset = Param->Swing_FPE_Offset;
462                     gCurrentHipOffset = Param->Push_HipOffset;
463                     gCurrentLiftOffset = 0;
464
465                     // Left leg leading, so push with right
466                     Packet->WalkingState = STATE_RIGHT_PUSH;
467
468                     gWorkingVarA = Packet->ServoRightKneeRef;
469                     gWorkingVarB = Packet->LeftFootPosx
470                                    - Packet->RightFootPosx;
471
472                 } else if ( (Packet->FPEGndPos - Packet->LeftFootPosx) >
473                     (((float) Param->Standing_FPE_Trans_Thres)*0.001f) ) {
474
475                     gCurrentFPEOffset = Param->Swing_FPE_Offset
476                                         + DisturbanceFPEOffset;
477                     gCurrentHipOffset = Param->Push_HipOffset;
478                     gCurrentLiftOffset = 0;
479
480                     Packet->WalkingState = STATE_RIGHT_LIFT;
481                 } else if ( (Packet->RightFootPosx - Packet->FPEGndPos) >
482                     (((float) StepBehindThreshold)*0.001f)) {
483
484                     gCurrentFPEOffset = -Param->Swing_FPE_Offset
485                                         -3*DisturbanceFPEOffset;
486                     gCurrentHipOffset = -Param->Push_HipOffset;
487                     gCurrentLiftOffset = DisturbanceLiftOffset;
488
489                     Packet->WalkingState = STATE_LEFT_LIFT;
490                 }
491             }
492         }
```

```
493     } else if (Packet->WalkingState == STATE_LEFT_PUSH) {
494
495         if ( (Packet->FPEGndPos - Packet->RightFootPosx) >
496             (float) Param->Push_FPE_Trans_Thres*0.001f ) {
497
498             // FPE in front of standing foot so transition to lift,
499             // but perform lift calculations immediately
500
501             Packet->WalkingState = STATE_LEFT_LIFT;
502
503             gCurrentFPEOffset = Param->Swing_FPE_Offset;
504             gCurrentHipOffset = Param->Push_HipOffset;
505             gCurrentLiftOffset = 0;
506
507             if (Packet->FootContactSensors & bitRightContact) {
508                 Packet->ServoRightHipRef = Packet->ServoRightHipPos
509                     + Encoder3Scaled + gCurrentHipOffset;
510             }
511
512             Packet->ServoLeftHipRef  = Packet->ServoLeftHipPos;
513             Packet->ServoLeftHipGain = Param->Lift_SwingLeg_Hip_Gain;
514
515             a = (Packet->HipPosy - ((float) Param->Lift_Height)
516                 *0.001f)*10000.0f;
517             b = (float) (Packet->ServoLeftHipPos + Encoder3Scaled
518                 - HipOffsetLookup(Packet->ServoLeftKneePos))*5.12f;
519             Packet->ServoLeftKneeRef = KneeAngleLookup( (int)
520                 (a/GetCos((int) b)));
521             Packet->ServoLeftKneeGain = Param->Lift_SwingLeg_Knee_Gain;
522
523         } else if ( (Packet->LeftFootPosx - Packet->FPEGndPos) >
524             (((float) StepBehindThreshold)*0.001f) ) {
525
526             // FPE behind push foot so the robot is falling backwards.
527             // Lift standing foot.
528
529             Packet->WalkingState = STATE_RIGHT_LIFT;
530
531             gCurrentFPEOffset = -Param->Swing_FPE_Offset
532                                 -3*DisturbanceFPEOffset;
533             gCurrentHipOffset = -Param->Push_HipOffset;
534             gCurrentLiftOffset = DisturbanceLiftOffset;
535
536             // Standing hip regulating torso vertical
537             if (Packet->FootContactSensors & bitLeftContact) {
538                 Packet->ServoLeftHipRef = Packet->ServoLeftHipPos
539                     + Encoder3Scaled + gCurrentHipOffset;
540             }
541             Packet->ServoLeftHipGain = Param->Push_StandingLeg_Hip_Gain;
```

```
542             Packet->ServoLeftKneeRef  = Param->Push_StandingLeg_Knee_Pos;
543             Packet->ServoLeftKneeGain = Param->Push_StandingLeg_Knee_Gain;
544
545             Packet->ServoRightHipRef  = Packet->ServoRightHipPos;
546             Packet->ServoRightHipGain = Param->Lift_SwingLeg_Hip_Gain;
547
548             a = (Packet->HipPosy - ((float) Param->Lift_Height)
549                 *0.001f)*10000.0f;
550             b = (float) (Packet->ServoRightHipPos + Encoder3Scaled
551                 - HipOffsetLookup(Packet->ServoRightKneePos))*5.12f;
552             Packet->ServoRightKneeRef = KneeAngleLookup( (int)
553                 (a/GetCos((int) b)));
554             Packet->ServoRightKneeGain = Param->Lift_SwingLeg_Knee_Gain;
555
556         } else {
557             // else continue with push kinematics
558
559             if (Packet->FootContactSensors & bitRightContact) {
560                 Packet->ServoRightHipRef = Packet->ServoRightHipPos
561                     + Encoder3Scaled + gCurrentHipOffset;
562             }
563             Packet->ServoRightHipGain = Param->Push_StandingLeg_Hip_Gain;
564
565             Packet->ServoRightKneeRef  = Param->Push_StandingLeg_Knee_Pos;
566             Packet->ServoRightKneeGain = Param->Push_StandingLeg_Knee_Gain;
567
568             Packet->ServoLeftHipRef  = Packet->ServoLeftHipPos;
569             Packet->ServoLeftHipGain = Param->Push_SwingLeg_Hip_Gain;
570
571             Packet->ServoLeftKneeGain = Param->Push_SwingLeg_Knee_Gain;
572
573             // Add length
574             gWorkingVarA = gWorkingVarA + Param->Push_RateConstant*0.1f;
575
576             if (gWorkingVarA > 0.0) {
577                 gWorkingVarA = 0.0;
578             }
579
580             Packet->ServoLeftKneeRef = (short) gWorkingVarA;
581
582             a = LegLengthLookup(Packet->ServoLeftKneeRef);
583             b = LegLengthLookup(Packet->ServoRightKneeRef);
584
585             gWorkingVarC = acos( (gWorkingVarB*gWorkingVarB - a*a - b*b)
586                 /(-2*a*b))*254.65;
587             Packet->ServoLeftHipRef = Packet->ServoRightHipRef
588                 - gWorkingVarC - HipOffsetLookup(Packet->ServoRightKneeRef)
589                 + HipOffsetLookup(Packet->ServoLeftKneeRef);
590         }
```

```
591     } else if (Packet->WalkingState == STATE_LEFT_LIFT) {
592         // Standing hip regulating torso vertical
593         if (Packet->FootContactSensors & bitRightContact) {
594             Packet->ServoRightHipRef = Packet->ServoRightHipPos
595                 + Encoder3Scaled + gCurrentHipOffset;
596         }
597 
598         Packet->ServoLeftHipRef  = Packet->ServoLeftHipPos;
599         Packet->ServoLeftHipGain = Param->Lift_SwingLeg_Hip_Gain;
600 
601         a = (Packet->HipPosy - ((float) (Param->Lift_Height
602             + gCurrentLiftOffset))*0.001f)*10000.0f;
603         b = (float) (Packet->ServoLeftHipPos + Encoder3Scaled
604             - HipOffsetLookup(Packet->ServoLeftKneePos))*5.12f;
605 
606         Packet->ServoLeftKneeRef = KneeAngleLookup( (int)(a/GetCos((int) b)));
607         Packet->ServoLeftKneeGain = Param->Lift_SwingLeg_Knee_Gain;
608 
609         if (Packet->LeftFootPosy > ((float)(Param->Lift_Height +
610             gCurrentLiftOffset))*((float) Param->Lift_Trans_Thres)*0.00001f){
611 
612             Packet->WalkingState = STATE_LEFT_SWING;
613         }
614 
615     } else if (Packet->WalkingState == STATE_LEFT_SWING) {
616 
617         if (Packet->FootContactSensors & bitRightContact) {
618             Packet->ServoRightHipRef = Packet->ServoRightHipPos
619                 + Encoder3Scaled + gCurrentHipOffset;
620         }
621         Packet->ServoRightHipGain = Param->Swing_StandingLeg_Hip_Gain;
622 
623         a = (float) (atan( (Packet->FPEGndPos -Packet->HipPosx + ((float)
624             gCurrentFPEOffset)*0.001f)/Packet->HipPosy)*800.0f/3.14159f);
625 
626         Packet->ServoLeftHipRef = HipOffsetLookup(Packet->ServoLeftKneePos)
627             - Encoder3Scaled + (short) a;
628         Packet->ServoLeftHipGain = Param->Swing_SwingLeg_Hip_Gain;
629 
630         a = (Packet->HipPosy - ((float)(Param->Lift_Height + gCurrentLiftOffset))
631             *0.001f)*10000.0f;
632         b = (float) (Packet->ServoLeftHipPos + Encoder3Scaled
633             - HipOffsetLookup(Packet->ServoLeftKneePos))*5.12f;
634         Packet->ServoLeftKneeRef = KneeAngleLookup( (int) (a/GetCos((int) b)));
635 
636         if (abs(Packet->ServoLeftHipPos - Packet->ServoLeftHipRef) <
637             Param->Swing_Trans_Thres) {
638 
639             // Hip (being the slowest servo to move) is sufficiently close
640             Packet->WalkingState = STATE_LEFT_DROP;
641         }
```

```
642     } else if (Packet->WalkingState == STATE_LEFT_DROP) {
643
644         // Standing hip regulating torso vertical
645         if (Packet->FootContactSensors & bitRightContact) {
646             Packet->ServoRightHipRef = Packet->ServoRightHipPos
647                 + Encoder3Scaled + gCurrentHipOffset;
648         }
649
650         a = (float) (atan( (Packet->FPEGndPos -Packet->HipPosx + ((float)
651             gCurrentFPEOffset)*0.001f)/Packet->HipPosy)*800.0f/3.14159f);
652
653         Packet->ServoLeftHipRef = HipOffsetLookup(Packet->ServoLeftKneePos)
654             - Encoder3Scaled + (short) a;
655
656         Packet->ServoLeftKneeRef = Param->Drop_SwingLeg_Knee_Pos;
657         Packet->ServoLeftKneeGain = Param->Drop_SwingLeg_Knee_Gain;
658
659         if (Packet->FootContactSensors & bitLeftContact) {
660
661
662             if (Packet->Mode == WALKING_MODE) {
663                 Packet->WalkingState = STATE_RIGHT_PUSH;
664
665                 Packet->ServoLeftHipRef = Packet->ServoLeftHipPos;
666                 Packet->ServoRightKneeRef = Packet->ServoRightKneePos;
667                 Packet->ServoRightKneeGain = Param->Drop_StandingLeg_Knee_Gain;
668
669                 gWorkingVarA = Packet->ServoRightKneeRef;
670                 gWorkingVarB = Packet->LeftFootPosx - Packet->RightFootPosx;
671
672
673             } else {
674
675                 Packet->WalkingState = STATE_STANDING;
676
677                 Packet->ServoLeftHipRef = Packet->ServoLeftHipPos;
678
679                 Packet->ServoRightHipGain = Param->Standing_All_Gain;
680                 Packet->ServoRightKneeGain = Param->Standing_All_Gain;
681                 Packet->ServoLeftHipGain = Param->Standing_All_Gain;
682                 Packet->ServoLeftKneeGain = Param->Standing_All_Gain;
683
684             }
685         }
```

```
686     } else if (Packet->WalkingState == STATE_RIGHT_PUSH) {
687
688         if ( (Packet->FPEGndPos - Packet->LeftFootPosx) > ( ((float)
689             Param->Push_FPE_Trans_Thres)*0.001f) ) {
690
691             // FPE in front of standing foot so transition to lift,
692             // but perform lift calculations immediately
693
694             Packet->WalkingState = STATE_RIGHT_LIFT;
695
696             gCurrentFPEOffset = Param->Swing_FPE_Offset;
697             gCurrentHipOffset = Param->Push_HipOffset;
698             gCurrentLiftOffset = 0;
699
700             // Standing hip regulating torso vertical
701             if (Packet->FootContactSensors & bitLeftContact) {
702                 Packet->ServoLeftHipRef = Packet->ServoLeftHipPos
703                     + Encoder3Scaled + gCurrentHipOffset;
704             }
705
706             Packet->ServoRightHipRef  = Packet->ServoRightHipPos;
707             Packet->ServoRightHipGain = Param->Lift_SwingLeg_Hip_Gain;
708
709             a = (Packet->HipPosy - ((float) Param->Lift_Height)
710                 *0.001f)*10000.0f;
711             b = (float) (Packet->ServoRightHipPos + Encoder3Scaled
712                 - HipOffsetLookup(Packet->ServoRightKneePos))*5.12f;
713             Packet->ServoRightKneeRef = KneeAngleLookup( (int)
714                 (a/GetCos((int) b)));
715             Packet->ServoRightKneeGain = Param->Lift_SwingLeg_Knee_Gain;
716
717         } else if ( (Packet->RightFootPosx - Packet->FPEGndPos) >
718             (float) StepBehindThreshold*0.001f ) {
719
720             // FPE in front of standing foot so transition to lift,
721             // but perform lift calculations immediately
722
723             Packet->WalkingState = STATE_LEFT_LIFT;
724
725             gCurrentFPEOffset = -Param->Swing_FPE_Offset
726                                 -3*DisturbanceFPEOffset;
727             gCurrentHipOffset = -Param->Push_HipOffset;
728             gCurrentLiftOffset = DisturbanceLiftOffset;
729
730             if (Packet->FootContactSensors & bitRightContact) {
731                 Packet->ServoRightHipRef = Packet->ServoRightHipPos
732                     + Encoder3Scaled + gCurrentHipOffset;
733             }
734             Packet->ServoRightHipGain = Param->Push_StandingLeg_Hip_Gain;
```

```
735             Packet->ServoRightKneeRef  = Param->Push_StandingLeg_Knee_Pos;
736             Packet->ServoRightKneeGain = Param->Push_StandingLeg_Knee_Gain;
737
738             Packet->ServoLeftHipRef  = Packet->ServoLeftHipPos;
739             Packet->ServoLeftHipGain = Param->Lift_SwingLeg_Hip_Gain;
740
741             a = (Packet->HipPosy - ((float) Param->Lift_Height)
742                 *0.001f)*10000.0f;
743             b = (float) (Packet->ServoLeftHipPos + Encoder3Scaled
744                 - HipOffsetLookup(Packet->ServoLeftKneePos))*5.12f;
745             Packet->ServoLeftKneeRef = KneeAngleLookup( (int)
746                 (a/GetCos((int) b)));
747             Packet->ServoLeftKneeGain = Param->Lift_SwingLeg_Knee_Gain;
748
749         } else {
750
751             // else continue with push kinematics
752
753             if (Packet->FootContactSensors & bitLeftContact) {
754                 Packet->ServoLeftHipRef = Packet->ServoLeftHipPos
755                     + Encoder3Scaled + gCurrentHipOffset;
756             }
757             Packet->ServoLeftHipGain = Param->Push_StandingLeg_Hip_Gain;
758
759             Packet->ServoLeftKneeRef  = Param->Push_StandingLeg_Knee_Pos;
760             Packet->ServoLeftKneeGain = Param->Push_StandingLeg_Knee_Gain;
761
762             Packet->ServoRightHipRef  = Packet->ServoRightHipPos;
763             Packet->ServoRightHipGain = Param->Push_SwingLeg_Hip_Gain;
764
765             Packet->ServoRightKneeGain = Param->Push_SwingLeg_Knee_Gain;
766
767             // Add length
768             gWorkingVarA = gWorkingVarA + Param->Push_RateConstant*0.1f;
769
770             // To max of ~L1+L2
771             if (gWorkingVarA > 0.0) {
772                 gWorkingVarA = 0.0;
773             }
774
775             Packet->ServoRightKneeRef = (short) gWorkingVarA;
776
777             a = LegLengthLookup(Packet->ServoRightKneeRef);
778             b = LegLengthLookup(Packet->ServoLeftKneeRef);
779
780             gWorkingVarC = acos( (gWorkingVarB*gWorkingVarB - a*a - b*b)
781                 /(-2*a*b))*254.65; ///3.14159f*800;
782             Packet->ServoRightHipRef = Packet->ServoLeftHipRef
783                 - gWorkingVarC - HipOffsetLookup(Packet->ServoLeftKneeRef)
784                 + HipOffsetLookup(Packet->ServoRightKneeRef);
785         }
```

```
786     } else if (Packet->WalkingState == STATE_RIGHT_LIFT) {
787         // Standing hip regulating torso vertical
788         if (Packet->FootContactSensors & bitLeftContact) {
789             Packet->ServoLeftHipRef = Packet->ServoLeftHipPos
790                 + Encoder3Scaled + gCurrentHipOffset;
791         }
792 
793         Packet->ServoRightHipRef  = Packet->ServoRightHipPos;
794         Packet->ServoRightHipGain = Param->Lift_SwingLeg_Hip_Gain;
795 
796         a = (Packet->HipPosy - ((float) (Param->Lift_Height
797             + gCurrentLiftOffset))*0.001f)*10000.0f;
798         b = (float) (Packet->ServoRightHipPos + Encoder3Scaled
799             - HipOffsetLookup(Packet->ServoRightKneePos))*5.12f;
800         Packet->ServoRightKneeRef = KneeAngleLookup((int)(a/GetCos((int) b)));
801         Packet->ServoRightKneeGain = Param->Lift_SwingLeg_Knee_Gain;
802 
803         if ( Packet->RightFootPosy > ((float) (Param->Lift_Height +
804             gCurrentLiftOffset))*((float) Param->Lift_Trans_Thres)*0.00001f) {
805 
806             Packet->WalkingState = STATE_RIGHT_SWING;
807         }
808 
809     } else if (Packet->WalkingState == STATE_RIGHT_SWING) {
810         //Packet->Debug1 = Packet->FootContactSensors & bitLeftContact;
811 
812         if (Packet->FootContactSensors & bitLeftContact) {
813             Packet->ServoLeftHipRef = Packet->ServoLeftHipPos
814                 + Encoder3Scaled + gCurrentHipOffset;
815         }
816         Packet->ServoLeftHipGain = Param->Swing_StandingLeg_Hip_Gain;
817 
818         a = (float) (atan( (Packet->FPEGndPos -Packet->HipPosx + ((float)
819             gCurrentFPEOffset)*0.001f)/Packet->HipPosy)*800.0f/3.14159f);
820 
821         Packet->ServoRightHipRef = HipOffsetLookup(Packet->ServoRightKneePos)
822             - Encoder3Scaled + (short) a;
823         Packet->ServoRightHipGain = Param->Swing_SwingLeg_Hip_Gain;
824 
825         a = (Packet->HipPosy - ((float)(Param->Lift_Height + gCurrentLiftOffset))
826             *0.001f)*10000.0f;
827         b = (float) (Packet->ServoRightHipPos + Encoder3Scaled
828             - HipOffsetLookup(Packet->ServoRightKneePos))*5.12f;
829         Packet->ServoRightKneeRef = KneeAngleLookup( (int) (a/GetCos((int) b)));
830 
831         if (abs(Packet->ServoRightHipPos - Packet->ServoRightHipRef) < 10) {
832 
833             // Hip (being the slowest servo to move) is sufficiently close
834             Packet->WalkingState = STATE_RIGHT_DROP;
835         }
```

```
836     } else if (Packet->WalkingState == STATE_RIGHT_DROP) {
837
838         // Standing hip regulating torso vertical
839         if (Packet->FootContactSensors & bitLeftContact) {
840             Packet->ServoLeftHipRef = Packet->ServoLeftHipPos
841                 + Encoder3Scaled + gCurrentHipOffset;
842         }
843
844         a = (float) (atan( (Packet->FPEGndPos -Packet->HipPosx + ((float)
845             gCurrentFPEOffset)*0.001f)/Packet->HipPosy)*800.0f/3.14159f);
846         Packet->ServoRightHipRef = HipOffsetLookup(Packet->ServoRightKneePos)
847             - Encoder3Scaled + (short) a;
848
849
850         Packet->ServoRightKneeRef = Param->Drop_SwingLeg_Knee_Pos;
851         Packet->ServoRightKneeGain = Param->Drop_SwingLeg_Knee_Gain;
852
853
854
855         if (Packet->FootContactSensors & bitRightContact) {
856
857
858             if (Packet->Mode == WALKING_MODE) {
859                 Packet->WalkingState = STATE_LEFT_PUSH;
860
861                 Packet->ServoRightHipRef = Packet->ServoRightHipPos;
862                 Packet->ServoLeftKneeRef = Packet->ServoLeftKneePos;
863                 Packet->ServoLeftKneeGain = Param->Drop_StandingLeg_Knee_Gain;
864
865
866                 gWorkingVarA = Packet->ServoLeftKneeRef;
867                 gWorkingVarB = Packet->RightFootPosx - Packet->LeftFootPosx;
868
869
870             } else {
871
872                 Packet->WalkingState = STATE_STANDING;
873
874
875                 Packet->ServoRightHipRef = Packet->ServoRightHipPos;
876
877                 Packet->ServoRightHipGain = Param->Standing_All_Gain;
878                 Packet->ServoRightKneeGain = Param->Standing_All_Gain;
879                 Packet->ServoLeftHipGain = Param->Standing_All_Gain;
880                 Packet->ServoLeftKneeGain = Param->Standing_All_Gain;
881             }
882         }
883     }
884 }
```