Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots

by

Jerry E. Pratt

M.Eng, Massachusetts Institute of Technology (1995)
B.S., Massachusetts Institute of Technology (1994)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2000

Author .....
Department of Electrical Engineering and Computer Science
May 5, 2000

Certified by .....
Gill A. Pratt
Assistant Professor of Electrical Engineering and Computer Science, MIT
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots

by

Jerry E. Pratt

Submitted to the Department of Electrical Engineering and Computer Science
on May 5, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Walking is an easy task for most humans and animals. Two characteristics which make it easy are the inherent robustness (tolerance to variation) of the walking problem and the natural dynamics of the walking mechanism. In this thesis we show how understanding and exploiting these two characteristics can aid in the control of bipedal robots. Inherent robustness allows for the use of simple, low impedance controllers. Natural dynamics reduces the requirements of the controller.

We present a series of simple physical models of bipedal walking. The insight gained from these models is used in the development of three planar (motion only in the sagittal plane) control algorithms. The first uses simple strategies to control the robot to walk. The second exploits the natural dynamics of a kneecap, compliant ankle, and passive swing-leg. The third achieves fast swing of the swing-leg in order to enable the robot to walk quickly ($1.25\frac{m}{s}$). These algorithms are implemented on Spring Flamingo, a planar bipedal walking robot, which was designed and built for this thesis. Using these algorithms, the robot can stand and balance, start and stop walking, walk at a range of speeds, and traverse slopes and rolling terrain.

Three-dimensional walking on flat ground is implemented and tested in simulation. The dynamics of the sagittal plane are sufficiently decoupled from the dynamics of the frontal and transverse planes such that control of each can be treated separately. We achieve three-dimensional walking by adding lateral balance to the planar algorithms. Tests of this approach on a real three-dimensional robot will lead to a more complete understanding of the control of bipedal walking in robots and humans.

Thesis Supervisor: Gill A. Pratt
Title: Assistant Professor of Electrical Engineering and Computer Science, MIT

Acknowledgments

I would like to thank my advisor, Gill Pratt, for creating an ideal work environment and for providing great ideas and advice during my six years in the Leg Laboratory. Working here has been fun and rewarding due to the freedom, support, and respect Gill has given me.

Thanks to Tomás Lozano-Pérez for pointing out the many flaws and omissions in my arguments and helping to clarify what I really accomplished over the course of this thesis.

Thanks to Steve Massaquoi for pointing me in the right direction with many references on the biological aspects of this thesis and also for giving good all-around advice.

A lot of people provided crucial help in the design and construction of Spring Flamingo. The actuators were based on a design and prototype by Mike Wittig for his Undergraduate Thesis. Robert Ringrose helped with the development software which was a modification of the lab's Creature Library and other lab simulation software. Mike Wessler made additional improvements to the robot interface software. Dave Robinson designed various versions of the feet, constructed strain gage conditioning circuits, and helped with design advice and machining. Dan Paluska designed the final version of the foot, helped design and construct the ramp for rough terrain experiments, and helped with assembly and design advice. Peter Dilworth provided design advice. Ann Torres helped with machining and named the robot based on its appearance. Allen Parseghian and Chee Meng Chew provided advice and help on the 3D simulation.

Thanks to Rodger Kram and Max Donelan for discussions and advice on how best to present the work on limits to maximum walking speed.

Thanks to the members of the Leg Laboratory who have supported me and become friends over the years. I couldn't think of a more unique and talented group of people to work with. Thanks to the many proofreaders: Ben, Mike, Allen, Chee Meng, Dan, Greg, Bruce, Chris, Teri, John, and Dave who helped point out many errors and gave great advice on making the document more readable. Thanks to the alumni of the Leg Laboratory who have provided a great deal of infrastructure which was important for this project - particularly all the simulation software.

Thanks to all the members of the Flanking Ito's and Juicy Chicken for all the fond memories.

I would like to thank my close friends Eric Amundsen, Brindha Muniappan, Len Granowetter, and Amy Rochelle for keeping me sane these last few years. There's no greater way to get one's mind off work then spending six hours in an SUV named Lucy and making it seem like six minutes. Also thanks to Len and Amy and all the others who agreed to partake in the fast walking experiments.

Thanks to my parents, family, teachers, coaches, and friends. The support and encouragement through the years has been terrific.

This thesis is dedicated to my best friend, Megan Benson Pratt.

This research was supported in part by the Defense Advanced Research Projects Agency under contract number N39998-00-C-0656 and the National Science Foundation under contract numbers IBN-9873478 and IIS-9733740.

Contents

1 Introduction **15**
- 1.1 Thesis . . . 15
- 1.2 Synopsis . . . 16
- 1.3 Motivation . . . 17
- 1.4 Bipedal Walking is Difficult (When Viewed as a General Dynamical System) . . . 18
- 1.5 Bipedal Walking is Easy (When Viewed as a Specific Mechanism) . . . 18
  - 1.5.1 Exploiting Inherent Robustness . . . 19
  - 1.5.2 Exploiting Natural Dynamics . . . 19
- 1.6 Experimental Robot . . . 20
- 1.7 Thesis Contributions . . . 20
- 1.8 Thesis Outline . . . 20
- 1.9 Note on Data in this Thesis . . . 21

2 Background **23**
- 2.1 Powered Bipedal Walking Robots . . . 23
  - 2.1.1 Waseda . . . 23
  - 2.1.2 P2 and P3 . . . 26
  - 2.1.3 Toddler . . . 26
  - 2.1.4 Moscow State University Biped . . . 27
  - 2.1.5 SD-2 . . . 27
  - 2.1.6 Biper . . . 27
  - 2.1.7 Meltran . . . 28
  - 2.1.8 Timmy . . . 28
  - 2.1.9 Powered Robot Summary . . . 29
- 2.2 Passive Dynamic Bipedal Walkers . . . 29
- 2.3 Virtual Model Control . . . 31
- 2.4 Series Elastic Actuators . . . 32
- 2.5 Conclusions . . . 32

3 Simple Models of Bipedal Walking **35**
- 3.1 Center of Mass and Center of Pressure . . . 35
- 3.2 Inverted Pendulum Models . . . 36
- 3.3 Multi Joint Pendulum Model . . . 38
- 3.4 Adding a Foot . . . 38
- 3.5 Flywheel Models . . . 39
- 3.6 Acrobot Model . . . 41
- 3.7 Massless Leg Biped Model . . . 42
- 3.8 A Word about Singularities . . . 43
- 3.9 Planar Biped with Distributed Mass . . . 44
- 3.10 Summary of Simple Models . . . 45

4 Exploiting Inherent Robustness **47**
4.1 Control Algorithm Properties . . . 48
4.2 Simple Control Strategies for Bipedal Walking . . . 48
4.2.1 Height Stabilization . . . 49
4.2.2 Pitch Stabilization . . . 49
4.2.3 Speed Stabilization . . . 50
4.2.4 Swing-Leg Placement . . . 50
4.2.5 Support Transitions . . . 51
4.3 Virtual Actuator Implementation for a Planar Biped With Feet and Ankles . . . 51
4.3.1 Single Leg Implementation . . . 52
4.3.2 Dual Leg Implementation . . . 54
4.4 Simple Control Strategies Applied to a Bipedal Walking Robot . . . 55
4.4.1 Walking Algorithm . . . 56
4.4.2 Robustness of Walking Algorithm . . . 58
4.4.3 Self-Stabilizing Speed . . . 60
4.5 Blind Walking over Hills and Rolling Terrain . . . 60
4.5.1 Estimating the Slope of Hills . . . 60
4.5.2 Simple Control Strategies for Hills . . . 60
4.5.3 Results on Hills . . . 62
4.6 Conclusions . . . 62

5 Exploiting Natural Dynamics **65**
5.1 Natural Dynamic Mechanisms . . . 65
5.1.1 Knee Cap . . . 65
5.1.2 Compliant Ankle . . . 66
5.1.3 Passive Swing-Leg . . . 67
5.2 Simulation Algorithm . . . 68
5.3 Robot Algorithm . . . 69
5.4 Conclusions . . . 71

6 Limits to Speed in Bipedal Walking **75**
6.1 Non-Dimensional Parameters: Froude Number and Pratt Number . . . 75
6.2 Stride Length and Stride Time Scaling with Speed . . . 76
6.3 Limits to Stride Length . . . 76
6.3.1 Groucho Running . . . 77
6.4 Limits to Stride Time . . . 79
6.4.1 Three-Legged Walking . . . 80
6.4.2 What Limits Swing-Time? . . . 83
6.5 Predicting Maximum Walking Speed . . . 83
6.6 Discussion . . . 84
6.6.1 Scaling Maximum Speed with Gravity (Will we Walk on Mars? Or will we Hop?) . . . 84
6.6.2 Dynamically Similar Walking . . . 87
6.6.3 Discussion of Experimental Methods . . . 88
6.6.4 Relevance to Bipedal Walking Robots . . . 88

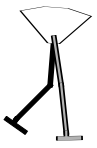
7 Fast Walking Algorithm **89**
7.1 Fast Walking Algorithm . . . 89
7.2 Swing-Leg Control . . . 90
7.3 Results . . . 92
7.4 Conclusions . . . 95

8 3D Bipedal Walking Simulation **99**
8.1 3D Simulation Algorithm . . . 99
8.1.1 Lateral Stability . . . 101
8.1.2 3D Simulation Results . . . 104
8.2 Conclusions . . . 107

9 Conclusions **109**
9.1 Discussion . . . 110
9.1.1 How far can this approach get us? . . . 110
9.1.2 Approaches to Developing Bipedal Walking Robots . . . 110
9.2 Further Work . . . 111
9.2.1 3D Walking Robot . . . 111
9.2.2 Understanding Human Walking . . . 112
9.2.3 Central Pattern Generators . . . 112
9.2.4 Multi-Joint Muscle Groups . . . 112
9.2.5 Producing a Practical Robot . . . 113
9.2.6 Learning to Walk . . . 113
9.2.7 Summary . . . 113

A Experimental Apparatus **121**
A.1 Overall Robot Setup . . . 121
A.2 Joint Technical Data . . . 127
A.3 Actuators . . . 127
A.4 Joint Torque De-Coupling . . . 130
A.5 Electronics . . . 130
A.6 Calibration . . . 131
A.7 Conclusion . . . 131

B Adaptive Control of Swing Leg **151**
B.1 Equations of Motion . . . 151
B.2 Adaptive Control . . . 153
B.3 Results . . . 154



List of Figures

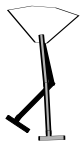
1-1 Spring Flamingo taking a stroll with the author looking on 16

2-1 Some previous powered bipedal walking robots. 24
2-2 A passive dynamic walker developed by Andy Ruina's lab at Cornell University . . . 29
2-3 Virtual Model Control applied to Spring Turkey. 31
2-4 Schematic of a Series Elastic Actuator 33

3-1 Picture illustrating the Center of Pressure. 35
3-2 Picture illustrating the similarity between walking and an inverted pendulum. 36
3-3 Simple pendulum model of bipedal walking during single support. 37
3-4 Simple inverted pendulum with linear actuator. 38
3-5 Multi-joint model with point mass and point foot. 39
3-6 Walking model with point mass body and foot with actuated knee and ankle. 39
3-7 Inverted pendulum model with flywheel. 40
3-8 Inverted pendulum with flywheel and linear actuator. 41
3-9 Acrobot model with only one actuated degree of freedom and inertial body. 41
3-10 Massless leg biped model and equivalent free body diagram. 43
3-11 Robot model with distributed mass. 45

4-1 Virtual Model implementation on a single leg. 52
4-2 Spring Flamingo, a planar bipedal walking robot. 55
4-3 [REAL] State machine used in Spring Flamingo's walking algorithm. 56
4-4 [REAL] Spring Flamingo walking data. 59
4-5 [REAL] Elapsed time snapshot of Spring Flamingo walking data. 59
4-6 Graphical definition of global slope and local slope 61
4-7 [REAL] Photograph of Spring Flamingo walking over a ramp with 15^o upslope and downslope. 62
4-8 [REAL] Spring Flamingo walking data while walking over a 15^o ramp. 63
4-9 [REAL] Video frames of Spring Flamingo walking over rolling terrain with 15^o upslopes and downslopes. 63

5-1 Diagram illustrating kneecap advantages. 66
5-2 Diagram illustrating compliant ankle. 66
5-3 Diagram illustrating passive swing. 67
5-4 [SIM] Simulation Algorithm. 68
5-5 [SIM] Elapsed time snapshot of the simulated planar robot walking data. 70
5-6 [SIM] Planar simulation data exploiting natural dynamics. 70
5-7 Spring Flamingo photograph. 71
5-8 [REAL] Physical robot algorithm exploiting natural dynamics. 72
5-9 [REAL] Elapsed time snapshot of physical robot walking data exploiting natural dynamics. 72
5-10 [REAL] Physical robot walking data exploiting natural dynamics. 73
5-11 [REAL] Specific resistance of Spring Flamingo when walking exploiting its natural dynamics. 74



6-1 Scaling of stride length and stride time with speed in normal human walking. 77
6-2 McGeer's rimless wheel model of walking. 78
6-3 Results of Groucho Running experiments. 79
6-4 Rotating pendulum model for estimating walking speed limit. 79
6-5 Cartoon of standard human bipedal walking. 80
6-6 Cartoon of human walking with three legs. 81
6-7 Three-legged walking showing each leg separately for clarity. 81
6-8 Stop frame animation of three-legged walking. 82
6-9 Video frames of a test subject (the author) three-legged walking on a treadmill. . . . 82
6-10 Results of fast walking, three-legged walking, and three-legged Groucho Running. . . 83
6-11 Swing Limits Data. 84
6-12 Results from maximum velocity walking experiment. 85
6-13 Results from maximum velocity walking experiments with ankle weights. 86
6-14 Human walking speed limits as a function of gravity. 87

7-1 REAL Fast walking algorithm. 90
7-2 REAL Data showing thigh and shin tracking during swing while walking $1.1\frac{m}{s}$. . . . 93
7-3 REAL Data from Spring Flamingo walking at $1.1\frac{m}{s}$. 94
7-4 REAL Elapsed time snapshot of Spring Flamingo walking at $1.1\frac{m}{s}$. 95
7-5 REAL Data showing control of walking speed. 96
7-6 REAL Data showing disturbance rejection due to being pushed. 97
7-7 REAL Relative stride lengths and stride times achieved for various walking trials of Spring Flamingo, compared to data from human subjects. 97

8-1 SIM Three dimensional bipedal walking simulation. 100
8-2 SIM 3D Simulation Algorithm 101
8-3 Simple pendulum model of the dynamics of the 3d simulation in the frontal plane. . 103
8-4 Range of Capture Angle vs. lateral velocity for a simple pendulum model with ankle torque. 104
8-5 SIM Elapsed-time snapshot of the 3d simulated robot walking data. 105
8-6 SIM Elapsed-time isometric snapshots of the 3d simulated robot walking data showing lateral displacement of the pelvis. 105
8-7 SIM 3D simulation walking data. 106

A-1 Experimental setup. 122
A-2 Side view of Spring Flamingo. 123
A-3 Top view of Spring Flamingo. 124
A-4 Front view of Spring Flamingo. 125
A-5 Hip and knee detail. 126
A-6 Ankle and foot detail. 126
A-7 Hip, knee, and ankle range of motion. 127
A-8 Actuator top view. 128
A-9 Actuator Bottom view. 129
A-10 Experimentally determined bode diagram of actuator force response. 129
A-11 Electronics Layout. 131
A-12 DSP Board, Analog Board, and Breakout Board forming the on-board computer system. 132
A-13 Force Control Board which implements a PD controller to control the actuator output force. 132
A-14 Power Board and Strain Gage Conditioning Board. 133
A-15 "Legplot", a graphical display and analysis program showing plots of data uploaded from Spring Flamingo's on-board computer system. 133
A-16 Spring Flamingo with broken leg. 134
A-17 Breakout Board Schematic Page 1 of 6. 135
A-18 Breakout Board Schematic Page 2 of 6. 136

A-19 Breakout Board Schematic Page 3 of 6. 137
A-20 Breakout Board Schematic Page 4 of 6. 138
A-21 Breakout Board Schematic Page 5 of 6. 139
A-22 Breakout Board Schematic Page 6 of 6. 140
A-23 Power Board Schematic Page 1 of 2. 141
A-24 Power Board Schematic Page 2 of 2. 142
A-25 Force Control Board Schematic Page 1 of 2. 143
A-26 Force Control Board Schematic Page 2 of 2. 144
A-27 Strain Gage Conditioning Board Schematic Page 1 of 2. 145
A-28 Strain Gage Conditioning Board Schematic Page 2 of 2. 146

B-1 Dynamic model of swing leg. 152
B-2 Swing leg tracking using adaptive control. 155
B-3 Tracking errors for adaptive control of swing leg. 156
B-4 Dynamic parameter estimates for adaptive control of swing leg. 157
B-5 Gravitational parameter estimates for adaptive control of swing leg. 157

List of Tables

4.1 REAL Control Parameters for Spring Flamingo's Walking Algorithm. 57
4.2 REAL Additional parameters for Spring Flamingo's walking algorithm over rough terrain. 61

5.1 SIM Physical parameters and controller parameters of the simulated planar bipedal walker. 69

6.1 Vital Statistics of Human subjects used in the walking experiments. 78

8.1 SIM Physical parameters of the simulated 3D bipedal walker. 99
8.2 SIM Control system parameters of the simulated bipedal walker. 102

A.1 Continuous torque and maximum speed specifications for the joints. 127
A.2 Maximum torques and speeds during fast walking. 127
A.3 Parts List of off-the-shelf mechanical and major electrical parts. 147
A.4 Suppliers, Page 1 of 2. 148
A.5 Suppliers, Page 2 of 2. 149

Chapter 1

Introduction

1.1 Thesis

Building and controlling bipedal walking robots can help us to understand how humans walk. The symbiosis between controlling robots and understanding humans can be achieved by using a robot to test theories on how humans walk, to further understand the biomechanics of both robots and humans, to discover limitations to walking in both humans and robots, and to suggest testable hypotheses on how humans control walking.

Testing Theories

Robots can be used to test theories on how humans walk. Since we can access the internals of the robot more easily than the internals of a human, testing theories on a robot is fairly easy, once the robot is built and working. In this thesis, we use strategies for standing and balancing that are similar to those discovered by researchers in biomechanics. We modify the center of pressure on the foot to balance, which is similar to the "ankle strategy" used by humans [Kuo & Zajac (1993*b*,*a*)], thus testing the usefulness of that strategy. As more biomechanical hypotheses on human walking are proposed, robots may be a useful tool for testing them.

Understanding Biomechanics

An understanding of the biomechanics of walking is important for building and controlling bipedal robots. In the process of controlling the robot in this thesis, we analyzed simple models of walking. Since humans must obey the same laws of physics as a robot, understanding the physics of one helps us understand the physics of the other.

Understanding Limitations in Walking

Some of the same elements that limit speed, efficiency, and grace in a human probably limit robots as well. By building robots which have good performance, we start to understand the limiting factors of performance. If these limits apply to human walking, then we start to understand human walking more fully. In this thesis, we examine the limits to maximum walking speed and find that the minimum swing time of the swing-leg is a major limiting factor in walking, both for humans and robots.

Suggesting Biomechanical Experiments

Good strategies for controlling a bipedal walking robot may also be good strategies for controlling human walking. In this thesis, we use a number of strategies for controlling height, pitch, speed, support transitions, and the swing-leg. These strategies may be good candidate hypotheses for how humans control walking and could be tested in a biomechanics laboratory.

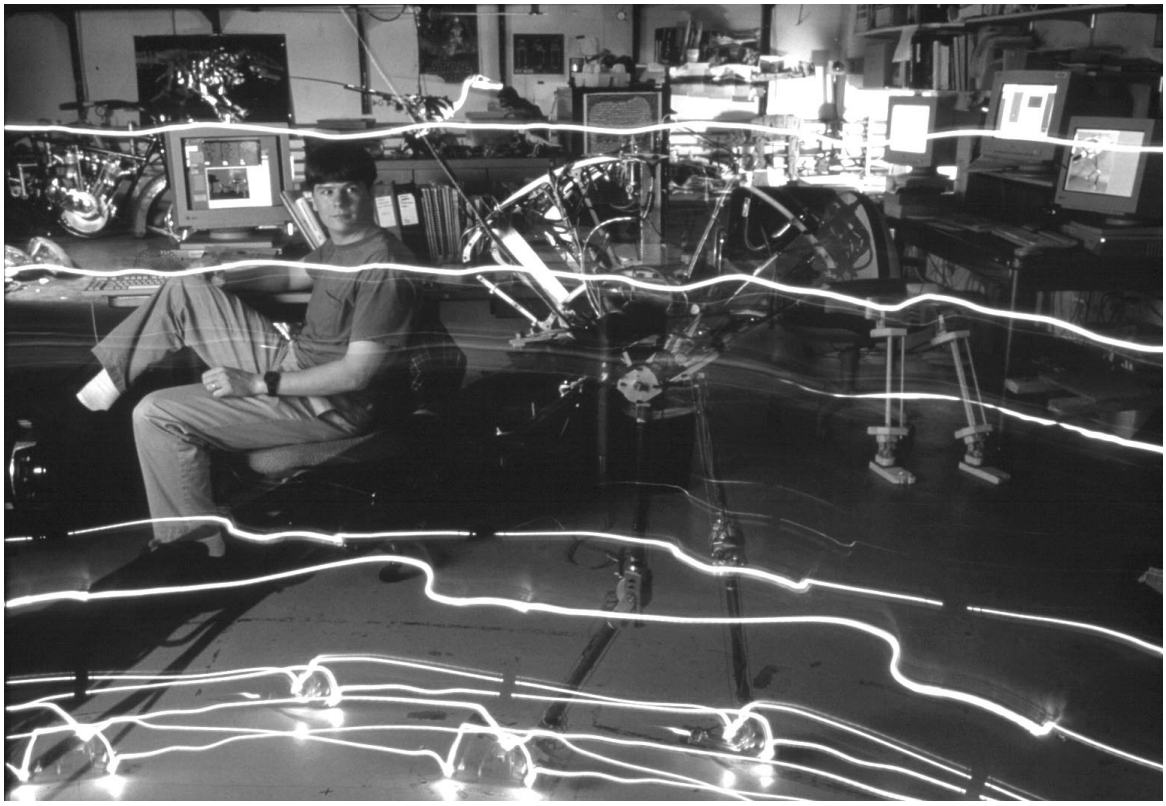


Figure 1-1: Spring Flamingo taking a stroll with the author looking on. Lights trace out the motion of the top and middle of the robot's body, the hip, knee, and ankle joints, and the toe and heel of the feet. Photo by Peter Menzel, Copyright 1999. Reproduced with permission.

1.2 Synopsis

This thesis presents three algorithms for controlling a planar bipedal walking robot called Spring Flamingo (Figure 1-1) and one algorithm for controlling a 3D bipedal walking simulation. The algorithms are all based on simple physical models of walking and employ simple control strategies. For instance, the control strategies for the first planar algorithm are

- Maintain a constant stance leg length by pushing up until hitting the knee cap.
- Maintain a constant level pitch using a virtual spring-damper mechanism with constant set point.
- Transition from double support to single support when the body's forward position becomes further than a preset distance from the rear foot or closer than a preset distance from the front foot.
- Transition from single support to double support when the body's forward position becomes further than a preset distance from the support foot.
- Swing the non-stance leg such that the foot will be placed a desired stride length away from the support foot.
- Increase the nominal stride length as the robot walks faster.
- Delay transition to double support if the robot is walking too slowly. Conversely, initiate transition to double support sooner if the robot is walking too quickly.
- Maintain the center of pressure of the support foot approximately below the center of mass, moving it forward if walking too quickly or backward if walking too slowly.
- During double support shift the load toward the back leg if walking too slowly or toward the front leg if walking too quickly.

These strategies are implemented with simple control tools including Virtual Model Control [Pratt (1995)], and linear control laws. The parameters of these equations are tuned manually. Since they have a clear physical interpretation, tuning is straightforward. A state machine is used to cycle in and out those strategies which are applicable to different phases of the walking cycle.

In the second algorithm, the natural dynamics of the walking mechanism are exploited. The natural mechanisms considered are a knee-cap joint limit which makes height control easy (already exploited in the first algorithm), a compliant ankle which naturally moves the center of pressure on the foot forward as the robot moves forward, and a passively swinging swing-leg.

Fast walking is achieved in the third algorithm by focusing on swinging the swing-leg as fast as possible. This requires actively driving the swing-leg. To do so, we use feed-forward inverse dynamics (computed torque) along with feedback to make the hip track a minimum jerk trajectory [Flash & Hogan (1985)] during swing. The knee remains passive during the first half of swing and tracks a spline trajectory during the second half of swing. Using this method we achieved walking speeds up to $1.25\frac{m}{s}$ which, while a moderate speed for a human, is quite fast for contemporary bipedal robots.

The 3D algorithm builds on the planar algorithms with lateral balance controlled with foot placement and ankle torque. This algorithm is verified in simulation. The algorithm is currently being adapted for a real biped, called M2.

This work shows that simple control algorithms can successfully control bipedal walking robots. The robots walk smoothly and efficiently and appear natural. The algorithms that exploit natural dynamics show how natural mechanisms can be used to simplify the algorithm while enhancing efficiency. The fast walking algorithm verifies that fast walking can be achieved if swing time can be reduced through active control. The 3D algorithm verifies that these techniques are applicable not just to planar walkers but also to a 3D robot.

1.3 Motivation

There are over 20 billion bipedal walking machines in the world today, yet no one fully understands how they work. Biologists have gained knowledge of the mechanics of walking and muscle firing patterns. However, researchers are only starting to understand the control of bipedal animals. Recently, engineers have begun designing, building, and controlling bipedal walking robots. These machines and their control have enormous potential for helping to test control strategies that animals might employ and for suggesting new experiments.

While the main character of this thesis is a bipedal robot called Spring Flamingo, the main theme is to further the understanding of bipedal walking, not just to maximize the performance of a single robot. We believe that not only is understanding bipedal walking as important a goal as building bipedal walking machines, but that emphasizing understanding will more quickly lead to maximizing performance and producing useful walking machines in the future.

Understanding bipedal walking can take many forms. In this thesis, we promote understanding of walking through simple physical models and simple control algorithms which relate to those models. The algorithms are kept functionally transparent, such that it is easy to understand the purpose of any fragment of the algorithm and any control parameters based on their underlying physical meaning.

An additional benefit of keeping the algorithms simple and minimizing the required control effort is that it makes it likely that the algorithms are biologically plausible, meaning that they could be implemented in a reasonable amount of biological hardware. However, even if control strategies used on the robot are similar to those of biological creatures, the exact way the strategies are implemented on a biological creature may be very different from how they are implemented on the robot. How biological creatures implement control is still an open question.

Nature tends to exploit the relevant aspects of a specific problem. Likewise, instead of attempting to develop techniques that are generally applicable across the spectrum of robotics, we focus only on bipedal walking. This allows us to simplify the control by exploiting elements which are specific to bipedal walking. Some of these elements include the specific nature of the bipedal walking problem

and the natural dynamics of bipedal mechanisms.

1.4 Bipedal Walking is Difficult (When Viewed as a General Dynamical System)

There are several characteristics of bipedal walking robots that make them seemingly difficult to control:

- Non-linear dynamics
- Multi-variable dynamics
- Naturally unstable dynamics
- Limited foot-ground interaction
- Discretely changing dynamics
- Subjective performance evaluation

The first three characteristics make synthesizing a controller using traditional control techniques difficult, but do not rule it out. Many "textbook" control problems are non-linear, multi-variable, and naturally unstable. It is the last three of these characteristics that move bipedal walking outside the range of traditional techniques.

Limited foot-ground interaction is a distinctive feature of bipedal walking. It is what makes walking different from the control of robotic arms, which have several traditional control methods available. Feet can only push on the ground, not pull. Also, the torque about the foot is limited as the foot will rotate over its toe or its heel if too much torque is applied. Because of this, limited control action can occur during a stride of walking. In particular, the forward velocity of the robot cannot be instantaneously controlled.

The dynamics of a bipedal walker change as it transitions from single support to double support and back to single support. Since the dynamical equations are not continuous, determining Lyupanov functions or applying other traditional techniques is challenging.

The performance measure of a bipedal walker is not as well-defined as that of typical robotic systems. For example, the performance of an industrial robot arm is often measured by how well it can follow a given trajectory. In bipedal walking, performance should not be based on trajectory following, as the exact trajectory does not really matter. To a first order approximation, performance is a one-bit measure – does the robot walk or does it fall down? Some secondary considerations such as smoothness and efficiency are easy to quantify, but those such as grace and biological similarity are more subjective. How does one quantify the emotional response of a kindergarten class upon seeing a robot walk?

Because bipedal walking is a challenging control problem, we have taken the approach of determining a controller based on the specific physics of bipedal walking, rather than attempting to develop a general approach which is applicable to other classes of robots. By concentrating specifically on bipedal walking, we can focus on the factors that make it unique and can be exploited in the control of walking. Some of these factors make bipedal walking easy, despite seeming challenging from a control synthesis point of view.

1.5 Bipedal Walking is Easy (When Viewed as a Specific Mechanism)

There are several characteristics of walking which make it easy. These include the inherent robustness of the bipedal walking problem and the natural dynamics of the bipedal walking mechanism. These characteristics can be exploited in the control of bipedal walking robots.

1.5.1 Exploiting Inherent Robustness

The inherent robustness of the bipedal walking problem can be exploited in the control of walking. By inherent robustness, we mean that specific trajectories, precision, and repeatability are not important in walking. The resultant motion can vary considerably between individuals and even in the same individual from step to step and it does not matter. Bipedal walking is achieved if the robot gets from point A to point B without falling down. From a mathematical perspective, this is equivalent to saying that there is an enormous set of trajectories through state space which can be considered satisfactory. Simple control techniques can be used to achieve one of these trajectories.

The specific nature of the walking problem can also be exploited in control. Bipedal walking occurs when the feet (at the distal end of the legs) are placed one in front of the other in a repeating pattern, with legs alternating between support and swing. Simple models of this technique can be used to develop insight and determine control strategies. For instance, the simplest model of bipedal walking is an inverted pendulum rising and falling. We know from experience that the pendulum slows down as it is rising and speeds up as it is falling. By adding an articulated body and an actuated foot to this model, we can make an analogy between the center of mass and the mass of the pendulum and between the center of pressure on the foot and the pivot of the pendulum. To a first order approximation, whenever the center of mass is in front of the center of pressure, the robot accelerates, and whenever the center of mass is behind the center of pressure, the robot decelerates.

These and other simple models can be exploited to determine simple strategies for bipedal walking. For example, in order for the robot to speed up, it can increase the amount of time that the center of mass of the robot is in front of the center of pressure, either through foot placement, ankle torque, force distribution during double support, or body lean.

1.5.2 Exploiting Natural Dynamics

Many robotics researchers have recognized the usefulness of designing a mechanism such that the natural dynamics performs much (if not all) of the task, and the rest is left to the controller. In this way, instead of viewing the mechanics strictly as a dynamical system to be controlled, it is often treated as part of the solution itself. In this thesis, we try to maintain this view and consider the bipedal walking mechanism and some of its natural dynamical mechanisms as part of the control solution.

Many researchers [Adolfsson et al. (1998), Fowble & Kuo (1996), Garcia et al. (1998), Goswami et al. (1997), McGeer (1990*a*)], have exploited natural dynamics in order to make walking machines which are fully passive. These devices rely completely on their dynamics, and interaction with gravity and the ground, in order to walk. Schaal & Atkeson (1993) showed that open-loop strategies can be successfully used in robot juggling tasks. Williamson (1999) exploited the natural dynamical interaction between neural-like oscillators and mechanical arms to perform tasks such as playing with a slinky, turning a crank, and drumming. Raibert (1986) exploited the natural dynamics of a springy pogo-stick-like leg in his running machines. Playter & Raibert (1994) showed that unstable layout somersault maneuvers can be stabilized by the addition of arms connected to the body with passive springs. Ringrose (1997) showed that monopedal, bipedal, and quadrupedal hoppers utilizing a constant speed motor, springy leg, and curved foot can hop without any sensing or control. Moran (September 1996) reports on some of these approaches and describes the PAMS (passive aerodynamically stabilized magnetically damped satellite), a satellite designed to stably hold an orientation while rotating about the Earth, without the need for thrusters, sensors, or control.

This prior work in exploiting natural dynamics motivates the approach described in Chapter 5 of this thesis. Three different natural mechanisms are exploited. A knee cap prevents the leg from inverting, which makes control of height simple. A compliant ankle makes the center of pressure on the foot travel forward with the center of mass of the body. And the natural swing dynamics of the leg makes swing control simple and natural looking.

1.6 Experimental Robot

Experiments on a real robot (Figure 1-1) are emphasized in this thesis. While we also use simulation extensively, we find that building a real robot forces the resolution of issues that might otherwise be overlooked in simulation. For example, sensor noise and actuator limitations are important issues, both in building real robots and understanding biological systems. An algorithm that is not robust to noise might work well in simulation, but may not be applicable to a real robot or an animal. While noise and sensor limitations can be modeled in simulation, there may be other aspects which are overlooked or are difficult to model, such as link compliance, foot-ground interaction, and stiction. Building a robot eliminates the need to model these effects as the real world will account for them. Also, it is often useful to interact with the robot to get a feel for whether the algorithm is performing as expected and to get a feel for how control parameters may be changed. For instance, suppose one wishes to tune a parameter corresponding to the torsional spring gain on a pitch servo. By physically rotating the robot's body and feeling how the robot responds, one can decide if that parameter needs to be increased or decreased. Thus the robot provides a user interface which would be very difficult to reproduce in simulation. Besides, building robots is fun and educational, and one of the best means for inspiring future generations of engineers.

1.7 Thesis Contributions

The contributions of this thesis are:

1. The presentation of simple models of bipedal walking which aid in understanding the mechanics of bipedal walking.
2. The design and construction of a planar bipedal walking robot, called Spring Flamingo.
3. The development of simple control algorithms which allow Spring Flamingo to walk on flat ground and rolling terrain.
4. The demonstration that natural dynamic mechanisms can make control of bipedal walking easier to achieve and more natural looking, thereby bridging the gap between passive dynamic walkers and powered walkers.
5. The presentation of the limits to top speed in bipedal walking.
6. The development of a fast walking algorithm that results in speeds up to $1.25\frac{m}{s}$, which is quite fast for contemporary bipedal walking robots.
7. The development, in simulation, of a simple control algorithm for 3D bipedal walking on flat ground.

1.8 Thesis Outline

This thesis proceeds as follows:

Chapter 2 gives background for this thesis, consisting of previous powered and passive bipedal walking robots, Virtual Model Control, and Series Elastic Actuators.

Chapter 3 presents several simple models of bipedal walking. These models help build physical intuition into walking, which aids in algorithm development.

Chapter 4 presents a simple control algorithm for planar bipedal walking. The algorithm is physically based on the simple models presented in Chapter 3. The algorithm uses a state machine and simple control strategies. Simple modifications are made to the algorithm for walking over slopes and rolling terrain.

Chapter 5 presents an algorithm that relies on the natural dynamics of a knee-cap, compliant ankle, and passive swing-leg. These mechanisms simplify height control, speed control, and swing-leg

control. Speed was found to be self-stabilizing for slow speeds. That is, there was no explicit control mechanism which fed back walking speed.

Chapter 6 hypothesizes that minimum swing time is a major limiting factor of maximum walking speed at Earth's gravity. Evidence to support that hypothesis is presented. This knowledge is used in the fast walking algorithms of Chapter 7.

Chapter 7 presents an algorithm for walking fast. The algorithm focuses on swinging the swing-leg quickly. To do so we use feed-forward inverse dynamics (computed torque) plus active feedback.

Chapter 8 presents an algorithm for 3D walking. The algorithm is an extension of the planar algorithms by adding lateral speed control through lateral foot placement and ankle torque.

Chapter 9 concludes the thesis with discussion and suggestions for future work.

Appendix A gives an overview of the experimental setup.

Appendix B describes the adaptive control method used to determine the dynamic parameters of the swing-leg for the control in Chapter 7.

1.9 Note on Data in this Thesis

This thesis includes data both from simulations and from the real robot. The figure captions indicate the source of the data. Figures with simulated data are marked SIM while figures with data from the real robot are marked REAL. This approach is borrowed from Williamson (1999).

Chapter 2

Background

In this Chapter we review previous powered and passive bipedal walking robots which have influenced this thesis and two technologies, Virtual Model Control and Series Elastic Actuators, which are used throughout this thesis.

There are many powered bipedal walking robots, some of which will be reviewed in Section 2.1. The existence of these robots and some of their control techniques have motivated much of the work and the approach followed in this thesis.

Passive dynamic walkers, reviewed in Section 2.2, are an interesting class of walking robots which have no actuation, sensing, or control systems. These robots walk down a shallow ramp, powered only by gravity. They motivate the natural dynamic mechanisms discussed in Chapter 5.

Virtual Model Control, reviewed in Section 2.3, is a descriptive control language which uses simulated mechanical elements, connected to a real robot, in order to control the robot to perform a task. Virtual Model Control aids the implementation of intuitive control strategies discussed in Chapter 4.

Series Elastic Actuators, reviewed in Section 2.4, are force-controllable actuators which enable much of the experimental work in this thesis. They allow for the implementation of Virtual Model Control, which requires high force fidelity and bandwidth, and the exploitation of natural dynamics which requires low actuator impedance. They also provide high shock tolerance which prevents foot-ground impulses from damaging the robot.

2.1 Powered Bipedal Walking Robots

There are a number of powered bipedal walking robots that have been built by various groups throughout the world. A handful exploit the inherent robustness of walking by using simple control methods. A few of them benefit from the natural dynamics of the mechanism, though none explicitly exploits the natural dynamics.

The robots fall into two broad categories. Many play back pre-recorded joint trajectories. Others use heuristic control approaches. Some of them combine both playback and heuristic control. Below we describe some of the robots which have had an impact on this thesis.

2.1.1 Waseda

Researchers at Waseda University have been working on powered bipedal walking robots since 1969. They have built a series of robots; the first were static walkers, and the most recent robots are dynamic walkers. All of the robots share the common feature that they play back pre-recorded joint trajectories. The recent robots all use trunk motion to dynamically compensate for moments so that the center of pressure on the foot follows a desired trajectory.

WL-5 [Kato & Tsuiki (1972)] was an 11 degree-of-freedom static walker which walked with a 20 centimeter step at 2 minutes per step. The robot was controlled by playing back pre-recorded joint trajectories. The only feedback was bang-bang hydraulic position control of the joints.

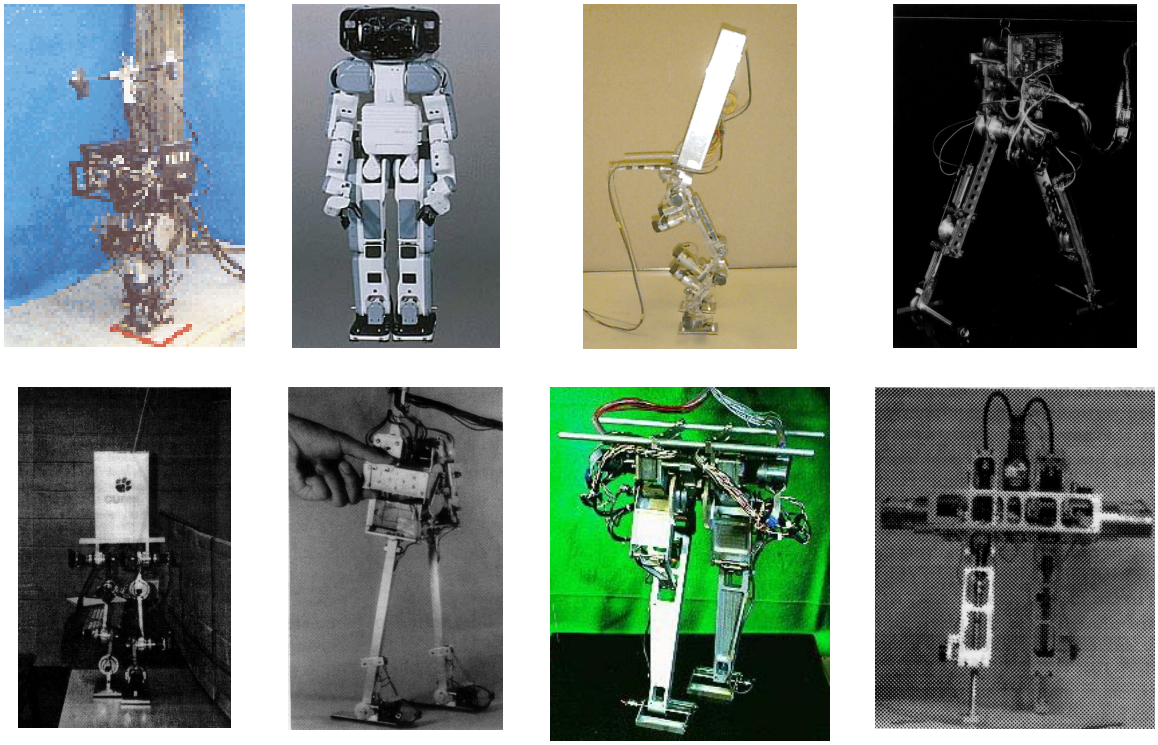


Figure 2-1: Some previous powered bipedal walking robots. From left to right are WL-10RV1 from Waseda, P2 from Honda, Toddler from UNH, the Moscow State University Biped, SD-2 from Clemson and Ohio State, Biper from University of Tokyo, Meltran II from Mechanical Engineering Lab in Tsukuba, and Timmy from Harvard.

WL-9RD [Kato et al. (1981)], a 10 degree-of-freedom biped, walked quasi-dynamically with a 45 centimeter step at 9 seconds per step. During single support, pre-recorded static joint trajectories were played back. During exchange of support, the robot dynamically transferred support from one leg to another by following the inverted pendulum dynamic trajectory, that is, by falling onto the new support leg.

WL-10RD [Takanishi et al. (1985)], a 12 degree-of-freedom biped, walked dynamically by playing back joint trajectories which result in dynamic walking. To ensure stability, the resultant path of the zero moment point (ZMP), also known as the center of pressure (COP), was computed before walking. As long as the ZMP remains strictly inside the support polygon of the support foot, the robot can be treated as though it is a fixed manipulator. During single support, the only feedback control was the control of the joint trajectories. During the changeover phase, all the joints were fixed except for the ankles which employed torque feedback to transfer the center of mass from one foot to the other. Strain gauges on the motor shaft were used during this stage to estimate the torque on the joint. Using this algorithm, WL-10RD walked at 1.5 seconds per step with a step length of 40 centimeters.

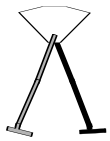
WL-12RIII [Takanishi et al. (1990*a*)] employed a heavy trunk with 2 degrees of freedom to ensure dynamically that the ZMP stayed within the support polygon of the foot. The robot had 8 degrees of freedom in total. Before walking, desired lower limb and ZMP trajectories were determined. Trunk motion was determined by iteratively solving the approximate dynamic equations of motion using the desired lower limb and ZMP trajectories. During walking, these pre-recorded trajectories were played back without any additional feedback or on-line control. With this method, the robot achieved walking on stairs with a step height of 10 centimeters at a rate of 2.6 seconds per step and over a trapezoidal terrain with an inclination of ± 10 degrees at a rate of 1.6 seconds per step. On-line control was added to compute the trunk motion when an external force was applied to the robot [Takanishi et al. (1990*b*)]. The trunk motion was altered in real time from the pre-recorded trajectory to maintain the pre-recorded stance leg and ZMP trajectories. To return the trunk motion to the pre-recorded trajectory, the step length of the swing-leg was changed, which required the trajectory of the joints of the swing-leg to be changed on-line. The robot was able to adapt to forces up to 100N, applied for 0.3 seconds, while stepping at 0.64 seconds per step.

WL-12RV [Yamaguchi et al. (1993)] improved over WL-12RIII by adding another degree of freedom to the trunk, bringing the trunk to 3 degrees of freedom and the robot to a total of 9 degrees of freedom. The control method was similar to WL-12RIII. Limb and ZMP trajectories were precomputed. The trunk trajectory was then calculated to ensure the desired limb and ZMP trajectories. An added criterion was to minimize the yaw torque between the foot and ground. The trajectories were then played back with no other on-line feedback. It was reported that adding the third axis to the trunk motion improved the stability of the robot and allowed it to walk 50 percent faster with a 0.3m step at 0.54 seconds per step.

WL-12RVI and WL-12RVII [Yamaguchi et al. (1994, 1995, 1996)] were improved over WL-12RV by adding a compliant foot which could detect forces and moments. The previous control method was used for flat ground walking. However, when the robot encountered a small step 12 mm high, the foot detected the change in terrain and the preset joint trajectories were modified to account for the height change. After adapting to the terrain, the robot returned to its preset trajectories.

WL-13 and WL-14 [Yamaguchi & Takanishi (1997), Yamaguchi et al. (1998)] used antagonistic joints employing non-linear spring mechanisms to control both position and impedance of a joint. The control method was the same as WL-12RVII, except that during the swing phase the impedance of the joints of the swing-leg were reduced. With this approach, the required power was reduced as the natural pendulum dynamics of the swing-leg were exploited somewhat. However, since pre-recorded trajectories were used, the benefit of this technique depended on how well those trajectories matched the passive dynamic trajectories.

Wabian [Yamaguchi et al. (1999), Setiawan et al. (1999)] is a humanoid bipedal walking robot. An anthropomorphic body is used in lieu of the compensating trunk of the WL-12, WL-13 and WL-14 series. There are 14 degrees of freedom in the arms, 6 in the eyes and neck, 6 in the lower legs, 3 in the trunk, and 6 in the hands, for a total of 35 degrees of freedom. The walking control is similar to that used for WL-13 and WL-14 except that arm trajectories can now be used. Joint



trajectories of the arms and legs and ZMP trajectories are pre-recorded. Compensatory motions of the trunk are then determined using a dynamic model. The preset walking pattern is then played back using joint position control during walking. During swing the impedance of the joints in the swing-leg is lowered. Wabian can dynamically walk, dance in place, and be led by a human pushing on its arms.

In summary, Waseda's walking algorithms rely on playing back pre-recorded joint and trunk trajectories which produce the desired ZMP trajectory in simulation. In order to adapt to terrain changes or external disturbances, the desired trajectories are altered. The early robots all used rigid position control at the joint level. The most recent robots have a higher degree of compliance control. We believe the compliance can be exploited more fully to produce more natural-looking walking. However, this may require relaxing the requirement of precise trajectory tracking which is still at the heart of the controllers.

2.1.2 P2 and P3

P2 (and its successor P3) was developed by Kazuo Hirai, Masato Hirose, Yuji Haikawa, Toru Takenaka and colleagues at the Honda Wako Research Center from 1986 to the present [Hirai et al. (1998)]. The robot has 12 actuated degrees of freedom in the lower body - three in each hip, one in each knee, and two in each ankle. The robot is primarily controlled by playing back pre-recorded joint trajectories acquired from direct measurements of human subjects. Three additional controllers modify the trajectory in order to maintain balance in light of disturbances, terrain, or modeling errors. A ground reaction force controller modifies joint angle trajectories to achieve the desired zero moment point and thus conform to uneven terrain. A model ZMP controller shifts the desired ZMP by changing the ideal body trajectory when the robot is about to tip over. A foot landing position controller changes the stride length to compensate for changes in the body trajectory made by the model ZMP controller. With this control scheme the robot can walk fairly fast (1.1 meters per second) and walk over 10-degree inclines. The robot can also walk up and down stairs and turn in place.

Honda's control method is similar to the one used by the Waseda group with WL-10RD. The resultant walking appears a bit more natural, however, for several reasons. First, the trajectories were recorded by motion capture from a real human rather than derived through trial and error. Second, high speed was emphasized as a goal of the project. Third, by using force/torque sensors in the feet, the ankle was made compliant through the ground reaction force controller which allows the robot to conform to uneven surfaces.

2.1.3 Toddler

Toddler was developed by Tom Miller, Andrija Kun, P. Latham, and colleagues at the University of New Hampshire from 1994 to the present [Kun & Miller (1998, 1996), Miller (1994)]. The robot has ten actuated degrees of freedom - two on each hip, one on each knee, and two on each ankle. A gait generator plays back preprogrammed posture sequences which are initiated by sensory triggers. The desired posture is then modified by five CMAC neural networks. The first CMAC modifies the sideways (lateral) lean angle to correct for errors in the leg lift duration. The second CMAC modifies the forward/backward lean of the hips to keep the measured center of pressure in the middle of the foot. The third and fourth CMACs learn adjustments to the front/back and sideways desired lean angles so that the actual lean angles better match the desired posture. The final reference posture then is converted to desired joint angles with a simple inverse kinematics mapping that assumes the sagittal and frontal planes are decoupled. The fifth CMAC is used to change the lateral roll of the ankles during double support to keep the center of pressure in the middle of the foot during double support. The commanded joint angles are then modified by a reactive controller that uses feedback from gyroscopes and accelerometers to modify the desired joint angles to better track the desired lean angles and velocities. The final joint trajectories are tracked using PID servos.

This approach is appealing because it does not require precise dynamic models of the robot. Instead, the CMAC neural networks learn trajectory modifications based on the state of the robot

in order to achieve the desired posture trajectories. Once learning is done, if the CMAC weights are frozen, then the controller can be seen as having both feed-forward trajectories and reactive control. The reactive control can be stated qualitatively as follows: lean forward/backward and side to side to keep the center of pressure in the middle of the foot, and rotate the ankles to control lateral speed and to keep the center of pressure in the middle of the foot. The CMAC neural networks can be seen as automatic parameter tuners for these reactive controllers. The robot walks very slowly in the forward direction (maximum speed is 0.12 meters per second) due to its small stride length (9 centimeters maximum). However, it walks very dynamically in the frontal plane since the pendulum tipping dynamics dominate. It seems possible to extend the UNH approach in order to make the sagittal plane walking faster and more graceful.

2.1.4 Moscow State University Biped

The Moscow State University Biped was developed by A. Grishin, A. Formal'sky, A. Lensky, S. Zhitomirsky, V. Budanov and colleagues [Grishin et al. (1994)]. The robot was a planar biped having only two actuated degrees of freedom. The leg lengths were controlled by one actuator, such that the total length of the legs remained constant. The angle between the two legs was controlled by another actuator. Polynomial reference trajectories were determined for the leg lengths and hip angle. The parameters of the polynomials were chosen through iteration such that the resultant motion was consistent with the dynamic equations of motion. During walking the reference trajectories were then played back. Errors in the motion were compensated for by on-line control during double support. The double support trajectories and the time to transition to single support were modified so that the initial conditions of the robot at the beginning of the single support state would be closer to the pre-recorded trajectories.

Since this robot has point feet, it must walk dynamically and exhibits motion consistent with its natural dynamics. With only two actuated degrees of freedom, though, it appears rather unnatural. A four degree of freedom robot at Moscow State is currently working but has not been reported yet.

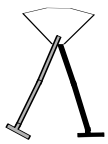
2.1.5 SD-2

SD-2, also called CURBI, was developed at Clemson University and Ohio State University by Yuan F. Zheng and colleagues. The robot had four degrees of freedom; two in the hips and two in the ankles. The control technique was play-back of pre-recorded trajectories that result in static walking [Golden & Zheng (1990), Zheng & Shen (1990)]. Pure playback was successful for flat ground and stairs. For sloped surfaces, the foot was fitted with force sensors to detect the slope blindly (without vision or previous information). The pre-recorded hip trajectories were then modified such that the robot leaned forward if going uphill and backward if going downhill. In this way, the other joint trajectories could remain unchanged while still walking statically.

[Yi & Zheng (1996)] investigated reduced ankle power. Due to reduced power, the robot could not accurately follow the pre-recorded ankle trajectory. A displacement was added to the hip trajectory to compensate for errors in the ankle trajectory. Essentially, forward speed and displacement were controlled by leaning forward or backward to change the center of mass location.

2.1.6 Biper

Biper was developed by Hirofumi Miura and Isao Shimoyama at the University of Tokyo in 1984 [Miura & Shimoyama (1984)]. Five different versions of the robot were developed with various degrees of freedom. They all used the same control method. The key to the control method was to assume small motions so that the equations of motion of the robot could be linearized, to allow for the use of linear control theory. First the equations of motion were computed and linearized. The equations of motion consist of both the continuous differential equations which determine motion during the single support phase and the discrete differential equations which describe step to step transition conditions. Next, joint trajectories were determined which were consistent with the equations of motion. The torque trajectories required to achieve the joint trajectories were then determined.



Error dynamics were then computed. Since the system is linear, the error dynamics are the same as the original dynamics, hence they need to be stabilized. During single support, joint torques were used to compensate for errors in the joint trajectories. Foot placement was used to stabilize the dynamics from step to step. The final dynamics were computed and shown to be stable, based on the final eigenvalues of the system.

This approach is one of the few which can be theoretically proven to result in stable walking. Unfortunately, because of the assumption of linearity, only small step sizes can be taken. With the exception of small strides, the Biper robots appear fairly natural since the natural pendulum dynamics are allowed to prevail. Trajectory errors are tolerated and compensated for by controlling speed via foot placement.

2.1.7 Meltran

Meltran was developed by Shuuji Kajita, Kazuo Tani, Akira Kobayashi, and Tomio Yamaura at the Mechanical Engineering Laboratory (MEL) in Tsukuba between 1990 and 1996 [Kajita & Tani (1996), Kajita & K.Tani (1996, 1995), Kajita et al. (1992), Kajita & Tani (1991*b*,*a*), Kajita et al. (1990)]. The robot is a planar bipedal walker with 6 actuated degrees of freedom. A simplified dynamic model was used which assumed that the legs were massless, the upper body was a rigid inertial element and the robot walked with the center of mass at a constant height. With these assumptions, the dynamics are linear and equations of motion can be determined in closed form. Height and pitch are controlled by servoing joint angles based on inverse kinematics. Ankle torque is used to compensate for modeling errors so that the forward velocity follows the estimated forward velocity based on the equations of motion with zero ankle torque. Forward speed is controlled from step to step by foot placement. With the simplified dynamic equations of motion, a quantity which remains constant during a stride can be determined. This constant is called the Orbital Energy. Foot placement is determined such that the Orbital Energy of the next step corresponds to the desired velocity. With this control technique, the robot walked at a speed of approximately 0.5 meters per second. Forward speed could be commanded in real time with a joystick. The algorithm was extended for walking over moderate steps. As long as the center of mass follows a line, the equations of motion remain linear and the orbital energy constant. Therefore only minor modifications needed to be made to the algorithm. Ground height was detected using a range sensor attached to the front of the robot.

MEL's control method was rather elegant, in that it allowed the natural dynamics of the robot to dictate the forward velocity rather than following precise trajectories. However, walking at a constant height is inefficient since kinetic energy decreases during the first half of the stride and increases during the second half while potential energy remains constant. Meltran also did not fully exploit its torque-controllable ankles. Ankle torque was used only to correct for modeling errors. It was not used for speed control, or energy injection at toe-off.

2.1.8 Timmy

Timmy was developed by Eric Dunn and Robert Howe at Harvard in 1994 [Dunn & Howe (1994, 1996)]. The robot is a planar walker with four degrees of freedom. The hips are driven with electric motors and the legs are driven with pneumatic cylinders. Dunn and Howe derived conditions for smooth exchange of support, with one solution being a constant height trajectory.

Dunn and Howe used a heuristic control approach. Speed was controlled by foot placement, based on a symmetry argument. Height was controlled by servoing leg length, based on inverse kinematics. Pitch was servoed using hip torque on the stance leg. The swing-leg was servoed to follow a cubic spline trajectory, ending with the desired step length.

Height, step length and speed could be changed by the user. The robot's top walking speed was approximately 0.3 meters per second with a step length of 20 centimeters. Because the robot had point feet, it appeared fairly natural, as the natural pendulum dynamics of the robot were exhibited.

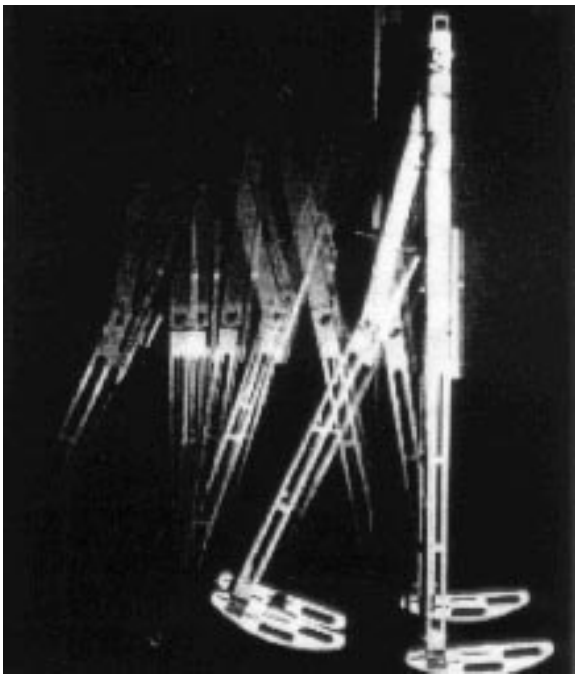


Figure 2-2: A passive dynamic walker developed by Andy Ruina's lab at Cornell University. The walker walks down shallow slopes with no sensors, computation, or actuation.

2.1.9 Powered Robot Summary

We see that previous powered bipeds fall into two broad categories: those which predominately use pre-recorded trajectory playback and those which predominantely use an algorithmic controller.

The control techniques used in this thesis fall in the algorithmic category. The algorithms presented will most closely match those of Dunn and Howe's Timmy and Kajita's Meltran with three major differences:

1. Instead of using a constant height trajectory, we straighten the leg during stance so that a compass-like gait is achieved. The exact height trajectory is not important.

2. Instead of controlling speed by only foot placement, we also use support ankle torque, body pitch, and toe-off ankle torque.

3. Instead of rigidly position-controlling the swing-leg, in Chapter 5 we let it swing freely, exploiting its natural dynamics. These differences result in walking that is faster, more robust, and appears more natural.

2.2 Passive Dynamic Bipedal Walkers

Passive dynamic walkers are a class of mechanisms which can walk smoothly and stably down a shallow incline with no sensing, control, or actuation. These passive dynamic walkers exhibit a steady walking cycle which is smooth and efficient and appears incredibly natural.

McGeer was the first to study these robots in depth [McGeer (1990*a*)]. He developed a method for analyzing stability, and implemented a few different designs. Several groups have expanded on McGeer's work to further analyze passive dynamic walkers and find other walking machines which fall into this class. Coleman, Ruina and colleagues and Thuilot, Goswami and colleagues studied bifurcation and chaos of passive gaits due to parameter variation [Coleman et al. (1997*a*), Thuilot et al. (1997)]. Ruina's group at Cornell University have discovered and built several different passive

walkers, one of which is shown in Figure 2-2. Smith and Berkemeier discovered and analyzed a passive quadrupedal walking mechanism [Smith & Berkemeier (1997)]. Adolfsson, Dankowicz, and colleagues started examining 3D passive walking but retained the planarizing effect of overlapping feet [Adolfsson et al. (1998)]. Fowble and Kuo showed that a simple control law consisting of lateral foot placement was enough to stabilize a 3D passive bipedal walker [Fowble & Kuo (1996)].

While there have been many different studies on passive dynamic walking, they all employ a variant of McGeer's original method of analysis:

- Model the system under study and derive equations of motion. These equations include continuous differential equations governing the continuous motion during a support phase; discrete impact equations governing the impact during exchange of support; and the conditions which cause a discrete impact.
- Solve the continuous equations and combine with the impact equations, resulting in a single equation, called the stride function. The stride function is a Poincare map relating the state during one part of a step with the state during the same part of the the next step.
- Find fixed points of the stride function and linearize it about these fixed points, resulting in a linear Jacobian matrix.
- Compute eigenvectors and eigenvalues of this Jacobian matrix to determine modes and stability of the system.
- Examine effects of parameter variation on the number of fixed points and their stability. This leads to bifurcation diagrams.
- Determine the region of convergence and study how parameter variation effects the region of convergence.
- Create an experimental machine. Tweak its parameters based on numerical and real experimentation.
- Draw conclusions.

This method can be applied analytically to simple systems, but requires numerical simulation and experimentation for more complicated systems. Fixed points, corresponding to periodic gait cycles, can be found through a numerical search by starting the simulation with various initial conditions and examining convergent trajectories. Jacobian matrices can be experimentally determined by disturbing the simulation from the periodic gait and recording state/next-state pairs. A number of such pairs can be obtained under various disturbances and used to solve for a linear matrix fit of the Jacobian.

Many in the passive dynamic walking community have suggested ways in which a passive machine could be equipped with actuators to increase its performance. McGeer suggested using plantar flexion to enable a passive walker to walk on flat ground or uphill. Goswami, et. al [Goswami et al. (1997)] presented a simple control law for increasing the region of convergence of a two link passive walker. Van der Linde [der Linde (1998)] built a machine which walks on level ground by pumping energy each step into a passive mechanism. Camp [Camp (1997)] showed that open loop ankle torque during toe-off results in stable walking on flat ground. However, no one has yet built a passive dynamic walking machine which has the capabilities of powered robots, such as being able to walk at various speeds on various terrain.

Passive dynamic research has motivated much of the work in this thesis, particularly the approach described in Chapter 5. However, we approach the problem from the opposite direction. Instead of starting with a fully passive machine, we exploit the natural dynamics of a powered bipedal robot in order to simplify the control and make its gait appear more natural. In this way, we can use a mechanical design that is more similar to animals in structure than the passive dynamic walkers and also may be able to achieve a wider range of walking capabilities. Chapter 5 describes our approach which combines powered and passive bipedal walking.

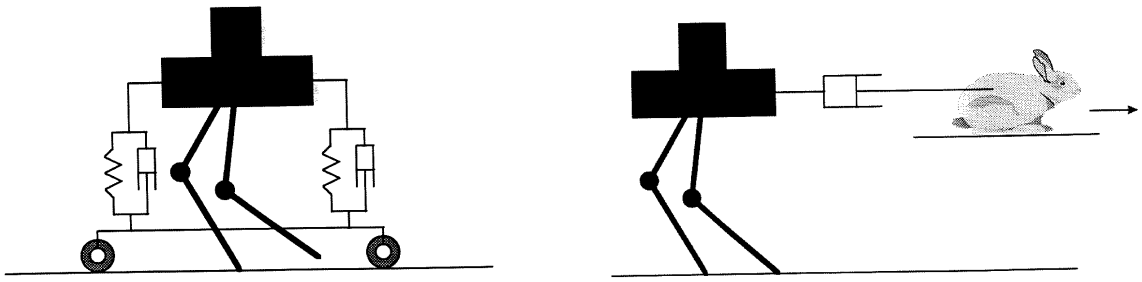


Figure 2-3: Virtual Model Control applied to Spring Turkey. A virtual "granny walker" (left) is used to support the robot and keep it balanced. A damper connected to a "dog-track bunny" (right) is used to control forward speed during double support. A virtual "reciprocating gait orthosis" (not shown) is used to make the swing-leg mirror the stance leg.

2.3 Virtual Model Control

A major goal of this thesis is to keep the control algorithms easy to understand. This is achieved by making use of intuitive control strategies and implementing them in an intuitive manner. Virtual Model Control [Pratt (1995)] is a tool which helps achieve this goal.

Virtual Model Control is a language for describing interactive force behaviors. This control technique uses simulations of virtual mechanical components, attached between two point on the robot, to generate real actuator torques (or forces). For a serial link chain, the transformation is simply,

$$\tau = J^T F \tag{2.1}$$

where F is the force produced by the virtual component, J is the Jacobian relating the two attachment frames of the virtual component, and τ is the joint torques.

The applied joint torques create the same effect that the virtual components would have created, had they physically existed, thereby creating the illusion that the simulated components are connected to the real robot. Such components can include simple springs, dampers, dashpots, masses, latches, bearings, non-linear potential and dissipative fields, or any other component which outputs a force based on its state. Virtual components can even contain adaptive and learning elements [Pratt (1994)]. Virtual Model Control borrows ideas from Virtual Reality, Hybrid Position-Force Control [Raibert & Craig (1981)], Stiffness Control [Salisbury (Dec. 1980)], Impedance Control [Hogan (1985)], and the Operational Space Formulation [Khatib (1986)].

Virtual Model Control was used to control Spring Turkey [Pratt et al. (1997)], a planar bipedal walking robot, and a simulated hexapod [Torres (1996)]. The hexapod had 18 degrees of freedom (2 in each hip and one in each ankle). It could walk in any direction while turning, traversing terrain, and balancing an inverted pendulum on its back. Spring Turkey had four degrees of freedom. There was an actuated hip and knee on each leg. The robot had no feet or ankles, and could not walk very robustly. Figure 2-3 shows the virtual components used to control Spring Turkey. A virtual "granny walker" is used to support the robot and keep it balanced. The springs push back in order to maintain height and pitch while the dampers prevent oscillation. A damper connected to a "dog-track bunny" is used to control forward speed during double support. If the robot is traveling at a different speed than the bunny, the damper applies a correcting force to the robot. A virtual "reciprocating gait orthosis" is used to make the swing-leg mirror the stance leg by applying correcting forces to the swing-leg. A state machine was used to cycle in the various components depending on the current phase of the gait. Using these simple components Spring Turkey was able to walk at $0.5\frac{m}{s}$. In this thesis, Virtual Model Control is used as one of the tools for controlling Spring Flamingo.

2.4 Series Elastic Actuators

In order to exploit the natural dynamics of the bipedal mechanism, it is important that the inertia and friction of the actuators does not interfere with the dynamics of the robot. In other words, it is important to be able to make the impedance of the actuators low. However, this is difficult to do with standard electric servos.

Electric motors work at high speed and low torque while muscle works at low speed and high force. To make motors behave more like muscles, they need to be geared down, which increases their friction and reflected inertia. Therefore, the dynamics of most robots are dominated by their actuators instead of the dynamics of their structure. Feedback control is typically used to overcome the large apparent inertia in making the joints track trajectories. The actuators can therefore be seen as rigid position sources.

However, rigid trajectory tracking is usually unnatural as the prescribed trajectories are based on cubic splines which makes the motion look "too good" and hence "robotic". It is inefficient since large torques are required whenever the trajectories deviate too much from the natural dynamic trajectories. It is not robust since it requires accurate models of the environment whenever there is contact between the robot and the environment. Such contacts occur on every step in bipedal walking. Robots that follow rigid trajectories, such as Waseda's earlier bipeds, cannot tolerate bumps in the ground as little as a centimeter in height.

Most of the more successful bipedal walking robots employ compliance in one form or another. Waseda's Wabian uses opposing actuators with nonlinear springs in order to vary joint impedance. Honda's P2 and P3 use force/torque sensors on springy feet and vary the joint trajectories in order for the foot to comply with the ground. Several robots have point feet or limited ankle torque. This effectively gives the ankle compliance (zero stiffness in the case of point feet). At the extreme, the passive dynamic walkers have high compliance (zero impedance) at all of their joints.

Ideally, a bipedal robot would have perfect torque sources at its joints. In that case, each joint could vary in compliance from being completely limp (zero torque) to being a rigid position source using high gain feedback. Perfect torque sources are impossible to achieve. However, we believe bipedal walking only requires moderate stiffnesses compared to industrial robot arms. Therefore, the bandwidth of the torque sources can be limited.

Series Elastic Actuators [Pratt & Williamson (1995), Williamson (1995), Robinson et al. (1999)] are a good solution for approximating torque sources at the joints. They are force controllable actuators with moderate dynamic range (ratio of largest applicable force to smallest resolvable force) and moderate bandwidth. The underlying idea in Series Elastic Actuators is to purposefully place an elastic element (spring) in series with the output of the actuator, after the gear reduction (see Figure 2-4). By measuring the spring deflection, the output force can be computed using Hooke's Law. The motor is then servoed such that the actual output force matches the desired force. The more compliant the elastic element, the more the inertia and stiction of the motor and gear train can be masked, resulting in low output impedance.

Series Elastic Actuators are used in all of the Leg Laboratory's bipedal walking robots and our new running robots. They allow for the implementation of Virtual Model Control, which requires control of joint torques. The actuators used in Spring Flamingo have a dynamic range (ratio of largest applicable force to smallest resolvable force) of over 100 and a bandwidth of 15 Hz. Our most recent implementation has a dynamic range of over 300 and a bandwidth of 45 Hz.

2.5 Conclusions

There are many examples of powered bipedal walking robots and passive bipedal robots. While researchers have recommended combining the two approaches for some time, this thesis is the first example we are aware of in which an experimental robot was constructed that explicitly takes advantage of the benefits of both powered and passive robots. To do so requires actuators with negligible dynamics compared to the natural dynamics of the robot. We use Series Elastic Actuators which have low impedance and friction. They have sufficient bandwidth and dynamic range for

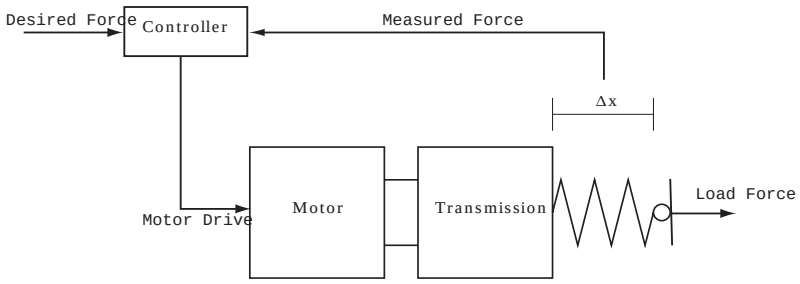
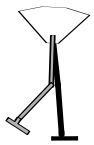


Figure 2-4: Schematic of a Series Elastic Actuator. A spring is placed between the transmission and the load. By measuring spring deflection, the load force can be estimated. A controller drives the motor based on the error between the desired force and the measured force. This technique allows for good force fidelity and bandwidth and also provides shock tolerance since the transmission is protected from shock loads by the spring.

bipedal walking and are sufficiently adequate as torque sources to use in implementing Virtual Model Control, one of the tools used in the algorithm development of Chapters 4 to 8.

Before describing the algorithms, however, we first introduce some simple models of bipedal walking in the next chapter and examine limits to maximum walking speed in Chapter 6. Understanding the walking models will assist in understanding why the algorithms work. Understanding limits to maximum walking speed will clarify the motivation for the approach used to control the robot to walk fast in Chapter 7.



Chapter 3

Simple Models of Bipedal Walking

Before jumping into bipedal walking algorithms, let's start by examining simple planar models of bipedal walking. The models are simple so that we can draw intuition from them without getting bogged down in the mathematics. We start with a simple inverted pendulum and progress to more detailed models. By gaining a firm understanding of the dynamics of bipedal walking, we will be able to better understand why the walking algorithms in Chapters 4 to 8 work.

3.1 Center of Mass and Center of Pressure

Understanding the concepts of center of mass and center of pressure will help us to understand simple models of bipedal walking.

The center of mass is the distance-weighted average location of the individual mass particles in the robot:

$$\vec{P}_{com} = \frac{\sum_i \vec{P}_{m_i} m_i}{\sum_i m_i} \tag{3.1}$$

where $\vec{P_{m_i}}$ is the location of mass particle i, and m_i is the mass of particle i.

The center of pressure (Figure 3-1) is the distance-weighted average location of the individual pressures on the foot:

$$\vec{P}_{cop} = \frac{\sum_i \vec{P}_{p_i} p_i}{\sum_i p_i} \tag{3.2}$$

Figure 3-1: Picture illustrating the Center of Pressure. The center of pressure (circle) is the distance weighted average of the pressures on the feet (arrows).

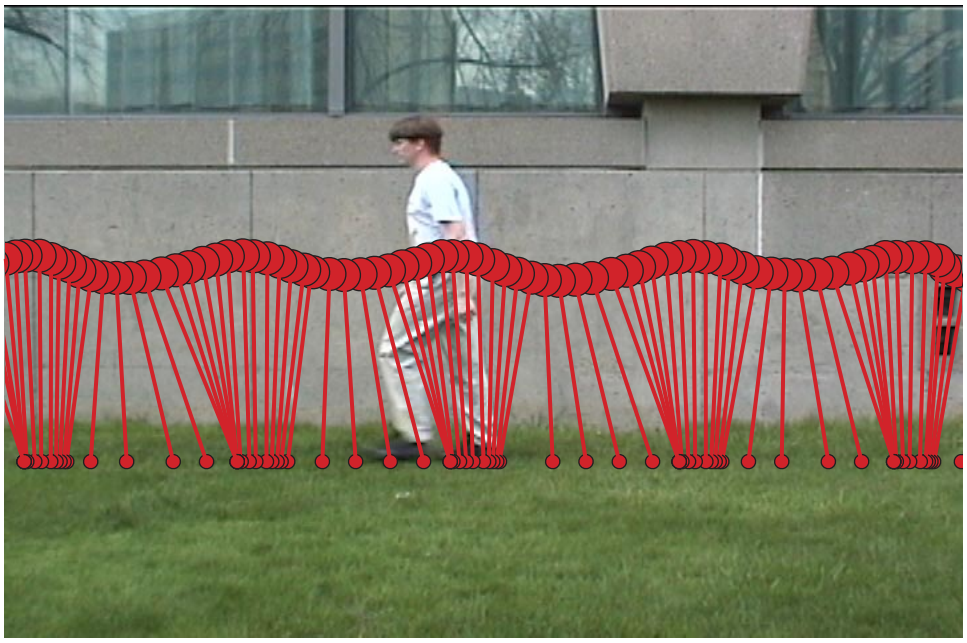


Figure 3-2: Picture illustrating the similarity between walking and an inverted pendulum. Pendulum pivot point is placed approximately at the center of pressure on the foot. Pendulum mass is placed approximately at the center of mass.

where $\vec{P_{p_i}}$ is the location of ground pressure i, and p_i is the value of its pressure.

We will show that the dynamics of walking is very similar to the dynamics of an inverted pendulum if the center of mass and the center of pressure are compared to the mass of the pendulum and the pivot point of the pendulum.

3.2 Inverted Pendulum Models

Figure 3-2 is a picture illustrating the similarity between walking and an inverted pendulum. The picture was created frame by frame by "eyeballing" video of the author as he walked across the screen. A pendulum is superimposed on the image with the pivot point placed approximately at the center of pressure on the foot and the mass of the pendulum placed approximately at the center of mass of the author. We see that there is much similarity and therefore, understanding the dynamics of an inverted pendulum will help us understand the dynamics of walking.

Figure 3-3 shows an inverted pendulum of length l with a point mass, m, at its end. It is in a gravity field and has a frictionless pin joint at its base. The mass corresponds to the body mass of a bipedal walker and the pendulum to its leg. This is admittedly a limited model of bipedal walking, yet it can give us insight and is the base of more detailed models.

We know from physical intuition that the pendulum has an equilibrium point in the straight up position and will accelerate in the direction of whichever side it is on. The further from vertical it is, the faster it will accelerate. Mathematically, this can be verified by deriving the equations of motion of the pendulum. To do so, we note that the there is no torque at the pivot joint, so the only thing affecting the angular momentum about the pivot is the torque due to gravity. The angular momentum about the pivot is

$$H_0 = ml^2\dot{\theta}_1 \tag{3.3}$$

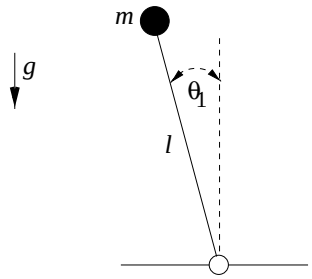


Figure 3-3: Simple pendulum model of bipedal walking during single support. m corresponds to body mass, l to leg length, and θ_1 to stance leg angle.

The change in angular momentum due to gravity is

$$\dot{H}_0 = mgl \sin \theta_1 \tag{3.4}$$

Differentiating H_0 and substituting, we get the equations of motion of a pendulum:

$$\ddot{\theta_1} = \frac{g}{l} \sin \theta_1 \tag{3.5}$$

The insight we can draw from the inverted pendulum model is as follows. Suppose the mass is traveling from left to right. If the mass is on the left of the pivot point, it will slow down as it rises, converting kinetic energy into gravitational potential energy. If it is on the right of the pivot point it will speed up, converting gravitational potential energy back into kinetic energy.

Now suppose we add a linear actuator along the length of the pendulum, as shown in Figure 3-4. The force on the point mass lies along the length of the leg. The acceleration of the mass in the radial direction depends on the actuator force, F, the gravitational force, and the fictitious centrifugal force due to the rotation of the pendulum.

The dynamics in the rotational direction can be computed as above, except this time the derivative of angular momentum will have an additional term due to the possibility of the length of the pendulum changing. The linear dynamics can be derived by noting that the forces along the length of the pendulum include the component due to the actuator (F), the component due to gravity ($mg \cos \theta_1$), and the component due to ficticious centrifical forces ($ml\dot{\theta}_1^2$).

The result is that the rotational dynamics of the pendulum are not directly influenced by F. However, the length and length derivative do play a role:

$$ml^2\ddot{\theta_1} = mgl \sin \theta_1 - 2ml\dot{l}\dot{\theta}_1 \tag{3.6}$$

$$m\ddot{l} = F - mg \cos \theta_1 + ml\dot{\theta_1}^2 \tag{3.7}$$

The only effect the leg actuator has on the rotational dynamics is that it can change the pendulum (leg) length, l and length derivative, $\dot{l}$. Pulling in the mass will accelerate the rotation while extending the mass will decelerate it just as it does for a child pulling in or extending their legs while twisting on a tire swing.

Since only gravity affects the angular momentum, the pendulum is not stabilizable in the upright position by the actuator. Also, assuming l is restricted to some bound and must always be greater than zero, the usefulness of moving the mass in and out to control angular rotation of a biped is limited during normal walking. Thus, the insight of the inverted pendulum extends to the case of the inverted pendulum with a linear actuator. If the mass is traveling from left to right then it will slow down if on the left side of the pivot point or speed up if on the right side.

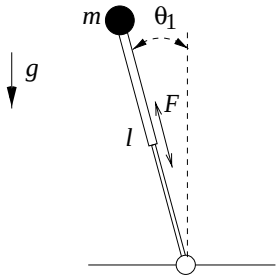


Figure 3-4: Simple inverted pendulum with linear actuator. The rotational dynamics are independent of the actuator force and only depend on g, l, and θ_1.

3.3 Multi Joint Pendulum Model

Now suppose instead of a linear actuator we have one or more rotational joints, as in the left side of Figure 3-5 where we have two actuated joints. We still assume that the only mass is a point mass at the end of the last link. We can show that the dynamics of this model are identical to those of the pendulum with a linear actuator (right side of Figure 3-5) provided that the model is not in the configuration of all joints being straight.

If there is only one actuated joint, then a torque of τ on the model on the left side of Figure 3-5 will result in dynamics identical to applying $F = \frac{\tau}{R_{lever}}$ to the model on the right side, where R_{lever} is the perpendicular lever arm distance from the joint to the line between the point mass and ground pivot.

With more than one actuated joint, the two systems are still equivalent in the l and θ_1 directions. However, the one on the left has redundant degrees of freedom and thus can take on an infinite number of configurations for each l and θ_1. The equations of motion in the l and θ_1 directions are

$$ml^2\ddot{\theta}_1 = mgl\sin\theta_1 - 2ml\dot{l}\dot{\theta}_1 \tag{3.8}$$

$$m\ddot{l} = f(\tau_2, \tau_3, \theta_1, \theta_2, \theta_3) - mg\cos\theta_1 + ml\dot{\theta}_1^2 \tag{3.9}$$

where f is the nonlinear mapping from joint torques to effective force on the point mass along the direction from the ground pivot to the mass. Note the similarity of these equations and Equations 3.6 and 3.7. The rotational dynamics are still independent of the torques of the actuators and are only influenced by the indirect changes in l and $\dot{l}$. Since gravity is still the only force that affects the angular momentum about the pivot point, the system is still not controllable.

Therefore, the insight of the inverted pendulum extends to this case too. If the mass is traveling from left to right then it will slow down if on the left side of the pivot point or speed up if on the right side. It is important to stress that the model has only a single point mass and no inertia about its center of mass. As we will see when we add inertia to the mass, the radial and rotational dynamics may become coupled, as the total momentum will include both the momentum of the mass around the pivot point and the momentum of the inertia about the center of mass. Therefore, the system will become controllable. First though, we show how a foot and actuated ankle changes the dynamics such that the system is controllable.

3.4 Adding a Foot

We see that with a point foot (limp pivot about the ground) and a point mass body, the dynamics of the robot in the rotational direction (θ_1) are independent of the robot mechanism and nothing can be done to control rotational speed. If we add a foot, however, we can apply torques to the ground, effectively changing the pendulum pivot point, and therefore affect forward speed.

The diagram on the left of Figure 3-6 is a model with actuated ankle and knee joints and a foot.

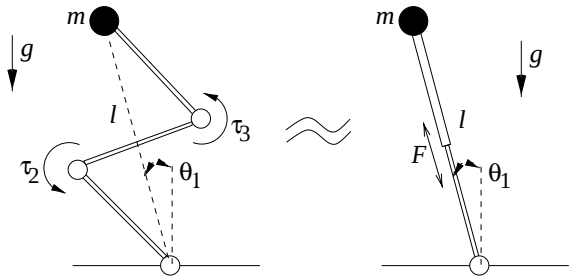


Figure 3-5: Multi-joint model with point mass and point foot(left) is equivalent to the linear actuator model (right) as long as both joints are not straight. Rotational dynamics are independent of actuator torques (τ_2 and τ_3) and depend only on g, l, and θ_1.

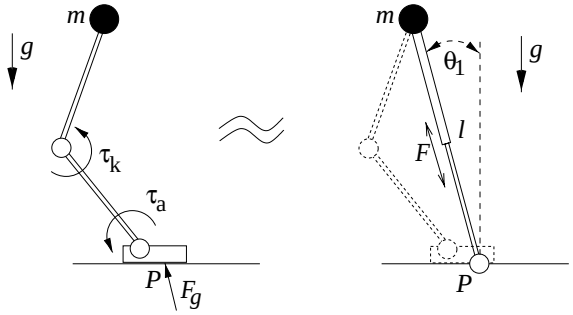


Figure 3-6: Walking model with point mass body and foot with actuated knee and ankle (left). P is the location of the center of pressure on the foot. The model with foot is dynamically equivalent to the inverted pendulum with linear actuator model (right). However, the foot allows us to effectively change the location of the pendulum pivot point, as long as it stays inside the foot projection.

Point P is the location of the center of pressure on the foot, with F_g being the ground reaction force. P must lie somewhere on or inside the projection of the foot on the ground. P is the point at which if the foot was replaced with a point contact with no torque, the dynamics of the system would stay the same.

We can show that the system on the left of Figure 3-6 is dynamically equivalent to the one on the right, which has been analyzed above. Its dynamics can still be written as in Equations 3.8 and 3.9. However, now we have the freedom to move the location of the effective point contact with the ground (center of pressure) as long as we keep it under the foot.

From Equation 3.8, we know that if the center of pressure is to the left of the point mass then the mass will accelerate to the right. If the center of pressure is to the right of the point mass, then the mass will accelerate to the left.

We can arbitrarily control the location of the center of pressure through joint torques, as long as it stays within the projection of the foot. By moving the center of pressure, we have some freedom to control the rotational dynamics. We can change the rotational velocity and even balance the point mass over the foot.

3.5 Flywheel Models

In all of the above models, we assumed that the mass was concentrated at a point. Therefore, no moments could be applied to the mass. To maintain zero torque about the ground pivot (or center of pressure if a foot was available), the force on the body was constrained to lie along the line joining the ground pivot and the point mass.

In the model on the left of Figure 3-7, we give inertia J to the mass and put a rotational joint at

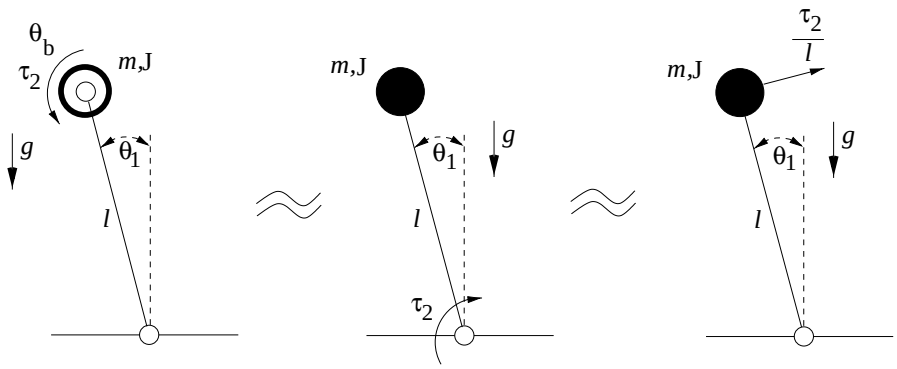


Figure 3-7: Inverted pendulum model with flywheel of mass m and inertia J (left). Applying a torque of τ_2 on the flywheel is equivalent to applying an ankle torque of τ_2 on the model in the middle or an external force of $\frac{\tau_2}{l}$ on the model on the right.

the center of mass, creating a "flywheel". The resulting equations of motion are

$$ml^2\ddot{\theta}_1 = mgl\sin\theta_1 - \tau_2 \tag{3.10}$$
$$J\ddot{\theta}_b = \tau_2 \tag{3.11}$$

where θ_b is the angle of the flywheel with respect to the ground, $\theta_b = \theta_1 + \theta_2$. In the middle diagram of Figure 3-7 we remove the flywheel, fixing the inertia to the link and instead have an actuated ankle with torque τ_2. In the diagram on the right we also fix the flywheel but instead apply an external force of $\frac{\tau_2}{l}$ from the world to the mass, perpendicularly to the leg.

The inverted pendulum dynamics of all three systems in Figure 3-7 are equivalent, governed by Equation 3.10. Thus if we ignore θ_b, the angle of the flywheel on the left, all three systems are dynamically equivalent.

From the above we see that a flywheel can be used to control the rotational dynamics of an inverted pendulum. By adding a linear actuator, (Figure 3-8), all three motions of the inertia are controllable. The equations of motion of the model on the left are

$$ml^2\ddot{\theta}_1 = mgl\sin\theta_1 - 2ml\dot{l}\dot{\theta}_1 - \tau_2 \tag{3.12}$$
$$m\ddot{l} = ml\dot{\theta_1}^2 - mg\cos\theta_1 + F \tag{3.13}$$
$$J\ddot{\theta}_b = \tau_2 \tag{3.14}$$

The schematic on the right of Figure 3-8 shows a free body diagram of the inertia. F_r, $F_\perp$, and τ_b are external forces and torques on the body. The two diagrams in Figure 3-8 are dynamically equivalent if the forces on the free body diagram on the right are

$$F_\perp = \frac{\tau_2}{l}, \quad F_r = F, \quad \tau_b = \tau_2 \tag{3.15}$$

It is important to understand the difference between τ_b and τ_2. τ_2 is an internal torque applied between the pendulum link and the flywheel. τ_b is an external torque applied between the world and the flywheel. Applying τ_2 between the pendulum link and the flywheel is dynamically equivalent to applying τ_b and $F_\perp$ between the world and the flywheel.

F_r can be independently applied, while τ_b and $F_\perp$ are coupled. This coupling is due to the requirement that their combined moment about the pendulum pivot point must be zero, $\tau_b + F_\perp l = 0$, which is also equivalent to saying that the angular momentum about the pivot point is only affected by gravity.

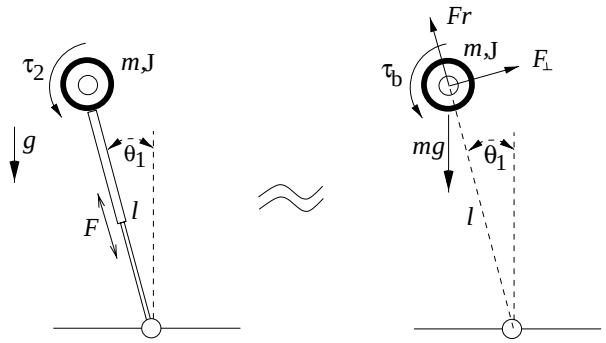


Figure 3-8: Inverted pendulum with flywheel and linear actuator (left). Free body diagram (right) showing external force and torque equivalents. The two systems are dynamically equivalent if $F_\perp = \frac{\tau_2}{l}$, $F_r = F$, and $\tau_b = \tau_2$.

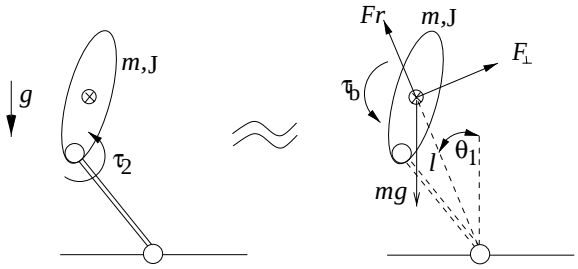


Figure 3-9: Acrobot model with only one actuated degree of freedom and inertial body. The system is controllable as hip torque, τ_2 affects all three dynamic variables.

3.6 Acrobot Model

In the previous section we saw that a flywheel can be used to arbitrarily control the rotational motion of an inverted pendulum. Of course, bipeds do not have flywheels on them. Many joints have range of motion limits which prevent them from spinning up like flywheels. Others joints, such as shoulders may be used to "windmill" limbs but the maximum rotational speed is quickly achieved. While it might be interesting to put a flywheel on a bipedal walking robot, we do not consider it in this thesis as it is biologically implausible. We use the flywheel example only for the basis of understanding further models.

Now that we understand how flywheel torques can affect the dynamics, we take a step back and analyze the Acrobot model (Figure 3-9) which gets its name from an acrobat on a high bar. This model has been used as a demonstration system for many swing and and balancing control techniques [Spong (1994), Boone (1997)]. The limp pivot joint represents the hands which cannot apply torque to the bar. The actuated joint represents the shoulder and waist joints which can apply torque τ_2. In our case, the lower limp joint represents point foot contact with the ground, while the upper joint represents a hip or knee joint.

As long as the actuated joint of the acrobot is bent, the equations of motion are

$$ml^2\ddot{\theta}_1 = mgl\sin\theta_1 - 2ml\dot{l}\dot{\theta}_1 - \tau_2 \tag{3.16}$$

$$m\ddot{l} = ml\dot{\theta}_1^{\,2} - mg\cos\theta_1 + \frac{\tau_2}{R_{lever}} \tag{3.17}$$

$$J\ddot{\theta}_b = \tau_2 \tag{3.18}$$

where R_{lever} is the perpendicular lever arm distance from the joint to the line between the center of mass and ground pivot. On the right of Figure 3-9 is an equivalent free body diagram of the acrobot

with dynamical equations

$$ml^2\ddot{\theta}_1 = mgl\sin\theta_1 - 2ml\dot{l}\dot{\theta}_1 - lF_\perp \tag{3.19}$$

$$m\ddot{l} = ml\dot{\theta_1}^2 - mg\cos\theta_1 + F_r \tag{3.20}$$

$$J\ddot{\theta}_b = \tau_b \tag{3.21}$$

where

$$F_\perp = \frac{\tau_2}{l}, \quad F_r = \frac{\tau_2}{R_{lever}}, \quad \tau_b = \tau_2 \tag{3.22}$$

We see that τ_2 affects all three degrees of freedom of the acrobot. Indeed, the acrobot is controllable about its equilibrium positions in which the center of mass lies above the pivot ($\theta_1 = 0$). In order to affect the rotational inverted pendulum dynamics, a torque (τ_b) must be applied to the body. This torque effectively creates the perpendicular force on the center of mass (F_{perp}):

$$F_\perp = \frac{\tau_b}{l} \tag{3.23}$$

This relationship is a result of the pivot joint having zero torque. Equivalently, we still have the situation where only gravity affects the angular momentum about the pivot point ($\dot{H} = mgl\sin\theta_1$), but now the angular momentum includes the momentum of the body about its center of mass, $J\dot{\theta}_b$.

3.7 Massless Leg Biped Model

We are now ready to examine a full, albeit simplified, bipedal model. Figure 3-10 shows a model of a bipedal walker in the single support phase with massless legs and no swing-leg. The diagram on the right shows a free body diagram with the equivalent forces and torques on the body inertia.

The equations of motion of the biped are

$$ml^2\ddot{\theta}_1 = mgl\sin\theta_1 - 2ml\dot{l}\dot{\theta}_1 - lF_\perp \tag{3.24}$$

$$m\ddot{l} = ml\dot{\theta_1}^2 - mg\cos\theta_1 + F_r \tag{3.25}$$

$$J\ddot{\theta}_b = \tau_b \tag{3.26}$$

where

$$F_\perp = f_1(\tau_a, \tau_k, \tau_h, \theta_a, \theta_k, \theta_h), \quad F_r = f_2(\tau_a, \tau_k, \tau_h, \theta_a, \theta_k, \theta_h), \quad \tau_b = f_3(\tau_a, \tau_k, \tau_h, \theta_a, \theta_k, \theta_h) \tag{3.27}$$

where f_1, f_2, f_3 are nonlinear mappings from joint torques to forces between the center of pressure and the center of mass. These mappings are complicated functions, but for now they can be ignored. What is important is the constraints on τ_b and $F_\perp$. From previous discussions, we have shown that F_r can be arbitrarily controlled but that τ_b and $F_\perp$ are related by

$$F_\perp = \frac{\tau_b}{l} \tag{3.28}$$

so that the net torque at the center of pressure (point P) is zero. Therefore, if there is a torque on the body, τ_b, then there will also be an induced force $F_\perp$ perpendicular to the line from the center of pressure to the center of mass. We can rewrite Equation 3.24 as

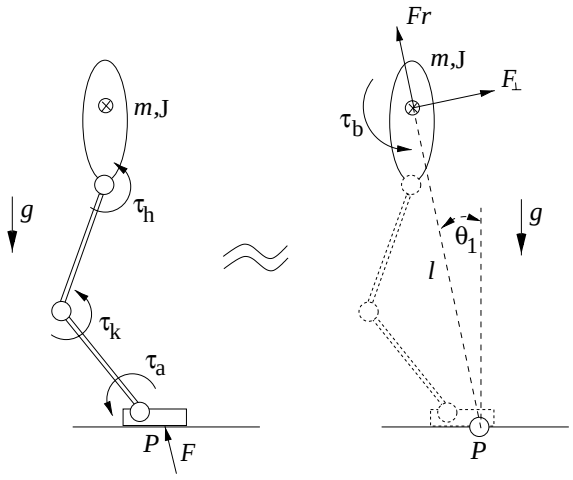


Figure 3-10: Massless leg biped model (left) and equivalent free body diagram (right). The biped has body mass m and inertia J. Point P is the location of the center of pressure on the foot. The two systems are equivalent through a mapping between joint torques and free body forces and moment. Due to the constraint of zero moment about the center of pressure (also called Zero Moment Point), we get the constraint on the free body diagram forces of $F_\perp = \frac{\tau_b}{l}$.

$$ml^2\ddot{\theta}_1 = mgl\sin\theta_1 - 2ml\dot{l}\dot{\theta}_1 - J\ddot{\theta}_b \tag{3.29}$$

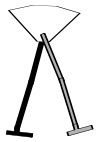
Examining this equation, we see that there are three ways that we can affect the rotational dynamics of the biped about its center of pressure. The first and most effective is to change the location of the center of pressure, P, thereby directly redefining the angle θ_1. By moving the center of pressure, we can control the forward speed, but only to the extent that the center of pressure remains inside the foot. The second method is to rotationally accelerate the inertia of the body (pitch forward or backward). The third and least effective is to pull in or push out the mass (change l).

In normal walking, the body of the robot does not experience significant rotational acceleration ($\ddot{\theta}_b \approx 0$), i.e. $F_\perp \approx 0$. Additionally, the length of the leg remains fairly constant ($\dot{l} \approx 0$). Therefore, the rotational dynamics of the robot about its center of pressure is governed by the pendulum equation $\ddot{\theta}_1 = \frac{g}{l}\sin\theta_1$. Thus, changing the effective rotation point of the pendulum dynamics by moving the location of the center of pressure (point P) is the most effective way to control the forward velocity of a biped during single support.

3.8 A Word about Singularities

In all of the examples above that compared systems involving rotational actuators with those involving translational actuators (Figures 3-5, 3-6, 3-9, 3-10), there was a caveat: the systems are equivalent as long as the rotational joints are not straight, thus avoiding singular configurations. There are many mathematical ways to deal with singularities but for the present purposes we are trying to understand the various bipedal models. The following will suffice to ensure that all the previous comparisons are *exactly* dynamically equivalent.

The model comparisons are all dynamically equivalent, even with the joints straight, provided that both models retain the restraining structure of the robot. For example, the two bipedal models in Figure 3-10 are exactly equivalent as long as the model on the right contains the effects of the leg and foot links. In other words, suppose we add to the robot (left), between the center of pressure on the foot and the center of mass on the body, an unconstrained three degree of freedom actuation



system that applies F_r, τ_b, and $F_\perp$ to the body (with $F_\perp = \frac{\tau_b}{l}$). If we keep the leg links (and their retaining structure) but apply zero torque at each rotational joint, then the two systems will be exactly dynamically equivalent.

If the leg is straight, then there can be no velocity in the direction of the leg. This is true in *both* models. For example, suppose that the center of pressure is at the ankle and the leg is straight. Then no matter what F_r is in the right diagram of Figure 3-10, it will have no dynamical effect on the system. If the system was really built, any F_r would simply put an internal load on the leg links. In fact, the corresponding torques in the left diagram required to achieve any F_r in the right diagram are all zero in this singular configuration. Mathematically, this corresponds to an infinite number of solutions for F_r in equation 3.7.

3.9 Planar Biped with Distributed Mass

All of the previous models assumed that the robot had either a point mass or a single inertial element. Here we consider the effects of distributed mass. The previous analysis dealing with rotational pendulum dynamics can be summarized by stating that only the torque due to gravity will affect the angular momentum of the robot about its center of pressure on the foot:

$$\dot{H}_{tot} = mgl \sin\theta_1 \tag{3.30}$$

The angular momentum of the robot consists of the angular momentum of its center of mass rotating about the center of pressure on its foot plus the angular momentum of internal masses rotating about the center of mass:

$$H_{tot} = H_0 + H_{cm} = ml^2\dot{\theta}_1 + H_{cm} \tag{3.31}$$

Differentiating, we get

$$\dot{H}_{tot} = ml^2\ddot{\theta}_1 + 2ml\dot{l}\dot{\theta}_1 + \dot{H}_{cm} \tag{3.32}$$

For example, with the model in Figure 3-10, we have $\dot{H}_{cm} = J\ddot{\theta}_b$ so that

$$ml^2\ddot{\theta}_1 = mgl \sin\theta_1 - 2ml\dot{l}\dot{\theta}_1 - J\ddot{\theta}_b \tag{3.33}$$

Since $J\ddot{\theta}_b = \tau_b = lF_\perp$, we see that Equation 3.33 is the same as Equation 3.24.

For a biped with distributed mass we have

$$H_{cm} = \sum_i m_i(\vec{v_i} \times \vec{r_i}) + J_i w_i \tag{3.34}$$

where $\vec{v_i}$ is the velocity of the i-th body, $\vec{r_i}$ is the vector from the center of mass of the robot to the i-th body, J_i is the inertia of the i-th body, and w_i is the angular velocity of the ith body.

If we ignore gravity, then H_{tot} will be conserved, so that if we change H_{cm}, then $ml^2\dot{\theta}_1$ will change by the same amount in the opposite direction. This is what gives us the freedom to balance by "throwing mass around". To understand this more fully, consider the following examples. Figure 3-11 shows a model of a biped with distributed mass. If we pitch the upper torso forward (left), rotating it in the clockwise direction, then the center of mass will accelerate backward in the counter-clockwise direction. If we accelerate a leg quickly forward (right), rotating it in the counter-clockwise direction, then the center of mass will accelerate forward in the clockwise direction.

The ratio between the inertia of the internal body part being accelerated and the inertia of the biped's center of mass about the center of pressure will determine how much acceleration occurs. This ratio is typically less than 1.0 since anything larger would require the body part to extend

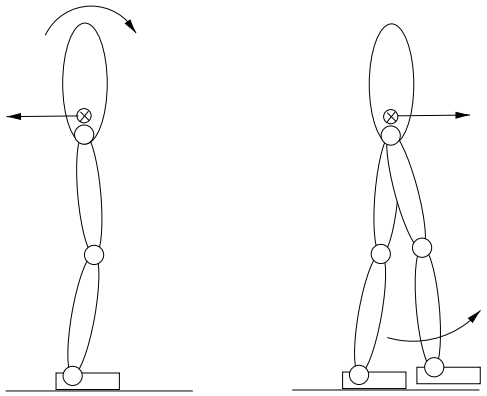


Figure 3-11: Robot model with distributed mass. Rotationally accelerating an internal inertia will cause an opposite acceleration of the inertia of the center of mass about the center of pressure on the foot. For example if the robot on the left accelerates its torso by pitching forward, then the center of mass will accelerate backward. If the robot on the right accelerates its swing-leg forward, then the center of mass will accelerate forward to conserve angular momentum about the center of pressure.

out further than the height of the robot or person. This is the case for a tightrope walker with a balancing bar. The bar can be accelerated in order to balance through restoring accelerations of the entire tightrope walker about the rope. However, without a balancing bar, the amount a biped can control rotational dynamics by throwing around its internal mass is limited.

Since the requirements of walking do not allow for arbitrary acceleration of the swing-leg and torso, we do not use acceleration of internal mass as a control mechanism in this thesis. However, acceleration of internal mass is probably useful for regaining balance after being pushed during standing, or when walking without the benefit of feet and actuated ankles. This is evident when tightrope walkers without a balancing bar lunge to regain balance.

3.10 Summary of Simple Models

The insight from the simple models of this Chapter can be summarized as follows:

- If the momentum of the robot about its center of mass is not changing and the length from the center of pressure to the center of mass is not changing, then the dynamics of the robot will obey the simple pendulum dynamics: $\ddot{\theta_1} = \frac{g}{l} \sin \theta_1$.
- The center of pressure on the foot can be moved to change the effective pivot point of the inverted pendulum.
- Changing the momentum of internal inertias in the robot (swinging legs, etc.) will cause an equal and opposite change in momentum of the center of mass about the effective pivot point of the inverted pendulum.

Thus, if we are interested in controlling the forward progress of the robot during single support, there are a few mechanisms we can use. We can directly move the center of pressure on the foot. Moving it forward will make the robot travel more slowly than it would otherwise. Moving it backward will increase its speed. We can change the location of the center of mass of the robot. By leaning forward, the center of mass of the robot will move forward and the robot will move faster than if the mass was straight up. By leaning backward the robot will slow down. Finally, we can accelerate internal mass to change the angular acceleration. For example, accelerating the swing-leg quickly forward will cause the body to accelerate forward.

The net change in forward velocity over a stride depends on how long the center of mass is in front of the center of pressure on the foot as opposed to how long it is behind it. Thus we can also change the speed of the robot over a number of strides, through foot placement, by changing θ_1 between the beginning of the stride and the end. For example, if we want to slow down over a stride, we can take a long stride length and transition to single support early.

All of the models considered in this chapter were confined to the sagittal plane. However, lateral balance can be achieved in a similar fashion. Foot placement can be used to initially place the center of pressure on the foot laterally and ankle torque can be used to move the center of pressure within the foot. Accelerating the swing-leg laterally can be used to balance when moving the center of pressure will not suffice.

The simple models in this chapter provide insight into the mechanisms that are useful for dynamic control of a biped. This insight will be used in Chapter 4 in the formulation of simple control strategies which are used in the algorithms presented in this thesis.

Understanding the models presented in this chapter not only help us to understand the control strategies used in future chapters, but also the control strategies that have been discovered that humans employ. For example, humans seem to use two strategies to regain balance when pushed during standing [Kuo & Zajac (1993*b*,*a*)]. These are referred to as the "ankle strategy", defined as acceleration about the ankle joint only, and the "hip strategy" defined as accelerations of the ankle and hip in opposite directions, with the hip accelerating about three times more than the ankle. The ankle strategy was found to be favored in response to slow perturbations, while the hip strategy was found to be favored in response to fast perturbations, or when the center of mass was displaced beyond the support region of the foot.

These strategies are quite understandable, considering the simple models of walking presented above. The ankle strategy moves the location of the center of pressure in order to regain balance while the hip strategy pitches the body inertia in order to make the center of mass accelerate in the opposite direction. The hip strategy is resorted to when the ankle strategy is not sufficient, due to the requirement that the center of pressure must stay inside the projection of the foot. With the robot (Chapter 4), we use the ankle strategy to balance during standing by simply changing the location of the center of pressure on the foot based on the position and velocity of the body.

Chapter 4

Exploiting Inherent Robustness

Walking is a moderately easy and inherently robust task. Specific trajectories, precision, and repeatability are not important in walking. The resultant motion can vary considerably between individuals and even in the same individual from step to step. Thus, simple control strategies that can be explained in simple terms can be used to control bipedal walking robots, despite non-optimal tracking performance.

Five conditions have to be met for a planar bipedal robot to walk. Height needs to be stabilized. Pitch needs to be stabilized. Speed has to be stabilized. The swing-leg needs to move so that the feet are in locations that allow for the stability of height, pitch, and speed. Finally, transitions from support leg to support leg must occur at appropriate times. If these five objectives are achieved, then the robot will walk.

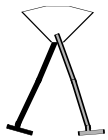
The converse is also true. Any controller that results in stable walking must meet these five objectives. Regardless of the specific control technique, something must act to stabilize height, pitch, and speed, swing the swing-leg, and transition appropriately from double to single and single to double support.

A number of different control strategies can be used to achieve each of these five objectives. Further, each strategy can be implemented in a variety of ways. In this chapter, we present several strategies for each objective that we have implemented on Spring Flamingo, our bipedal walking robot. We use the insight gained from the simple physical models presented in Chapter 3 to derive these strategies. This same insight makes the effectiveness of these strategies apparent. We describe how these control strategies can be implemented with Virtual Model Control [Pratt et al. (1997)], a robot control language which appeals to physical intuition.

We introduce a simple technique, called the "virtual toe point" constraint for approximately controlling the location of the center of pressure on the feet. The virtual toe point is the point along the foot at which zero torques are commanded. This is similar to a static version of the zero moment point [Vukobratovic et al. (1990)]. The virtual toe point constraint prevents the foot from over-rotating due to high ankle torques, while allowing the robot to go up on its toes to get an extended range of motion from the rear leg. The ability of the robot to go up on its toes allows it to walk with a straight leg which in turn increases energy efficiency.

We also present a simple scheme for dealing with force distribution between the two legs during the double support phase of walking. Instead of solving the force distribution problem exactly, we use a force distribution parameter relating percentage of load on each leg. This parameter can then be modified for speed control.

Using simple control strategies, the virtual toe point constraint, and the simple force distribution scheme, we have compelled Spring Flamingo, a seven link planar bipedal robot, to walk. The robot walks both slowly and quickly, walks over moderate obstacles, starts, and stops. With a few simple modifications to the algorithm, the robot walks over 15^o slopes and rolling terrain. It does this all blindly, i.e. with no a-priori knowledge of the terrain and no method to sense it except through contact with the feet.



4.1 Control Algorithm Properties

The control algorithms we develop in this chapter and throughout this thesis have the following properties:

- **Simple** The control laws employed are typically linear and low order with a few simple terms.
- **De-Coupled** The control problem is decomposed into the subproblems of height, pitch, speed, swing leg, and support transitions and independent controllers are developed for each part.
- **Physics-Based** The control laws and their parameters are physically meaningful and developed using physical intuition.
- **Low Impedance** The controllers are made "soft" by using low precision techniques and keeping control gain parameters low whenever possible. This allows the robot to conform to a disturbance or uneven terrain.

These control algorithm properties are similar to those used by Raibert and colleagues in their running robots [Raibert (1986)]. The running problem was decomposed into three parts: height, balance, and speed. A control rule was developed for each part using physical insight. The resulting rules are simple and can be described in simple terms: to control height, energy is pumped into the leg spring when the leg is fully compressed; to control pitch, the body is servoed, using hip torque, to be level to the ground when the stance leg is compressed; to control speed over a stride, the foot is placed further forward (to slow down) or further back (to speed up) from the neutral point in which speed is neither increased or decreased.

For walking, we will use the same decomposition plus explicitly add swing leg control and gait state transitions, which although were important in the running algorithms, were treated as side problems. However, since walking is fundamentally a different problem then running, the strategies and resulting control laws will differ from the running control laws.

Simple, de-coupled, physics-based, low impedance controllers can be powerful, easy to apply, and provide a high level of insight as to what is really important in the control of a robot. However, these controllers are often called "ad-hoc" or a "hack" because they are not mathematically based, and seldom rigorously proven to work. They are often considered to be "cheating" because parameters are usually hand tuned until desired performance is achieved.

However, use of this style of controller does not preclude mathematical analysis after the controller is developed. Unfortunately, analysis of legged robots is complicated for several reasons that are independent of getting a legged robot to perform a task. Automatic synthesis of a controller is even more difficult than analysis of an existing controller. New analysis and synthesis tools are being developed by many researchers but for the time being we believe that the limited capability of modern control theory and analysis should not be used as an excuse for the limited abilities of modern robots.

It is possible to use adaptive or learning techniques to automatically tune the control parameters. When the adaptation or learning is complete, it may then be possible to understand what was learned or why by examining parameters whose effects can be understood. For example, suppose the single support to double support transition distance parameter is changed by a learning algorithm that is attempting to maximize efficiency. One may then determine how that parameter affects efficiency.

In the following section we describe strategies based on physical insight that can be used to control bipedal walking robots. These strategies are easy to understand and easy to apply. Some of the strategies have been used to successfully compel Spring Flamingo, a planar bipedal robot, to walk, as described in Section 4.4.

4.2 Simple Control Strategies for Bipedal Walking

We now describe simple control strategies that can be used to achieve the five conditions required for planar bipedal walking: height stabilization, pitch stabilization, speed stabilization, swing-leg placement, and support transitions.

There are a number of methods that can be used to implement the control strategies. These include inverse kinematics, high gain servos, feed-forward control, impedance control, etc.

We use Virtual Model Control, a tool which relies on physical intuition, to implement the control strategies. This control technique uses simulations of virtual mechanical components to generate real actuator torques (or forces). These joint torques create the same effect that the virtual components would have created, thereby creating the illusion that the simulated components are connected to the real robot. Such components can include linear or non-linear springs, dampers, dashpots, masses, latches, bearings, potential and dissipative fields, or any other imaginable component that produces a force or torque based on its state.

In section 4.4 we discuss how the following control strategies are used to compel Spring Flamingo, a bipedal walking robot, to walk.

4.2.1 Height Stabilization

Stabilizing height is straightforward provided that a support leg is firmly on the ground. This will be the case if the swing-leg strategy and support leg strategies are working. There are many ways in which height can be stabilized. Two strategies are described here.

1. Maintain a constant height above the ground.
2. Maintain a constant stance leg length.

Maintaining a constant height can be implemented with a virtual spring-damper mechanism attached between the ground and the robot's body. The spring set point determines the height above ground that the robot maintains. The damper causes oscillations about that height to decay. A virtual vertical force of the weight of the body allows for lower virtual spring constants and decreased offset.

Maintaining a constant stance leg length can be done in a number of different ways. A simple method that does not require high-gain feedback is to push vertically on the body with a force a little larger than the weight of the robot. This will cause the robot to increase its height until the knee hits its joint limit (knee cap).

The first strategy was used to stabilize height in Spring Turkey [Pratt et al. (1997)], our previous walking robot. We successfully applied both strategies to Spring Flamingo but embraced the second one because it is much more efficient and more similar to biological walking. By walking with fully extended legs, there is a low torque requirement on the knee. The robot will walk with somewhat of a compass gait, in which potential energy and kinetic energy are out of phase and hence total mechanical energy is nearly constant.

4.2.2 Pitch Stabilization

As in height stabilization, stabilizing pitch is straightforward provided that we have a support leg firmly on the ground. Two strategies for stabilizing pitch are

1. Maintain a level pitch.
2. Follow a pitch trajectory.

Both strategies require feedback since the center of gravity is above the hip. If it were below the hip, we could rely on the natural dynamics that would already be stable. To control the pitch, we use a virtual torsional spring-damper mechanism. If a level pitch trajectory is desired, then the virtual spring set point is held constant. If a pitch trajectory is desired (perhaps to help control speed, as described below), then the set point is changed to match the desired pitch position. In this chapter, we used the level pitch strategy in the control of Spring Flamingo. In Chapter 7, we modify the desired pitch angle to help control speed.

4.2.3 Speed Stabilization

For dynamic bipedal walking, it is impossible to arbitrarily control the forward speed during a stride since the center of mass projection lies outside the support foot polygon during much of the stride. However, speed can be modified, within limits, during a stride, and can be modified from stride to stride using discrete control events.

Five strategies for stabilizing speed are

1. Change stride length.
2. Change transition events.
3. Vary the location of the center of pressure on the support foot.
4. Vary force distribution during double support.
5. Change the location of the body center of mass by pitching forward or backward.
6. Vary the amount of push during toe-off.
7. Apply energy at strategic times.

The first two strategies (stride length and transitions) are discrete controllers that will stabilize the speed over a number of strides. By changing stride lengths and transition events one can change the fraction of the stride in which the support leg is in front of the body (slowing down the robot) and the fraction of the stride in which the support leg is behind the body (speeding up the robot).

The third strategy (center of pressure) can be used to control speed during single support. The center of pressure can be placed forward, closer to the front of the foot, to slow the robot down and placed backward, closer to the heel, to speed it up. In this chapter we use the virtual toe point to do this and assume that the virtual toe point and the center of pressure are highly correlated.

The fourth strategy (force distribution) can only be used during double support. More force can be applied by the rear leg to speed up the robot and more force can be applied by the front leg to slow down the robot. This changes the effective location of the center of pressure of a virtual leg between the two real legs (the zero moment point).

The fifth strategy (pitching the body) can be used to speed up the robot by leaning forward or slow it down by leaning backward. This strategy is typically seen when a person leans forward when walking up a hill or leans back to brake when going down it.

The sixth strategy (toe-off push) can be used to control speed by varying how much the foot pushes against the ground during toe-off. Due to the leg being behind the body during toe-off, a push that effectively lengthens the leg length will propel the body forward, increasing its forward velocity. Modulation of this push can be used to control velocity.

The seventh strategy (applying energy at strategic times) is a bit trickier to apply. It may be possible to put energy into the system in one mode during part of the stride and later convert this energy into forward speed. For example, it may be possible to increase the potential energy of the system by having the robot go up onto its toes earlier and then convert that energy into forward kinetic energy during a later portion of the stride.

The first four strategies were used to stabilize Spring Flamingo's speed. The fifth and sixth strategies are added in Chapter 7 to achieve fast walking speeds. The last strategy is currently being investigated.

4.2.4 Swing-Leg Placement

In order to walk successfully, the swing-leg must swing quickly to its next support location. Fortunately, the exact placement is not important when walking on smooth ground.

Two possible strategies for swinging the swing-leg are

1. Servo the swing-leg, either as a function of time, or as a function of the state of the support leg.
2. Let the swing-leg swing passively, making sure it does not hit the ground.

Following a swing trajectory can be implemented with a virtual spring-damper mechanism attached between the body and the ankle of the robot. The spring set point can move along the trajectory, pulling the leg along to follow it.

The natural pendulum dynamics of the swing-leg are exploited in the second strategy. The leg will naturally swing forward, as long as the foot clears the ground.

In this chapter, the first strategy is successfully applied to Spring Flamingo. In Chapter 5 we use the passive swing trajectory. In Chapter 7 we achieve fast walking by actively swinging the hip but letting the knee swing passively during most of the swing phase.

4.2.5 Support Transitions

To continuously walk forward, the support legs must be alternated since a given leg can only support the body over a small range. For bipedal walking, double support to single support and single support to double support transitions must occur at appropriate times.

Four strategies for transitioning from double support to single support are

1. Transition to single support if the body is within a certain distance to the next support leg.
2. Transition to single support if the body is over a certain distance away from the previous support leg.
3. Transition to single support after being in double support for a certain amount of time.
4. Transition to single support when the force on the rear leg falls below a threshold.

These strategies can be implemented by measuring joint angles and transitioning based on joint angle thresholds or by computing the kinematics of the robot and transitioning on center of mass to foot distance thresholds. A state machine can be used to keep track of which support state the robot is in.

The first strategy will ensure that the next support leg will have a long enough support time that the other leg will be able to swing through in time. The second strategy will ensure that the rear leg has enough range of motion that the robot does not have to drag its rear leg. The third strategy simply transitions to single support after a given time. The fourth strategy ensures that the rear leg is not supporting any load before swinging.

In this chapter, we use the first two strategies together in the control of Spring Flamingo. When either event happens, the robot transitions to single support.

Two strategies for transitioning from single support to double support are

1. Transition to double support if the body is over a certain distance away from the support leg.

2. Transition to double support if the swing-leg has swung beyond a certain position or has slowed below a certain speed.

The first strategy ensures that the robot will transition onto a new support leg before the body moves too far from the current support leg. This will guarantee that the current support leg can safely support the body and hence stabilize height and pitch.

The second strategy will ensure that the next support leg has swung far enough that it is in a position to support the body when the time comes. This strategy also maximizes double support time as the robot will put its swing-foot down as soon as it has swung through rather than wait for the transition distance event to occur.

In this chapter, we used the first strategy in the control of Spring Flamingo. In Chapter 5, we use the second strategy, transferring to double support as soon as the swing-leg swings through and hits the ground.

4.3 Virtual Actuator Implementation for a Planar Biped With Feet and Ankles

We use Virtual Actuators [Pratt et al. (1996)] and Virtual Model Control [Pratt et al. (1997)], two techniques based on physical intuition, to implement some of the strategies of the previous section.

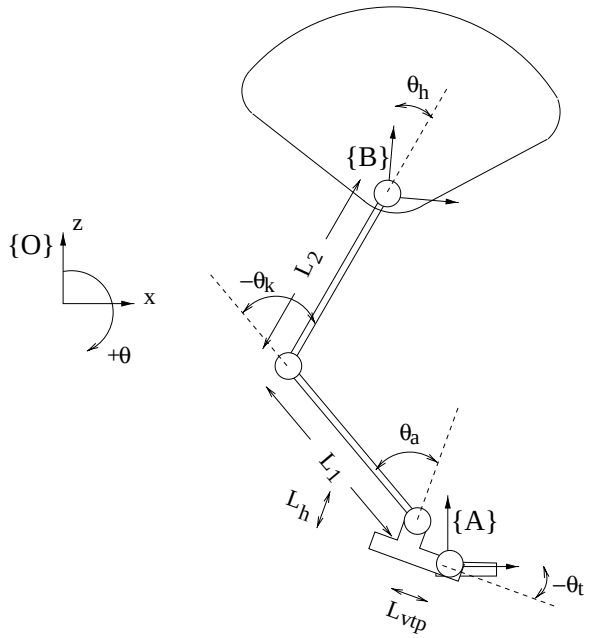


Figure 4-1: Virtual Model implementation on a single leg. Reaction frame {A} is assumed to be in the same orientation as reference frame {O} so that ${}^{O}_{A}R = I$. For this derivation, we assume the robot is in the configuration with the knee bent backward. However, the resulting equations hold whether the knee is bent backward or forward.

In this section we present the mathematics to implement virtual components on Spring Flamingo for the support leg in single support and both legs in double support, following the procedure described in [Pratt et al. (1996)].

4.3.1 Single Leg Implementation

Figure 4-1 shows a simple planar, five link, four joint, serial robot model that we use to represent a single leg of our walking robot. The toe joint and link do not exist on the real robot (Figure 4-2). They are used to represent the point on the foot at which no torque is applied. We refer to this as the "virtual toe point". It is similar to the center of pressure on the foot or the zero moment point [Vukobratovic et al. (1990)] except that it is a commanded quantity, not a measured one, and is based on static, not dynamic, considerations.

The virtual toe point can be used for control in the following intuitive sense. If it is desired to accelerate the robot backward (or reduce the forward acceleration) one can move the virtual toe point forward on the foot. Similarly, if it is desired to accelerate the robot forward (or reduce the backward acceleration) one can move the virtual toe point backward toward the heel.

We wish to connect a virtual component between the virtual toe point frame, {A}, and the body frame, {B}. The angles θ_t, θ_a, θ_k, and θ_h are those of the virtual toe, ankle, knee, and hip. The upper link (femur) is of length L_2, the lower link (tibia) is of length L_1, and the height of the foot is L_h. We assume that the virtual toe is flat on the ground, so that ${}^{O}_{A}R = I$. The commanded distance to the virtual toe point from the ankle is L_{vtp}.

The forward kinematic map from frame {A} to frame {B} is as follows,

$$ {}^{A}_{B}\vec{X} = \begin{bmatrix} x \\ z \\ \theta \end{bmatrix} = \begin{bmatrix} -L_{vtp}\, c_t - L_h\, s_t - L_1\, s_{t+a} - L_2\, s_{t+a+k} \\ -L_{vtp}\, s_t + L_h\, c_t + L_1\, c_{t+a} + L_2\, c_{t+a+k} \\ -\theta_h - \theta_k - \theta_a - \theta_t \end{bmatrix} \tag{4.1} $$

where $c_a \equiv \cos(\theta_a)$, $s_a \equiv \sin(\theta_a)$, etc. Partial differentiation with respect to the joint angles produces the Jacobian,

$$ {}^{A}_{B}J = \begin{bmatrix} J_{1,1} & J_{1,2} & J_{1,3} & 0 \\ J_{2,1} & J_{2,2} & J_{2,3} & 0 \\ -1 & -1 & -1 & -1 \end{bmatrix} \tag{4.2} $$

where,

$$ J_{1,3} = -L_2\, c_{t+a+k} $$

$$ J_{1,2} = J_{1,3} - L_1\, c_{t+a} $$

$$ J_{1,1} = J_{1,2} + L_{vtp}\, s_t - L_h\, c_t $$

$$ J_{2,3} = -L_2\, s_{t+a+k} $$

$$ J_{2,2} = J_{2,3} - L_1\, s_{t+a} $$

$$ J_{2,1} = J_{2,2} - L_{vtp}\, c_t - L_h\, s_t $$

The Jacobian relates the virtual velocity between frames A and B to the joint velocities

$$ {}^{A}_{B}\vec{X} = {}^{A}_{B}J\, \dot{\vec{\Theta}} \tag{4.3} $$

and the virtual force to joint torque,

$$ \vec{\tau} = ({}^{A}_{B}J)^T\, ({}^{A}_{B}\vec{F}) \tag{4.4} $$

where $\vec{\tau}$ is the joint torque vector and $\vec{F}$ is the virtual force vector.

Next we add the constraint of an unactuated toe, $\tau_t = 0$, since we desire zero actuated torque about the virtual toe point. This will constrain the direction in which virtual forces can be applied. With the virtual toe point constraint, Equation 4.4 is,

$$ \begin{bmatrix} 0 \\ \tau_a \\ \tau_k \\ \tau_h \end{bmatrix} = \begin{bmatrix} J_{1,1} & J_{2,1} & -1 \\ J_{1,2} & J_{2,2} & -1 \\ J_{1,3} & J_{2,3} & -1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_x \\ f_z \\ f_\theta \end{bmatrix} \tag{4.5} $$

The control strategies described previously control height by applying vertical force and control pitch by applying torque to the body. Forward speed is indirectly controlled by varying the location of the center of pressure and the center of mass and by stride length and toe-off force. Since we are more concerned about applying forces in the vertical direction and torques about the body than we are concerned about applying horizontal forces we specify f_z and f_θ and solve for f_x

$$f_x = \begin{bmatrix} \frac{-J_{2,1}}{J_{1,1}} & \frac{1}{J_{1,1}} \end{bmatrix} \begin{bmatrix} f_z \\ f_\theta \end{bmatrix} \tag{4.6}$$

Plugging equation 4.6 back into equation 4.5, we get

$$\begin{bmatrix} \tau_a \\ \tau_k \\ \tau_h \end{bmatrix} = \begin{bmatrix} \frac{-J_{1,2}J_{2,1}}{J_{1,1}} + J_{2,2} & \frac{J_{1,2}}{J_{1,1}} - 1 \\ \frac{-J_{1,3}J_{2,1}}{J_{1,1}} + J_{2,3} & \frac{J_{1,3}}{J_{1,1}} - 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} f_z \\ f_\theta \end{bmatrix} \tag{4.7}$$

Throughout the above derivation we have assumed that the toe is flat on the ground and that we can measure all angles. Because there is no toe (it is virtual) we cannot measure its angle with the ground. Instead we measure the body angle via a potentiometer on a boom or a gyroscope and compute what the toe angle would be if the toe was flat on the ground,

$$\theta_t = -\theta - \theta_h - \theta_k - \theta_a \tag{4.8}$$

We now have a simple set of equations for determining joint torques given virtual forces. These equations will be used in the next section in the control of a bipedal walking robot during the single support phase.

4.3.2 Dual Leg Implementation

In the previous subsection we discussed virtual actuator implementation for a single leg (when the robot is in single support). Here we examine the double support case.

As in [Pratt et al. (1996)] we could construct a constraint matrix and solve exactly for the force distribution between the two legs such that any arbitrary F_x, F_z, F_θ force vector could be commanded. However, we decide to use a simpler method because

- There is no solution when the feet are together, as the constraint matrix is not invertible in such a configuration.
- It is unlikely that biological creatures solve the force distribution problem exactly.
- Solving the force distribution problem exactly is unnecessary.
- The method presented below appeals more to intuition.

Instead of solving the force distribution problem exactly we simply distribute the force between the two legs with a force distribution parameter, α, such that,

$$\begin{bmatrix} f_z \\ f_\theta \end{bmatrix}_{left} = \alpha \begin{bmatrix} F_z \\ F_\theta \end{bmatrix}, \quad \begin{bmatrix} f_z \\ f_\theta \end{bmatrix}_{right} = (1-\alpha) \begin{bmatrix} F_z \\ F_\theta \end{bmatrix} \tag{4.9}$$

where $0 \leq \alpha \leq 1$

As in the single support case, we do not command forces in the x direction. Instead we command forces in the z and θ directions, decide the force distribution between the two legs, and solve for the joint torques using Equation 4.7 for each leg.

We choose α such that the forces are divided between the legs in a natural way. If the robot's body is directly above the left leg, all forces are provided by the left leg ($\alpha = 1$). Similarly, if the robot's body is directly above the right leg, all forces will be provided by the right leg ($\alpha = 0$). If the robot's body is between the left and right legs, the forces will be divided with a linear relationship

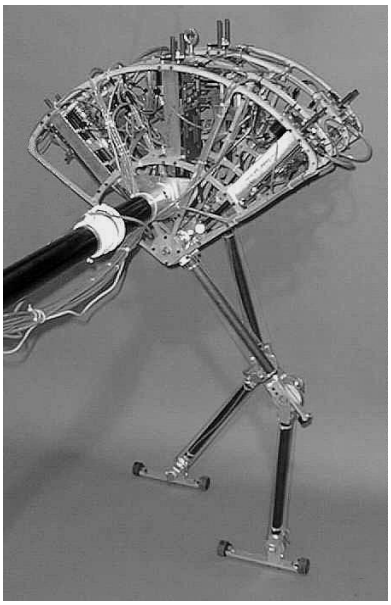


Figure 4-2: Spring Flamingo, a planar bipedal walking robot. There are six force controlled actuators attached to the body. Power is transmitted to the hips, knees, and ankles via cables. A boom prevents motion in the lateral, roll, and yaw directions. However, the boom does not support the robot in the sagittal plane.

$$\alpha = \frac{x_{right}}{x_{left} + x_{right}} \tag{4.10}$$

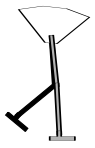
where $x_{left} \geq 0$ is the horizontal distance from the left leg to the body and $x_{right} \geq 0$ is the horizontal distance from the right leg to the body. If the legs are both close together, it is very much like the single support case and we simply set $\alpha = 0.5$, dividing the forces evenly between the two legs.

We can modify the force distribution parameter α for control in the following way. To accelerate forward, we put more of the force distribution on the rear leg. To accelerate backward, we put more of the force distribution on the front leg. For example we can use the simple control law $\alpha = \alpha_0 + b(\dot{x}_d - \dot{x})$ to help regulate velocity when the left leg is the rear leg.

In the next section we will use this control strategy to help regulate forward velocity during double support.

4.4 Simple Control Strategies Applied to a Bipedal Walking Robot

Figure 4-2 is a photograph of Spring Flamingo, a planar bipedal walking robot. The robot has an actuated hip, knee, and ankle on each leg. An unactuated boom constrains Spring Flamingo's roll, yaw, and lateral motion, thereby reducing it to a planar robot. All of Spring Flamingo's motors are located in its upper body, with power being transmitted to the joints via cable drives. Series Elastic Actuation [Pratt & Williamson (1995)] is employed at each degree of freedom, allowing for accurate application of torques and a high degree of shock tolerance. The maximum torque that can be applied to the hips and ankles is approximately 18 Nm, while approximately 24 Nm can be applied to the knees. The force control bandwidth we achieve is approximately 15 Hz. Spring Flamingo weighs approximately 30 lbs (14 kg) and stands 3 ft (1 m) tall from floor to hip.



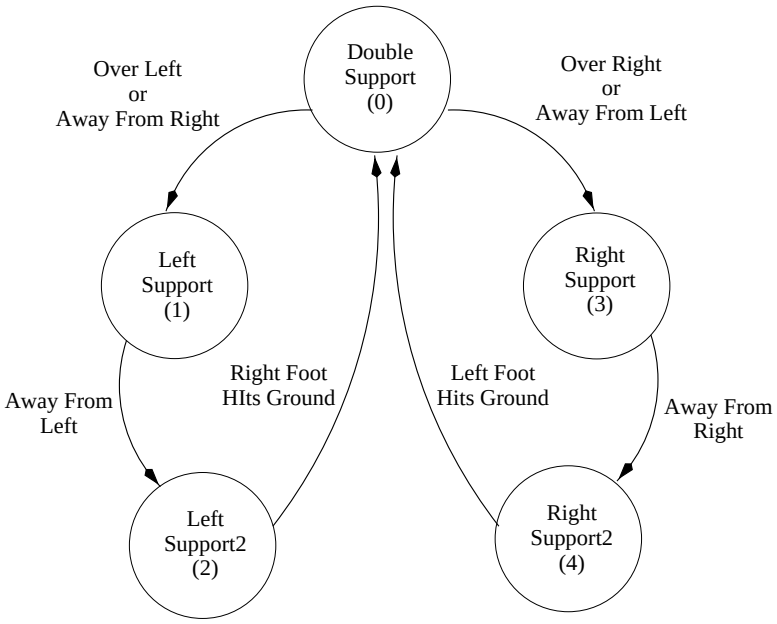


Figure 4-3: REAL State machine used in Spring Flamingo's walking algorithm.

Potentiometers at the hips, knees, ankles, and boom measure joint angles and body pitch. Compression springs are used in the joint actuators to implement Series Elastic Actuation. Linear potentiometers measure the stretch in the springs.

4.4.1 Walking Algorithm

The simple control strategies used on Spring Flamingo are:

- Maintain a constant stance leg length by pushing up until hitting the knee cap.
- Maintain a constant level pitch using a virtual spring damper mechanism with constant set point.
- Transition from double support to single support when the body's forward position becomes further than a prespecified distance from the rear foot or closer than a prespecified distance from the front foot.
- Transition from single support to double support when the body's forward position becomes further than a certain distance from the support foot.
- Swing the non-stance leg such that the foot is roughly placed a desired stride length away from the support foot.
- Increase the nominal stride length as the robot walks faster.
- Delay transition to double support if the robot is walking too slowly. Conversely, initiate transition to double support sooner if the robot is walking too quickly.
- Maintain the center of pressure of the support foot approximately below the center of mass, moving it forward if walking too quickly or backward if walking too slowly.
- During double support shift the load towards the back leg if walking too slowly or towards the front leg if walking too quickly.

To implement Spring Flamingo's walking algorithm, we use a simple set of virtual components and a state machine shown in Figure 4-3. The various virtual spring, damper, and force variables and walking parameters are chosen using physical insight and a manual search. For example, the virtual spring-damper for pitch control is

Table 4.1: REAL Control Parameters for Spring Flamingo's Walking Algorithm. The listed range is the reasonable amount that the parameters can vary by while the robot remains well-behaved.

Parameter	Value	Range
Height Control		
Virtual Z Anti-Gravity Force	110 N	90-120
Virtual Z Damper	$200\frac{N}{m/s}$	100-250
Pitch Control		
Virtual Pitch Spring	$60\frac{Nm}{rad}$	30-80
Virtual Pitch Damper	$10\frac{Nm}{rad/s}$	4-15
Forward Speed Control		
Nominal Velocity	$0.4\frac{m}{s}$	0.0-0.7
Virtual Toe Point Gain	$0.3\frac{m}{m/s}$	0.0-0.5
Double Support Transfer Ratio Gain	$0.3\frac{\%}{m/s}$	0.0-1.5
Double to Single Support Transition Distance Gain	$0.3\ \frac{m}{m/s}$	0.0-0.5
Single to Double Support Transition Distance Gain	$0.3\ \frac{m}{m/s}$	0.0-0.5
Swing-Leg Control		
Virtual Swing-Leg X Spring	$25\frac{N}{m}$	10-40
Virtual Swing-Leg X Damper	$3\frac{N}{m/s}$	1-5
Virtual Swing-Leg Z Spring	$150\frac{N}{m}$	100-200
Virtual Swing-Leg Z Damper	$8\frac{N}{m/s}$	2-14
Support Transitions		
Double to Single Support Rear Leg Transition Distance	$0.26m$	0.20-0.30
Double to Single Support Front Leg Transition Distance	$0.05m$	0.01-0.10
Single to Double Support Transition Distance	$0.16m$	0.10-0.24
Nominal Stride Length	$0.36m$	0.24-0.42

$$\tau_b = K_{pitch}(\theta_d - \theta) - B_{pitch}\dot{\theta} \tag{4.11}$$

where τ_b is the torque on the body through the stance leg, θ is the body lean angle, θ_d is the virtual spring set-point, K_{pitch} is the virtual spring stiffness, and B_{pitch} is the virtual damper parameter. We manually tune K_{pitch} and B_{pitch} until the amount the robot resists body twist and the rate at which oscillations die out is acceptable.

The algorithm control parameters are listed in Table 4.1 along with their tuned values and reasonable ranges. This range represents the reasonable amount that the parameter can vary by while the robot remains well-behaved. For some parameters, the robot can continue to walk throughout this range. For others, the robot cannot walk if the parameter is near the boundaries of the range. All the parameters can be varied, one at a time, by at least 10% or all at once by 5% without disrupting stable walking.

The vertical force to control height is calculated to be a little larger than the weight of the robot. Many parameters are tuned by physically examining their effects (resistance to being pushed on, decay rate, etc.) until the desired effects are achieved and the robot walks successfully. These walking parameters consist of nominal stride length, transition distances, swing-leg gains, and velocity gains on the transition distances, stride length, virtual toe points, and double support loading ratio. From the time the robot was built until the moment it could continuously walk, approximately 40 iterations were performed over a span of 3 weeks.

Spring Flamingo is initialized balancing with its feet together. It starts walking by lifting up one leg and transitioning into the single support phase. At no time is external intervention required. The robot stops by setting its desired velocity to zero after walking a given distance.

Figure 4-4 shows experimental data from Spring Flamingo while walking. The graphs on the left show (from top to bottom) the body's horizontal position (x), horizontal velocity, and state. The graphs on the right show the body's vertical position (z), pitch (θ), and the mechanical power being exerted at the joints. The mechanical power is computed as the sum of the absolute value of the torque times angular velocity at each joint.

Spring Flamingo walked continuously at approximately 0.6 m/s. The data shows approximately 10 steps (left to right or right to left support transitions) in 8 sec, giving a step time of 0.8 seconds. The height fluctuated about 3 cm as the robot walked using a compass-like gait. The pitch was confined to ± 0.04 radians (± 2.1 deg). The mechanical power averaged about 15 Watts. However, due to the inefficiencies of the motors, transmissions, and power electronics, the electrical power consumed was much higher. In fact, the power measured from the wall outlet in which the robot was attached was about 200 Watts!.

The data in Figure 4-4 is plotted in graphical form in Figure 4-5. The snapshots in Figure 4-5 are approximately 0.5 seconds apart. Lines are drawn to show the path of the tips of the feet and the hip trajectory.

4.4.2 Robustness of Walking Algorithm

The walking control algorithm discussed above is robust to moderate external forces, rough terrain, and parameter variation. Spring Flamingo can be pushed fairly hard in either direction, temporarily changing its speed by about 25%, recovering to the original speed within a few steps. The robot can walk up and down slopes of approximately 5^0 without any change in the algorithm and without being informed of (or detecting) the presence of the slope. Each of the control parameters can be changed by 10% or more while still maintaining stable walking.

The most common failure mode occurs when the robot is pushed forward too hard and cannot recover. The robot will typically take several short, choppy steps further increasing its speed and finally falling. A biological creature in this situation typically recovers by running a few steps and slowing down. By running, the animal benefits from an ariel phase which provides more time to swing a leg through to a position that allows recovery. Unfortunately, Spring Flamingo cannot run and hence has no recourse when its speed increases above its natural walk-to-run transition speed.

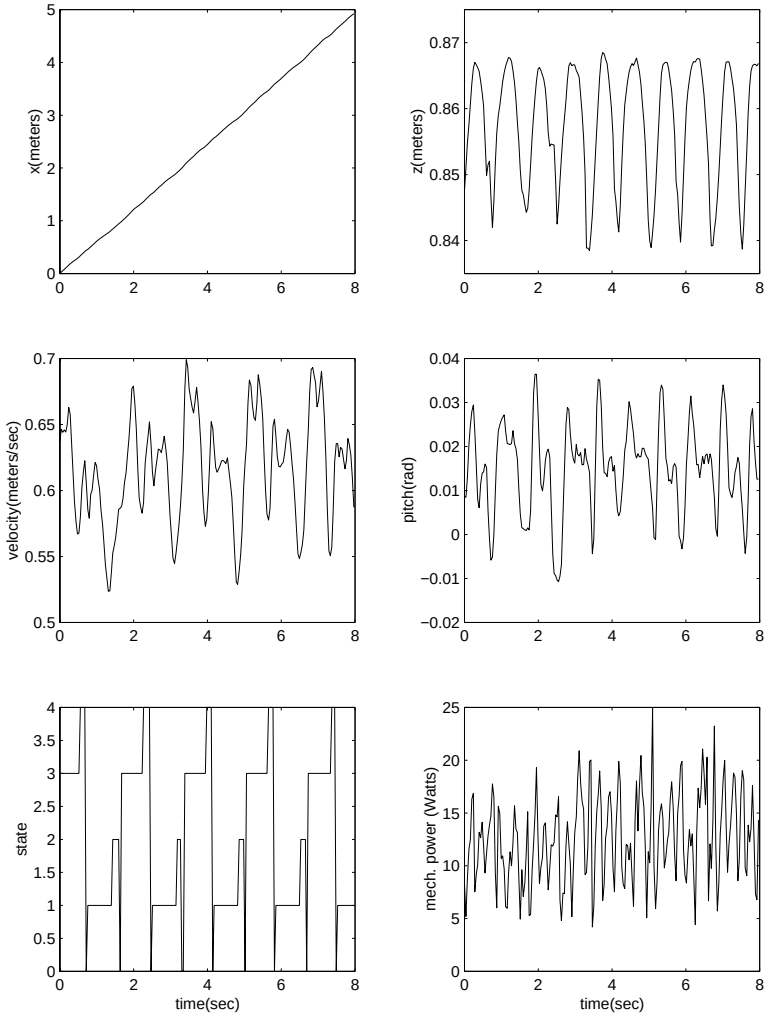


Figure 4-4: REAL Spring Flamingo walking data. Left graphs display, from top to bottom the horizontal position (x) and horizontal velocity of the body and the state of the state machine. Right graphs display the vertical position (z) and pitch (θ) of the body and mechanical power being applied to the joints.

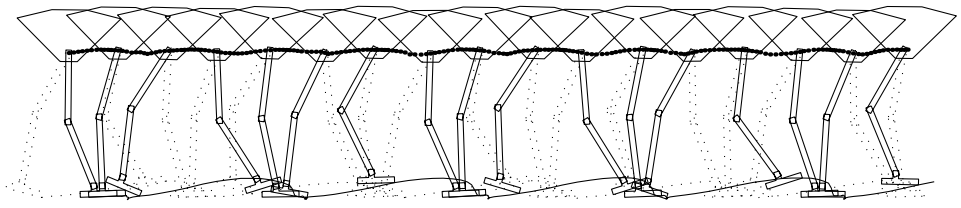


Figure 4-5: REAL Elapsed time snapshot of the Spring Flamingo walking data in Figure 6. The drawings of the robot are spaced approximately 0.5 seconds apart. The left leg is dotted while the right leg is solid. Lines show the path of the tips of the feet and the hip trajectory. The robot walks from left to right.

4.4.3 Self-Stabilizing Speed

The above algorithm used several controllers to stabilize speed. In another algorithm, we successfully compelled Spring Flamingo to walk stably without any feedback on the forward speed. We used the speed control strategy "Take longer strides as the robot walks faster" but we never explicitly programmed it. Instead, we used low gains on the swing-leg such that overshoot was significant and the natural dynamics of the system played a large role in where the leg was placed. As the robot walked faster it naturally took longer strides without explicitly being told to. The speed-dependent stride lengths then self-stabilized the forward walking speed.

With this self-stabilizing speed algorithm, the robot walked continuously while being robust to external pushes. However, we could only get this algorithm to work for slow speed walking. In Chapter 5 we describe how we were able to get the robot to walk $0.5\frac{m}{s}$ without explicit speed control.

4.5 Blind Walking over Hills and Rolling Terrain

Simple control strategies which appeal to physical intuition can be used to control a biped to walk over moderate hills and rolling terrain. In this section we describe how we extended the above algorithm to control Spring Flamingo to walk over a 15^o ramp and rolling terrain consisting of alternating 15^o uphills and downhills.

No sensors were added to the robot and prior information about the slope was not used in controlling the robot. Detecting and measuring the slopes was achieved entirely through the feet. Therefore, we refer to this as blind walking.

4.5.1 Estimating the Slope of Hills

There are two ways to estimate the ground slope through the kinematics of the robot. The first, which we call the local slope, is the angle that the foot makes, with respect to horizontal, when it is flat on the ground. The second, which we call the global slope, is the slope of the line connecting the two ankles during double support. Figure 4-6 shows graphically the definition of local and global slope. This means of slope estimation is the same as that used in [Chew et al. (1999)].

The local slope and the global slope are not equal in general, and in fact may even have differing signs, for example when transitioning from uphill to downhill. The global slope seems to be a better estimate of the real slope of the hill since it takes into consideration more distance. However, it can only be measured during double support. The local slope has a predictive quality when walking over smooth terrain in which the slope does not change drastically such as those considered here. Here we simply use the average of the local and global slope for the estimation of the ground slope. During single support we use the global slope estimated during the previous double support phase.

4.5.2 Simple Control Strategies for Hills

The following simple control strategies are added to the flat ground algorithm to control the robot to walk over rough terrain:

- Increase desired velocity when going uphill and decrease it when going downhill.
- Lean forward when walking uphill, lean backward when walking downhill.
- Put weight more on the toes when walking uphill.
- Bend the support knee when going downhill to help the swing-leg hit the ground.
- Lift leg higher when going uphill to prevent stubbing a toe.

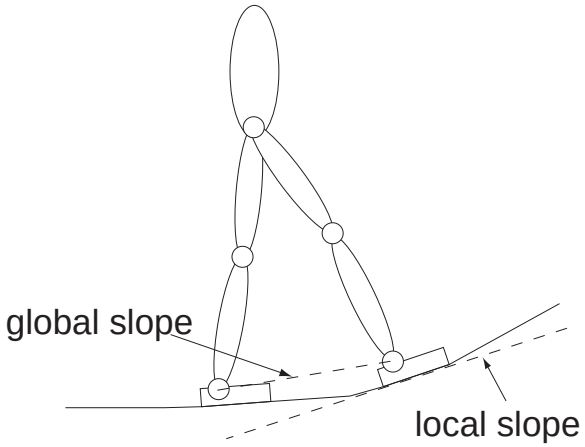


Figure 4-6: Graphical definition of global slope and local slope. In general, the global and local slopes will be different. The global slope may be a better estimate, but the local slope has predictive qualities for smoothly rolling terrain.

Table 4.2: REAL Additional parameters for Spring Flamingo's walking algorithm over rough terrain.

Parameter	Value
Desired Velocity Multiplier	$0.7\frac{m/s}{rad}$
Desired Pitch Multiplier	$0.15\frac{rad}{rad}$
Virtual Toe Point Multiplier	$0.125\frac{m}{rad}$
Desired Knee Bend Rate	$5.0\frac{rad}{sec}$

The first four of these strategies use the parameters listed in Table 4.2 to modulate the parameters used in the flat terrain algorithm.

Changing the desired velocity is an easy way to effectively alter a number of other parameters at the same time. This results in the robot putting more effort into walking uphill and resist speeding up while going downhill. We do this with the desired velocity multiplier, such that the desired velocity is what it would be in the flat ground algorithm plus the desired velocity multiplier times the estimated slope of the ground.

Previously we discussed how body lean can be used to speed up or slow down. Here we increase desired pitch to lean into the hill and decrease it to lean back when going down a hill with the desired pitch multiplier. This parameter augments the desired velocity multiplier. Both parameters were tuned so that the robot didn't change its speed between uphill, downhill, and flat terrain.

Putting weight more forward on the toes when walking uphill was found to be useful in order to get the robot to go onto its toes when walking uphill. The virtual toe point multiplier changes the location of the virtual toe point depending on the estimated slope. This parameter was tuned such that the robot toed-off more naturally when walking uphill.

When going downhill cautiously, one bends their support leg in order to lower their swing-leg to the level of the ground. To achieve this effect with the robot, we have the support knee servo down starting when the body passes over the support ankle. The desired knee angle decreases at the desired knee bend rate until the swing-leg hits the ground.

When going uphill, the swing-leg must be lifted higher to prevent stubbing the toe. The desired height of the swing-leg is simply increased such that it will be at the proper height at touchdown assuming the estimated slope is correct. To prevent toe-stubbing when the estimated slope is not correct, we simply increased the desired swing height for flat terrain.

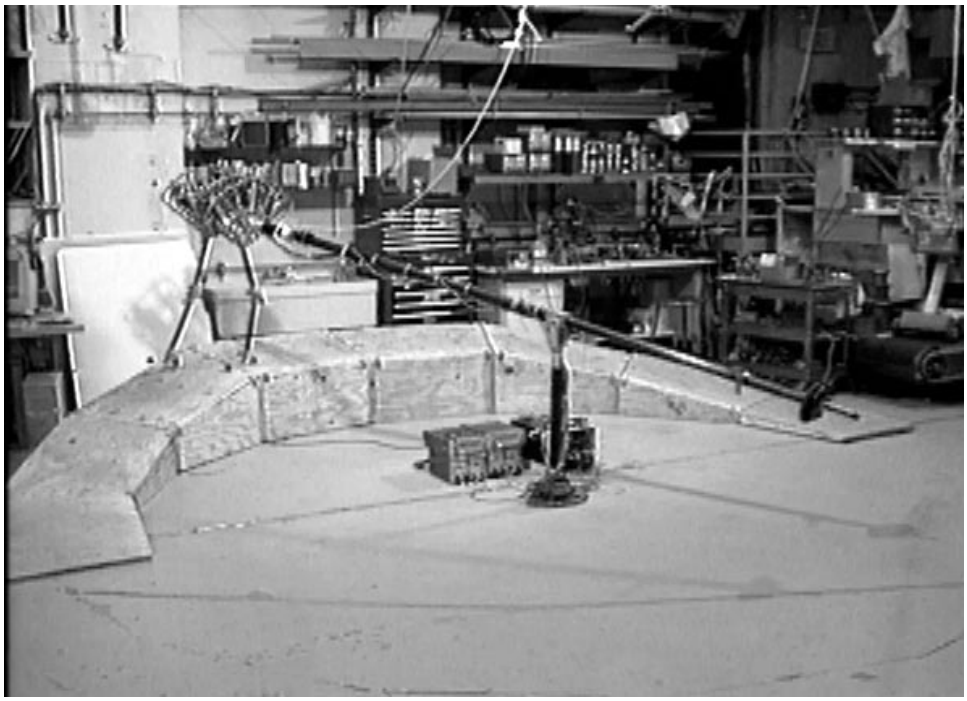


Figure 4-7: REAL Photograph of Spring Flamingo walking over a ramp with 15^o upslope and downslope. The robot uses only its feet to detect the terrain. While the boom constrains the robot to the sagittal plane, it does not support it in that plane.

4.5.3 Results on Hills

Figure 4-7 shows Spring Flamingo walking over a ramp that starts with a 15^o upslope, goes flat, and then has a 15^o downslope. Data are plotted in Figure 4-8. The robot walks approximately $0.35\frac{m}{s}$. The body pitch increases when going uphill and decreases when going downhill, showing that the strategy of bending forward or backward depending on the slope is in effect. The estimated slope has errors but still accurately tracks the general uphill and downhill progression. The mechanical power jumps to about $25W$ when walking uphill and downhill with a peak power of approximately $50W$. It falls back to about $15W$ average when the hill flattens. This is expected as the joints must supply positive work to increase potential energy when walking uphill and must absorb negative work to decrease potential energy when walking downhill.

Figure 4-9 shows video frames of Spring Flamingo walking over rolling terrain consisting of alternating 15^o upslopes and downslopes. With the algorithm described above, the robot successfully traversed this terrain. On a typical day, the robot could traverse the terrain over 20 times without falling.

4.6 Conclusions

Spring Flamingo walked continuously using a simple set of control strategies that were developed using physical insight and intuition. These strategies are easy to develop, are easy to understand, and are easy to implement. In short, planar bipedal walking is easy to achieve despite it being difficult to synthesize a controller mathematically.

By tuning the control parameters by hand, one gains insight into how the parameters relate to the resultant walking. This insight helps speed up the tuning process and suggests control algorithm improvements.

Spring Flamingo gained several advantages over Spring Turkey [Pratt et al. (1997)] by the use

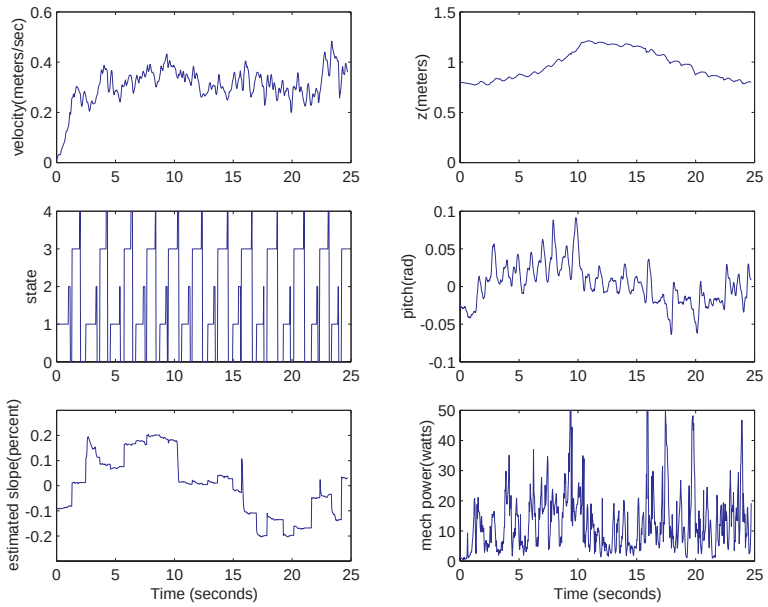


Figure 4-8: REAL Spring Flamingo walking data while walking over a 15^o ramp.

Figure 4-9: REAL Video frames of Spring Flamingo walking over rolling terrain with 15^o upslopes and downslopes. Frames proceed left to right with 0.5 seconds between frames.

of feet and actuated ankles. Since Spring Turkey had only point feet, it could not balance on one foot, had to walk with bent knees, and had large velocity fluctuations during each stride. Spring Flamingo exhibits smaller velocity fluctuations as it keeps its virtual toe point directly underneath when the center of mass passes over the support foot. It walks more efficiently as a compass gait with straight legs can be used without the worry of losing range of motion on the rear leg. Also, with feet and ankles, the robot is able to stand and balance on one leg, which was not presented in this thesis.

Stable, robust, and efficient planar bipedal walking can be achieved with simple control strategies and simple control techniques. Spring Flamingo walked over moderate slopes with no change to its level ground algorithm. For steeper slopes, five simple control strategies were used. Four of these strategies simply modified a parameter of the flat terrain algorithm proportionately to the estimated slope. The other strategy was to bend the support leg when going downhill in order to get the swing-leg to reach the ground.

In the next chapter we show how natural dynamic mechanisms can be used to simplify the control further. One of the mechanisms, knee limits, aided in making height control of the algorithms discussed in this chapter simpler. The next chapter will discuss the benefits more fully.

Chapter 5

Exploiting Natural Dynamics

A powerful practice in machine design is to make use of natural dynamics that make control simpler and more efficient. For example, most planes are designed to have wings so that they naturally fly stably, requiring only a simple power source and simple control. Early locomotives used fly-ball governors, a mechanical feedback device, to help maintain constant speed. Throttles naturally govern speed without explicit feedback. Satellites and rifle bullets spin to stabilize their trajectory.

Animals have evolved similar mechanisms that exploit natural dynamics. Birds have wings which allow for stable gliding. Fish have hydro-dynamically stable shapes and are neutrally buoyant. Running animals have springy legs. And walking animals have knee limits, compliant ankles, and legs that swing passively.

Many researchers [Goswami et al. (1997), McGeer (1990*a*), Garcia et al. (1998), Adolfsson et al. (1998), Fowble & Kuo (1996)], starting with McGeer and his passive dynamic walker, have exploited natural dynamics in order to make walking machines that are fully passive. These devices rely completely on their dynamics and interaction with gravity in order to walk.

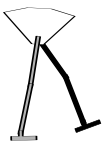
Passive walkers have limitations, of course, such as the need to walk down a slope and the inability to control speed. Powered robots [Hirai et al. (1998), Raibert (1986), Pratt & Pratt (1998*b*), Pratt et al. (1997), Dunn & Howe (1996), Kajita & K.Tani (1996), Kun & Miller (1996), Miura & Shimoyama (1984), Yamaguchi et al. (1994), Yi & Zheng (1996)] can avoid these limitations. However, the control of powered bipedal robots has often been very complicated and the resultant motion often looks unnatural and is inefficient. Many of the controllers for powered robots are model based, requiring an accurate model of the dynamics of the robot and often an accurate model of the environment in order to work. Several of the robots use trajectory tracking approaches which require pre-specified trajectories of either the body or the joints themselves.

In this chapter, we describe an approach to powered bipedal walking which exploits the natural dynamics of the robot and requires only simple control. We exploit three different natural mechanisms. We use a knee-cap to prevent the leg from inverting, which makes control of height easy. We use a compliant ankle limit so that the center of pressure on the foot travels forward with the center of mass of the body. And we exploit the natural swing dynamics of the leg to make swing control simpler and natural looking. This approach was used to control Spring Flamingo, as described below. The robots walk fairly naturally and efficiently.

5.1 Natural Dynamic Mechanisms

5.1.1 Knee Cap

Walking with straight support legs is more efficient than with bent legs since energy requirements in muscles and motors are monotonically related to the torque at the joint, even if there is no velocity. However, since the leg must support the weight of the body, a straight leg poses an interesting challenge. Figure 5-1 illustrates the issue. When the body is directly over the foot (A), no torque is required at the knee. However, this is an unstable latch configuration. If the knee moves slightly



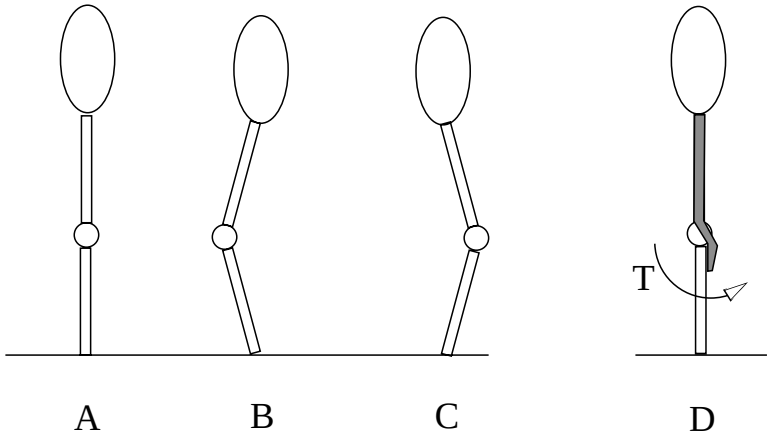


Figure 5-1: Diagram illustrating kneecap advantages. Without a kneecap, a biped with a straight support leg is in an unstable buckling configuration (A). Feedback control may result in chatter between knee inflections B and C due to delay, etc. With a kneecap (D), a constant torque with no feedback is enough to stablize the system against buckling.

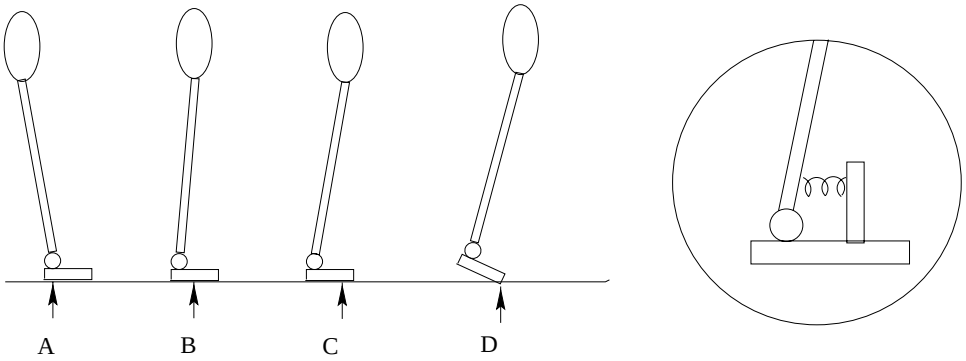


Figure 5-2: Diagram illustrating compliant ankle. In normal walking, the center of pressure on the foot travels forward as the center of mass travels forward (A-D). A compliant ankle (insert) can naturally achieve this effect. However, energy injection at toe off requires actuation.

either way, the leg will buckle (B or C). It is challenging to control this situation. Due to controller non-idealities (bandwidth limitation, delays, etc.) a straight knee controller will typically exhibit chatter between configurations B and C. This is similar to the challenges encountered in haptic devices when trying to simulate a hard surface such as a wall. We want the knee to straighten quickly, but then stop abruptly as though it hit a virtual limit stop. Instead of simulating a virtual limit stop, however, it is easier, more efficient, and more stable to simply have a real limit stop.

We add a knee cap (D) to simplify the control and make the resultant motion smoother and more efficient. In nature, this role is filled by ligaments which prevent the knee from inverting. With a knee limit, a very simple control technique to keep the leg straight is to apply a constant torque so that the knee pushes against the stop. Of course other techniques can be used. Also, if the line of force between the center of mass and the center of pressure on the foot passes in front of the kneecap, the knee will naturally be locked against the kneecap without any actuator torque.

5.1.2 Compliant Ankle

Feet and ankles provide many benefits to bipedal walking. They reduce velocity fluctuations since the center of pressure on the foot can travel forward, staying below the center of mass of the body. They also help to control speed and to inject energy at the end of the stride through toe off.

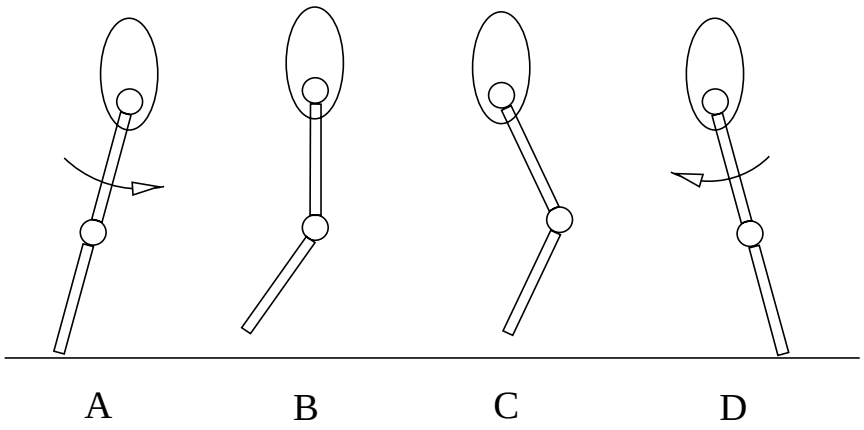


Figure 5-3: Diagram illustrating passive swing. Swing is initiated (A) through a forward torque on the hip, supplied either by hip actuators or gravity. The leg can swing passively (B - C) until swing is stopped (D) through a backward torque on the hip, again supplied either by hip actuators or gravity.

The torque at the ankle can be controlled actively. However, torque requirements can be quite high, since the foot provides a significant lever arm when the center of pressure is near the toe. A compliant ankle provides most of the benefits of a foot and ankle but without the torque requirements. An actuator can then be used in addition to the passive ankle for fine control and energy injection at toe off. Figure 5-2 illustrates the situation. In configuration A, the center of mass is behind the foot and there is zero ankle torque. In configurations B and C, the center of mass is traveling forward. The ankle torque increases, thereby moving the center of pressure of the foot forward from the heel to the toe. In configuration D, the robot goes into toe off, releasing the energy stored in configurations B and C and perhaps injecting some more, through active torques, to maintain walking. The inset shows a simple spring configuration which can give the ankle the desired compliance.

Choosing an adequate spring torque versus displacement curve is important in order to make the center of pressure start moving forward at the desired instance and at the desired rate. In the simulation discussed below we use a quadratic spring ($\tau = k(\theta - \theta_0)^2$) and tune the stiffness parameter. For the robot we use a rubber stop and adjust the position of the stop for best results.

5.1.3 Passive Swing-Leg

In our previous algorithm for Spring Flamingo, described in Chapter 4 [Pratt & Pratt (1998*b*)], as with most powered bipedal walkers, we used control techniques similar to those used for robotic arms to control the swing-leg along a trajectory to a desired landing position. However, with a suitable leg, the natural swing dynamics are such that once the swing starts, the leg will continue without any intervention, as illustrated in Figure 5-3. Gravity alone can be used to initiate swing, as in the case of the passive dynamic walkers. Hip torque can be added in order to make the leg swing faster as in animals.

We use the passive swing properties of the leg in the control of our simulation and robot. In both cases, the hip is servoed forward to a desired angle and the knee is allowed to swing freely. At the end of the swing, moderate damping is added to the knee to prevent it from banging into the knee cap and finally it is locked once it hits the knee cap.

In order for the natural swing property to exist on Spring Flamingo, we had to invert the direction of knee bend. Previously, the robot had a backward bent joint, meant to model the ankle of a bird. However, the robot did not have the joint corresponding to a birds knee. By giving the robot a human style leg configuration, it obtains passive swing characteristics.

In the next sections, we describe algorithms which exploit the three natural mechanisms, de-

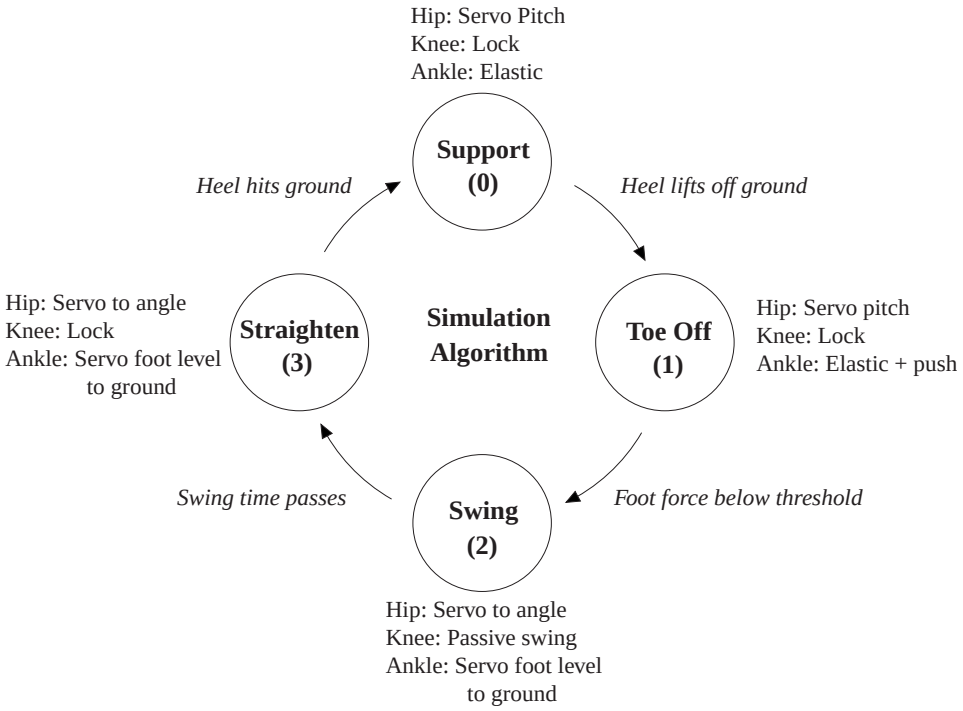


Figure 5-4: SIM Simulation Algorithm. Each leg has a state machine which is in one of four states. State transition conditions and actions in each state are shown.

scribed above, in the control of a simulated and a physical bipedal robot.

5.2 Simulation Algorithm

We use the natural dynamic mechanisms described above in the control of a simulated seven link bipedal robot. The simulation has an actuated hip, knee, and ankle on each leg. It is confined to walk in the sagittal plane. The simulation parameters are listed in Table 5.1. Moments of inertia are measured about the center of mass. The center of mass of each leg link is approximately in the center of the link.

The simulation algorithm is summarized in Figure 5-4. Each leg acts separately and has a simple state machine. The leg can be in either Support, Toe Off, Swing, or Straighten states. In Support and Toe Off states, the hip is used to servo body pitch to maintain balance and the knee is locked to maintain height. In Support state, the ankle is unactuated - only the passive ankle compliance is present. During Toe Off state, the ankle is servoed to an angle using a Proportional-Derivative (PD) controller in addition to its passive compliance. The transition from Support to Toe Off occurs when the heel lifts off the ground due to the passive compliance of the ankle.

The robot transitions from Toe Off to Swing when the force on the foot falls below a certain threshold. In both Swing and Straighten states the hip is servoed to an angle using a PD controller and the foot is servoed to be level with the ground so that the robot does not stub its toe. In Swing state, the knee is damped while in Straighten state the knee is locked straight using a PD controller.

The robot transitions from Swing to Straighten state after a constant amount of time passes. Finally, the robot transitions from Straighten to Support state when the heel of the swing-leg hits the ground.

The simulation parameters were first manually tuned, and then fine tuned using a genetic algorithm with efficiency as its cost function. Efficiency was computed as distance traveled divided by total joint energy after ten seconds of walking. Total joint energy was computed by integrating the total joint power which is the sum of the absolute values of the mechanical power at each joint:

Table 5.1: SIM Physical parameters and controller parameters of the simulated planar bipedal walker. The controller parameters were tuned both manually and with a genetic algorithm.

Physical Parameter	Value
Total mass	14.2 kg
Body mass	12.0 kg
Hip to body center of mass	0.20 m
Body moment of inertia	0.10 kg m^2
Upper leg mass	0.46 kg
Upper leg moment of inertia	0.13 kg m^2
Upper leg length	0.42 m
Lower leg mass	0.31 kg
Lower leg moment of inertia	0.0095 kg m^2
Lower leg length	0.42 m
Foot mass	0.35 kg
Foot moment of inertia	0.0014 kg m^2
Foot height	0.04 m
Foot length forward	0.17
Foot length back	0.06

Controller Parameter	Manual	GA
hip swing angle	0.5 rad	0.56
hip gain	15.0	12.9
hip damp	1.0	1.76
hip hold angle	0.35	0.40
knee lock gain	1.0	0.68
knee damping	0.5	0.68
foot force threshold	20.0	22.5
swing time	0.4	0.45
ankle push gain	4.0	5.07
ankle compliance	1000	384
pitch gain	100	(same)
pitch damp	20	(same)
knee stance gain	30	(same)
knee stance damp	10	(same)

$$E_{tot} = \int P_{tot}\, dt, \quad P_{tot} = \sum_{joints} |P_{joint}|, \quad P_{joint} = \tau_{joint}\, \dot{\theta}_{joint} \tag{5.1}$$

After a couple generations, naturally looking simulations resulted. Some of them achieved a smooth transfer of support by coordinating toe off and heel strike. One of those solutions was chosen to be presented here as it appeared quite natural. A time elapsed animation is shown in Figure 5-5. The drawings on the left show the swing phase of one leg. The drawings on the right show several steps. The right leg is dotted while the left leg is solid. Lines show the path of the tips of the feet and the hip trajectory. Several variables are plotted graphically in Figure 5-6. In the first row, left to right, are the state of the left leg (solid) and right leg (dotted), forward distance traveled (x), and velocity. In the second row are total power, body height, and pitch. In the last row are hip, knee, and ankle power of the left leg.

We see that the simulated robot walked at a moderate speed (approximately 0.8 m/s) and had a natural looking gait. The simulation requires less than 20 watts of joint power on average. Most of the energy consumption comes from the hips at the beginning of swing phase. It is interesting that the algorithm does not contain any explicit speed control mechanism, yet speed is stabilized. We speculate that this is due to the natural system dynamics, in the same way that speed is naturally stabilized in the passive dynamic walkers.

5.3 Robot Algorithm

Figure 5-7 contains photographs of Spring Flamingo, a seven link planar bipedal walking robot. The photo on the left is a close up of the knee joint and kneecap. The photo on the right is a close up of the compliant ankle stop. We exploit these two natural mechanisms, and the passive swing in the control of the robot.

The robot algorithm is summarized in Figure 5-8. As in the simulation algorithm, each leg acts separately and has a simple state machine. The robot algorithm differs from the simulation algorithm in two ways. First, because the robot does not have ground force sensors yet, we use geometric state transition conditions for initiation of swing. Second, instead of controlling each joint

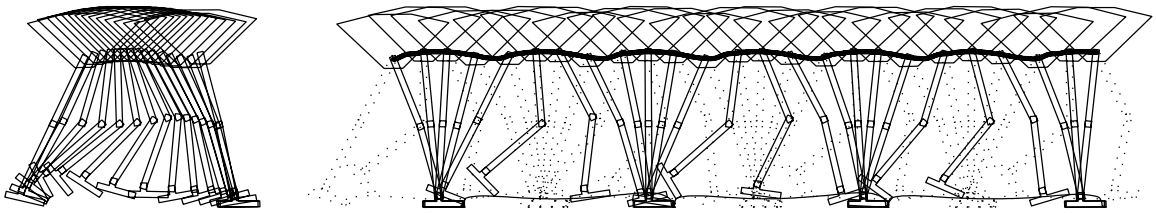


Figure 5-5: SIM Elapsed time snapshot of the simulated robot walking data. The drawings on the left are spaced approximately 0.08 seconds apart and show the swing phase of one leg. The drawings on the right are spaced approximately 0.25 seconds apart. The right leg is dotted while the left leg is solid. Lines show the path of the tips of the feet and the hip trajectory. The robot walks from left to right at approximately $0.8\frac{m}{s}$.

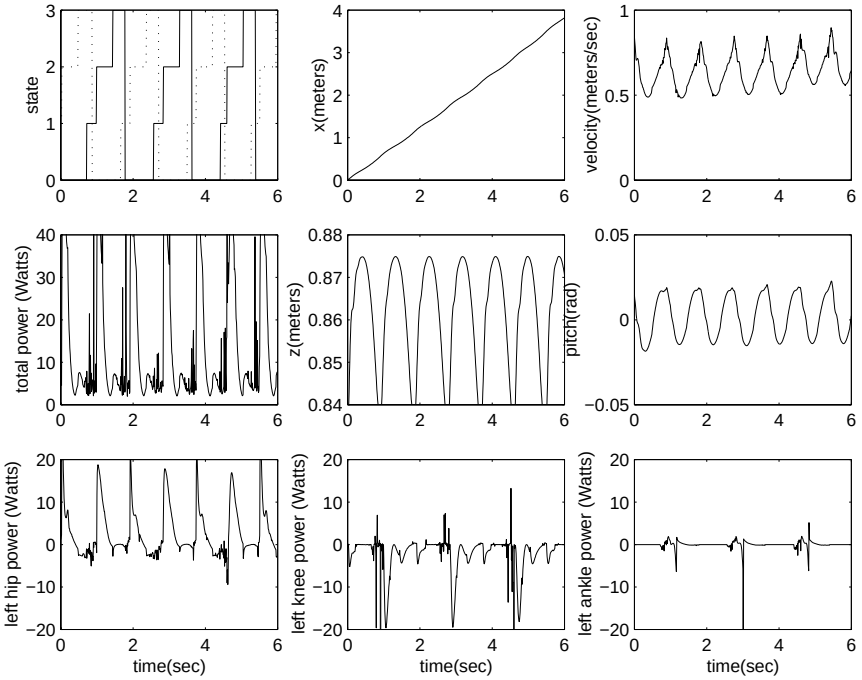


Figure 5-6: SIM Planar simulation data exploiting natural dynamics. The first row contains, left to right, state of the legs, forward distance traveled, and forward velocity. The second row contains total power, body height, and body pitch. The last row contains joint power at the hip, knee, and ankle of the left leg.

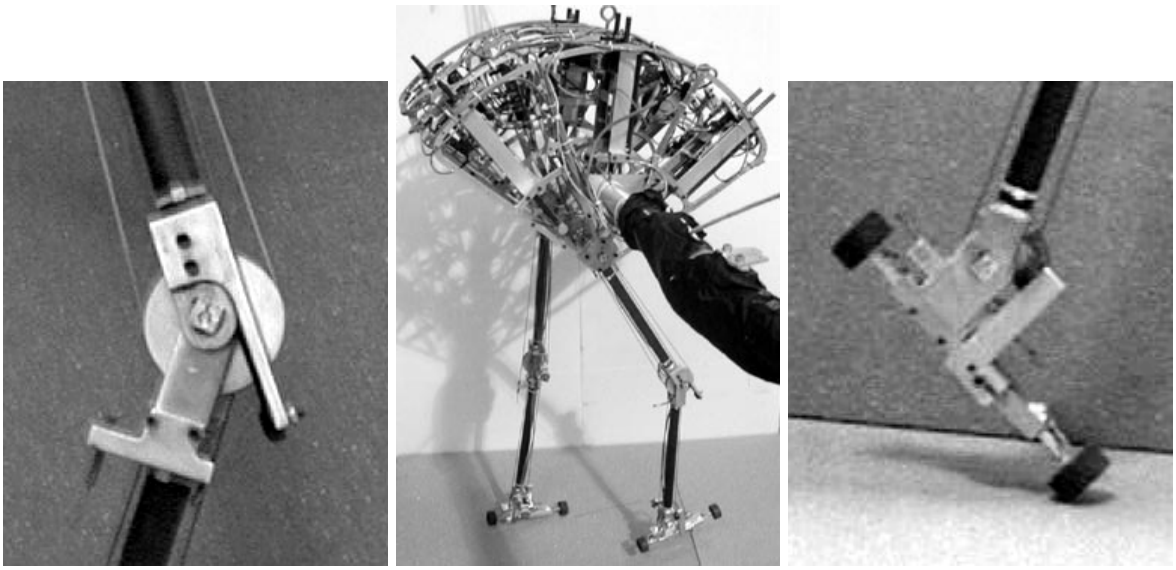


Figure 5-7: Spring Flamingo photograph. There are six force controlled actuators attached to the body. Power is transmitted to the hips, knees, and ankles via cables. A boom prevents motion in the lateral, roll, and yaw directions. A knee-cap (left) prevents the knee from hyper-extending, while a compliant ankle limit (right) helps naturally transfer the center of pressure forward as the robot travels forward over the foot.

separately, in a decoupled manner, we use Virtual Model Control [Pratt et al. (1997), Pratt & Pratt (1998*b*)] in the control of pitch and height.

The control parameters for the physical robot were manually tuned over the course of two days. Stable solutions were found that were fairly robust such that the robot walked continuously on level ground. A time elapsed animation of the robot is shown in Figure 5-9. The drawings on the left show the swing phase of one leg. The drawings on the right show several steps. Several variables are plotted graphically in Figure 5-10. In the first row, left to right, are the state of the left leg (solid) and right leg (dotted), forward distance traveled (x), and velocity. In the second row are total power, body height, and pitch. In the last row are hip, knee, and ankle power of the left leg.

We see that the physical robot walked at a casual speed (approximately 0.5 m/s) and had a natural looking gait. As in the simulation, it is interesting that the algorithm does not contain any explicit speed control mechanism, yet speed is stabilized. The robot requires less than 5 Watts of joint power on average. As in the simulation, most of the energy consumption comes from the hips at the beginning of swing phase. We can compare the energy consumption to other vehicles by computing the specific resistance

$$\epsilon = \frac{P}{WV} = \frac{5}{9.81 * 14 * 0.5} = 0.07 \tag{5.2}$$

This specific resistance is plotted versus velocity in Figure 5-11 along with other vehicles. However, this power is mechanical power at the joints, not electrical input power. If we measure the power from the wall outlet, over 200 Watts is required! This is due to inefficies in the power supplies, motor amplifiers, and motors. Making the robot more efficient would take a lot of further engineering work.

5.4 Conclusions

Spring Flamingo walked continuously by exploiting the natural dynamics of a kneecap, compliant ankle, and passive swing-leg. Two very simple algorithms are used in simulation and on the physical

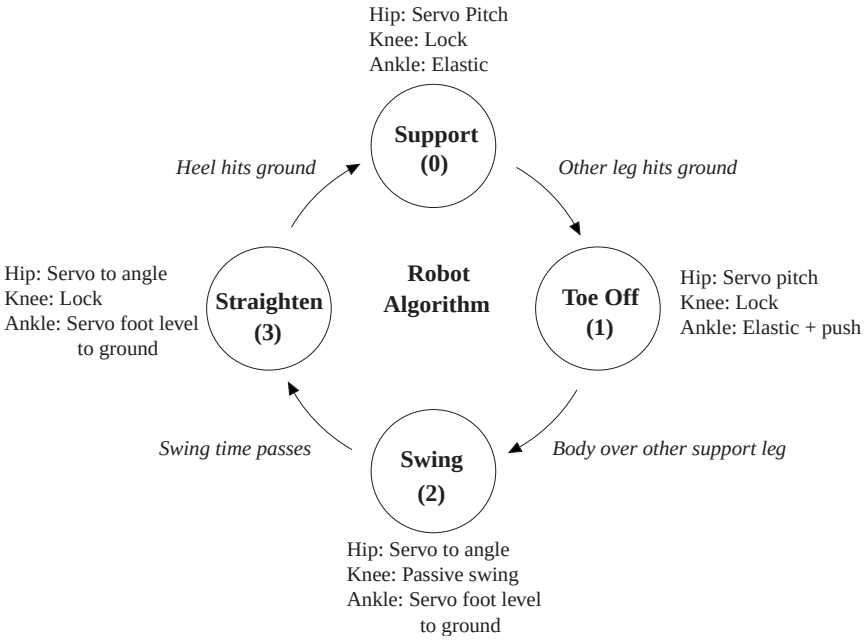


Figure 5-8: REAL Physical robot algorithm exploiting natural dynamics. Each leg has a state machine which is in one of four states. State transition conditions and actions in each state are shown.

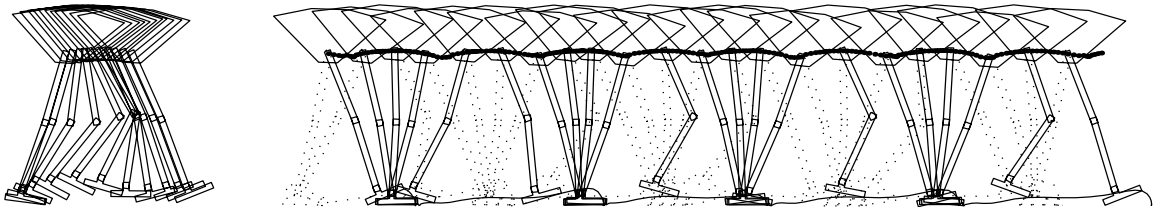


Figure 5-9: REAL Elapsed time snapshot of the physical robot walking data exploiting natural dynamics. The drawings on the left are spaced approximately 0.07 seconds apart and show the swing phase of one leg. The drawings on the right are spaced approximately 0.4 seconds apart. The robot walks from left to right at approximately $0.5\frac{m}{s}$.

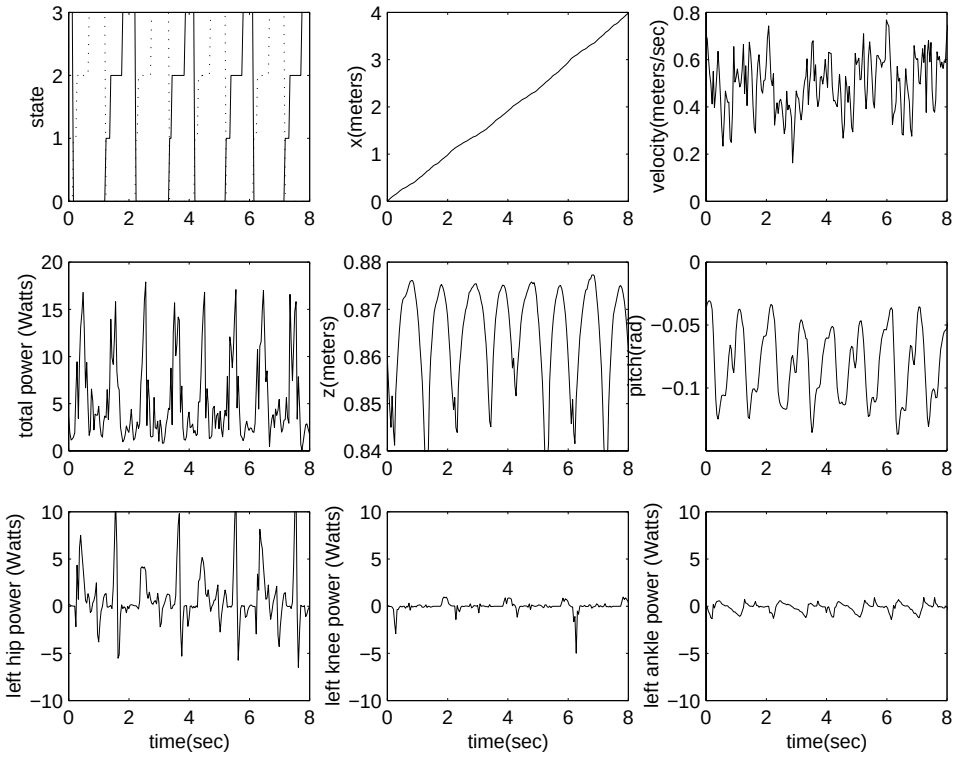


Figure 5-10: REAL Physical robot walking data exploiting natural dynamics.

robot. The resultant motion is fairly smooth and efficient. This work may help bridge the gap between passive dynamic walkers and powered bipedal robots.

Both the simulation and the robot settle on a stable speed of walking. The simulation walks approximately 0.8 m/s while the robot walks approximately 0.5 m/s. However, nowhere in either controller is speed explicitly controlled. We believe that the speed is stabilized in a similar way to passive dynamic walking machines. That is, if the robot goes too fast, it naturally takes a longer step due to the natural swing-leg dynamics and hence slows down on the next step. Similarly, if the robot moves too slowly, it naturally takes a shorter step and hence speeds up on the next step.

In order to exploit the natural dynamics of a walking robot, it is important that the inertia and friction of the actuators does not dominate the dynamics of the legs. On Spring Flamingo we use Series Elastic Actuators which result in very little unwanted dynamics. These actuators have good force dynamic range and low force offset which are important in natural and efficient walking.

Both the simulation and the physical robot required tuning of parameters in order to walk. The simulation was tuned first manually and then fine tuned using a genetic algorithm with efficiency as the cost function. The genetic algorithm found a set of parameters which nicely synchronized toe off of the support leg with heel strike of the swing-leg such that a smooth transfer of support occurred and minimal energy was lost. The robot was only tuned manually but still walked fairly well. Its joint power averaged less than 5 Watts.

Spring Flamingo demonstrates that natural mechanisms can be exploited to make control of bipedal robots simple and the resultant motion natural looking and efficient. In Chapter 7, we expand on this algorithm to get the robot to walk quickly ($1.25\frac{m}{s}$). We do this through active control of velocity, more active control of the swing-hip, and more active control of the ankle during toe off. In Chapter 8, we describe a similar algorithm for three dimensional bipedal walking.

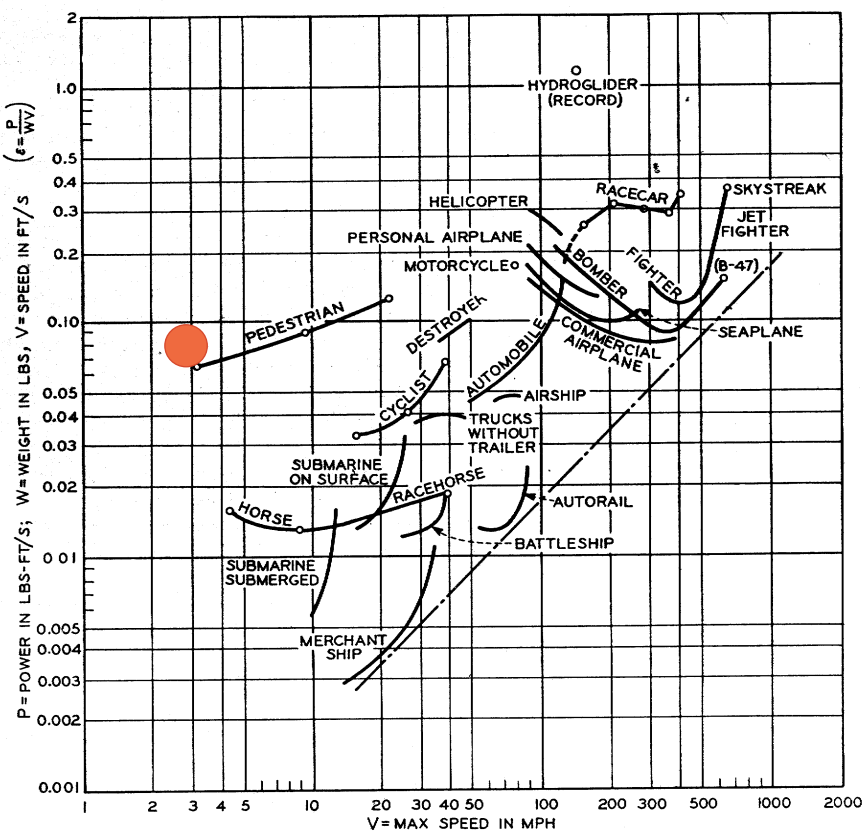


FIG. 3 SPECIFIC RESISTANCE OF SINGLE VEHICLES

Figure 5-11: REAL Specific resistance of Spring Flamingo (filled circle) when walking exploiting its natural dynamics. The robot walks at the speed of a slow pedestrian and has approximately the same specific resistance. However, this value is computed from mechanical power and not from input electrical power. Figure taken from Gabriella and Von Karmen (1950).

Chapter 6

Limits to Speed in Bipedal Walking

Speed appeals to the imagination and competitive spirit. It excites and challenges inventors and engineers to devise new methods of transport and improve on existing ones, if for no other reason than to conquer speed records. In the process of conquering such records, the limits to speed become more apparent and more fully understood.

In the case of bipedal walking, building and controlling fast robots demands a deeper understanding of bipedal walking than it does for slow robots. Fast walkers require dynamic balance while slow ones may walk statically; they require more robustness as disturbances have a greater dynamic effect and less time is available to correct for disturbances. Building fast bipedal walking robots not only makes them more useful as machines but also ensures that the developer understands the issues which are important in bipedal walking.

Koechling & Raibert (1988) studied the limits to speed in bipedal running robots. They cited leg length, leg stiffness, maximum leg force, leg sweep angle and joint velocity as the limiting factors in running speed. However, most of their analysis focussed on the stance leg. Minimum swing-time was not cited as a limiting factor in running speed. Perhaps this was overlooked as their robot had powerful hydraulic legs that could swing sufficiently fast.

Bipedal walking is very different than bipedal running. Since there is no aerial phase, the stride length is limited to how far the body can move while being supported on a leg and the stride time is limited by how fast the swing-leg can be positioned for the next stride. In this chapter we hypothesize that minimum swing-time and maximum stride length are the major limiting factors to bipedal walking speed, both robotic and human, on Earth.

As a person increases their walking speed they hit a point where they are walking with maximum stride length and minimum swing-time. At higher speeds, they can not get the swing-leg out in front fast enough. If speed is to be increased further, a running or skipping gait must be used. Such gaits provide an aerial phase which allows for both a longer stride length and more time to swing the leg, since the swing-leg can swing both during the stance phase and the aerial phase.

6.1 Non-Dimensional Parameters: Froude Number and Pratt Number

Walking data is commonly expressed in dimensionless quantities, in order to normalize for different shapes and sizes of walkers. For example, stride length is often normalized with leg length to give relative stride length.

Speed is often normalized with gravity and leg length to produce a dimensionless quantity called the Froude Number. This quantity can be viewed as a ratio between centrifugal and gravitational forces, or as a ratio between kinetic and potential energies:

$$Fr = \left(\frac{mv^2}{L_{leg}}\right)/(mg) = \frac{v^2}{gL_{leg}} \tag{6.1}$$

where m is body mass, g is gravitational acceleration, v is velocity, and L_{leg} is leg length. (In some of the figures in this chapter we will plot various quantities against the Froude Number.)

Froude Number is often used as a scaling parameter in comparing animals of different sizes. A maximum Froude Number limit is often cited as why astronauts had a hard time walking on the moon: As gravity goes down, maximum velocity must go down in order to maintain a constant Froude Number. At higher Froude Number, centrifugal force rises above gravitational force and the astronauts must hop.

However, we hypothesize that another dimensionless quantity is important for determining maximum walking speed at Earth's gravity. This quantity, which we refer to here as the Pratt Number, is velocity normalized with minimum swing-time and leg length:

$$Pr = \frac{T_{swmin}\, v}{L_{leg}} \tag{6.2}$$

where T_{swmin} is the minimum time it takes for the person to swing their leg a given distance while *not* walking. In this Chapter, we choose a distance of $0.75m$ (in hindsight it would have been better to choose a percentage of leg length).

We found that at maximum walking speed for several human subjects there was more variation in the maximum Froude Number than the maximum Pratt Number. In Section 6.5 we show results comparing the predictive value of these two non-dimensional parameters for maximum walking speed.

6.2 Stride Length and Stride Time Scaling with Speed

Average walking velocity is the stride length divided by the stride time,

$$V = \frac{L_{stride}}{T_{stride}} \tag{6.3}$$

Therefore, to increase speed, one can either take a longer stride or decrease the stride time. Humans do both as they increase their speed. Figure 6-1 shows data collected from 10 individuals whose vital statistics are shown in Table 6.1.

The subjects were asked to walk on a 100 meter track at several different speeds from very slowly to as fast as they could possibly walk without significantly bending their knees. The subjects were recorded on video tape. The average speed, stride length, and stride times were computed using the total distance, total time, and number of steps.

In Figure 6-1, normalized stride length, $\frac{L_{stride}}{L_{leg}}$, (left) and stride time (right) are plotted against Froude Number. Maximum stride length was approximately 1.1 times the leg length while minimum stride time was approximately 0.35 to 0.40 seconds. Maximum Froude number achieved varied from 0.6 to 1.3.

6.3 Limits to Stride Length

Since walking contains a double support phase, the geometrical maximum stride length is twice the length of the leg plus the distance between the hips. Of course, such a stride would require the hip height to lower to the level of the ground and the pelvis to rotate 180^o every step! From Figure 6-1, we see that our test subjects have a stride length of approximately 90% of their leg length when walking with a normal gait at a casual pace. At maximum speed, their stride length increases to approximately 1.1 times their leg length.

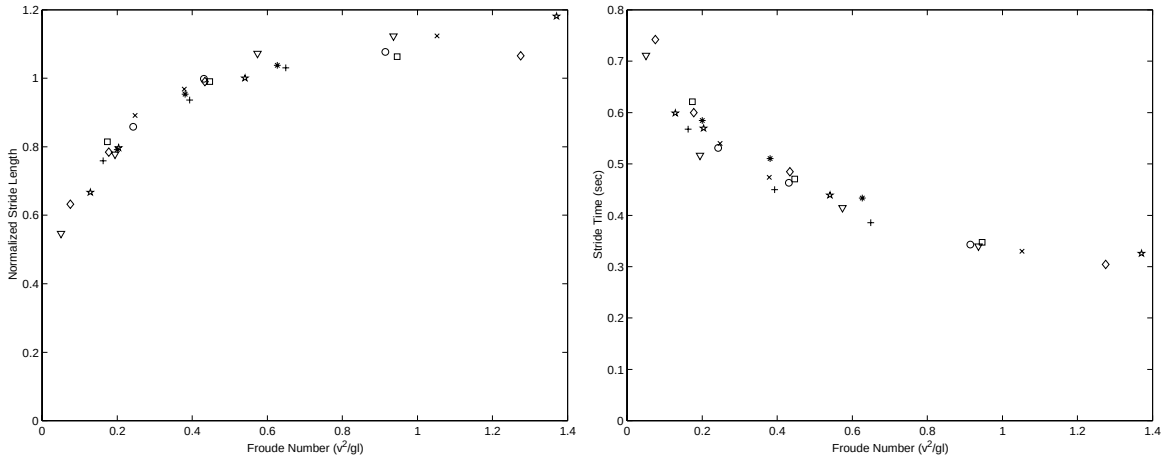


Figure 6-1: Scaling of stride length and stride time with speed in normal human walking. At maximum speed the maximum stride length is approximately 1.1 times the leg length and the minimum stride time is approximately 0.35 seconds.

The rimless wheel model of walking [McGeer (1990*a*)] provides insight into the stride length limit. Suppose a rimless wheel (Figure 6-2) with rim separation angle of $2\alpha_0$ is rolling on flat ground. As McGeer shows using conservation of angular momentum, each time a spoke hits the ground there is an impulse which reduces the rotation rate Ω of the wheel,

$$\frac{\Omega^+}{\Omega^-} = \frac{\cos 2\alpha_0 + r_{gyr}^2}{1 + r_{gyr}^2} = \eta \tag{6.4}$$

where r_{gyr} is the wheel's radius of gyration normalized by the leg length. Much of a human's mass is in the upper body which does not rotate like a rimless wheel would. Therefore, the effective radius of gyration is low and can be approximated as zero. In order to maintain a velocity then, we get $\alpha_0 < 90^o$ which limits the stride length $L_{stride} < \sqrt{2}L_{leg}$. At this limit, the rimless wheel would come to a standstill on each impact.

The rimless wheel model assumes point contact with the ground. In a human with a foot the effective pivot point (center of pressure) can lie anywhere inside the foot. Therefore the maximum stride length is increased by the length of the foot. Weyand et al. (1999) argues that this is one reason why humans take longer strides and are more efficient walkers than rheas, ostrich-like bipedal birds which have shorter feet than humans. Therefore, an upper bound on the stride length during normal walking is $\sqrt{2}L_{leg} + L_{foot}$. Of course the real limit will be lower than this since at this stride length velocity goes to zero on each impact. From Figure 6-1 we see that the maximum stride length for normal walking in the test subjects is approximately 1.2 times the leg length.

6.3.1 Groucho Running

The above analysis on maximum stride length was based on a rimless wheel model with a pure compass gait. By flattening the center of mass trajectory, longer stride lengths can be achieved as less energy is lost on the foot-ground impact. This is the case with Groucho Running [McMahon et al. (1987)]. During normal walking, the center of mass reaches a *maximum* height near mid-stride. For Groucho Running, the center of mass reaches a *minimum* height near mid-stride. Like normal running, potential and kinetic energy are in phase during Groucho Running. Therefore, even though there is no aerial phase, Groucho Running is more energetically similar to running than to walking.

In Groucho Running, maximum stride length should increase, resulting in a larger maximum speed. However, if Groucho Running is performed while limiting the stride length, speed should not be increased significantly, as bent knees increase walking speed mostly by increasing maximum

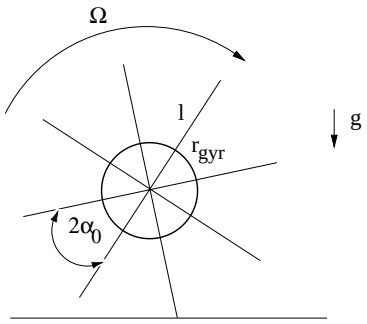


Figure 6-2: McGeer's rimless wheel model of walking. Leg separation angle is $2\alpha_0$. l is the leg length. r_{gyr} is the radius of gyration normalized with the leg length.

Subject	Leg Length	Height	Mass	Age	Sex
1	$0.91m$	$1.75m$	$67.7kg$	$24y$	Male
2	$0.89m$	$1.68m$	$75.0kg$	$33y$	Male
3	$0.89m$	$1.68m$	$65.9kg$	$26y$	Female
4	$1.07m$	$1.98m$	$114.0kg$	$29y$	Male
5	$0.99m$	$1.83m$	$82.7kg$	$27y$	Male
6	$1.02m$	$1.78m$	$77.3kg$	$28y$	Male
7	$0.84m$	$1.65m$	$63.6kg$	$29y$	Male
8	$0.89m$	$1.75m$	$54.5kg$	$33y$	Male
9	$0.97m$	$1.83m$	$72.7kg$	$28y$	Male
10	$0.84m$	$1.61m$	$50.0kg$	$29y$	Female

Table 6.1: Vital Statistics of Human subjects used in the walking experiments.

stride length. However, other secondary effects of bent knees are decreasing swing-leg inertia about the hip and having the knee collapse before swing, which may reduce swing-time.

To test the hypothesis that increased stride length is the primary benefit to maximum walking speed with bent knees, we had five human subjects walk in various fashions on a 100 meter track. They were asked to:

1. Walk as fast as possible without significantly bending one's knees.

2. Walk as fast as possible while significantly bending one's knees (Groucho Run).

3. Groucho Run with stride length approximately equal to stride length in the fast walking experiment.

Figure 6-3 shows maximum velocity, stride length and stride time for human subjects walking normally and Groucho Running. We see that the average stride length when walking fast but normally is approximately 1.1 times the leg length. When Groucho Running, the stride length increases, usually resulting in an increase in velocity.

When Groucho Running with a reduced stride length, the subjects only achieved a speed slightly faster than their top normal walking speed. In all cases, the stride time was lowest in the normal walking case. Therefore, we hypothesize that it is mostly the increase in stride length and not other factors which makes Groucho Running faster than normal walking.

We conclude that limited stride length limits human walking speed. Groucho Running can be used to extend the stride length but is typically not used by humans since it is less efficient than normal running.

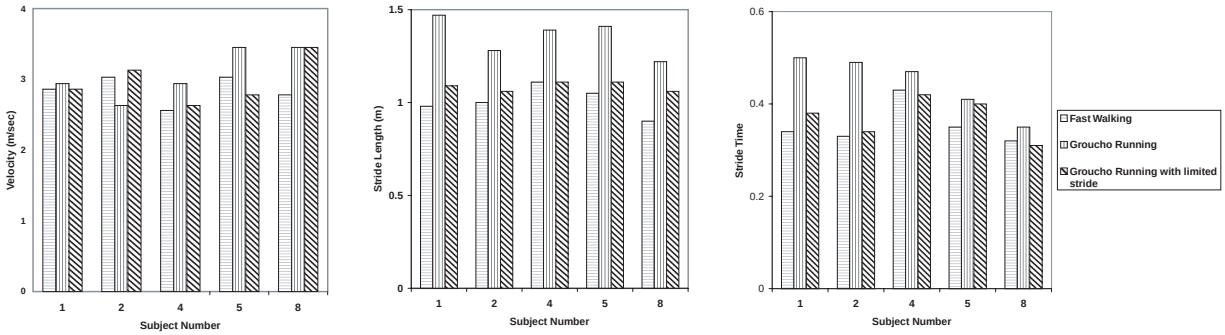


Figure 6-3: Results of Groucho Running experiments. Groucho Running is faster than normal walking since stride length is increased. However, stride time is longer when Groucho Running. When stride length is artificially limited, Groucho Running is no faster than normal walking, since stride time is longer.

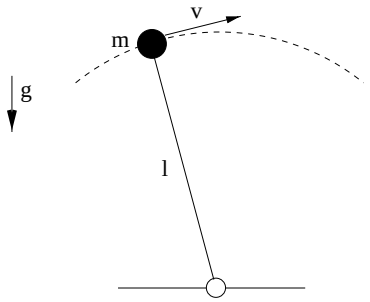


Figure 6-4: Rotating pendulum model for estimating walking speed limit. Centrifugal force of $\frac{mv^2}{l}$ will equal the gravity force of mg at the top of the arc if $v^2 = gl$ (Froude Number equals 1.0). At that speed the force on the foot will fall to zero.

6.4 Limits to Stride Time

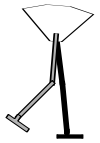
Figure 6-1 shows that at the fastest walking speed, stride time is approximately 0.35 to 0.40 seconds. There are several things which may limit minimum stride time:

1. Stance leg strength and power may limit how fast the body can move and still be supported by the stance leg.
2. At high speed the ground reaction force may fall due to centrifugal force from the body rotating like an inverted pendulum.
3. The swing-leg may have a minimum swing-time due to leg inertia, strength, and coordination.

The first of these is unlikely since a person can move much more quickly on the same support leg during running than during walking.

The second limitation is a real limitation in low gravity such as on the Moon. If the body is moving at velocity v, along a perfect arc as in Figure 6-4, then at the top of the arc, the upward centrifugal force is $\frac{mv^2}{l}$. This force will be greater than the gravity force mg when $v^2 = gl$, i.e. when the Froude Number, $\frac{v^2}{gl}$, is greater than 1.0. For a person with a leg length of $1.0m$, this corresponds to a maximum Earth speed of $3.3\frac{m}{s}$ which is above but close to the walk to run transition speed. However, the center of mass in normal walking does not follow a perfect arc but rather follows a significantly flattened arc [Lee & Farley (1998)].

We hypothesize that neither of the first two limitations are reached in normal human walking on Earth. Instead, we believe that the third limitation, minimum swing-time, is the limitation for stride



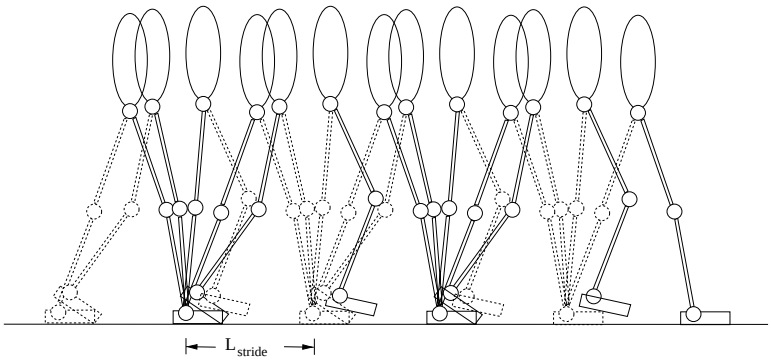


Figure 6-5: Cartoon of standard human bipedal walking. Two full walking cycles are shown. L_{stride} is the stride length. With standard walking swing-time is limited to the stride time.

time of human walking on Earth. When the speed is too great, one can not get the swing-leg out in front fast enough and must revert to running. We demonstrate that swing-time is a hard limit by showing that faster walking can be achieved by artificially allowing for longer swing-time. In fact, Froude Numbers of 1.5 can be achieved without significantly changing the stance leg characteristics or stride length. Higher numbers can be achieved if significant knee bend (Groucho Running) is allowed, even if longer stride lengths are not used. This may be due to flattening of the body trajectory, or perhaps at high speeds larger knee-break is required upon landing, or perhaps some other factor. We can achieve longer swing-times for a given speed through Three-Legged Walking.

6.4.1 Three-Legged Walking

Figure 6-5 shows a time elapsed cartoon drawing of normal bipedal walking. If we define the swing-leg to be in the swing phase at the same instant that the next stance leg touches the ground, then the swing-time is equal to the stance time. At fast walking speeds, humans tend to walk this way and the double support phase nearly vanishes. Therefore for this discussion we will assume no double support phase.

Suppose there was a human with three legs, as in Figure 6-6. At a fast walking speed, each of the three legs would be in stance phase for $\frac{1}{3}$ and swing for $\frac{2}{3}$ of the walk cycle. Figure 6-7 shows each of the legs separately while Figure 6-8 shows a cartoon animation of the walking. Assuming that the swing-time for a three-legged human is the same as for a two-legged human, each stride can be twice as fast and still allow time for the swing-leg to be in position for its new stance phase.

The only difference in the gait of each leg between the three-legged walker and the two-legged walker is the swing phase. Thus, if swing-time is a limiting factor in walking speed then the three-legged person should be able to walk up to twice as fast.

Three-legged walking can be simulated with a real person by equipping them with a pair of crutches to use as their third leg. Figure 6-9 shows a snapshot of the author walking on a treadmill performing three-legged crutching. We tested five subjects by having them walk in various fashions on a 100 meter track. The subjects were asked to

1. Walk as fast as possible without significantly bending one's knees.

2. Three-legged walk as fast as possible without significantly bending one's knees.

3. Three-legged walk as fast as possible while significantly bending one's knees.

The average speed, stride length, and stride time were computed as before. The results are shown in Figure 6-10.

We see that three-legged fast crutching was faster across subjects than normal fast walking. We hypothesize that the increase in speed came from a decrease in stride time which was due to having

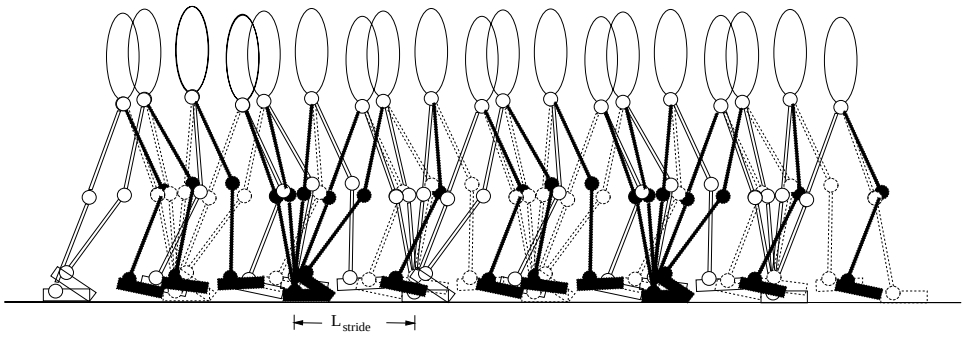


Figure 6-6: Cartoon of human walking with three legs. Two full walking cycles are shown. L_{stride} is the stride length. With three-legged walking swing-time is limited to twice the stride time. Therefore, if minimum swing-time is a limitation of human walking, a three-legged human should be able to walk twice as fast as a two-legged human.

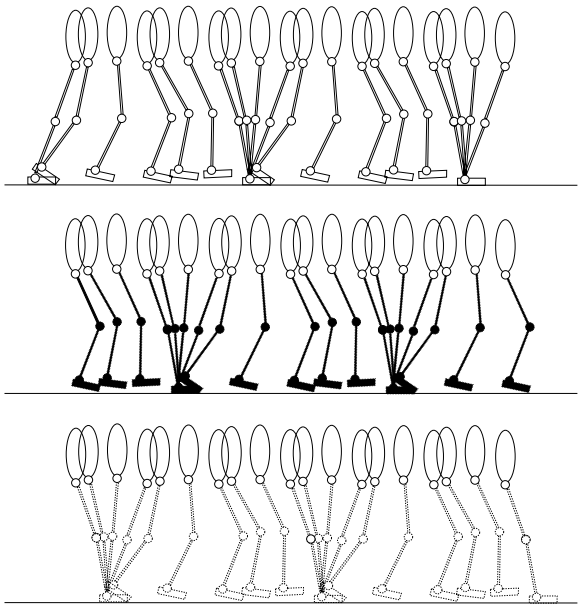
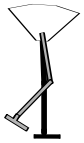


Figure 6-7: Three-legged walking showing each leg separately for clarity. We see that the swing-leg can move more slowly with respect to the stance leg than with two-legged walking.



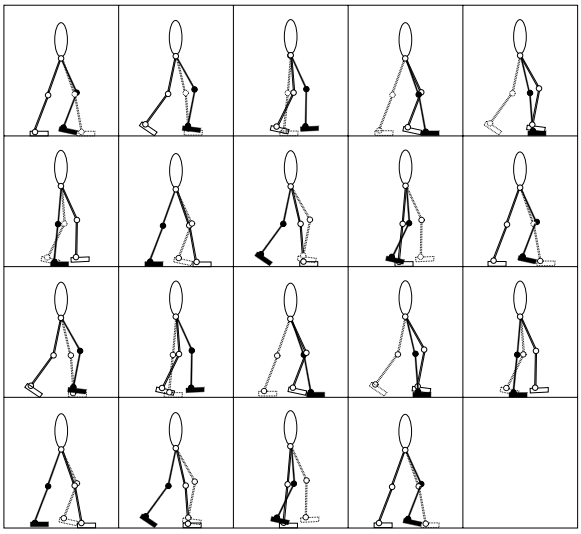


Figure 6-8: Stop frame animation of three-legged walking. Notice that in some frames there are two legs swinging at the same time and that in all frames at least one leg is swinging.

Figure 6-9: Video frames of a test subject (the author) three-legged walking on a treadmill. Images proceed left to right, top to bottom.

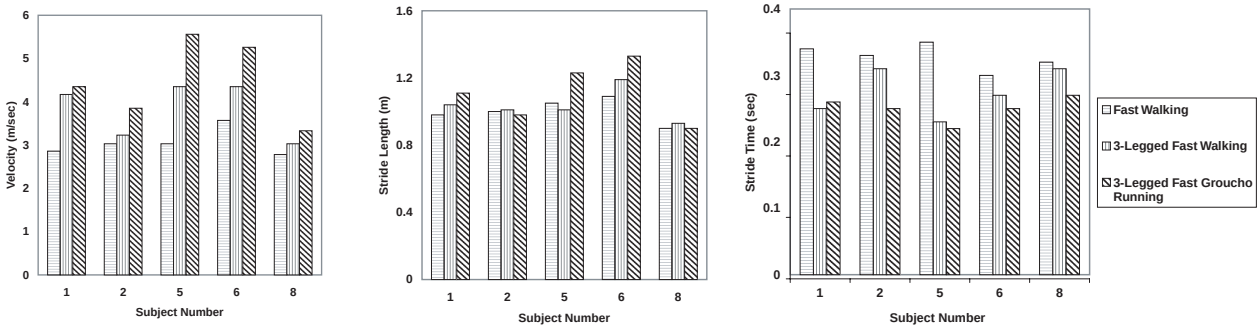


Figure 6-10: Results of fast walking, three-legged walking, and three-legged Groucho Running. Three-legged walking is faster than normal walking suggesting that swing-time is a limiting factor in walking speed. Stride time is decreased and stride length are increased in three-legged Groucho Running, making it even faster than normal walking.

twice as much time available to swing the leg. Stride length was about the same as in normal walking. This suggests that minimum swing-time is a limiting factor of top walking speed on Earth.

By combining the effects of three-legged walking and Groucho Running, the subjects were able to increase their speed even further. The fastest three-legged walking subject (the author) was able to move at a velocity of $5.56\frac{m}{s}$ (100 meters in 18 seconds – only 4 seconds slower than the subject could run 100 meters). Assuming the pace could be maintained, this time corresponds to a mile time of 4 minutes and 48 seconds, nearly a minute faster than the world record race walking mile! Of course, such a walk would not be allowed in speed walking races as knee bend is not allowed, nor are three legs.

6.4.2 What Limits Swing-Time?

In order for the swing-leg to complete a swing phase, the thigh, shin, and foot must be accelerated, swung through, and decelerated. The fastest this can be done may be limited by leg inertia, control bandwidth, maximum muscle speed, maximum muscle torque, and coordination of the swing-leg.

We asked 10 human subjects to step forward and backward as quickly as possible 10 times at three different stride lengths. The average time to swing the leg was calculated. Results are plotted in Figure 6-11. We see that minimum swing-time depends on swing distance.

Minimum swing-time is hard to predict for a given person since it not only depends on leg inertia but also on factors such as the ratio of fast twitch to slow twitch muscle fibers, coordination, skill, and training. We see that there is a large variation across subjects in minimum swing time and we cite this as a reason for the variation in maximum walking speed.

6.5 Predicting Maximum Walking Speed

Not surprisingly, the subjects with the minimum swing-time also had the fastest walking speed for their leg length. Therefore, it is reasonable to expect the Pratt Number to be a useful predictor of maximum walking speed. Figure 6-12 shows results from fast walking experiments with ten subjects. The top graphs show actual maximum walking speed and estimated speed based on the maximum Pratt number limit (left) and Froude number limit (right). Although the Pratt number prediction is a closer line fit, both seem like good candidates for predicting speed, especially considering the margins of error. We see that relative stride length and stride time is fairly constant between individuals. Maximum Pratt number is fairly constant while maximum Froude number varies a bit. We plot the square root of the Froude number to remove the squaring factor on velocity, resulting in a more fair comparison.

From Figure 6-12 there is not enough of a range of data to conclusively determine whether Pratt

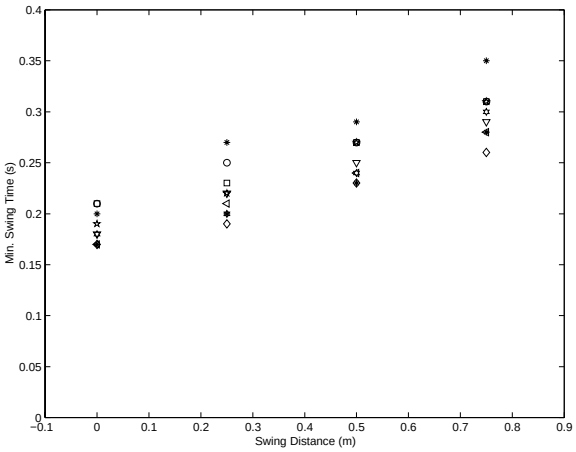


Figure 6-11: Swing Limits Data. Minimum swing time increases with swing distance. There is a large variation among subjects for minimum swing time. We argue that this variation is responsible for the variation in maximum walking speeds among different subjects.

Number or Froude Number is a better predictor of maximum walking speed. However, we can perform a simple experiment in which the subjects walk with various ankle weights. In this case the maximum speed predicted by Froude Number will not change since gravity and leg length both remain constant. However, Pratt Number will predict a slower speed since the minimum swing time should increase.

Figure 6-13 shows results from four subjects walking fast without ankle weights, with three different ankle weights, and with the ankle weights in a backpack. As the weight increased, both minimum swing time and stride time increased in such a way that stride time, relative to minimum swing time, remained fairly constant. Stride length remained fairly constant despite differing ankle weights. Therefore, Pratt number remained fairly constant while Froude number varied. Thus, in this case, Pratt Number is a much better predictor of maximum walking speed than Froude Number.

6.6 Discussion

We hypothesize and have shown evidence from our test subjects that the major limiting factors for bipedal walking speed on Earth are stride length and swing-time. For normal, non-Groucho walking, stride length is limited to approximately 1.1 times the leg length. Swing-time is limited by the inertia, strength and coordination of the swing-leg, but only slightly affected by gravity. For most subjects, minimum swing-time was between 0.25 and 0.35 seconds. For our subjects tested, Pratt Number varied under 15% while Froude Number varied by a factor of two. Therefore it seems that Pratt Number may be a better predictor of maximum walking speed than Froude Number on the Earth.

6.6.1 Scaling Maximum Speed with Gravity (Will we Walk on Mars? Or will we Hop?)

We have argued that at Earth's gravity, swing-time and stride length are the limiting factors for maximum walking speed. Since neither of these is highly dependent on gravity, the maximum walking speed near Earth's gravity should then be mostly independent of gravity.

At low gravity, Froude Number becomes important as losing foot-ground contact due to centrifugal force is the limiting factor in walking speed. This is cited as the major reason why the astronauts on the moon found a hopping gait preferable to a walking gait. The limiting value of Froude Number is is still an open question. From the simple pendulum model, we get $v^2 < gL_{leg}$ (Froude Number less than 1.0). But our fastest walking subject walked at Froude Numbers of about

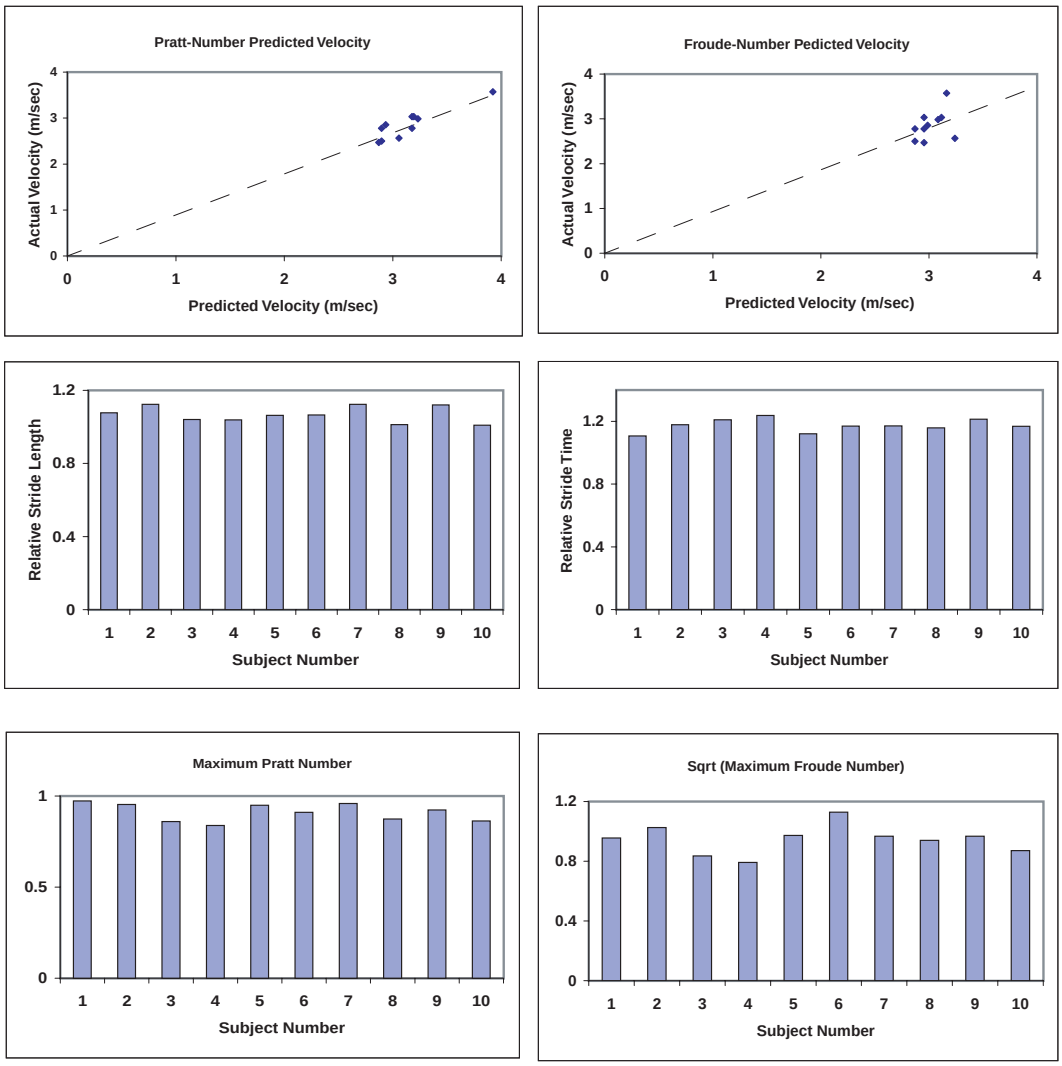


Figure 6-12: Results from maximum velocity walking experiments. Ten subject were asked to walk as fast as possible on a 100 meter track. Pratt Number appears as a potentially better predictor of maximum walking speed than Froude Number (upper row of graphs). There is low variation in the stride length relative to leg length and the stride time relative to minimum swing time (second row of graphs). Therefore there is low variation in the maximum Pratt Number, while there is more variation in the maximum Froude Number (lower row of graphs).

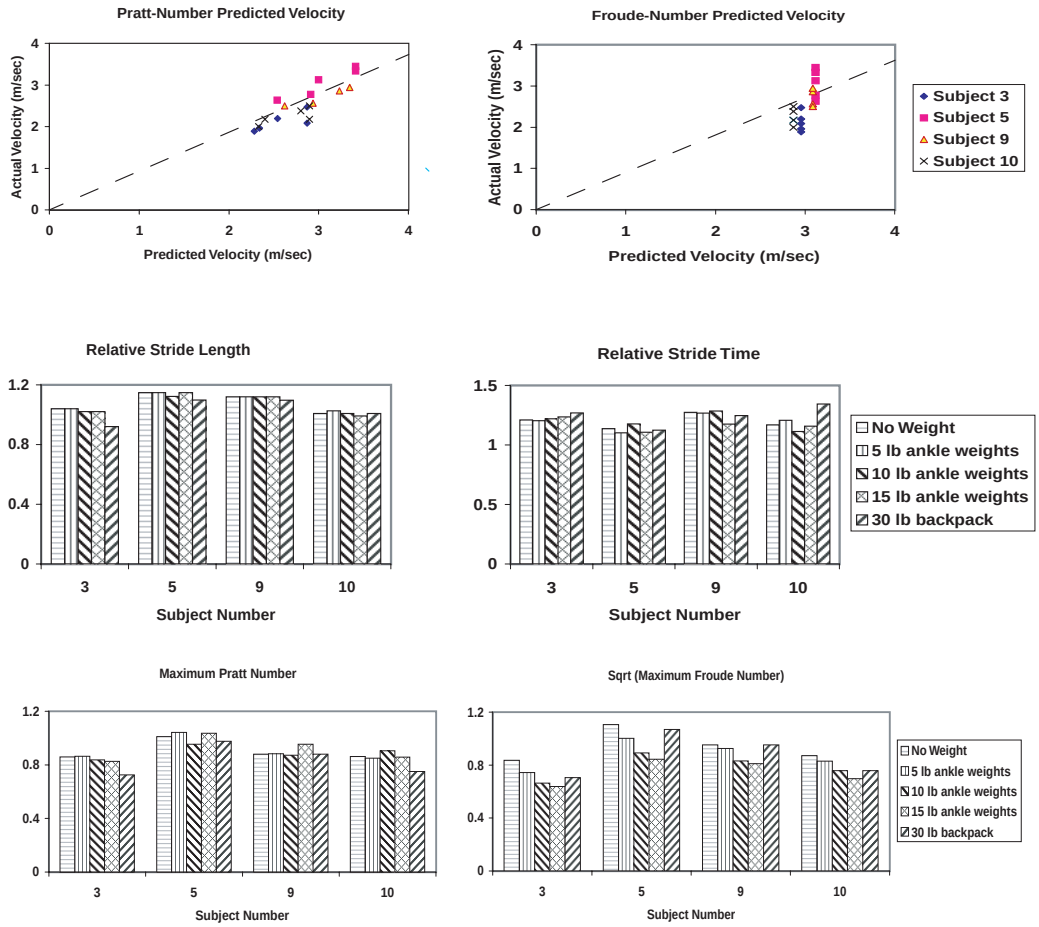


Figure 6-13: Results from maximum velocity walking experiments with ankle weights. Four subjects walked as fast as possible with no weight, 5 lb. ankle weights, 10 lb. ankle weights, 15 lb. ankle weights, and a 30 lb. backpack. Upper left graph shows predicted and actual velocity when using Pratt Number as a maximum velocity predictor. Upper right graph shows the same when using Froude Number. As more ankle weight was added, average velocity decreased, due mainly to increasing swing time. Stride length relative to leg length and stride time relative to minimum swing time remained fairly constant across subjects and weights (second row of graphs). Thus there was less variation in Pratt Number than in Froude Number (third row of graphs).

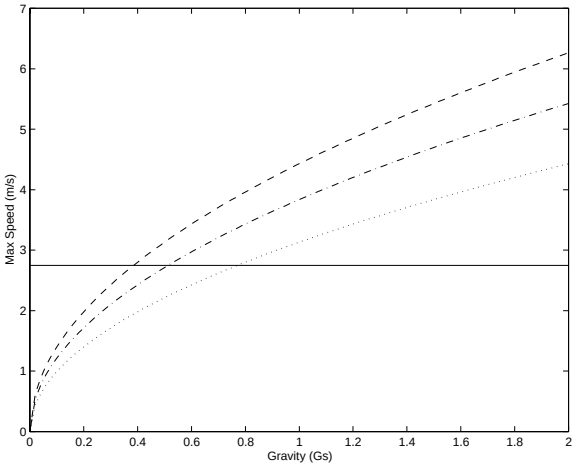


Figure 6-14: Human walking speed limits as a function of gravity. At low gravity losing foot-ground contact due to centrifugal force is the limiting factor. This is indicated by the dotted, dash-dotted, and dashed curves $v^2 < 1.0gl$, $v^2 < 1.5gl$, and $v^2 < 2.0gl$ with a leg length of 1.0 meters. At near Earth's gravity, maximum speed is independent of gravity since swing-time and stride length, which are mostly independent of gravity, are the limiting factors. These limits intersect between a gravity of 0.4 Gs and 0.8 Gs.

1.25 and our fastest 3-legged walking subject walked at Froude Numbers of about 1.5 and Groucho Ran at Froude Numbers over 2.0.

Thus we speculate that at lower gravities in which losing foot contact, and not swing-time, is the limiting factor in walking, Froude numbers between 1.0 and 1.5 may be achievable for normal walking and Froude numbers over 2.0 will be achievable for walking with significantly flattened trajectories or for Groucho Running.

Froude Number limits of 1.0, 1.5, and 2.0 are plotted in Figure 6-14 for a person with leg length of $1.0m$. Also plotted is the limit of $2.75\frac{m}{s}$, which is typical for $1.0m$ leg length. These limits meet at a gravity between 0.4 and 0.8 Gs. Mars gravity is about 0.4 Gs. It is interesting to wonder what the top walking speed on Mars would be.

Cavagna, Willems, and Heglund [Cavagna et al. (1998)] have shown, for a particular subject, that the energetically optimal walking speed on Mars for a particular subject is $1.51\frac{m}{s}$, down from $2.44\frac{m}{s}$ on Earth and that the work done per unit distance to move the center of mass is half that on Earth. However, they do not report on the top walking speed at various gravities. An interesting experiment would be to experimentally measure the gravity/speed curve of Figure 6-14. This could be done simply by having subjects walk as fast as possible while artificially simulating various gravities in NASA's KC-135 airplane, also known as the "Vomit Comet". (If looking for a potential test subject, please contact the author!)

6.6.2 Dynamically Similar Walking

McMahon and Bonner, in their book "On Size and Life" [McMahon & Bonner (1983)], give an excellent introduction to dimensional analysis and dynamic similarity. Following their explanation, two scaled objects are dynamically similar if all relevant dimensionless quantities are the same. For example two pendulums are dynamically similar if $\frac{T_{swing}^2 g}{l}$ and the swing angle are the same between them. Two submarines are dynamically similar if they are the same shape and if their Reynolds Number and Froude Number are the same.

Previous theories on dynamical similarity [Alexander & Jayes (1983)] have suggested that humans will walk in a dynamically similar fashion at equal Froude Numbers, $Fr = \frac{v^2}{gL_{leg}}$. Various studies support dynamic similarity when velocity and leg length are varied. However, Donelan & Kram

(1997) have shown that if gravity is changed, gaits are not dynamically similar at equal Froude Numbers.

We have shown in this chapter that maximum velocity normalized with minimum swing-time and leg length, $Pr_{max} = \frac{T_{swmin} v_{max}}{L_{leg}}$, is fairly constant among test subjects while maximum Froude Number has more variation. Since swing characteristics are important in bipedal walking, it may be important to keep both this quantity and the Froude Number constant in order to achieve dynamic similarity.

Suppose we simulate reduced gravity of 25% Earth's gravity. For a given person, leg length does not change. To keep Froude Number constant, velocity would have to be decreased by half. However, to keep velocity normalized with swing-time and leg length constant, swing-time would then have to double for dynamic similarity. To achieve this, one could add damping to the subjects hip or add inertia to the swing-leg. We believe that if Donelan and Kram's experiments are reproduced, keeping both dimensionless quantities in mind, then dynamic similarity might be better achieved.

Note that a third dimensionless quantity, $\frac{T_{swmin}^2 g}{L_{leg}}$, can be created by manipulating Fr and Pr. This quantity is constant for a passively swinging pendulum. Therefore, *if* the swing-leg dynamics were dictated by pendulum dynamics, dynamic similarity would probably be achieved at equal Froude Numbers when gravity was changed. However, the swing-leg does not scale with gravity like a passive pendulum since it is the inertial and muscular properties of the leg which determine minimum swing-time. Keeping both Pr and Fr constant as gravity varies forces swing-time to scale as if the swing-leg dynamics *were* dominated by the passive pendulum dynamics.

6.6.3 Discussion of Experimental Methods

While the results of this chapter are convincing, the experimental methods have something to be desired. Only a small number of subjects were tested with only a small number of trials. Insufficient practice in Groucho Running and three-legged crutching resulted in some skewed results due to some subjects still learning how to walk in this fashion. Experiments were performed on an outdoor track over different days with differing weather conditions. Therefore, we leave the results of this chapter as open hypotheses which require further testing.

6.6.4 Relevance to Bipedal Walking Robots

In order to make fast bipedal walking robots, we should thus concentrate on stride length and swing-time and a quick transition from swing to stance. In Chapter 7 we describe how we increased the speed of Spring Flamingo, our bipedal walking robot, from approximately $0.8\frac{m}{s}$ to $1.25\frac{m}{s}$ by increasing the stride length and reducing the swing-time. Speed was significantly less than maximum human walking speed because we could not get the swing-leg to swing any faster due to actuator saturation and control difficulties. From experience, we now believe that the most challenging part of achieving high speed bipedal walking is to minimize swing-time.

Chapter 7

Fast Walking Algorithm

In Chapter 5 we showed that the natural dynamics of the robot mechanism can be exploited in the control of walking. For slow walking the swing-leg can swing almost completely passively and speed can be self-stabilizing.

In this Chapter we are concerned with producing an algorithm that results in fast walking. To do so we actively control the swing-leg, actively control the robot's velocity, and more actively control the ankle during toe-off.

In Chapter 6 we argued that swing time is a limiting factor in top walking speed on the Earth. Up to this point, we have avoided using dynamic inversion or trajectory tracking. However, as a first step in achieving a fast swing time, we now decide to use inverse dynamics in addition to feedback in order for the swing thigh to track a minimum jerk trajectory. We still allow the knee to collapse and swing passively during the first half of swing. However, we take into consideration the dynamics of the entire swing-leg when controlling the thigh. During the second half of swing, we apply trajectory tracking to both the thigh and the shin.

This approach results in walking at speeds up to $1.25\frac{m}{s}$. Faster speeds were not achieved as torque and speed limits of the actuators made it difficult to swing the legs any faster.

7.1 Fast Walking Algorithm

The fast walking algorithm is very similar to those described in Chapters 4 and 5 with the exception of the swing-leg control and state transitions based on foot forces. The intuitive strategies employed are the following:

- **Height:** Maintain a constant stance leg length by pushing up until hitting the knee cap.
- **Pitch:** Maintain level pitch using a virtual spring damper mechanism with a set point. Set point may be moved for speed control.
- **Transitions:** Transition from Support state to Toe Off state when the heel lifts off the ground. Transition to the Swing state when the force on the toe falls below a threshold. Transition to the Hold state after swinging for a preset amount of time. Transition to the Support state once the swing-leg hits the ground.
- **Swing-Leg:** Initiate swing of the non-stance leg by tracking a minimum jerk trajectory at the thigh while the knee is passively damped. During the second part of swing track both a thigh and shin trajectory.
- **Speed:** Maintain the virtual toe point of the support foot approximately below the center of mass. Move it forward if walking too quickly or backward if walking too slowly. Lean forward if walking too slowly and backward if walking too quickly.

This algorithm is summarized in Figure 7-1. Since the algorithm is very similar to the other algorithms, in this chapter we concentrate on the quick swing-leg control.

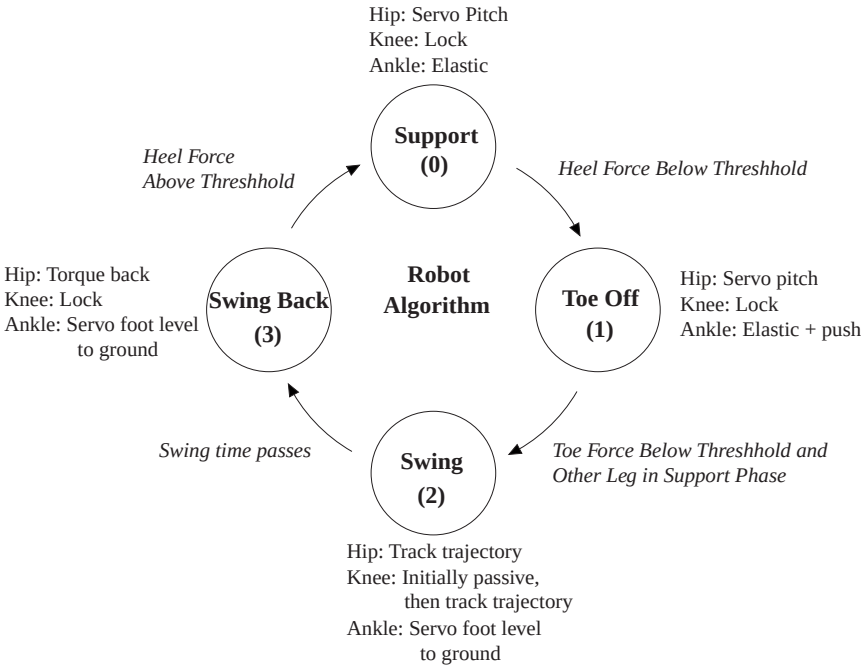


Figure 7-1: REAL Fast walking algorithm. Each leg has a state machine which is in one of four states. State transition conditions and actions in each state are shown.

7.2 Swing-Leg Control

To quickly swing the swing-leg, we use traditional trajectory tracking techniques commonly employed with robot arms. In Appendix B, we derive the dynamic equations of motion for the swing-leg. These equations can be written as

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \vec{g}(q) = \tau \tag{7.1}$$

where H is the mass matrix, C is the matrix of centrifugal and Coriolis terms, and $\vec{g}$ is the gravitational vector. q is the vector of thigh and shin positions:

$$q_1 = \theta_b + \theta_h \tag{7.2}$$
$$q_2 = \theta_b + \theta_h + \theta_k \tag{7.3}$$
$$\tag{7.4}$$

In Appendix B, we describe an adaptive control technique which we use for identifying the parameters of the equations of motion. When adaptation is turned off, the adaptive controller of Appendix B can be written as

$$\tau = \hat{H}\ddot{q}_d + \hat{C}\dot{q}_d + \hat{g} + K_P(q_d - q) + K_D(\dot{q}_d - \dot{q}) \tag{7.5}$$

where $\hat{H}$, $\hat{C}$, and $\hat{g}$ are the esitmated inertia and Coriolis matrices and gravitational vectors of the dynamics, K_P is a matrix of proportional feedback gains, and K_D is a matrix of derivative feedback gains. The three terms on the left are feed-forward terms while those on the right are feedback terms. The feedback gain matrices can be derived as

$$K_P = \begin{bmatrix} K_{D1}\lambda_1 + a_2\sin(q_k)\dot{q}_1\lambda_1 & 0 \\ 0 & K_{D2}\lambda_2 \end{bmatrix} \quad (7.6)$$

$$K_D = \begin{bmatrix} K_{D1} + (a_1 + a_2\cos(q_k))\lambda_1 & 0 \\ 0 & K_{D2} + a_3\lambda_2 \end{bmatrix} \quad (7.7)$$

Note that the gains are dependent on the angle of the knee and that one term is even dependent on the velocity of the thigh. The control gains used in this Chapter are the same as those used in Appendix B:

$$K_{D1} = 1.0 \quad (7.8)$$
$$K_{D2} = 0.7 \quad (7.9)$$
$$\lambda_1 = 16.0 \quad (7.10)$$
$$\lambda_2 = 10.0 \quad (7.11)$$

We use a minimum-jerk 5th order spline for the desired thigh trajectory [Flash & Hogan (1985)].

$$q_{1d}(t) = q_{1_0} + (q_{1_0} - q_{1d_f})(15\tau^4 - 6\tau^5 - 10\tau^3) \quad (7.12)$$
$$\dot{q}_{1d}(t) = \frac{1}{t_f}(q_{1_0} - q_{1d_f})(60\tau^3 - 30\tau^4 - 30\tau^2) \quad (7.13)$$
$$\ddot{q}_{1d}(t) = \frac{1}{t_f^2}(q_{1_0} - q_{1d_f})(180\tau^2 - 120\tau^3 - 60\tau) \quad (7.14)$$

where q_{1d} is the desired thigh angle, q_{1_0} is the actual thigh start angle, q_{1d_f} is the desired final thigh angle, t is the time since initiation of swing, t_f is the final time, and $\tau = \frac{t}{t_f}$. The thigh trajectory starts at the beginning of the swing phase. The thigh is swung forward along a minimum jerk trajectory to a preset position during the first part of swing. During the second part of swing it is swung back to a shallower angle.

Initially the shin swings passively, with damping at the knee in order to prevent the knee from collapsing too much. To achieve this, we simply make the desired thigh position and velocity equal to the actual position and velocity and compute what the desired acceleration should be in order to achieve the desired damping torque:

$$q_{2d}(t) = q_2(t) \quad (7.15)$$
$$\dot{q}_{2d}(t) = \dot{q}_2(t) \quad (7.16)$$
$$\ddot{q}_{2d}(t) = \frac{1}{\hat{H}_{22}}(-\hat{H}_{21}\ddot{q}_{1d} - \hat{C}_{21}\dot{q}_1 - \hat{C}_{22}\dot{q}_2 - \hat{g}_2 + \tau_2) \quad (7.17)$$
$$(7.18)$$

where $\hat{H}_{21}$, $\hat{H}_{22}$, $\hat{C}_{21}$, $\hat{C}_{22}$, and $\hat{g}_2$ are selected elements of the estimated inertial and Coriolis matrices and gravity vector.

After the shin swings passively for a set amount of time, we then have it follow a 3rd order spline trajectory [Craig (1989)]. The initial position and velocity are set to equal the current shin state. The final position and velocity are predetermined constant parameters.

$$
\begin{aligned}
q_{2d}(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 && (7.19)\\
\dot{q}_{2d}(t) &= a_1 + 2a_2 t + 3a_3 t^2 && (7.20)\\
\ddot{q}_{2d}(t) &= 2a_2 + 6a_3 t && (7.21)\\
& && (7.22)
\end{aligned}
$$

where

$$
\begin{aligned}
a_0 &= q_{2_0} && (7.23)\\
a_1 &= \dot{q}_{2_0} && (7.24)\\
a_2 &= \frac{3(q_{2_f} - q_{2_0})}{t_f^2} - \frac{2\dot{q}_{2_0} - \dot{q}_{2_f}}{t_f} && (7.25)\\
a_3 &= -\frac{2(q_{2_f} - q_{2_0})}{t_f^3} + \frac{\dot{q}_{2_0} + \dot{q}_{2_f}}{t_f^2} && (7.26)
\end{aligned}
$$

7.3 Results

Using this algorithm, Spring Flamingo occasionally walked at speeds up to $1.25\frac{m}{s}$. The robot walked reliably at speeds up to $1.1\frac{m}{s}$, results of which we present below. We also present results from varying the desired velocity and from recovering after being pushed.

Typical thigh and shin tracking results while walking at $1.1\frac{m}{s}$ are shown in Figure 7-2. The parameters we used for the desired trajectories for the first half of swing were:

$$
\begin{aligned}
q_{1_f} &= 0.55 && (7.27)\\
t_f &= 0.35 && (7.28)
\end{aligned}
$$

while the parameters for the second half of swing were

$$
\begin{aligned}
q_{1_f} &= 0.45 && (7.29)\\
q_{2_f} &= 0.50 && (7.30)\\
\dot{q}_{2_f} &= 0.0 && (7.31)\\
t_f &= 0.17 && (7.32)
\end{aligned}
$$

Swing was initiated at approximately 4.28 seconds. The knee was passively damped until approximately 4.46 seconds, at which time the shin tracked a 3rd order spline trajectory. We see that good tracking was achieved for both the thigh and shin until approximately 4.6 seconds, at which time the knee hit the knee limit stop, causing an un-modeled disturbance in both trajectories. Swing was complete at 4.8 seconds, at which time the Swing Back state was entered. The leg hit the ground and transitioned to the Support state at approximately 4.92 seconds, for a total time of swing of approximately 0.64 seconds.

Figure 7-3 shows five seconds of data from Spring Flamingo walking at $1.1\frac{m}{s}$, while a stick figure animation of the same data is shown in Figure 7-4 and also plotted in flip-book format at the bottom of this thesis. The top three graphs of Figure 7-3 show velocity, height, and pitch. The velocity fluctuates during each stride and between legs as the inner leg moves more slowly than the outer leg. The height exhibits an arc-like pattern as the robot walks with a compass-like gait. Pitch fluctuates both due to disturbances caused by each step and because it is actively used to control speed. The middle three graphs show the state machine state, thigh angle, and shin angle of the two legs. The bottom three graphs show mechanical power being applied at the left hip, knee, and ankle. We see that there are significant peaks in hip and knee power during swing and in the ankle during toe-off.

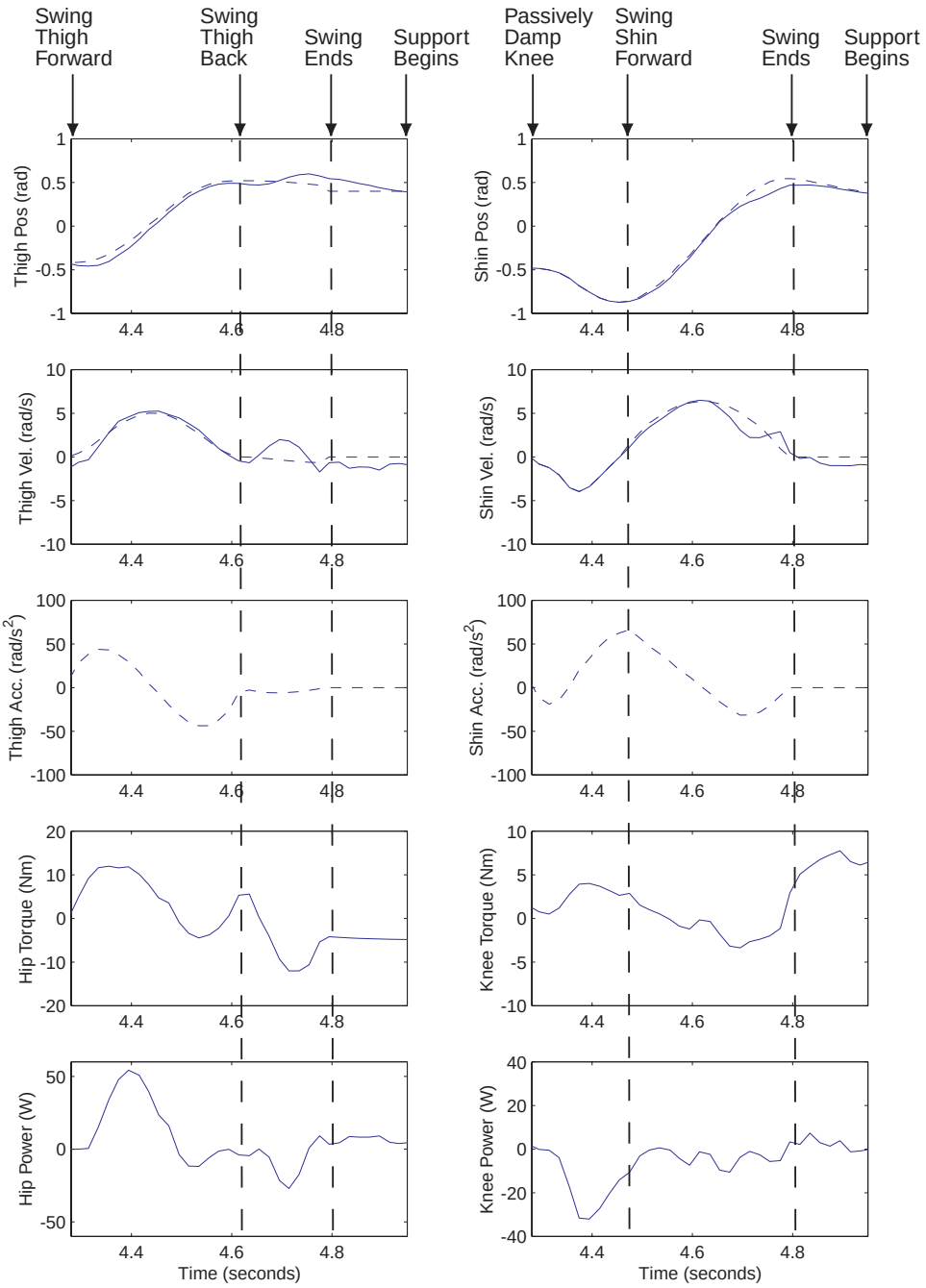


Figure 7-2: REAL Data showing thigh and shin tracking during swing while walking 1.1$\frac{m}{s}$. Dashed plots are desired values while solid plots are actual values. At approximately 4.7 seconds, the knee hits its limit stop which causes tracking errors. The duration from start of swing until the leg is straight is approximately 0.55 seconds while the duration from start of swing until touch-down is approximately 0.67 seconds. A large amounts of positive work is applied by the hip at the beginning of swing in order to accelerate it while a large amount of negative work is absorbed by the knee in order to prevent it from collapsing too much.

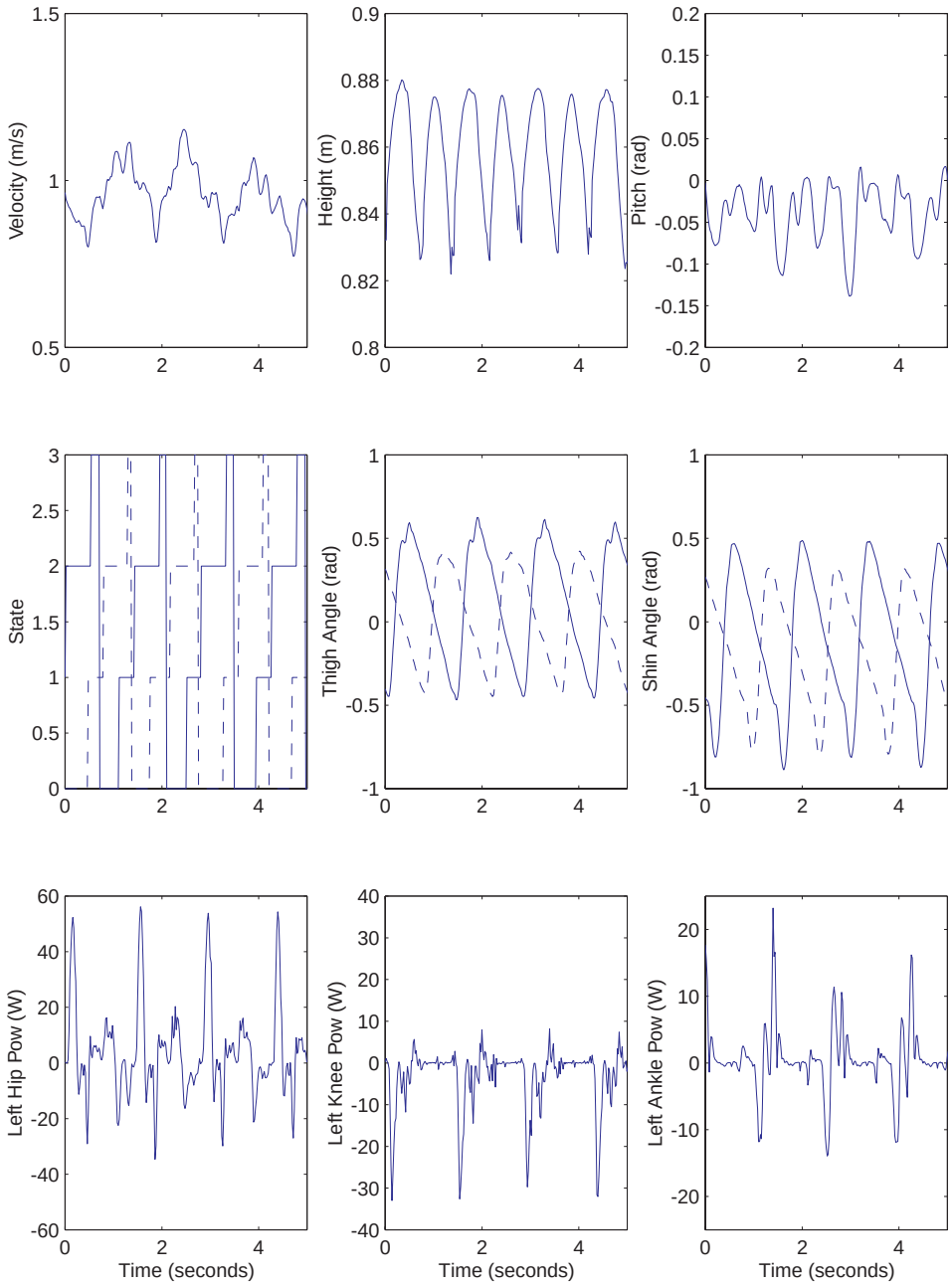


Figure 7-3: REAL Data from Spring Flamingo walking at $1.1\frac{m}{s}$. The top three graphs show velocity, height, and pitch. The middle three graphs show the state machine state, thigh angle, and hip angle of the two legs. The data for the right leg is dashed while the data for the left leg is solid. The bottom three graphs show mechanical power being applied at the left hip, knee, and ankle. This data is plotted as an elapsed time snapshot in Figure 7-4 and along the bottom of this document.

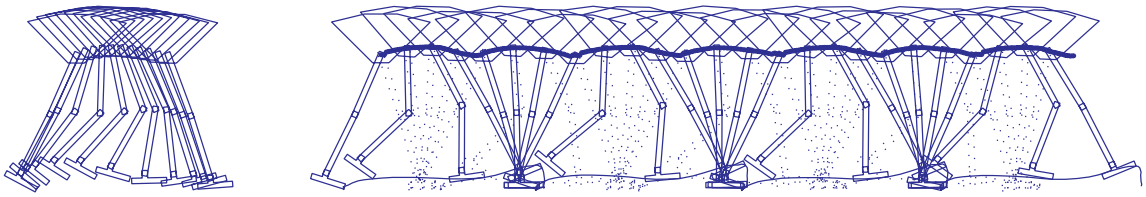


Figure 7-4: REAL Elapsed time snapshot of Spring Flamingo walking at $1.1\frac{m}{s}$. The drawings on the left are spaced approximately 0.06 seconds apart and show the swing phase of one leg. The drawings on the right are spaced approximately 0.2 seconds apart. The robot walks from left to right. Snapshots of this data spaced approximately 0.06 seconds apart are also plotted on the bottom of this thesis.

Figure 7-5 shows results from varying the desired velocity from $0.0\frac{m}{s}$ during standing to $0.9\frac{m}{s}$. We see that speed is tracked fairly well, although not with high precision. It takes a few steps for the robot to change from one speed to another.

Figure 7-6 shows disturbance rejection after being pushed. The desired speed is set at $0.6\frac{m}{s}$. At approximately 15 seconds, the robot is pushed backward, decelerating it to under $0.2\frac{m}{s}$. At approximately 25 seconds, the robot is pushed forward, accelerating it to over $0.9\frac{m}{s}$. In both cases, the robot regains the desired velocity in a few steps. The second push also creates a significant disturbance in both height and pitch, both of which the robot quickly recovers from.

7.4 Conclusions

By more quickly swinging the swing-leg using trajectory tracking methods, we were able to control the robot to walk up to $1.25\frac{m}{s}$. While this is a moderate speed for a human, it is quite fast for modern bipedal walking robots. Preliminary versions of Honda's P2 and P3 robots were able to walk up to $1.30\frac{m}{s}$. We know of no other bipedal robots which can walk over $1.0\frac{m}{s}$.

We were not able to achieve faster walking speeds as the motors of the robot were pushed beyond their rated specifications. From Figure 7-2 we see that the hip power exceeded $50W$ during swing. However, the hip motor is only rated for $30W$! At speeds over $1.0\frac{m}{s}$ the hip motors would heat up and start smoking after 3 laps of walking. It is curious to note that large amounts of positive work is applied at the hip at the same time large amounts of negative work is absorbed at the knee (lower 2 graphs of Figure 7-2). Since an individual actuator is employed at each joint, the total amount of work is large. Perhaps a multi-joint actuator which passes over both the hip and the knee could be used during the first half of swing and consume much less total power.

In Figure 7-7, we plot relative stride length and stride time vs. Froude Number for various walking trials of Spring Flamingo and for the human subjects of Chapter 6. The fastest stride times we could achieve with Spring Flamingo were 0.61 seconds. The longest relative stride lengths we could achieve were 0.99. Compared to the human test subjects who could walk with a stride time of as low as 0.3 seconds, and a relative stride length up to 1.1, we see there is much room for

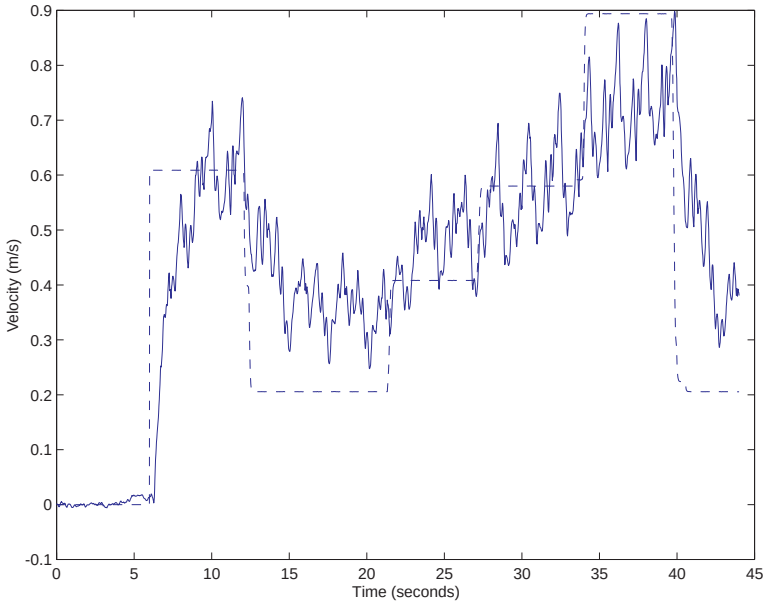


Figure 7-5: REAL Data showing control of walking speed. The actual speed (solid) follows the trend of the desired speed (dashed), although it does not track it exactly. The robot is standing and balancing for the first five seconds.

improvement.

Spring Flamingo's fastest Froude Number was 0.18 which falls way outside the range of maximum Froude Numbers for the human test subjects of Chapter 6 (Figure 6-12). However, if we calculate the maximum Pr,

$$Pr_{max} = \frac{T_{swmin} v_{max}}{L_{leg}} \tag{7.33}$$

we get $Pr_{max} = 0.52 * 1.25/0.89 = 0.73$, which is within 20% of the range of the human subjects in Figure 6-12. We use a minimum swing time of 0.52 seconds since it is the minimum duration of the desired spline trajectories the swing-leg tracks. However, in Chapter 6, we include the time to touch down in the swing time measurements. Therefore, a minimum swing time between 0.6 and 0.7 would be more realistic, resulting in a Pr range between 0.84 and 0.98, which matches the range of human subject values in Figure 6-12. This further supports the hypothesis that minimum swing time is the limiting factor in walking speed at Earth's gravity. Spring Flamingo's speed was limited mostly by the lack of power in its motors which in turn limited how fast it could swing its leg.

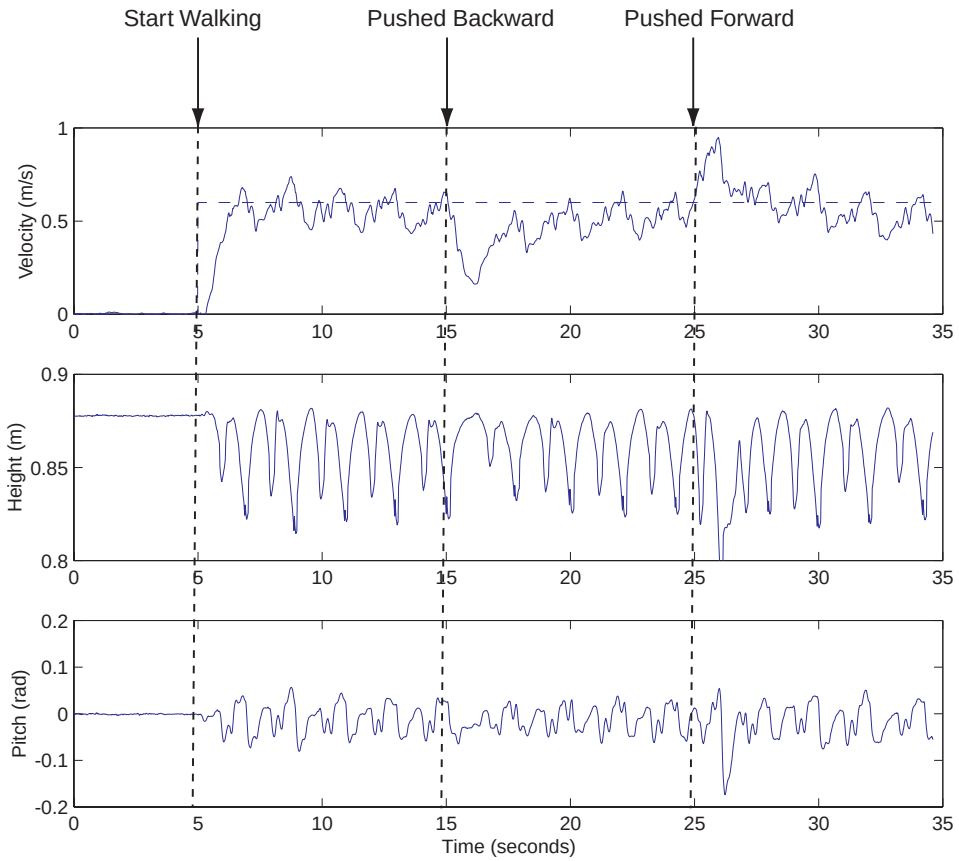


Figure 7-6: REAL Data showing disturbance rejection due to being pushed. The robot is pushed backward at approximately 15 seconds which decreases its speed and pushed forward at approximately 25 seconds which increases its speed. In both cases, the robot regains the desired speed of approximately $0.6\frac{m}{s}$. The second push creates a significant disturbance in both height and pitch, yet the robot quickly recovers.

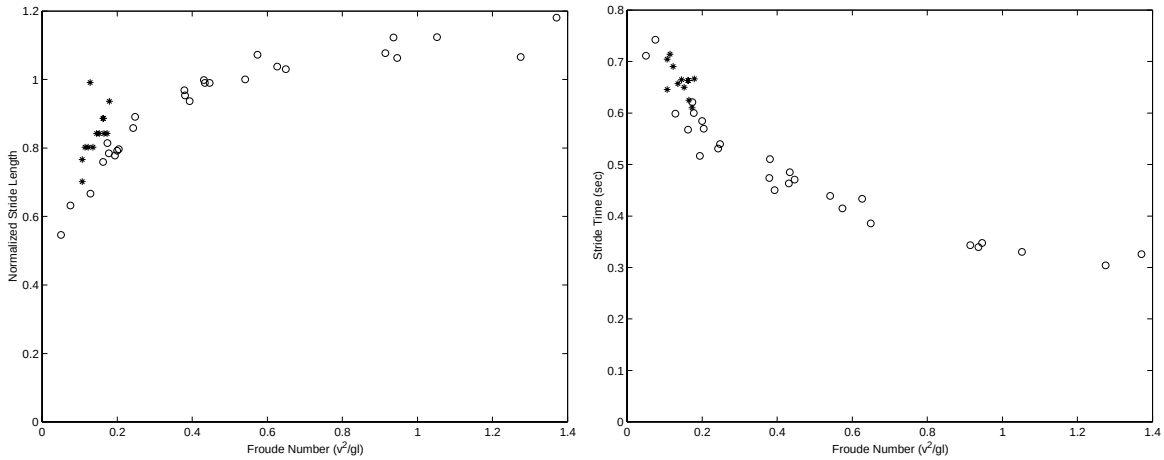


Figure 7-7: REAL Relative stride lengths and stride times achieved for various walking trials of Spring Flamingo (*), compared to data from human subjects (circles). We see that Spring Flamingo walks relatively slowly compared to the human subjects, achieving a maximum Froude Number of 0.18.

Chapter 8

3D Bipedal Walking Simulation

In Chapter 5 we showed that natural dynamics mechanisms could be exploited in the control of bipedal walking robots. In this chapter we use the same natural dynamic mechanisms and extend the algorithm to 3D walking. We use a knee cap to prevent the leg from inverting, which makes control of height easy. We use a compliant ankle limit so that the center of pressure on the foot travels forward with the center of mass of the body. And we exploit the natural swing dynamics of the leg to make swing control simple and natural looking.

The 3D algorithm uses the planar simulation algorithm described in Chapter 5 to control motion in the sagittal plane. Fowble & Kuo (1996) showed with a 3D passive dynamic walking model that the dynamics of the sagittal plane are sufficiently decoupled from dynamics of the frontal and transverse planes such that control of each can be treated separately. Therefore, we extend our planar algorithm by adding lateral balance through foot placement and ankle torque.

8.1 3D Simulation Algorithm

In this Chapter we use the natural dynamic mechanisms described in Chapter 5 in the control of a simulated seven-link, twelve degree-of-freedom, bipedal robot. The rigid body simulation (Figure 8-1) is implemented using the Creature Library, an in-house software package based on SD-Fast of Symbolic Dynamics (www.symdyn.com). The robot model has an actuated hip (3 degree-of-freedom), knee (1 degree-of-freedom), and ankle (2 degree-of-freedom) on each leg. The physical parameters (Table 8.1) were chosen to match those of a bipedal robot currently under construction. Note that the nearly half of the mass of the robot is in its legs.

The simulation algorithm is summarized in Figure 8-2, with controller parameters in Table 8.2. Pitch, roll, and yaw are defined as if the body were an airplane: pitch corresponds to bending forward, roll to bending to the side, and yaw to twisting about the vertical axis. Each leg acts

Length	hip to body	hip sep.	thigh	shin	foot length	width	height
(m)	.160	.184	.432	.432	.203	.089	.051

Mass	body	thigh	shin	foot
(kg)	12.75	2.74	2.70	0.66

Inertia	body Ixx,yy	Izz	thigh Ixx,yy	Izz	shin Ixx,yy	Izz	foot Ixx	Iyy	Izz
(kgm^2)	.23	.21	.044	.004	.054	.003	.001	.002	.002

Table 8.1: SIM Physical parameters of the simulated bipedal walker. The center of mass of all links are located at their centroid. The first length is the distance of the body center of mass above the hip. The second length is the separation distance between the hips. The moments of inertia are calculated about the center of mass of each link.

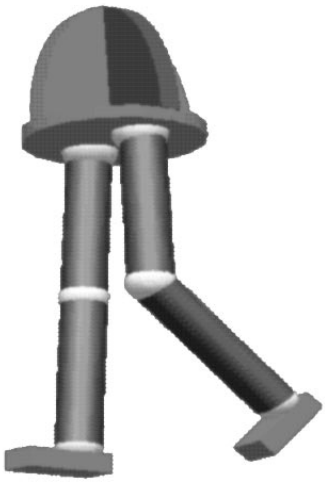


Figure 8-1: SIM Three dimensional bipedal walking simulation. The robot has twelve degrees of freedom: three in each hip, one in each knee, and two in each ankle. The simulation is based on a real robot, called M2, that we are currently building.

separately and has a simple state machine. The leg can be in either Support, Toe Off, Swing, or Straighten states. Below we describe the control laws in each state and the transition conditions between states.

In Support and Toe Off states, proportional-derivative (PD) controllers are used at the hip to servo body pitch, roll and yaw in order to maintain balance. Feed-forward torque is added to hip roll to offset the weight of the body:

```
tau.left_hip_pitch = -body_pitch_gain * (0.0 - q.pitch) + body_pitch_damp * qd.pitch;
tau.left_hip_roll = body_roll_gain * q.roll + body_roll_damp * qd.roll
                  + (HIP_OFFSET_Y * cos(q.roll) + BODY_CG_Z * sin(q.roll)) * ff_z;
tau.left_hip_yaw = body_yaw_gain * (q.yaw - q_d.yaw) + body_yaw_damp * qd.yaw;
```

where *tau* is the vector of joint torques, q is the vector of joint positions, and qd is the vector of joint velocities. Note that only the left leg is presented in all the code fragments.

In Support and Toe Off states, the knee is locked to maintain height:

```
tau.left_knee = knee_gain * (0.0 - q.left_knee) - knee_damp * qd.left_knee;
```

In Support state, the ankle pitch is unactuated - only the passive ankle compliance is present. The ankle roll is used to dampen lateral velocity, as described in Section 8.1.1. During Toe Off state, the ankle is servoed to an angle using a PD controller in addition to its passive compliance:

```
tau.left_ankle_pitch = push_gain * (push_set - q.left_ankle_pitch)
                     - push_damp * qd.left_ankle_pitch;
if (q.left_ankle_pitch < ankle_limit_set)
    tau.left_ankle_pitch = tau.left_ankle_pitch + ankle_limit_gain
                         * (ankle_limit_set - q.left_ankle_pitch)^2;
```

The transition from Support to Toe Off occurs when the heel lifts off the ground due to the passive compliance of the ankle.

The robot transitions from Toe Off to Swing when the force on the foot falls below a certain threshold. In both Swing and Straighten states the hip pitch is servoed to an angle using a PD controller:

```
Swing:  tau.left_hip_pitch = hip_gain * ((-hip_d - q.pitch) - q.left_hip_pitch)
                           - hip_damp * qd.left_hip_pitch;
Straighten:  tau.left_hip_pitch = hip_gain * ((-hip_hold - q.pitch) - q.left_hip_pitch)
                                - hip_damp * qd.left_hip_pitch;
```

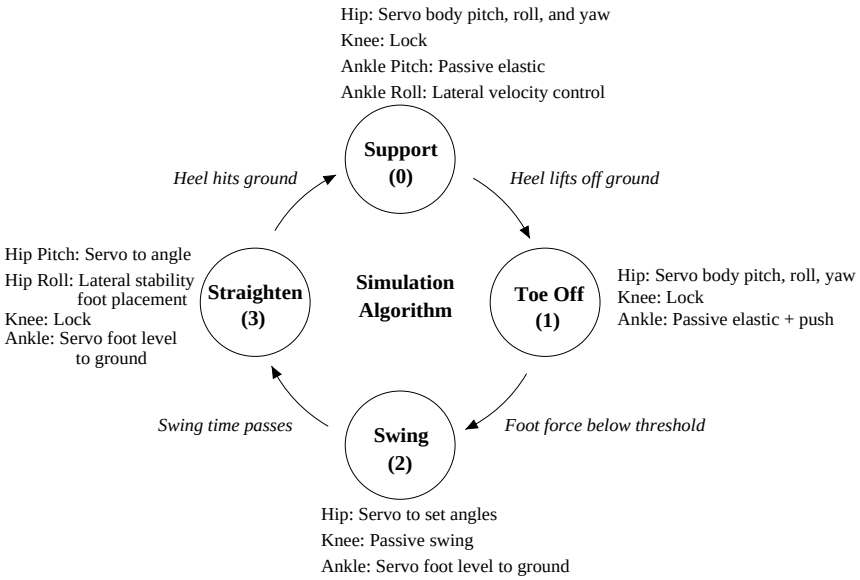


Figure 8-2: SIM 3D Simulation Algorithm. Each leg has a state machine which is in one of four states. State transition conditions and actions in each state are shown.

The foot is servoed to be level with the ground so that the robot does not stub its toe. In Straighten state, the hip roll is used for lateral foot placement, to control lateral velocity, as described in Section 8.1.1. In Swing state, the knee is lightly damped while in Straighten state the knee is locked straight using a PD controller:

```
Swing:      tau.left_knee = -swing_damp_knee * qd.left_knee;
Straighten:  tau.left_knee = straighten_gain_knee * (0.0 - q.left_knee)
                           - straighten_damp_knee * qd.left_knee;
```

The robot transitions from Swing to Straighten state after a constant amount of time passes. Finally, the robot transitions from Straighten to Support state when the heel of the swing leg hits the ground.

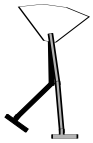
8.1.1 Lateral Stability

To control lateral stability in the frontal plane, we use foot placement and ankle torque. Since our control algorithm treats the frontal plane dynamics and the sagittal plane dynamics separately, as though they were decoupled, we use a conservative approach for the frontal plane. On each step we attempt to place the foot such that lateral velocity will be absorbed into potential energy as the body moves directly over the support foot. In this way the body is "captured" laterally so that no matter what the forward velocity of the robot is, it will not start tipping back laterally until the body has traveled forward over the support foot.

Figure 8-3 shows a simple pendulum model for determining foot placement. The initial kinetic energy is

$$E_k = \frac{1}{2}mv^2 \tag{8.1}$$

The change in potential energy when the mass transfers from its initial condition to its highest point is



Controller Parameter	Value
SUPPORT	
Body pitch gain	$100\frac{Nm}{rad}$
Body pitch damping	$20\frac{Nm}{rad/s}$
Body roll gain	$200\frac{Nm}{rad}$
Body roll damping	$20\frac{Nm}{rad/s}$
Body yaw gain	$30\frac{Nm}{rad}$
Body yaw damping	$4\frac{Nm}{rad/s}$
Knee gain	$30\frac{Nm}{rad}$
Knee damping	$10\frac{Nm}{rad/s}$
Passive ankle set point	$0.0rad$
Passive ankle spring constant	$400\frac{Nm}{rad^2}$
Ankle lateral set point	$0.087rad$
Ankle lateral speed gain	$0.36\frac{rad}{m/s}$
Ankle roll gain	$100\frac{Nm}{rad}$
Ankle roll damping	$0.10\frac{Nm}{rad/s}$
TOE OFF	
Ankle pitch set point	$0.3rad$
Ankle pitch gain	$30\frac{Nm}{rad}$
Ankle pitch damping	$0\frac{Nm}{rad/s}$
Swing transition foot force	$100N$

Controller Parameter	Value
SWING	
Hip pitch set point	$0.625rad$
Hip pitch gain	$23.7\frac{Nm}{rad}$
Hip pitch damping	$2.37\frac{Nm}{rad/s}$
Max hip pitch torque	$10Nm$
Hip roll gain	$92\frac{Nm}{rad}$
Hip roll damping	$23\frac{Nm}{rad/s}$
Hip yaw gain	$9.2\frac{Nm}{rad}$
Hip yaw damping	$5.75\frac{Nm}{rad/s}$
Knee damping	$0.25\frac{Nm}{rad/s}$
Ankle gain	$4.0\frac{Nm}{rad}$
Ankle damping	$0.5\frac{Nm}{rad/s}$
Swing time	$0.3sec$
STRAIGHTEN	
Hip pitch set point	$0.55rad$
Hip roll offset	$0.048rad$
Hip roll lateral speed gain	$0.36\frac{rad}{m/s}$
Hip roll lateral position gain	$2.0\frac{rad}{m}$
Knee gain	$1.0\frac{Nm}{rad}$
Knee damping	$1.2\frac{Nms}{rad}$

Table 8.2: SIM Control system parameters of the simulated bipedal walker.

$$\Delta E_p = mgl(1 - \cos\theta) \tag{8.2}$$

Setting the change in potential energy equal to the kinetic energy we get

$$\cos\theta = 1 - \frac{v^2}{2gl} \tag{8.3}$$

For small angle approximation, we get

$$\theta \approx \frac{v}{\sqrt{gl}} \tag{8.4}$$

We use equation 8.4 for determining lateral foot placement during the Straighten state. Hip roll is servoed to a set point proportional to the robot's lateral velocity (yd). The set point is also proportional to the robot's lateral position with respect to the support foot (r_ceny). This second term is a predictive one. The higher the lateral position, the faster the robot is accelerating, which means more hip roll torque will be required.

```
q_d.left_hip_roll = -q.roll + swing_roll_off + yd_gain * yd + ceny_gain * r_ceny;
tau.left_hip_roll = hip_k * (q_d.left_hip_roll - q.left_hip_roll)
                  - hip_b * qd.left_hip_roll;
```

If the leg touches down at the proper angle, the mass will come to rest laterally at the highest point, above the foot. However, there will be some error in the foot placement due to imperfections

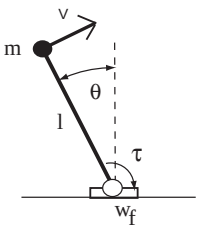


Figure 8-3: Simple pendulum model of the dynamics of the 3d simulation in the frontal plane. The robot mass, m is lumped at a point. The leg is of length l at an angle θ from vertical. The foot has a half width of w_f, with torque τ being applied at the ankle. This model is useful for studying the effects of foot placement and ankle torque on lateral velocity.

in the model and servo control limits. To overcome these errors, we use ankle torque, after the foot is placed.

The ankle torque is limited due to the small width of the feet. A static analysis gives

$$-mgw_f \leq \tau_{ankroll} \leq mgw_f \tag{8.5}$$

where w_f is half the width of the foot. Any torque outside this range will cause the foot to rotate about its edge. By examining the dynamics, one can show that the mass can be captured over the foot if

$$\frac{v}{\sqrt{gl}} - \frac{w_f}{l} \leq \theta \leq \frac{v}{\sqrt{gl}} + \frac{w_f}{l} \tag{8.6}$$

We can show this in the following way. Suppose the pendulum is tipped to the left (θ positive). The most restoring torque which can be added is $\tau = -mgw_f$, which will put the center of pressure fully on the left edge of the foot. The dynamics of the system will then be

$$ml^2\ddot{\theta} = mgl\cos(\theta) - mgw_f \tag{8.7}$$

Assuming a small angle approximation, the dynamics can be written as

$$\ddot{\theta} - \frac{g}{l}\theta = \frac{g}{l^2}w_f \tag{8.8}$$

There will be a fixed point at which the mass can rest (but be unstable) at $\theta^* = \frac{w_f}{l}$. Defining $\theta' = \theta - \frac{w_f}{l}$, the dynamics can be rewritten as

$$\ddot{\theta}' - \frac{g}{l}\theta' = 0 \tag{8.9}$$

These are the same dynamics as the inverted pendulum with no ankle torque. Thus, by the same energy argument above, we have $\theta' = \frac{v}{\sqrt{gl}}$ for capture to occur. Substituting back, we then get the range of capture angles in Equation 8.6.

$$\frac{v}{\sqrt{gl}} - \frac{w_f}{l} \leq \theta \leq \frac{v}{\sqrt{gl}} + \frac{w_f}{l} \tag{8.10}$$

Equations 8.4 and 8.6 are plotted in Figure 8-4. The parameter values we use are $w_f = 0.05m$, $l = 1.0m$, $g = 10.0\frac{m}{s^2}$. The middle line is the touch down angle required to have the mass stop at rest above the foot (capture angle). The top and bottom lines show the range of angles at which the mass

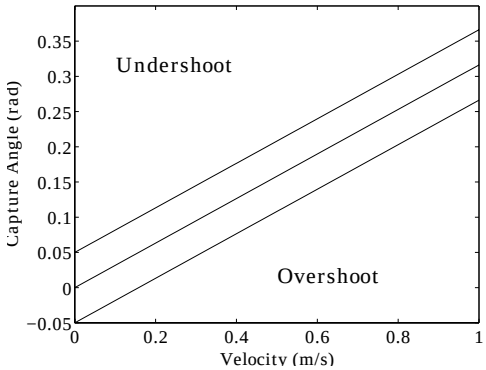


Figure 8-4: Range of Capture Angle vs. lateral velocity for a simple pendulum model with ankle torque. The middle curve is the leg angle required at foot placement in order for the mass to stop directly over the foot without using ankle torque. The outside curves show the range of leg angle which can result in capture with the use of ankle torque.

can be captured at the top with the aid of ankle torque. We see that ankle torque can compensate for errors in lateral foot placement, even with a narrow foot. In our simulation algorithm, we change the set point of the stance ankle roll based on the lateral velocity for capture, from Equation 8.4:

```
tau.left_ankle_roll = yd_ankle_gain
                      * ((-flat_ankle_roll - yd_gain * yd) - q.left_ankle_roll)
                      - ankle_b * qd.left_ankle_roll;
```

where `yd_ankle_gain` and `ankle_b` are proportional and derivative position gains, `flat_ankle_roll` is an offset position, and `yd_gain` is the amount to change the ankle position per unit lateral velocity. This control law can also be viewed as equivalent to a PD controller with constant set point and additional ankle torque proportional to lateral velocity if rewritten as

```
tau.left_ankle_roll = yd_ankle_gain * (-flat_ankle_roll - q.left_ankle_roll)
                      - ankle_b * qd.left_ankle_roll
                      - yd_ankle_gain * yd_gain * yd;
```

8.1.2 3D Simulation Results

The simulation parameters were first manually tuned, and then fine tuned using a genetic algorithm with efficiency as its cost function. Efficiency was computed as distance traveled divided by total joint energy after ten seconds of walking. Total joint energy was computed by integrating the total joint power which is the sum of the absolute values of the mechanical power at each joint:

$$E_{tot} = \int P_{tot}\, dt, \quad P_{tot} = \sum_{joints} |P_{joint}|, \quad P_{joint} = \tau_{joint}\, \dot{\theta}_{joint} \tag{8.11}$$

After several generations, natural looking walking resulted. Time-elapsed animations are shown in Figures 8-5 and 8-6. The drawings on the left show the swing phase of one leg. The drawings on the right show several steps. The right leg is dotted while the left leg is solid. Lines show the path of the tips of the feet and the hip trajectory. Results are plotted graphically in Figure 8-7.

We see that the simulated robot walked at a moderate speed (approximately 0.8 m/s) and had a natural looking gait. It is interesting that the algorithm does not contain any explicit speed control mechanism, yet speed is stabilized. We speculate that this is due to the natural system dynamics, in the same way that speed is naturally stabilized in passive dynamic walkers.

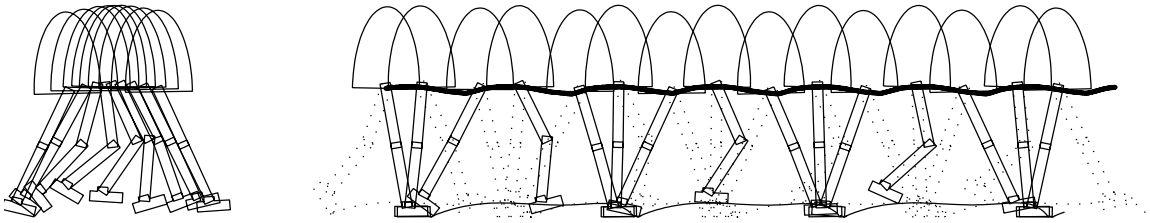


Figure 8-5: SIM Elapsed-time snapshot of the 3d simulated robot walking data. The left leg is dotted while the right leg is solid. Lines show the path of the tips of the feet and the hip trajectory. The snapshots on the left are spaced at 0.1 seconds and show one swing phase. The snapshots on the right are spaced at 0.4 seconds and show several steps.

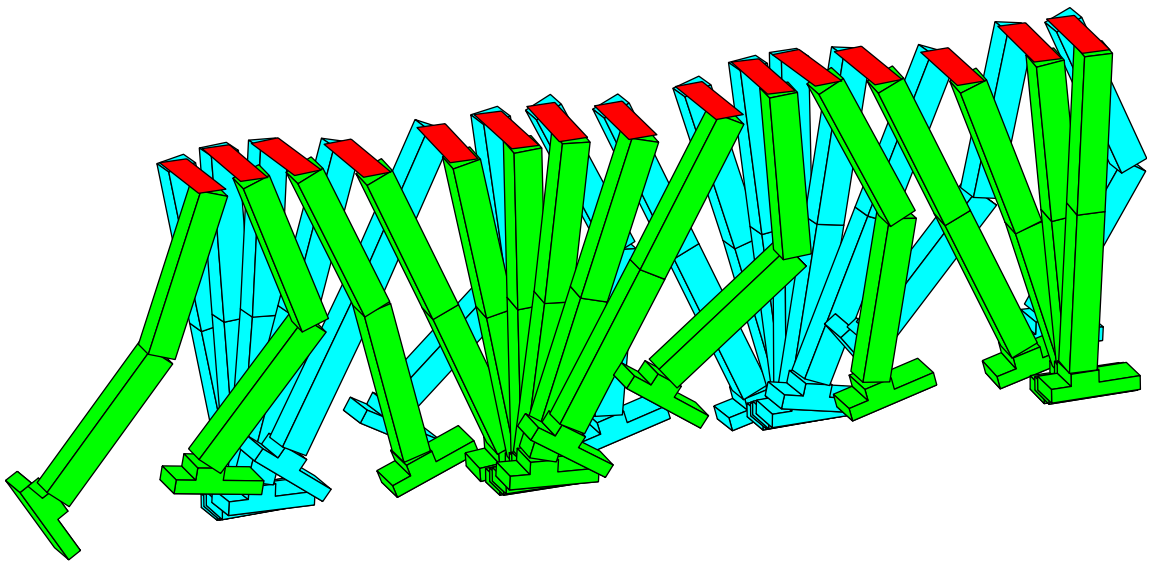


Figure 8-6: SIM Elapsed-time isometric snapshots of the 3d simulated robot walking data showing lateral displacement of the pelvis.

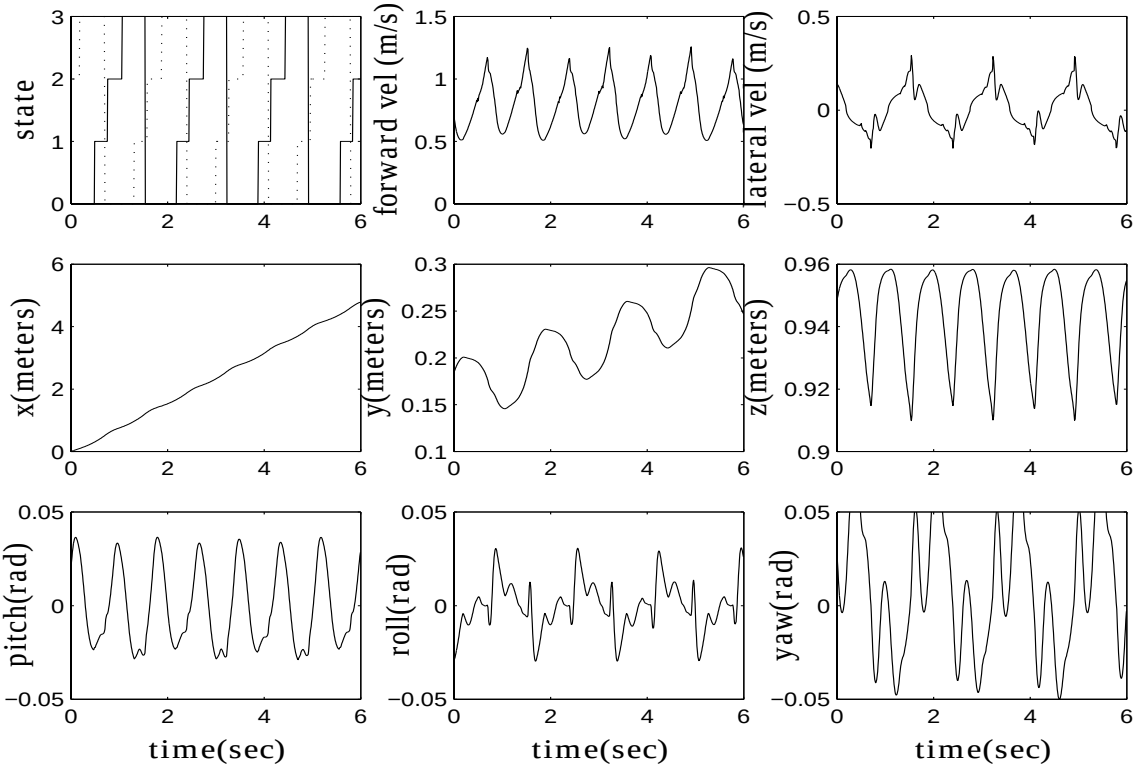


Figure 8-7: SIM 3D simulation walking data. The first row, from left to right, contains the state of the legs, forward velocity, and lateral velocity. The second row contains forward distance, lateral motion, and body height. The last row contains body pitch, roll, and yaw.

8.2 Conclusions

Three dimensional bipedal walking can be achieved by a simple control algorithm which exploits the natural dynamics of a kneecap, compliant ankle, and passive swing leg. Lateral foot placement and ankle torque can be used for lateral stability. The resultant motion is smooth and efficient. This work should be applicable to the real 3D bipedal robots, further bridging the gap between passive dynamic walkers and powered bipedal robots.

The simulation settles on a stable speed of walking of approximately 0.8 m/s. However, nowhere in the controller is speed explicitly controlled. We believe that the speed is stabilized in a similar way to passive dynamic walking machines. That is, if the robot goes too fast, it naturally takes a longer step due to the natural swing leg dynamics and hence slows down on the next step. Similarly, if the robot moves too slowly, it naturally takes a shorter step and hence speeds up on the next step. This behavior has also been observed in our experimental robot, Spring Flamingo, as described in Chapter 5.

Chapter 9

Conclusions

Exploiting Inherent Robustness

Because the walking problem is inherently robust, simple, imprecise, control techniques can be used to control bipedal walking. Spring Flamingo walked using a simple set of control strategies that were easy to develop, easy to understand, and easy to implement. These strategies needed to be only slightly modified in order to control the robot to traverse a ramp and rolling terrain. In short, planar bipedal walking is easy to achieve despite being difficult to analyze mathematically.

By tuning the control parameters by hand, one gains insight into how the parameters relate to the resultant walking. This insight in turn helps speed up the tuning process.

Exploiting Natural Dynamics

Much of the requirements of the walking controller can be achieved through the natural dynamics of the walking mechanism. Joint limits prevent the knee from inverting; a springy ankle naturally transfers the center of pressure forward as the robot walks forward; and the swing-leg swings naturally like a pendulum. We presented an algorithm for Spring Flamingo which exploits these natural mechanisms. This work will help bridge the gap between powered bipedal walkers which typically do not exploit their natural dynamics, and passive dynamic walkers, which rely completely on their natural dynamics in order to walk.

Walking Fast

We have provided evidence that in order to walk quickly, one must swing their swing-leg and transition it into a stance leg quickly. Therefore, in order to make Spring Flamingo walk quickly, we concentrated on actively swinging the leg using trajectory tracking techniques for the hip. By doing so, we achieved walking at velocities up to $1.25\frac{m}{s}$, which while being only moderate speed for a human, is quite fast for current walking robots.

3D Simulation

Three dimensional bipedal walking can be achieved by adding lateral balance to a planar algorithm. We used lateral foot placement and ankle torque to control lateral stability in a 3D walking simulation. This work should be applicable to the real 3D bipedal robots, further bridging the gap between passive dynamic walkers and powered bipedal robots.

9.1 Discussion

9.1.1 How far can this approach get us?

There are many limitations to the approach taken in this thesis for developing walking algorithms. For a robot with a different shape, size, or number of legs the algorithms will have to be significantly rewritten. If the properties of the robot change, the control parameters will need to be retuned. In order to perform a new task, an entirely new algorithm needs to be rewritten. By concentrating specifically on bipedal walking we have developed algorithms which are specific to the walking problem. However, we believe the principles of exploiting inherent robustness in the problem and natural dynamics in the solution are universal.

Furthermore, using what we have learned about walking and others have learned about running, it may now be feasible to attempt other tasks. We believe that the approach of developing simple algorithms based on an understanding of simple physical models can be extended to many exciting problems. These include jumping, skipping, and simple gymnastics maneuvers. As Hodgins and Raibert showed [Hodgins & Raibert (1990)], once you have a good running algorithm, performing flips requires only a simple extension. Perhaps performing robotic Tai-Chi will only require simple extensions to the walking algorithm.

Of course in order to tackle these challenging problems, it is wise to have a well stocked tool box of approaches. It is useful to explicitly understand the approaches which were used in this thesis, and others which may have aided, as described below.

9.1.2 Approaches to Developing Bipedal Walking Robots

Like any engineering problem, bipedal walking can be pursued with a number of different techniques. I list four broad categories of tools here. These are reverse engineering, automation, mathematical analysis, and intuition. Each tool has its advantages and disadvantages. Since bipedal walking is a fairly complex problem, it makes sense to try to use techniques from all four categories.

Unfortunately, some methods are often religiously promoted over others. For example, using intuition to develop heuristic controllers is often frowned upon because it is not rigorous. Mathematical analysis is often criticized because it is too limited and complicated. Reverse engineering nature is considered cheating by some and too limited since we do not know much about how animals control walking. Automation through learning or parameter search is not rigorous and does not give insight into why the learned parameters are good.

In this thesis, I consciously attempted to avoid such arbitrary arguments and embrace all four techniques to whatever extent I could. However, intuition was promoted the most as we felt that using simple control strategies based on physical intuition was the quickest route to getting the robot to walk. Future work will focus on using all four techniques to their fullest extent.

Intuition

Intuition is a powerful tool for developing bipedal control algorithms. Simple heuristic control laws can be developed such as "move the center of pressure on the stance foot forward in order to slow down." Intuition can be obtained through experience, by studying simple models of bipedal walking, and by simply asking what you, yourself, are doing when performing a certain maneuver such as balancing on one leg. The major advantage of using intuition in developing control laws is that the barrier between understanding bipedal walking and controlling bipedal walking is as small as possible.

In this thesis intuition was used extensively in developing control strategies, implementing them, and tuning their parameters. Keeping the control intuitive allows quick modification, debugging, and tweaking since all the control laws and parameters have a physical interpretation.

Reverse Engineering

Bipedal walking robots are copies of their natural counterparts. By putting two sticks on the end of a torso, we are already using reverse engineering. Of course we can extend reverse engineering since anything we can learn about how animals control walking can potentially be used to control bipedal robots. This knowledge is currently limited but is growing rapidly.

In this thesis reverse engineering was used to help answer more of the "what" than the "how". For instance, when pushed sideways one usually puts their leg out and places their foot in a place which allows them to slow down and catch their balance. The neural mechanisms of how this is achieved are currently unknown. Reverse engineering can motivate the intuitive strategy of "use foot placement to control lateral speed when moving too quickly sideways" but implementing this strategy requires more insight and experimentation. Biological systems also motivate the use of natural dynamic mechanisms to simplify the control of walking. In future work, as more is learned about the biology of walking, reverse engineering can be used to a fuller extent.

Mathematical Analysis

There is a lack of examples in which traditional control theory is used to synthesize control laws for bipedal walking robots. There are only limited examples of using traditional control theory to fully prove stability of existing control laws. These are both difficult problems and are being actively pursued by many researchers.

In this thesis, we used mathematical analysis to a more limited extent. Simple mathematical models of walking were used to help develop control rules for various aspects of bipedal walking and for increasing our understanding and intuition of bipedal walking. The qualitative nature of these models helped to develop control strategies, while the quantitative nature in some cases helped tune the parameters.

Mathematical control tools can also be used for accomplishing sub-tasks of the walking problem. For instance, in Chapter 7 and Appendix B we used adaptive control techniques to swing the hip quickly during swing in order to achieve faster walking.

Automation

Automation can be applied through learning, adaptation, or automatic search of parameter spaces. Automation techniques are often more powerful and general purpose than other techniques. However, they often are used as black boxes and thus sometimes do not aid in understanding the problem at hand, even if they produce a good solution.

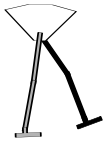
In this thesis we used both manual tuning and genetic algorithms to automatically search parameter spaces. The genetic algorithms were used only in simulation due to the large number of trials required. Adaptively tuning parameters and automatically learning control strategies is an exciting possible area for future work in bipedal walking.

9.2 Further Work

9.2.1 3D Walking Robot

All of the experimental work on a real robot in this thesis was for planar walking. Not only is a planar walker impractical for real world tasks, it also misses a lot of the flavor of human walking. Whether the goal of bipedal robot research is to make a practical robot, or to help study human walking, it is therefore important to make a fully 3D robot.

The 3D simulation was a start at applying the techniques of this thesis to 3D robots, but simulation has a lot to be desired. We are currently building a 12 degree-of-freedom (3 in each hip, one in each knee, 2 in each ankle) biped. Since the simulation parameters were determined based on the physical robot, we are confident that similar algorithms will work on the real robot. We are currently working on starting, stopping, and balancing when pushed sideways, and have been fairly



successful in simulation. Walking sideways and at an angle, and dynamically turning while walking are two challenging maneuvers which we will tackle after straight walking is achieved.

9.2.2 Understanding Human Walking

We believe that one of the most relevant applications for building a biomimetic robot, such as a bipedal walker, is to help understand its real-life counterpart. While the algorithms presented in this thesis were based more on an abstract model of walking than they were in accurately replicating human walking, many of the concepts apply to both. For example, understanding the simple models presented in Chapter 3 helps us to understand why humans use certain strategies to balance or walk. Understanding the limits to maximum walking speed with the robot motivated experiments on humans which showed that both robots and humans are limited by their minimum swing time.

As more is learned about human walking, the feasibility of the hypotheses can be tested on bipedal robots. Eventually a complete model of how humans control walking may be developed and verified on a robot. Besides being useful for verification, bipedal robots may also be useful for discovering the aspects of walking which are important and what to look for when studying human walking. Knowledge gained from human walking will help improve the algorithms used for the robots while knowledge gained from the robots will help further understand human walking. It therefore makes sense for more research groups to become multi-disciplinary with people experienced in biomechanics, neuroscience, robotics, and other fields working together for the common purpose of understanding the mechanics and control of human walking.

9.2.3 Central Pattern Generators

Central Pattern Generators (CPGs) are found in many animals which locomote in rhythmic ways. There is evidence that CPGs are used in swimming lamprey [Stein et al. (1997)], walking cats [Pearson (1976)], and even walking humans [Wickelgren (1998)].

Several researchers in the robotics community have showed that CPGs may be useful for robotic tasks as they have been shown to have very good entrainment, robustness and internal stability properties. Williamson (1999) used CPGs to control robotic arms which can perform simple juggling, turn a crank, and play with a slinky. Taga (1995*a*,*b*) used CPGs in the control of a bipedal walking simulation. However, we are unaware of anyone who has successfully applied Central Pattern Generators to a real bipedal robot. This is clearly an important area of future work.

9.2.4 Multi-Joint Muscle Groups

Animals have muscles which cross several joints [Kapandji (1982)]. For example, the gastrocnemius activates both the ankle and the knee; the rectus femoris activates both the knee and the hip. Perhaps multi-joint muscles have advantages in simplifying control or in achieving higher efficiency.

Simplifying Control

It may be possible to design and construct the actuators and actuation connection of a robot in order to simplify the control. For example, joints which have correlated torques during a portion of the walking gait may benefit from an actuator which passes over both joints with a lever arm ratio corresponding to the torque ratio. Perhaps instead of actively computing the desired torques, that actuator could be fired in more of a on-off fashion during that phase. Also, separate actuators may be wired in groups for the same effect. The robot design or hard-wiring may simplify the commands from the higher level control centers.

Efficiency

Looking at Figure 7-2 we see that during the swing phase when Spring Flamingo is walking fast, the hip does a lot of positive work while the knee does a lot of negative work. Since there are separate actuators for both joints, the total work done by the robot is the sum of the absolute values of the

hip and knee work. However, if there were a multi-joint actuator actuating both the hip and the knee and the lever arm at both joints was properly chosen, perhaps that actuator could apply the positive hip power minus the negative knee power, resulting in increased efficiency. This would be accomplished since the actuator would have to move a shorter distance due to the hip requiring actuator motion in one direction and the knee requiring actuator motion in the other direction during the first part of the swing phase.

9.2.5 Producing a Practical Robot

For a bipedal robot to be practical it must be able to survive falls and stand back up; and work for extended periods of time with on-board power. These are two challenging engineering problems. Batteries are still too heavy for the amount of energy they provide. Fuel cells have a lot of promise but are not yet economically available in the required sizes. Combustion engines or turbine generators could easily provide the required power but are noisy and create exhaust and thus are less practical in an indoor environment. Eventually an adequate energy source will be developed, especially since the automobile and cell phone markets will drive the technology.

Making the robot survive a fall will require extensive engineering design but can be done. Standing up from a fall will require either arms or some clever method such as grappling onto a nearby object.

All these issues are solvable but will take lots of work. Eventually we will be able to develop a biped which is mechanically robust and reliable and has a lot of endurance. These robots will be useful in situations where they can be tele-operated or semi-autonomous, such as amusement parks, nuclear power plants and other industrial settings, and as home entertainment robots. Being useful in autonomous situations, such as working around the house without supervision, however, will require many more advances in Artificial Intelligence.

9.2.6 Learning to Walk

Except for tuning simulation parameters with a genetic algorithm, all of the algorithms developed in this thesis were manually derived and manually tuned. In order to reproduce this work for another robot of a different size or configuration, the work would all have to be repeated. Machine Learning has the potential to make the process faster and perhaps even achieve better performance. Ideally, one could demonstrate a new maneuver to the robot and it would be able to practice on its own until it was able to perform the task. This has been accomplished to some extent with simple tasks such as balancing an inverted pendulum with a robot arm. Whether it's possible to teach more complicated tasks by demonstration without the learner having higher cognitive powers is an open question. However, there most likely is a way to use learning as a way to augment the manual process of algorithm development and tuning. Tuning parameters based on cost function maximization is one way, as was achieved using genetic algorithms in this thesis. Such techniques can surely be used to a fuller extent in future work.

9.2.7 Summary

This thesis has presented an approach to the control of bipedal walking which is simple and efficient and results in fairly smooth and natural walking. Several algorithms were presented for a planar walking robot and a 3D walking simulation.

In Chapter 2 we gave background for this thesis, consisting of previous powered and passive bipedal walking robots, Virtual Model Control, and Series Elastic Actuators.

In Chapter 3 we presented several simple models of bipedal walking. These models helped build physical intuition into walking, which aided in algorithm development.

In Chapter 4 we presented a simple control algorithm for planar bipedal walking. The algorithm was physically based on the simple models presented in Chapter 3. The algorithm used a state machine and simple control strategies based on physical intuition. Simple modifications were made to the algorithm for walking over slopes and rolling terrain.

In Chapter 5 we presented an algorithm which relied on the natural dynamics of a knee-cap, compliant ankle, and passive swing-leg. These mechanisms simplified height control, speed control, and swing-leg control. Speed was found to be self-stabilizing for slow speeds. That is, there was no explicit control mechanism which fed back forward walking speed.

In Chapter 6 we hypothesized that minimum swing time is a major limiting factor of maximum walking speed on Earth. Evidence to support that hypothesis was presented. This knowledge was used in the fast walking algorithms of Chapter 7.

In Chapter 7 we presented an algorithm for walking fast. The algorithm focused on swinging the swing-leg quickly. To do so we used feed-forward inverse dynamics (computed torque) plus active feedback.

In Chapter 8 we presented an algorithm for 3D walking. The algorithm was an extension of the planar algorithms with lateral speed control achieved through lateral foot placement and ankle torque.

While this thesis showed one approach for achieving bipedal walking, there are many other approaches and it is hard to say which is the best. There are many avenues for further work in controlling bipedal walking robots. Eventually this work will lead to better understanding of walking and more competent walking robots.

However, since humans are an obvious comparison with bipedal robots, there will always be more things to work on. Once walking and running and transitioning between the two are fully conquered, we can try ballet, Irish dance, long jumping, pole vaulting, mountain climbing, etc., etc. Studying bipedal maneuvers should keep researchers busy for centuries to come. Perhaps the best cost function to judge today's robots is how well they inspire future generations of walking robot builders to lead the charge.

Bibliography

Adolfsson, J., Dankowicz, H. & Nordmark, A. (1998), '3-d Stable Gait in Passive Bipedal Mechanisms', *Proceedings of 357 Euromech*.

Alexander, R. M. (1988), *Elastic Mechanisms in Animal Movement*, Cambridge University Press.

Alexander, R. M. & Jayes, A. S. (1983), 'A dynamic similarity hypothesis for the gaits of quadrupedal mammals', *Journal of Zoology, London* **201**, 135–152.

Boone, G. (1997), 'Minimum-time Control of the Acrobot', *IEEE International Conference on Robotics and Automation* pp. 3281–3287.

Camp, J. (1997), 'Powered "Passive" Dynamic Walking.', *Masters of Engineering Project Report, Cornell University*.

Cavagna, G. A., Willems, P. & Heglund, N. C. (1998), 'Walking on Mars', *Nature* **393**, 636.

Chew, C.-M., Pratt, J. & Pratt, G. (1999), 'Blind Walking of a Planar Bipedal Robot on Sloped Terrain', *IEEE International Conference on Robotics and Automation* **1**, 381–386.

Coleman, M. & Ruina, A. (April 1998), 'An uncontrolled walking toy that cannot stand still.', *Physical Review Letters* pp. 3658–3661.

Coleman, M., Garcia, M., Ruina, A. & Camp, J. (1997*a*), 'Stability and Chaos in Passive-Dynamic Locomotion', *1997 IUTAM Conference on New Applications of Nonlinear Dynamics and Chaos in Mechanics*.

Coleman, M. J., Chatterjee, A. & Ruina, A. (1997*b*), 'Motions of a Rimless Spoked Wheel: a Simple 3D System with Impacts', *Dynamics and Stability of Systems* **12**(3), 139–160.

Craig, J. J. (1989), *Introduction to Robotics: Mechanics and Control*, Addison-Wesley.

der Linde, R. Q. V. (1998), 'Active leg compliance for passive walking', *IEEE International Conference on Robotics and Automation* pp. 2339–2345.

Donelan, J. M. & Kram, R. (1997), 'The Effect of Reduced Gravity on the Kinematics of Human Walking: A Test of the Dynamic Similarity Hypothesis for Locomotion', *Journal of Experimental Biology* **200**, 3193–3201.

Dunn, E. & Howe, R. (1994), 'Towards Smooth Bipedal Walking', *IEEE International Conference on Robotics and Automation* pp. 2489–2494.

Dunn, E. & Howe, R. (1996), 'Foot Placement and Velocity Control in Smooth Bipedal Walking', *IEEE International Conference on Robotics and Automation* pp. 578–583.

Flash, T. & Hogan, N. (1985), 'The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model', *Journal of Neuroscience* **5**(7), 1688–1703.

Fowble, J. & Kuo, A. (1996), 'Stability and Control of Passive Locomotion in 3D', *Proceedings of the Conference on Biomechanics and Neural Control of Movement* pp. 28–29.

Gabrielli, G. & von Karman, T. (1950), 'What Price Speed?', *Mechanical Engineering* **72**, 775–781.

Garcia, M., Chatterjee, A. & Ruina, A. (1998), 'Speed, Efficiency, and Stability of Small-Slope 2D Passive Dynamic Bipedal Walking', *IEEE International Conference on Robotics and Automation* pp. 2351–2356.

Golden, J. A. & Zheng, Y. F. (1990), 'Gait Synthesis For The SD-2 Biped Robot To Climb Stairs', *International Journal of Robotics and Automation* **5**(4), 149–159.

Goswami, A., Espiau, B. & Keramane, A. (1996), 'Limit cycles and their stability in a passive bipedal gait', *IEEE International Conference on Robotics and Automation* **1**, 246–251.

Goswami, A., Espiau, B. & Keramane, A. (1997), 'Limit cycles in a passive compass gait biped and passivity-mimicking control laws', *Journal of Autonomous Robots*.

Grishin, A. A., Formal'sky, A. M., Lensky, A. V. & Zhitomirsky, S. V. (1994), 'Dynamic Walking of a Vehicle With Two Telescopic Legs Controlled by Two Drives', *International Journal of Robotics Research* **13**(2), 137–147.

Hirai, K., Hirose, M., Haikawa, Y. & Takenaka, T. (1998), 'The Development of Honda Humanoid Robot', *IEEE International Conference on Robotics and Automation*.

Hodgins, J. & Raibert, M. (1990), 'Biped Gymnastics', *International Journal of Robotics Research* **9**(2), 115–132.

Hogan, N. (1985), 'Impedance Control: An Approach to Manipulation: Part I - Theory, Part II - Implementation, Part III - Applications', *J. of Dynamic Systems, Measurement and Control* **107**, 1–24.

Jalics, L., Hemami, H. & Clymer, B. (1996), 'A Control Strategy for Adaptive Bipedal Locomotion', *IEEE International Conference on Robotics and Automation* pp. 563–569.

Kajita, S. & K.Tani (1995), 'Experimental Study of Biped Dynamic Walking in the Linear Inverted Pendulum Mode.', *IEEE International Conference on Robotics and Automation*.

Kajita, S. & K.Tani (1996), 'Adaptive Gait Control of a Biped Robot based on Realtime Sensing of the Ground Profile', *IEEE International Conference on Robotics and Automation* pp. 570–577.

Kajita, S. & Tani, K. (1991*a*), 'Study of Dynamic Biped Locomotion on Rugged Terrain - Derivation and Application of the Linear Inverted Pendulum Mode', *IEEE International Conference on Robotics and Automation* pp. 1405–1412.

Kajita, S. & Tani, K. (1991*b*), 'Study of Dynamic Biped Locomotion on Rugged Terrain - Theory and Basic Experiment', *Fifth International Conference on Advanced Robotics (ICAR'91)* pp. 741–746.

Kajita, S. & Tani, K. (1996), 'Experimental Study of Biped Dynamic Walking', *IEEE Control Systems Magazine* pp. Feb:13–19.

Kajita, S., Tani, K. & Kobayashi, A. (1990), 'Dynamic Walk Control of a Biped Robot along the Potential Energy Conserving Orbit', *IEEE International Conference on Intelligent Robots and Systems* pp. 789–794.

Kajita, S., Yamaura, T. & Kobayashi, A. (1992), 'Dynamic Walking Control of a Biped Robot Along a Potential Energy Conserving Orbit', *IEEE Transactions on Robotics and Automation* **6**(1), 431–438.

Kapandji, I. A. (1982), *The Physiology of the Joints*, Churchill Livingstone.

Kato, I. & Tsuiki, H. (1972), 'Hydraulically Powered Biped Walking Machine With A High Carrying Capacity', *Proceedings of the 4th International Symposium on External Control of Human Extremities* pp. 410–421.

Kato, T., Takanishi, A., Ishikawa, H. & Kato, I. (1981), 'The Realization Of The Quasi Dynamic Walking By The Biped Walking Machine', *Romansy '81* pp. 341–351.

Khatib, O. (1986), 'Real-Time Obstacle Avoidance for Manipulators and Mobile Robots', *IEEE Journal of Robotics and Automation* **5**(1), 90–98.

Khatib, O. (1987), 'A Unified Approach for Motion and Force Control of Robot Manipulators: the Operational Space Formulation', *IEEE Journal of Robotics and Automation* **3**(1), 43–53.

Koechling, J. & Raibert, M. (1988), 'How Fast Can a Legged Robot Run?', *Proceedings of the American Society of Mechanical Engineers Symposium on Robotics* **ASME Dynamic Systems and Controls Division Vol. 11**, 241–249.

Kram, R., Domingo, A. & Ferris, D. P. (1997), 'Effect of Reduced Gravity on the Preferred Walk-Run Transition Speed', *Journal of Experimental Biology* **200**, 821–826.

Kun, A. & Miller, W. T. (1996), 'Adaptive Dynamic Balance of a Biped Robot Using Neural Networks', *IEEE International Conference on Robotics and Automation* pp. 240–245.

Kun, A. L. & Miller, W. T. (1998), 'Unified Walking Control for a Biped Robot Using Neural Networks', *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference* pp. 283–288.

Kuo, A. D. & Zajac, F. E. (1993*a*), 'A Biomechanical Analysis of Muscle Strength as a Limiting Factor in Standing Posture', *Journal of Biomechanics* **26**, 137–150.

Kuo, A. D. & Zajac, F. E. (1993*b*), 'Human Standing Posture: Multi-Joint Movement Strategies Based on Biomechanical Constraints.', *Progress in Brain Research* **97**, 349–358.

Lee, C. R. & Farley, C. T. (1998), 'Determinants of the Center of Mass Trajectory in Human Walking and Running', *Journal of Experimental Biology* **201**, 2935–2944.

McGeer, T. (1990*a*), 'Passive Dynamic Walking', *International Journal of Robotics Research* **9**(2), 62–82.

McGeer, T. (1990*b*), 'Passive Dynamic Walking With Knees', *IEEE International Conference on Robotics and Automation* **2**, 1640–1645.

McGeer, T. (1991), 'Passive Dynamic Biped Catalogue', *Proceedings of the 2nd International Symposium of Experimental Robotics*.

McGeer, T. (1993), 'Dynamics and Control of Bipedal Locomotion', *Journal of Theoretical Biology* **163**, 277–314.

McMahon, T. A. & Bonner, J. T. (1983), *On Size and Life*, Scientific American Books.

McMahon, T. A., Valiant, G. & Frederick, E. C. (1987), 'Groucho Running', *Journal of Applied Physiology* **62**(6), 2326–2337.

Miller, W. T. (1994), 'Real Time Neural Network Control of a Biped Walking Robot', *IEEE Control Systems Magazine* pp. Feb:41–48.

Miura, H. & Shimoyama, I. (1984), 'Dynamic Walk of a Biped', *International Journal of Robotics Research* **3**(2), 60–74.

Mochon, S. & McMahon, T. A. (1979), 'Ballistic Walking: An Improved Model', *Mathematical Biosciences* **52**, 241–260.

Moran, B. (September 1996), 'Back to Basics', *New Scientist* pp. 32–35.

Olree, K. S. & Vaughan, C. L. (1995), 'Fundamental Patterns of Bilateral Muscle Activity in Human Locomotion', *Biological Cybernetics* **73**, 409–414.

Pearson, K. (1976), 'The Control of Walking', *Scientific American* **235**(6), 72–86.

Playter, R. R. (1994), Passive Dynamics in the Control of Gymnastic Maneuvers., PhD thesis, Massachusetts Institute of Technology.

Playter, R. R. & Raibert, M. H. (1994), 'Passively Stable Layout Somersaults.', *Proceedings of the Eighth Yale Workshop on Adaptive and Learning Systems* pp. 66–71. New Haven, CT.

Pratt, G. A. & Williamson, M. M. (1995), 'Series Elastic Actuators', *IEEE International Conference on Intelligent Robots and Systems* **1**, 399–406.

Pratt, J., Dilworth, P. & Pratt, G. (1997), 'Virtual Model Control of a Bipedal Walking Robot', *IEEE International Conference on Robotics and Automation* pp. 193–198.

Pratt, J. E. (1994), Learning Virtual Model Control of a Biped Walking Robot, Unpublished Project Report for MIT's Machine Learning Class (6.858).

Pratt, J. E. (1995), Virtual Model Control of a Biped Walking Robot, Master's thesis, Massachusetts Institute of Technology.

Pratt, J. E. & Pratt, G. A. (1998*a*), 'Exploiting Natural Dynamics in the Control of a Planar Bipedal Walking Robot', *Proceedings of the Thirty-Sixth Annual Allerton Conference on Communication, Control, and Computing* pp. 739–748.

Pratt, J. E. & Pratt, G. A. (1998*b*), 'Intuitive Control of a Planar Bipedal Walking Robot', *IEEE International Conference on Robotics and Automation*.

Pratt, J. E. & Pratt, G. A. (1999), 'Exploiting Natural Dynamics in the Control of a 3D Bipedal Walking Simulation', *Proceedings of the International Conference on Climbing and Walking Robots (CLAWAR99)*.

Pratt, J., Torres, A., Dilworth, P. & Pratt, G. (1996), 'Virtual Actuator Control', *IEEE International Conference on Intelligent Robots and Systems*.

Raibert, M. H. (1986), *Legged Robots That Balance.*, MIT Press, Cambridge, MA.

Raibert, M. H. & Craig, J. J. (1981), 'Hybrid Position/Force Control of Manipulators', *Journal of Dynamic Systems, Measurement, and Control*.

Ringrose, R. (1997), Self-Stabilizing Running, PhD thesis, Massachusetts Institute of Technology.

Robinson, D. W., Pratt, J. E., Paluska, D. J. & Pratt, G. A. (1999), 'Series Elastic Actuator Development for a Biomimetic Robot', *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*.

Rose, J. & Gamble, J. G. (1994), *Human Walking*, Williams and Wilkins.

Salisbury, K. (Dec. 1980), 'Active Stiffness Control of a Manipulator in Cartesian Coordinates', *19th IEEE Conference on Decision and Control* pp. 83–88.

Schaal, S. & Atkeson, C. G. (1993), 'Open loop stable control strategies for robot juggling.', *Proceedings of the IEEE International Conference on Robotics and Automation* **3**, 913–918.

Setiawan, S. A., Hyon, S. H., Yamaguchi, J. & Takanishi, A. (1999), 'Physical Interaction between Human And a Bipedal Humanoid Robot - Realization of Human-follow Walking', *IEEE International Conference on Robotics and Automation* pp. 361–367.

Sias, F. R. & Zheng, Y. F. (1990), 'How Many Degrees-Of-Freedom Does A Biped Need?', *IEEE International Conference on Intelligent Robots and Systems* pp. 297–302.

Slotine, J.-J. E. & Li, W. (1991), *Applied Nonlinear Control*, Prentice-Hall, Cambridge, MA.

Smith, A. C. & Berkemeier, M. D. (1997), 'Passive Dynamic Quadrupedal Walking', *IEEE International Conference on Robotics and Automation* **1**, 34–39.

Spong, M. W. (1994), 'Swing Up Control of the Acrobot', *IEEE International Conference on Robotics and Automation* pp. 2356–2361.

Stein, P. S. G., Grillner, S., Selvertston, A. I. & Sturart, D. G. (1997), *Neurons, Networks, and Motor Behavior*, MIT Press.

Stitt, S. & Zheng, Y. F. (1994), 'Distal Learning Applied to Biped Robots', *IEEE International Conference on Robotics and Automation* pp. 137–142.

Taga, G. (1995*a*), 'A Model of the Neuro-Musculo-Skeletal System for Human Locomotion. I. Emergence of Basic Gait.', *Biological Cybernetics* **73**, 97–111.

Taga, G. (1995*b*), 'A Model of the Neuro-Musculo-Skeletal System for Human Locomotion. II. Realtime Adaptability Under Various Constraints.', *Biological Cybernetics* **73**, 113–121.

Takanishi, A., Ishida, M., Yamazaki, Y. & Kato, I. (1985), 'The Realization Of Dynamic Walking By The Biped Walking Robot WL-10RD', *Proceedings of ICAR '85* pp. 795–801.

Takanishi, A., ok Lim, H., Tsuda, M. & Kato, I. (1990*a*), 'Realization of Dynamic Biped Walking Stabilized By Trunk Motion On A Sagittally Uneven Surface', *IEEE International Workshop on Intelligent Robots and Systems* pp. 323–330.

Takanishi, A., Takeya, T., Karaki, H. & Kato, I. (1990*b*), 'A Control Method For Dynamic Biped Walking Under Unknown External Force', *IEEE International Workshop on Intelligent Robots and Systems* pp. 795–801.

Tani, K., Ikeda, K., Yano, T., Kajita, S. & Matsumoto, O. (1993), 'The Concept of Model Free Robotics for Robots to Act in Uncertain Environments', *Proceedings of the 1993 IEEE/Tsukuba International Workshop on Advanced Robotics* pp. 85–90.

Thuilot, B., Goswami, A. & Espiau, B. (1997), 'Bifurcation and Chaos in a Simple Passive Bipedal Gait', *IEEE International Conference on Robotics and Automation*.

Torres, A. L. (1996), 'Implementation of Virtual Model Control on a Walking Hexapod'. Undergraduate Thesis, Massachusetts Institute of Technology.

Vukobratovic, M., Borovac, B., Surla, D. & Stokic, D. (1990), *Biped Locomotion: Dynamics, Stability, Control, and Applications*, Springer-Verlag, Berlin.

Weyand, P., Ty, J., Patek, E., Bellizzi, M. & Wright, S. (1999), Is Walking More Economical than Running for Bipeds?, In Review at Science.

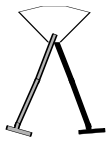
Wickelgren, I. (1998), 'Teaching the Spinal Cord to Walk', *Science* **279**, 319–321. January 16, 1998.

Williamson, M. M. (1995), Series Elastic Actuators, Master's thesis, Massachusetts Institute of Technology.

Williamson, M. M. (1998), 'Exploiting Natural Dynamics in Robot Control', *Proceedings of the Fourth European Meeting on Cybernetics and Systems Research (EMCSR '98)*.

Williamson, M. M. (1999), Robot Arm Control Exploiting Natural Dynamics, PhD thesis, Massachusetts Institute of Technology.

Yamaguchi, J. & Takanishi, A. (1997), 'Development of a Biped Walking Robot Having Antagonistic Driven Joints Using Nonlinear Spring Mechanism', *IEEE International Conference on Robotics and Automation* pp. 185–192.



Yamaguchi, J., Kinoshita, N., Takanishi, A. & Kato, I. (1996), 'Development of a Dynamic Biped Walking System for Humanoid - Development of a Biped Walking Robot Adapting to the Humans' Living Floor', *IEEE International Conference on Robotics and Automation* pp. 232–239.

Yamaguchi, J., Nishino, D. & Takanishi, A. (1998), 'Realization of Dynamic Biped Walking Varying Joint Stiffness Using Antagonistic Driven Joints', *IEEE International Conference on Robotics and Automation* pp. 2022–2029.

Yamaguchi, J., Soga, E., Inoue, S. & Takanishi, A. (1999), 'Development of a Bipedal Humanoid Robot - Control Method of Whole Body Cooperative Dynamic Biped Walking', *IEEE International Conference on Robotics and Automation* pp. 368–374.

Yamaguchi, J., Takanishi, A. & Kato, I. (1993), 'Development of a Biped Walking Robot Compensating For Three-Axis Moment By Trunk Motion', *IEEE International Conference on Intelligent Robots and Systems* pp. 561–566.

Yamaguchi, J., Takanishi, A. & Kato, I. (1994), 'Development of a Biped Walking Robot Adapting to a Horizontally Uneven Surface', *IEEE International Conference on Intelligent Robots and Systems* pp. 1156–1163.

Yamaguchi, J., Takanishi, A. & Kato, I. (1995), 'Experimental Development of a Foot Mechanism with Shock Absorbing Material for Acquisition of Landing Surface Position Information and Stabilization of Dynamic Biped Walking', *IEEE International Conference on Robotics and Automation* pp. 2892–2899.

Yi, K. Y. & Zheng, Y. F. (1996), 'Biped Locomotion by Reduced Ankle Power', *IEEE International Conference on Robotics and Automation* pp. 584–589.

Zheng, Y. F. & Shen, J. (1990), 'Gait Synthesis For The SD-2 Biped Robot To Climb Sloping Surfaces', *IEEE Transactions on Robotics and Automation* **6**(1), 86–96.

Appendix A

Experimental Apparatus

High quality experimental apparatus are crucial in legged robotics. Actuators must produce sufficient torque, speed, and power. The structure must be solid and reliable. Electronics must be sufficiently low in noise and be reliable. Interface software must be easy to use and responsive.

Unfortunately, many robotics projects fail not because of a lack of ideas, but because of poor mechanical design and construction. The amount of time, effort, skill, and money required to make a good robot is universally underestimated. Many legged robot projects are undertaken by computer scientists (as was this one). Therefore, it is extremely important to stress the importance of the mechanics. Common misperceptions are that the control is more important than the mechanics, or worse yet that clever control can make up for poor mechanics.

The legged robots which have been exceptionally successful – Raibert's hoppers, the Ohio State adaptive suspension vehicle, Honda P2 and P3 humanoids, and the Mechanical Engineering Lab's MELTRAN, to name a few, all have good design and construction in common. In some cases good mechanical construction has made the difference. For example, the Honda humanoids use a control algorithm very similar to that used by the Waseda robots over the last 30 years. However, public and even professional perception of the Honda robot seems to be much more positive than that for the Waseda robots.

For this thesis, good design and construction of the experimental robot and other hardware was explicitly stressed from the beginning. Over a year was spent on design and construction. One of the major goals was to produce a "robotic-workhorse" – one which can be used for many years with a high mean time between failure and a short down time. In this chapter we discuss the resulting experimental apparatus.

A.1 Overall Robot Setup

Figure A-1 shows the experimental setup. The robot, Spring Flamingo, is attached to a planarizing boom which constrains the robot to the sagittal plane and forces the robot to walk around a circle of 6 foot radius. While the robot's control computer is on-board, power supplies and motor amplifiers are off-board on the floor with power cables running along the boom. An interface computer communicates with the robot's digital signal processor through an RS-232 serial line.

Spring Flamingo weighs approximately 30 lbs (13.5 kg) and is 3 feet (90 cm) tall. The robot has an actuated hip, knee, and ankle on each leg for a total of six actuated degrees of freedom. The joints are actuated by identical Series Elastic Actuators [Pratt & Williamson (1995)] located in the upper body. Cable drives transmit power from the actuators to the joints. Rotary potentiometers measure the joint angles. Load Cells in the feet measure the ground reaction forces.

Figures A-3, A-2, and A-4 show the top, side, and front views of the robot's body. Electronics include a computer consisting of a Digital Signal Processing Board, Analog Board and Breakout Board; two Power Boards; two Strain Gage Conditioning Boards, and six Force Control Boards attached to the six actuators which are distributed about the body. Two potentiometers are mounted

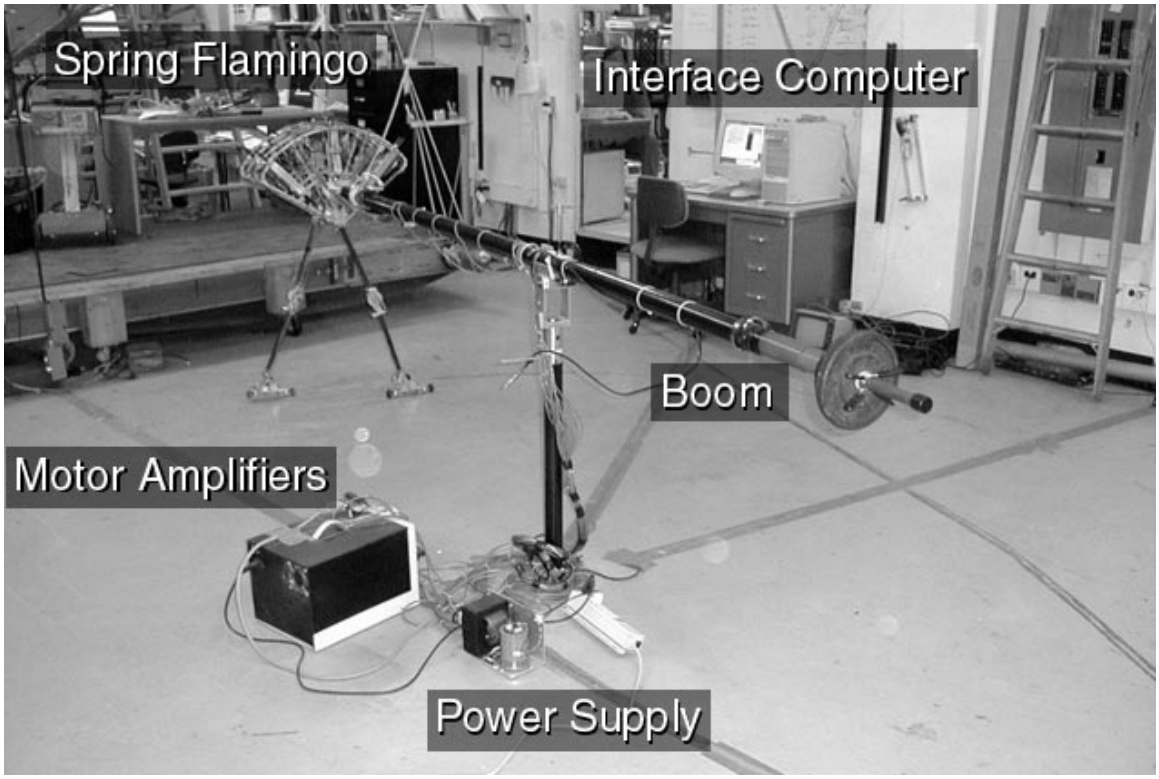


Figure A-1: Experimental setup. The robot, Spring Flamingo, is attached to a planarizing boom. Motor amplifiers and power supplies are located off-board with motor cables running along the boom. An interface computer communicates to the robot through an RS-232 serial cable.

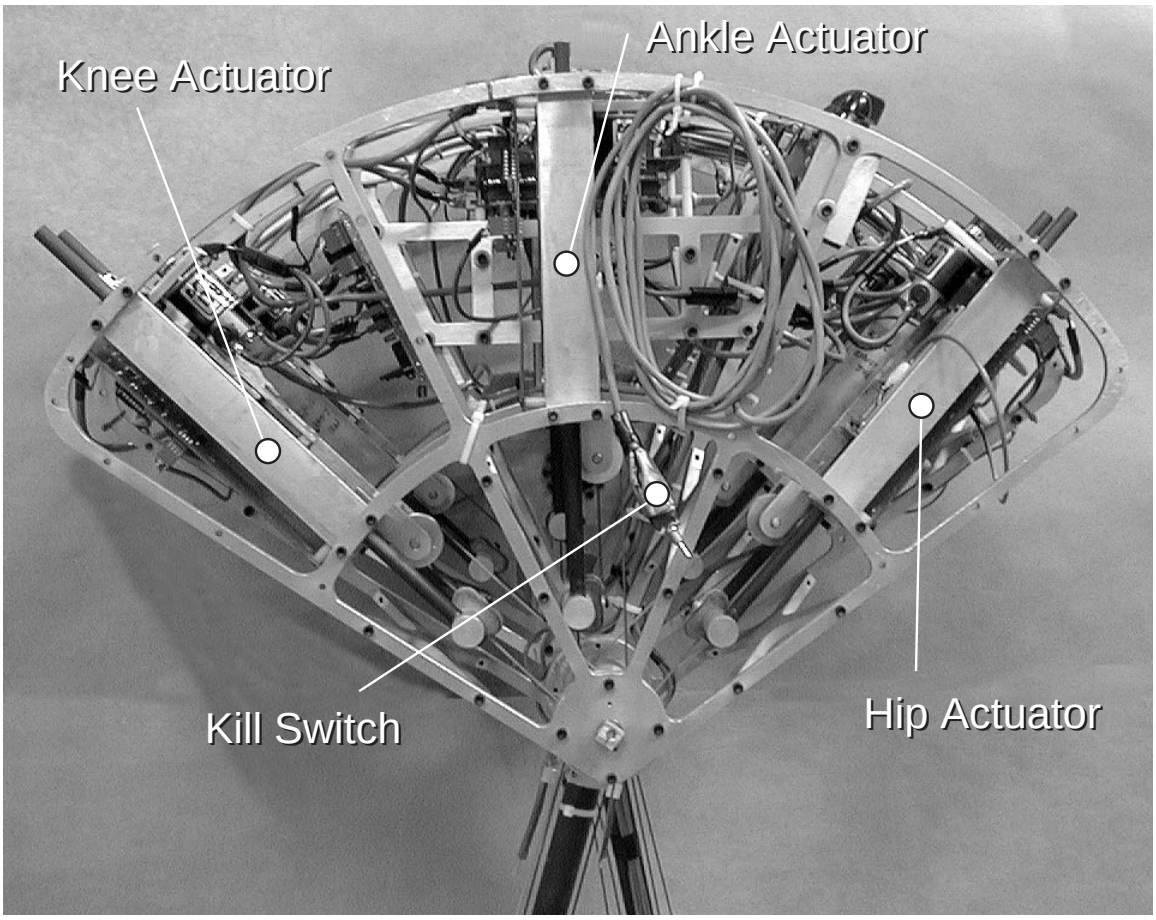


Figure A-2: Side view of Spring Flamingo. The actuators are located in the body, transmitting torque to the joints through cables. A kill switch located on the body is used to disable the motor amplifiers. There are also other external kill switches.

on the body for tuning control parameters while the robot is walking. A kill switch is attached to the robot for disabling power to the motors.

Figure A-5 shows details of the hip and knee. At the hip, one set of cables passes over a $1.75in$ diameter pulley and is fixed at the hip. The other two sets of cables pass over $1.2in$ diameter pulleys and continue on to the knee and ankle. At the knee one set of cables pass over a $2.5in$ diameter pulley and become fixed while the other set passes over a $1.2in$ diameter pulley and continue to the ankle (Figure A-6). At the ankle, the cables pass over a $1.75in$ diameter pulley and become fixed.

All of the pulleys freely rotate about bearings on a shaft. Each shaft is rigidly connected to the bottom link of each joint. A potentiometer measures the angle at each joint.

Limit stops prevent the knee from hyper-extending and help naturally transfer the center of pressure on the foot forward as the robot moves forward (see Chapter 5).

Load cells on the feet are used to measure the ground reaction forces and determine the location of the center of pressure. The load cells were not initially on the robot. In the control of Chapters 4 and 5, ground reaction forces were not used. Load cells were added and used in the fast walking control of Chapter 7.

Rubber wheels (Figure A-6) soften the landing of the foot on the ground, and produce traction. They also allow the foot to swivel with the sagittal plane of the body as the robot moves forward. This swivelling is due to the robot walking in a circle, while the links of the robot remain in the same plane, parallel to the end of the boom.

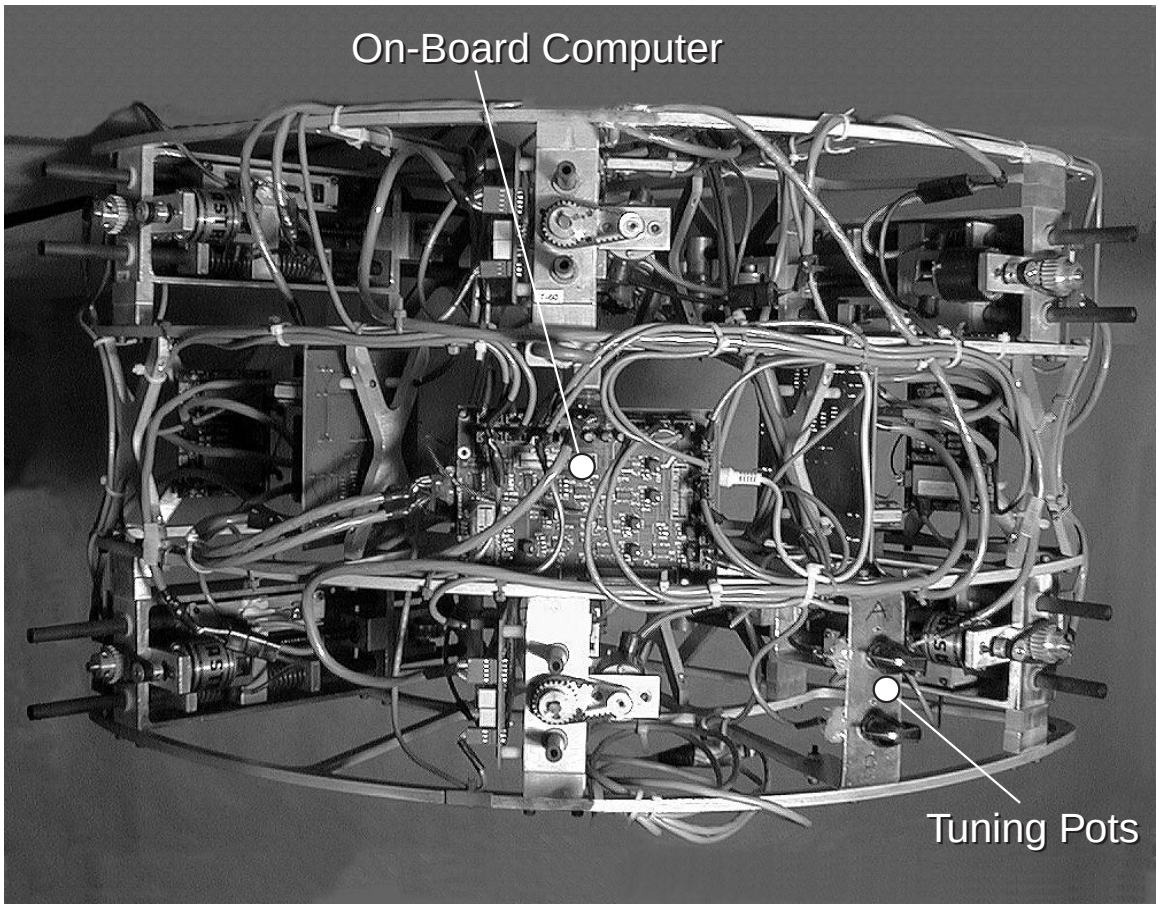


Figure A-3: Top view of Spring Flamingo. The on-board computer system consists of a Digital Signal Processor Board, an Analog Board, and a Breakout Board. Two tuning potentiometers are available for hand-tuning control parameters while the robot is walking.

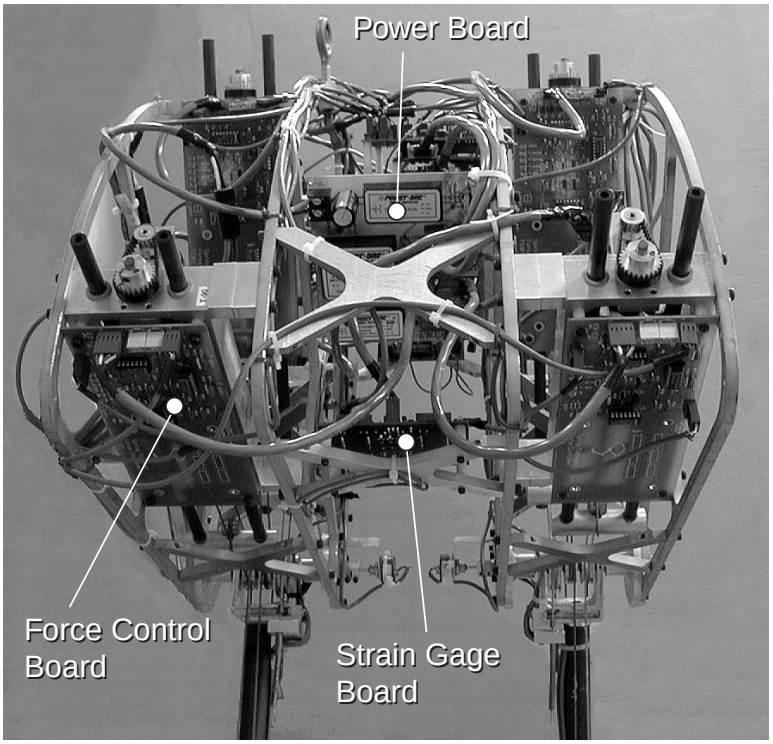


Figure A-4: Front view of Spring Flamingo. Two Power Boards supply power to the on-board computer, a Strain Gage Board, and six Force Control Boards. The Strain Gage Boards condition the signals coming from the load cells located in the feet. The force control boards control the actuator force with a proportional-derivative (PD) controller, servoing motor current as a function of force error.

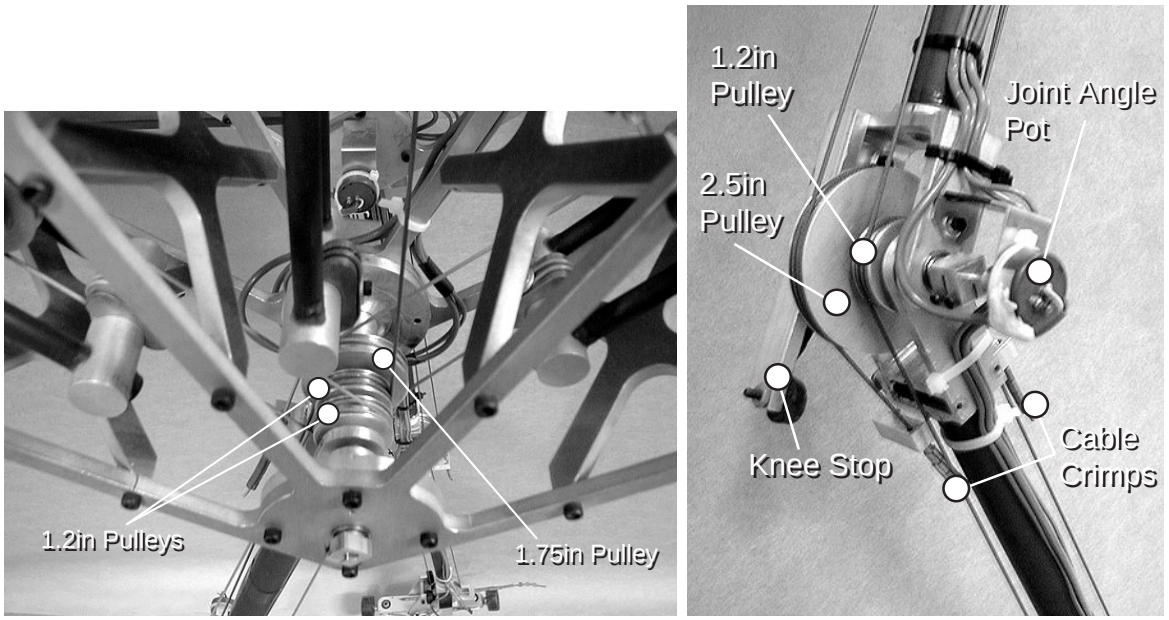


Figure A-5: Hip and knee detail. Cables from all three of the legs actuators pass over pulleys at the hip. Those passing over the $1.75in$ diameter pulley are then fixed to the upper link of the leg. The other two pass over $1.2in$ diameter pulleys and go on to the knee and ankle. At the knee one cable passes over a $2.5in$ pulley and is fixed to the leg's lower link. The other passes over a $1.2in$ pulley and continues to the ankle. A potentiometer at each joint measures the joint angle. A knee stop prevents the knee from inverting. The cables are all terminated with cable crimps.

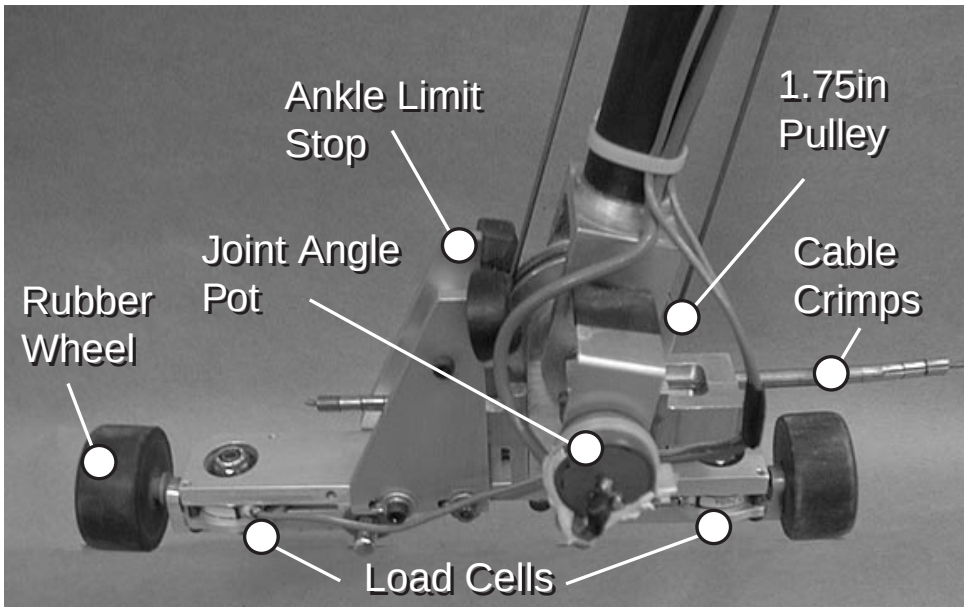


Figure A-6: Ankle and foot detail. The cable driving the ankle passes over a $1.75in$ diameter pulley and is fixed to the foot. A limit stop naturally causes the center of pressure on the foot to travel forward as the robot moves forward. Load cells measure the ground reaction force. Rubber wheels allow the foot to swivel. The first version of the foot did not have load cells and was used for the experiments of Chapters 4 and 5. The new version, shown here, was used for the experiments of Chapter 7.

Table A.1: Continuous torque and maximum speed specifications for the joints. During moderate speed walking, the robot stays within these limits.

Joint	Range	Torque	Speed
Hip	$\pm 45^o$	$14.4Nm$	6.9 rad/sec
Knee	$0^o - 90^o$	$20.6Nm$	6.9 rad/sec
Ankle	$15^o - 45^o$	$14.4Nm$	9.9 rad/sec

Table A.2: Maximum torques and speeds during fast walking. The torques at the joints occassionally exceed the continuous specifications by up to 66%.

Joint	Torque	Speed
Hip	$18Nm$	7.0 rad/sec
Knee	$28Nm$	7.0 rad/sec
Ankle	$24Nm$	10.0 rad/sec

A.2 Joint Technical Data

From previous work on Spring Turkey [Pratt (1994)] and from walking simulations, joint ranges of motion, torque, and speed specifications were determined. These specifications are listed in table A.1. Figure A-7 shows the robots joints at their maximum ranges of motion.

A.3 Actuators

Six identical Series Elastic Actuators [Pratt & Williamson (1995)] provide power to the joints. Series Elastic Actuators provide good force control with a large dynamics range, and shock tolerance. A spring is intentionally placed in series between the motor and the load. By measuring the spring deflection, the output force can be deduced. Feedback is then used to control the output force.

Spring Flamingo's Series Elastic Actuators are shown in Figure A-9 and A-8. A brushed DC motor is attached to a timing belt with a 2.5 reduction. The timing belt drives a ball screw with a lead of $0.3175mm/rev$ ($0.125in$ in/rev). The ball nut pushes a block which pushes on four die

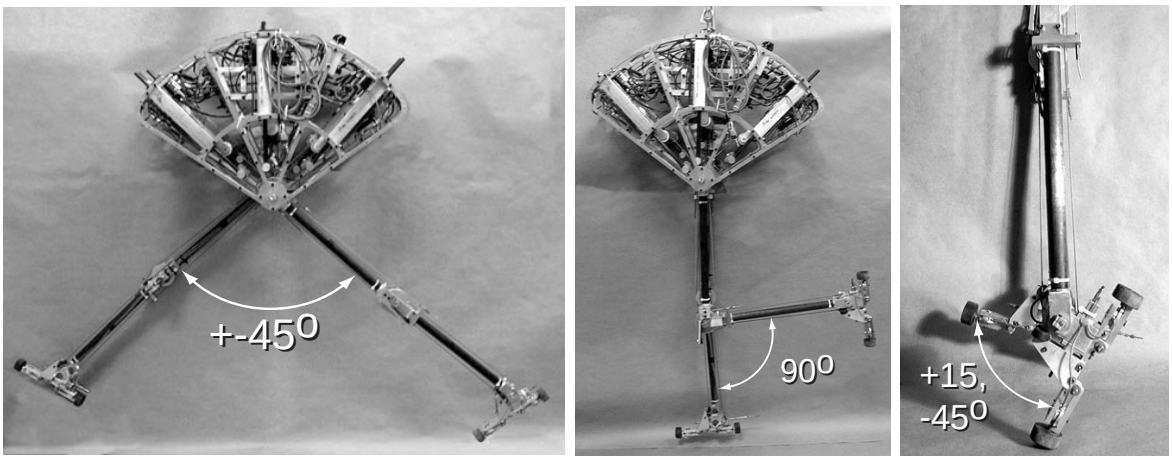
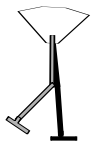


Figure A-7: Hip, knee, and ankle range of motion.



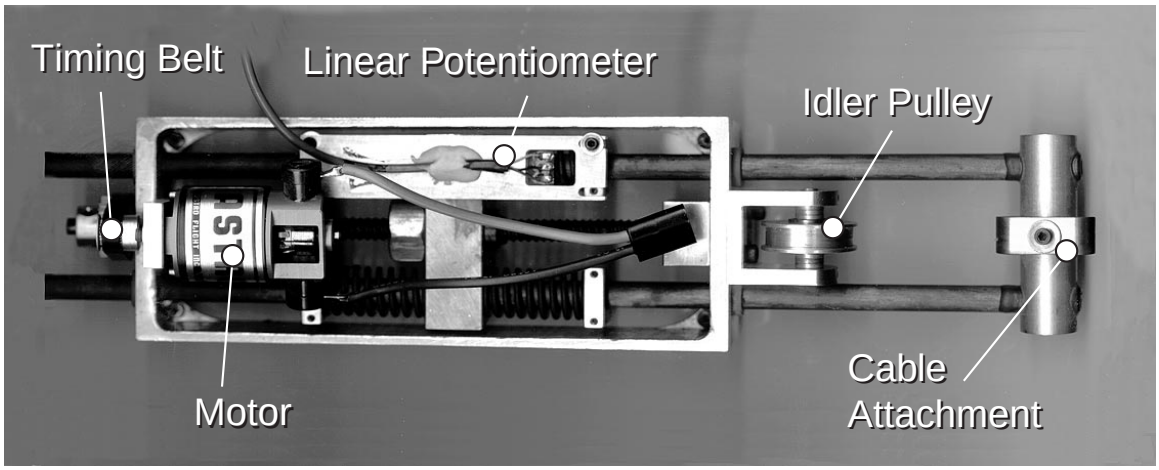


Figure A-8: Actuator top view. A motor drives a timing belt which is attached to a ball screw. The ball screw pushes a block which pushes on four die compression springs. The springs then push on clamps which are clamped to carbon fiber tubes which drive the output. In the robot a cable is attached to the actuator output and then passes over an idler pulley. A linear potentiometer measures the spring deflection and hence the actuator output force.

compression springs each with a spring constant of $30900\frac{N}{m}$ ($176\frac{lbs}{in}$). The four springs together thus have a combined spring constant of $123600\frac{N}{m}$ ($704\frac{lbs}{in}$). The springs push on clamps which are clamped to carbon fiber tubes. The output of the actuator is attached to a cable which winds around an idler pulley and travels to the joint.

Components were selected in order to meet the specifications in Table A.1. The motor has a maximum speed rating of $1095\frac{rad}{sec}$ (10464 RPM) and a continuous torque rating of $0.13Nm$ ($18.6oz-in$). After passing through the 2.5 reduction of the timing belt, and the $0.3175mm/rev$ reduction of the ball screw, this translates to 647 Newtons (146 pounds) of force, and a maximum speed of 0.221 meters per second (8.72 inches per second).

Since the hip and ankle pulleys are $0.044m$ ($1.75in$) in diameter and the knee pulley is $0.064m$ ($2.5in$), this corresponds to $14.4Nm$ maximum torque and $9.9rad/sec$ maximum speed at the hip and ankle and $20.6Nm$ and $6.9rad/sec$ at the knee.

During the moderate speed walking of Chapters 4 and 5, the joint torques and speeds stay within these ranges. However, during fast walking of Chapter 7, the continuous ratings of the actuator are significantly exceeded. Table A.2 show the values achieved during fast walking. We see that the speed limits are matched and the torque limits are exceeded by up to 66%. This results in the motors getting overheated within a couple minutes of walking. Occasionally a motor will burn out and be replaced.

The bandwidth (-3dB frequency) of the actuators is approximately 14 Hz. Force response bode diagrams, taken with no load motion, are shown in Figure A-10. This figure was acquired by commanding a chirp signal as the desired actuator force:

$$F_d = A\sin(2\pi 1.5t^2) \tag{A.1}$$

for $0 < t < 10$ seconds. This gives excitation frequencies from 0 to 30 Hz. A was kept low enough to prevent saturation. Five chirps were commanded while sampling the actual force output. These were then averaged to reduce random noise. The bode diagrams were generated using Matlab's spa command. One standard deviation confidence intervals are dashed. We see that the -3dB cutoff frequency is approximately $90\frac{rad}{s}$ (14Hz).

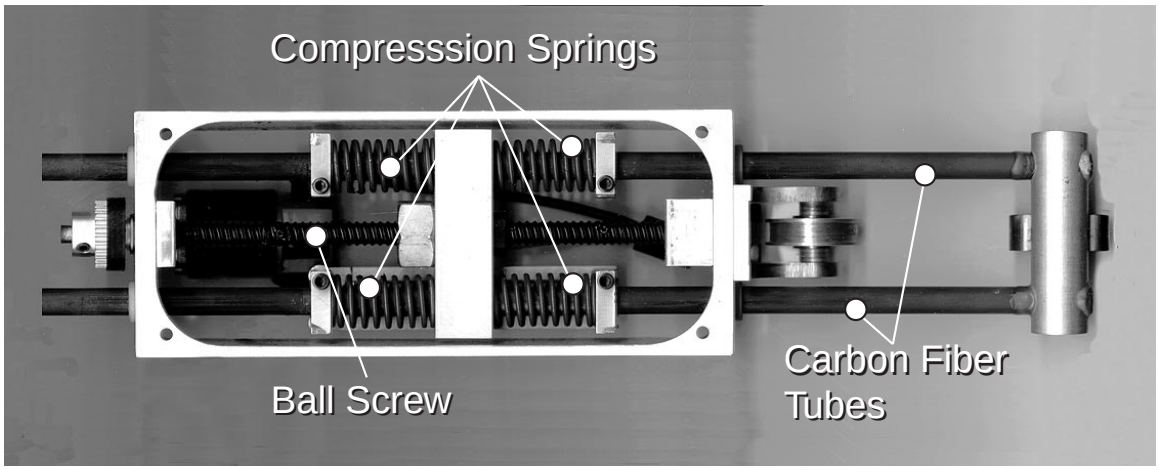


Figure A-9: Actuator Bottom view.

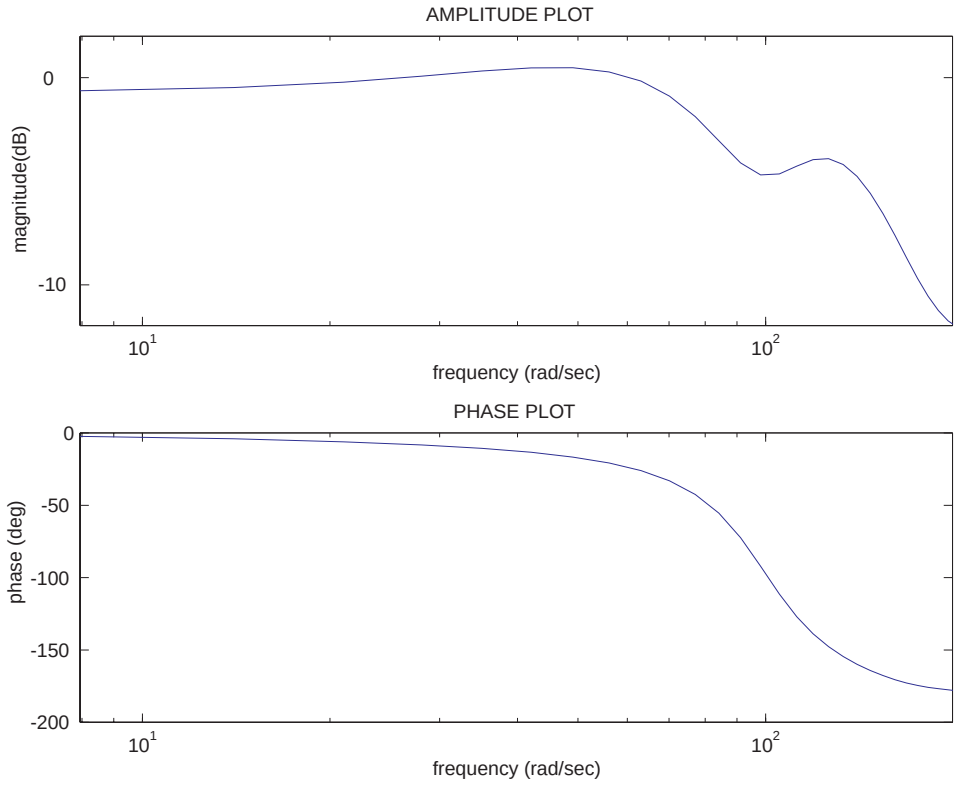


Figure A-10: Experimentally determined bode diagram of actuator force response. The -3dB magnitude cross-over point is at approximately $90\frac{rad}{s}$ (14Hz).

A.4 Joint Torque De-Coupling

Since the cables traveling to the ankle pass over the hip and the knee and since those traveling to the knee pass over the hip, the resultant torque at a joint is a function of all three of the leg's actuator forces:

$$\begin{bmatrix} \tau_h \\ \tau_k \\ \tau_a \end{bmatrix} = \begin{bmatrix} R_h & R_c & R_c \\ 0 & R_k & R_c \\ 0 & 0 & R_a \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{A.2}$$

where τ_h, τ_k, and τ_a are the torques at the hip, knee, and ankle; R_h, R_k, R_a are the radii of the pulleys with terminating cables at the hip, knee, and ankle; R_c is the radius of the pulleys with cables continuing to the lower joints; and f_1, f_2, and f_3 are the actuator forces with cables terminating at the hip, knee, and ankle. The sizes of the pulleys are $R_h = 0.0222m(0.875in)$, $R_k = 0.0318m(1.25in)$, $R_a = 0.0222m(0.875in)$, $R_c = 0.0152m(0.6in)$. To determine what the actuator forces should be, given desired joint torques, we must invert the matrix in Equation A.2:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{R_h} & \frac{-R_c}{R_h R_k} & \frac{R_c^2}{R_h R_k R_a} - \frac{R_c}{R_h R_a} \\ 0 & \frac{1}{R_k} & \frac{-R_c}{R_k R_a} \\ 0 & 0 & \frac{1}{R_a} \end{bmatrix} \begin{bmatrix} \tau_h \\ \tau_k \\ \tau_a \end{bmatrix} \tag{A.3}$$

A.5 Electronics

Figure A-11 shows the layout of Spring Flamingo's electronic components. The control program is run at a update rate of 3.5 milliseconds on a Digital Signal Processing Board made by Digital Designs and Systems, Inc. (www.dideas.com). The DSP Board has a Texas Instruments TMS-C32 DSP chip and 4 Megabytes of RAM. The DSP is stacked on an Analog Board, also made by Digital Designs, and a Breakout Board, made by the author. The Analog Board has 32 channels of analog to digital conversion, 32 channels of digital to analog conversion and 16 digital I/O lines, which is more than sufficient for Spring Flamingo. The Breakout Board filters the incoming signals from the joint potentiometers and differentiates them to obtain velocity information. It also reads the foot forces from the Strain Gage Boards and sends the desired actuator forces to the Force Control Boards. Photographs of these boards are shown in Figure A-12. Schematics for the Breakout Board are in Figures A-17 through A-22.

The DSP, Analog, Breakout, and Strain Gage Boards are powered by the Power Board shown in Figure A-14 with schematic in Figures A-23 and A-24.

The Strain Gage Board is shown in Figure A-14, with schematics in Figures A-27 and A-28. It conditions the signals coming from the two load cells in each foot.

The output from the algorithm running on the DSP is the desired force in each of the actuators. This signal is sent from the Breakout Board to the Force Control Boards. The Force Control Boards, shown in Figure A-13, close the force control loop of the actuators. Linear potentiometers measure the force in each actuator. The Force Control Board implements a PD controller on this signal and sends a signal representing the desired current to the motor amplifiers. The PD gains were tuned such that the closed loop response of the actuators had a roll-off frequency of $20Hz$ and a damping ratio of 0.7.

An interface computer communicates user commands with the Digital Signal Processor and receives data from the robot over an RS-232 serial line. While the robot is walking, all the joint angles, velocities, control parameters, and any intermediate variables can be recorded on the robots internal memory at any rate up to once every 3.5 milliseconds. This data can then be uploaded to

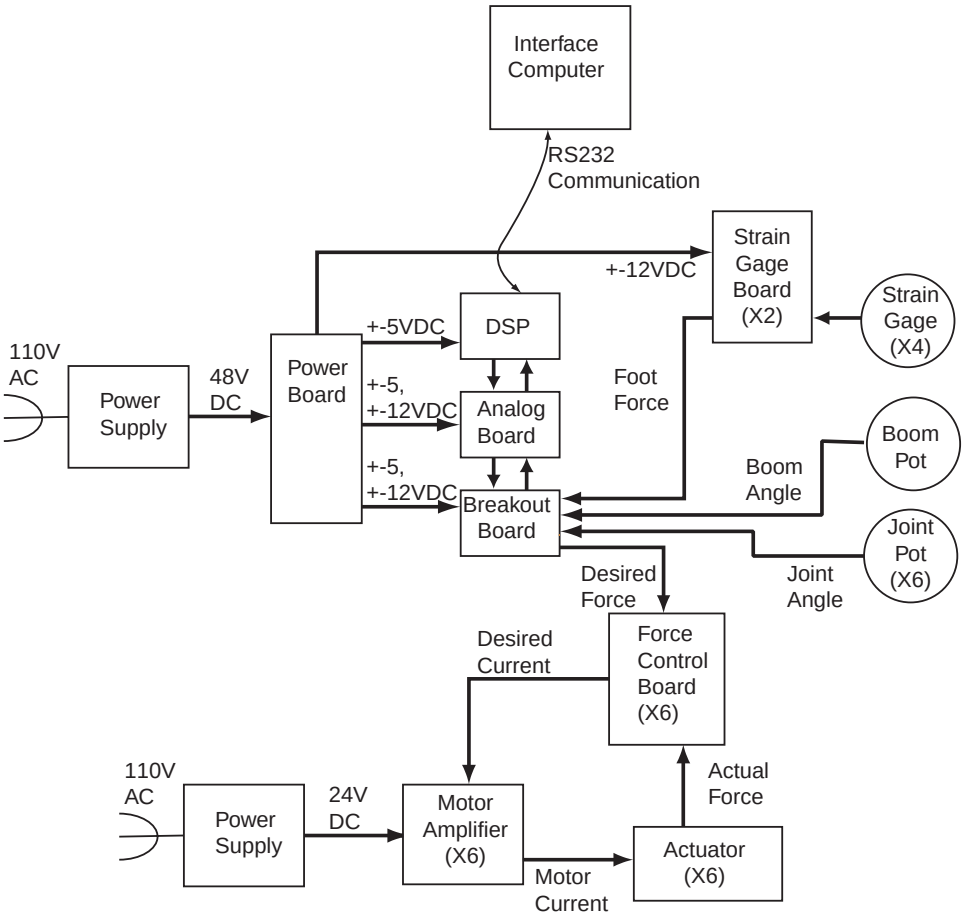


Figure A-11: Electronics Layout. See text for details.

the interface computer, which can display the data in graphical format, as shown in Figure A-15. The data can also be imported into Matlab or other graphing and analysis packages.

A.6 Calibration

All of the sensors and actuators were calibrated either experimentally or analytically based on specifications. The joint potentiometers were calibrated by measuring the voltage reading with the joint straight and bent at 90^o. Since the measured slopes were all within 15%, we chose an average value and used it for all the joint potentiometers. Joint velocity calibration was determined analytically, based on the computed gain of the differentiation circuits. Actuator force was calibrated experimentally by measuring the force on a spring scale for various output voltages. Since the rate was within 15% for all the actuators, we chose an average value and used it for all the actuators. Foot force sensors calibration was determined analytically by specifications from the sensor manufacturer and computed gains of the strain gage conditioning circuit. Zero offset calibration for all the sensors was achieved by suspending the robot with all of its joints straight and averaging 1000 sensor readings.

A.7 Conclusion

Spring Flamingo meets all the required specifications determined by physical simulations. The robot is mechanically sound and has reliable electronics hardware. Thus it has achieved the original goal

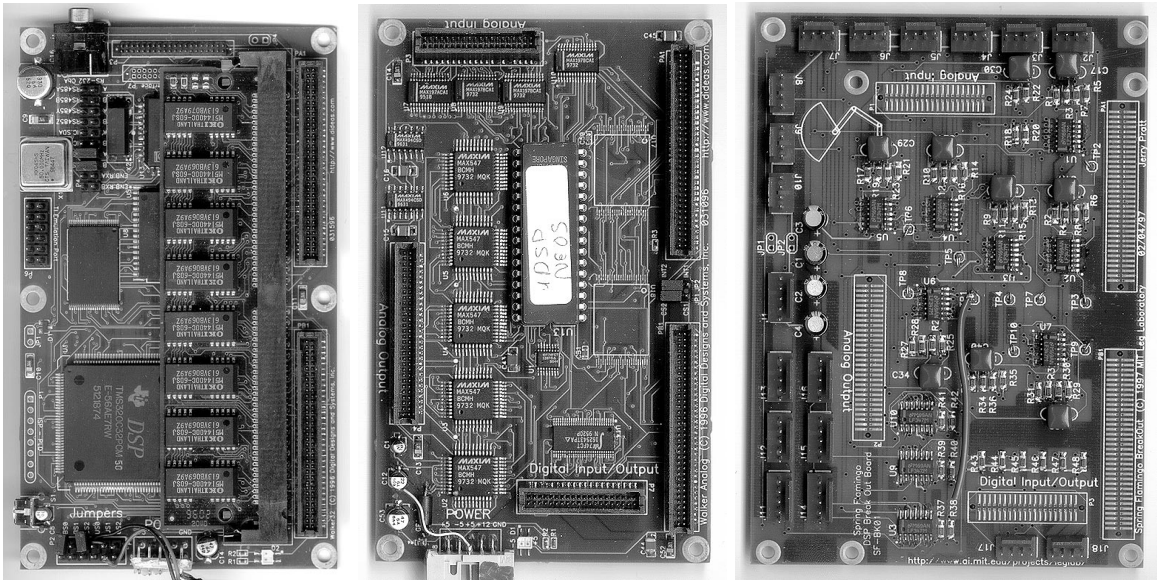


Figure A-12: DSP Board, Analog Board, and Breakout Board forming the on-board computer system.

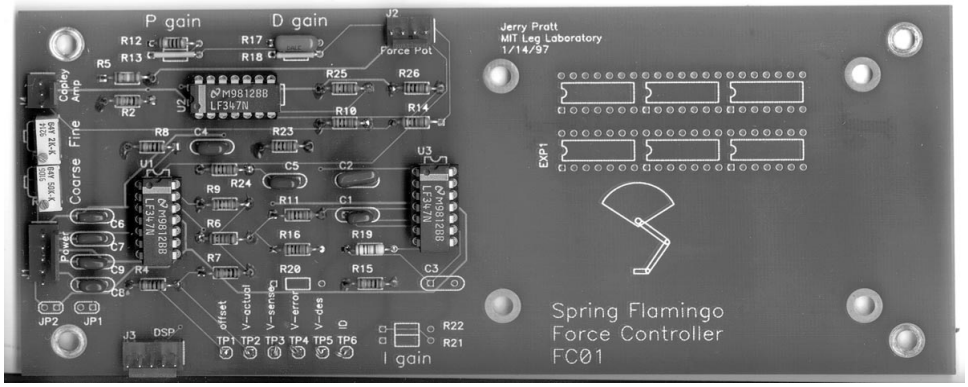


Figure A-13: Force Control Board which implements a PD controller to control the actuator output force.

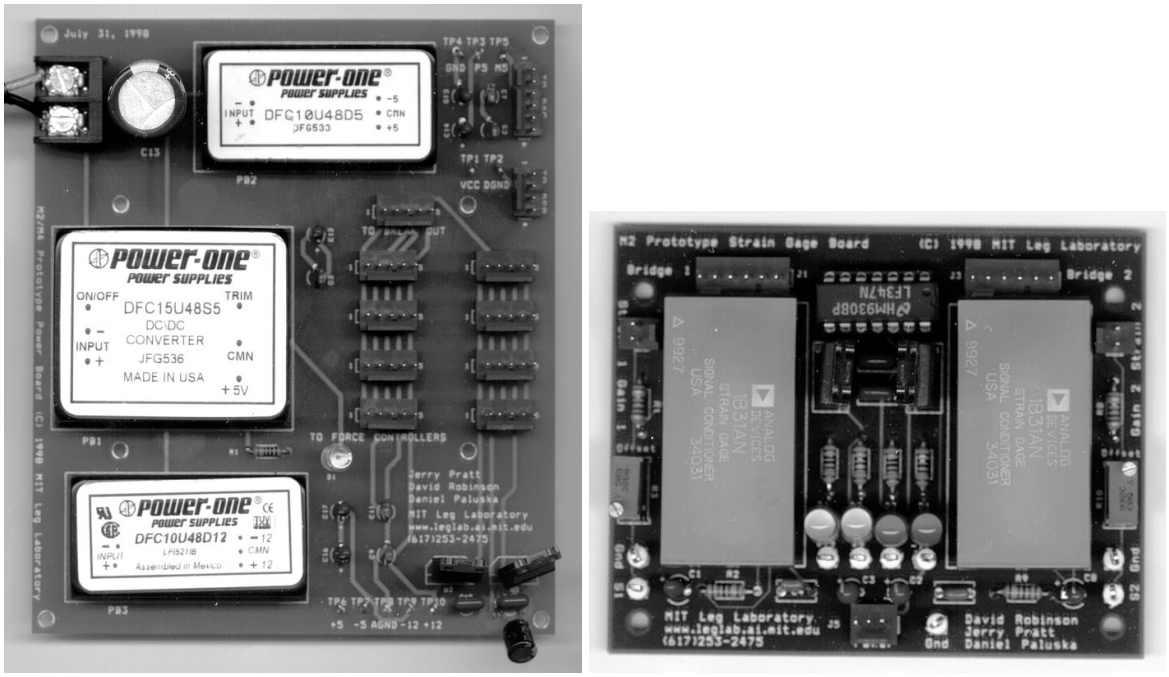


Figure A-14: Power Board (left) and Strain Gage Conditioning Board (right). The Power Board powers the other low-power electonics. The Strain Gage Board conditions the signal from the load cells in the feet.

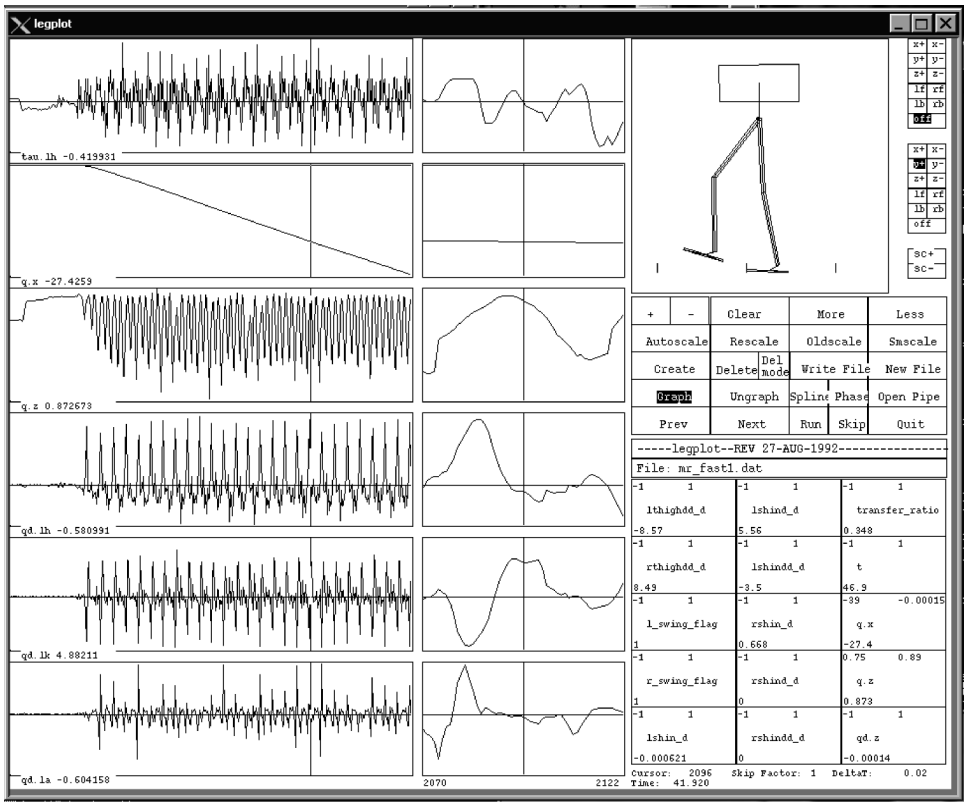


Figure A-15: "Legplot", a graphical display and analysis program showing plots of data uploaded from Spring Flamingo's on-board computer system.

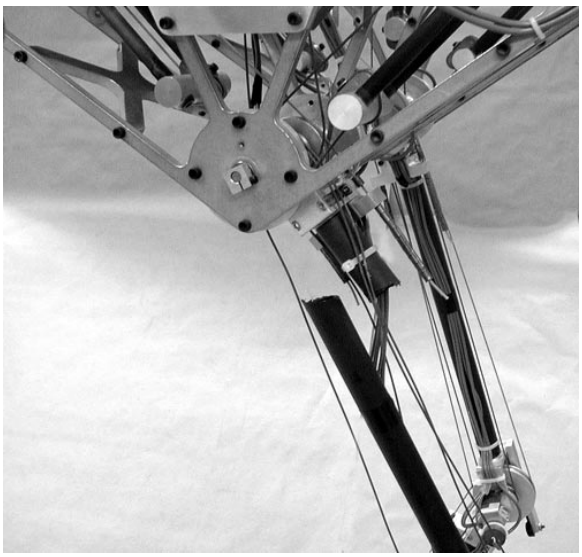


Figure A-16: Spring Flamingo with broken leg. This was the only significant failure of the robot. Luckily, the carbon fiber tube was easily removed by heating the glue to 250 degrees and replaced with a spare.

of becoming a robotic workhorse, operating for 3 years with very little down time. We estimate that the robot has walked over 15 miles in that time! Much of the credit for the success of the robot, as with all successful robots lies in the quality design and construction. We spent over a year designing and building the robot. Parts cost was approximately $25,000. Major off-the-shelf mechanical and electrical parts are listed in Table A.3. Parts suppliers are listed in Table A.4 and A.5.

The only significant failure was a broken leg which occurred due to the leg tube striking the body whenever the robot fell. Eventually a crack was formed and the tube split in two, as shown in Figure A-16. Occassionally a cable breaks which requires a 30 minute repair job.

Slow communication from the robot to the interface computer, due to a slow (38Kbaud) RS-232 serial line requires 1 minute to download the robot program and up to 8 minutes for recorded data to be uploaded to the computer. This caused some delay in analysis, tuning, and debugging turn-around. Our new robots all have ethernet communications which effectively eliminate this delay.

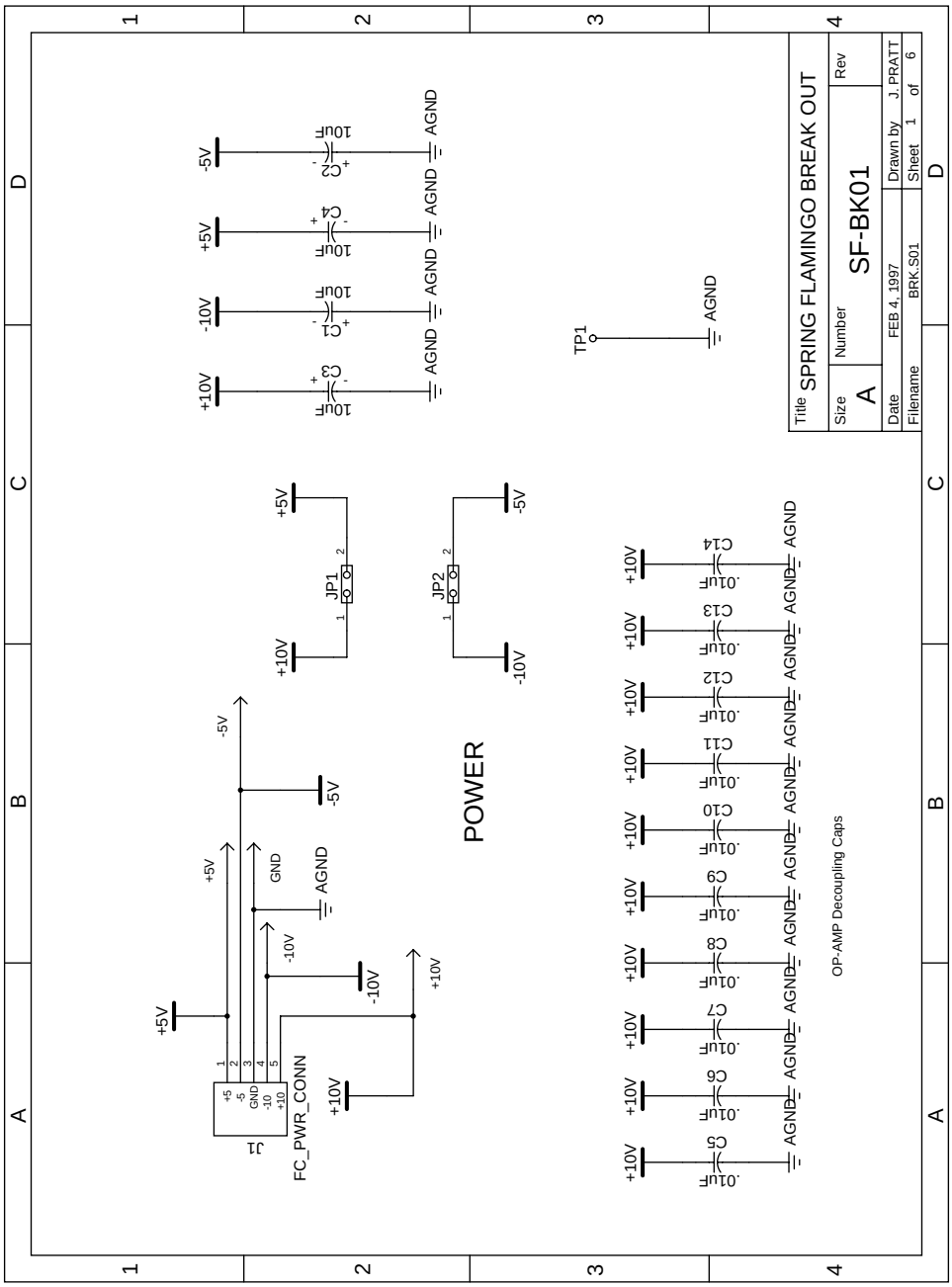


Figure A-17: Breakout Board Schematic Page 1 of 6.

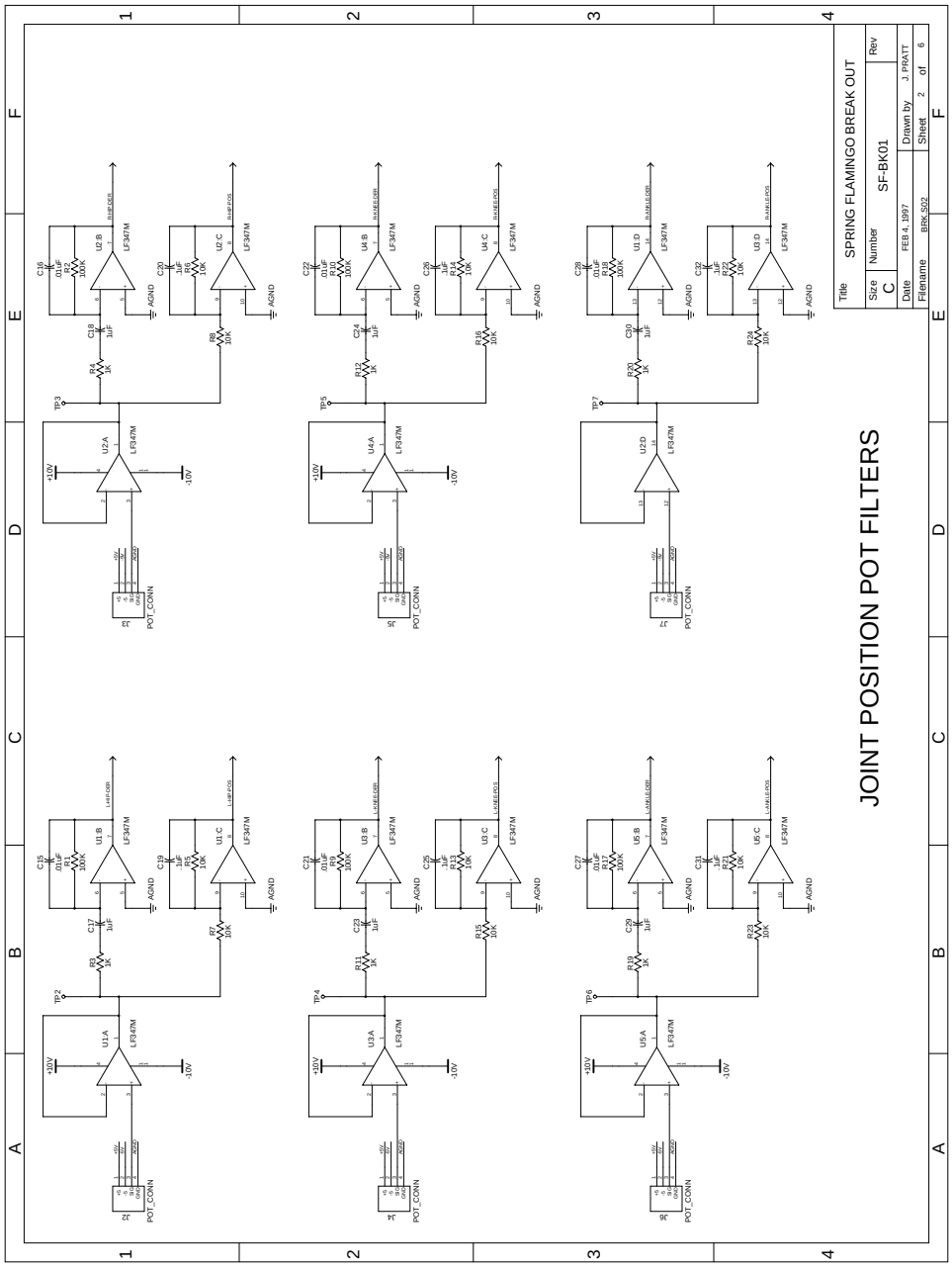


Figure A-18: Breakout Board Schematic Page 2 of 6.

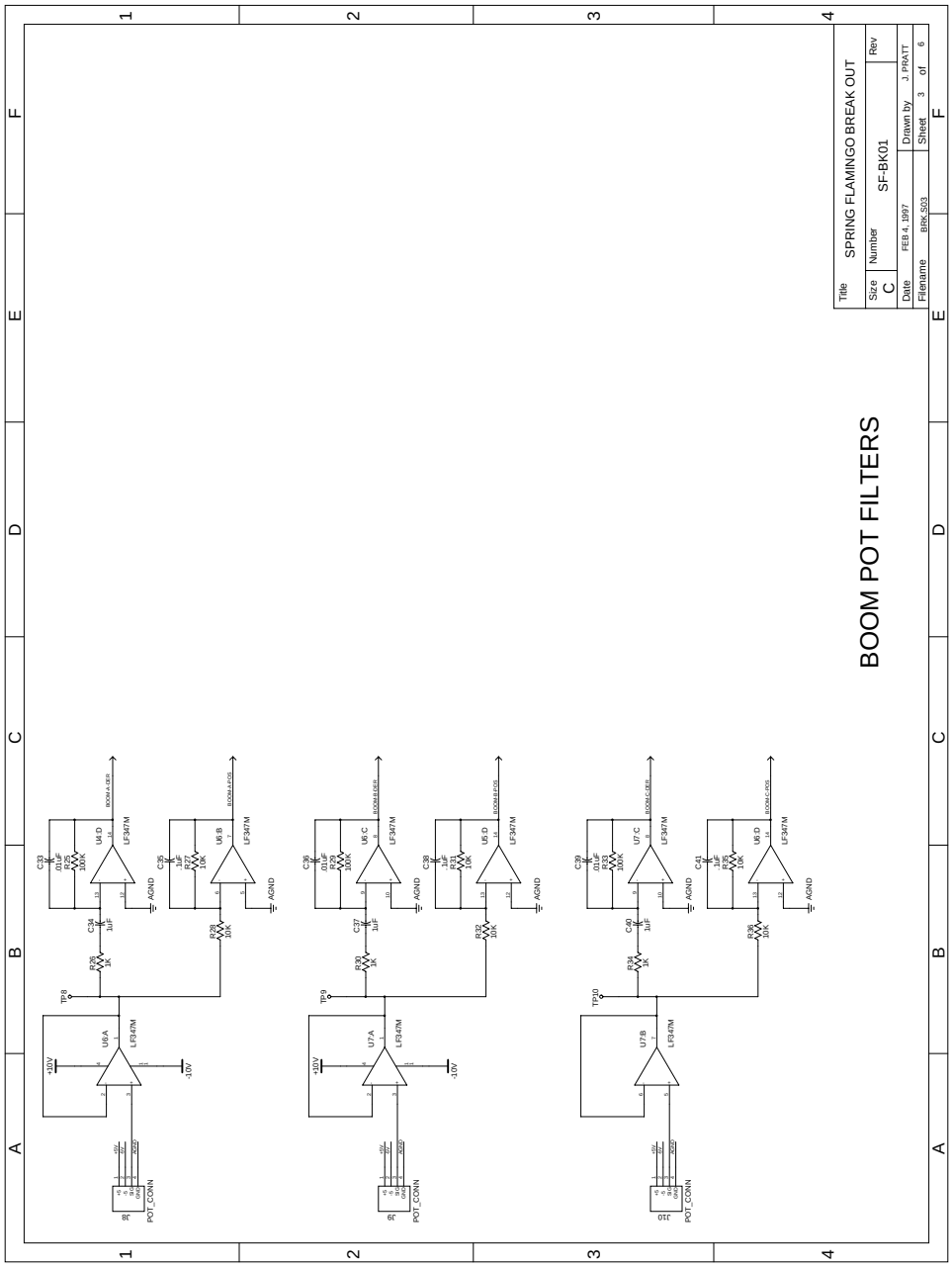


Figure A-19: Breakout Board Schematic Page 3 of 6.

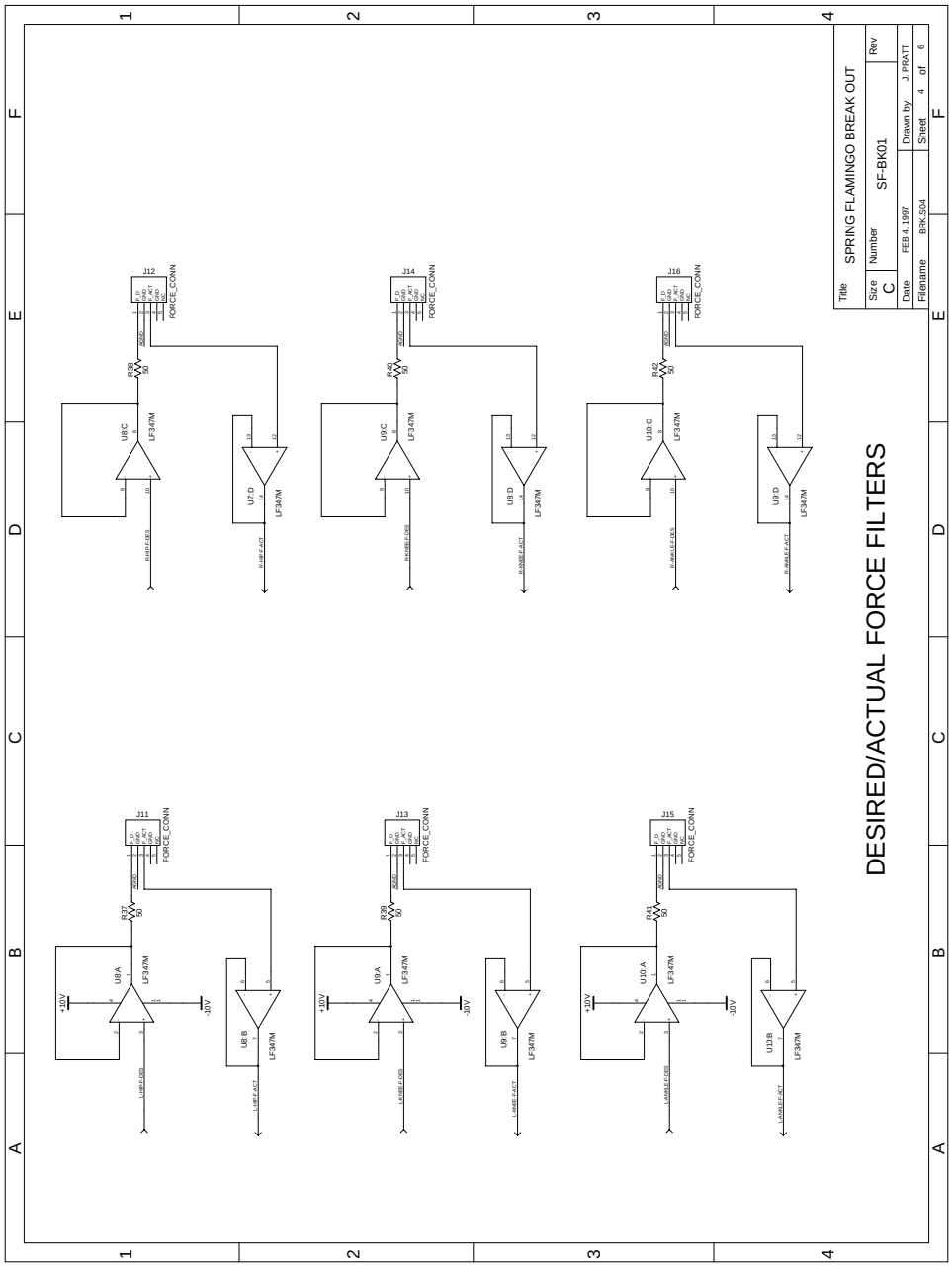


Figure A-20: Breakout Board Schematic Page 4 of 6.

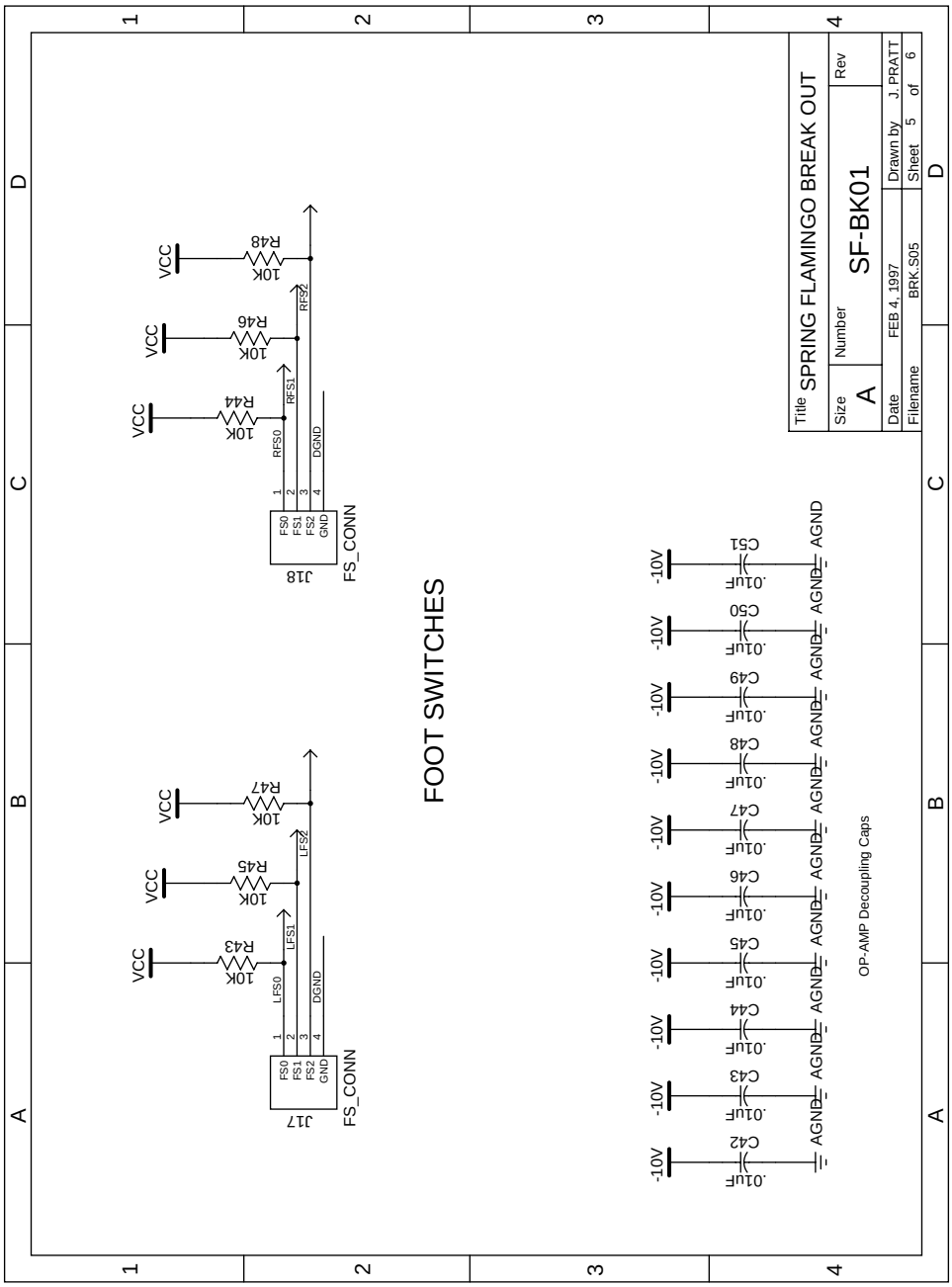


Figure A-21: Breakout Board Schematic Page 5 of 6.

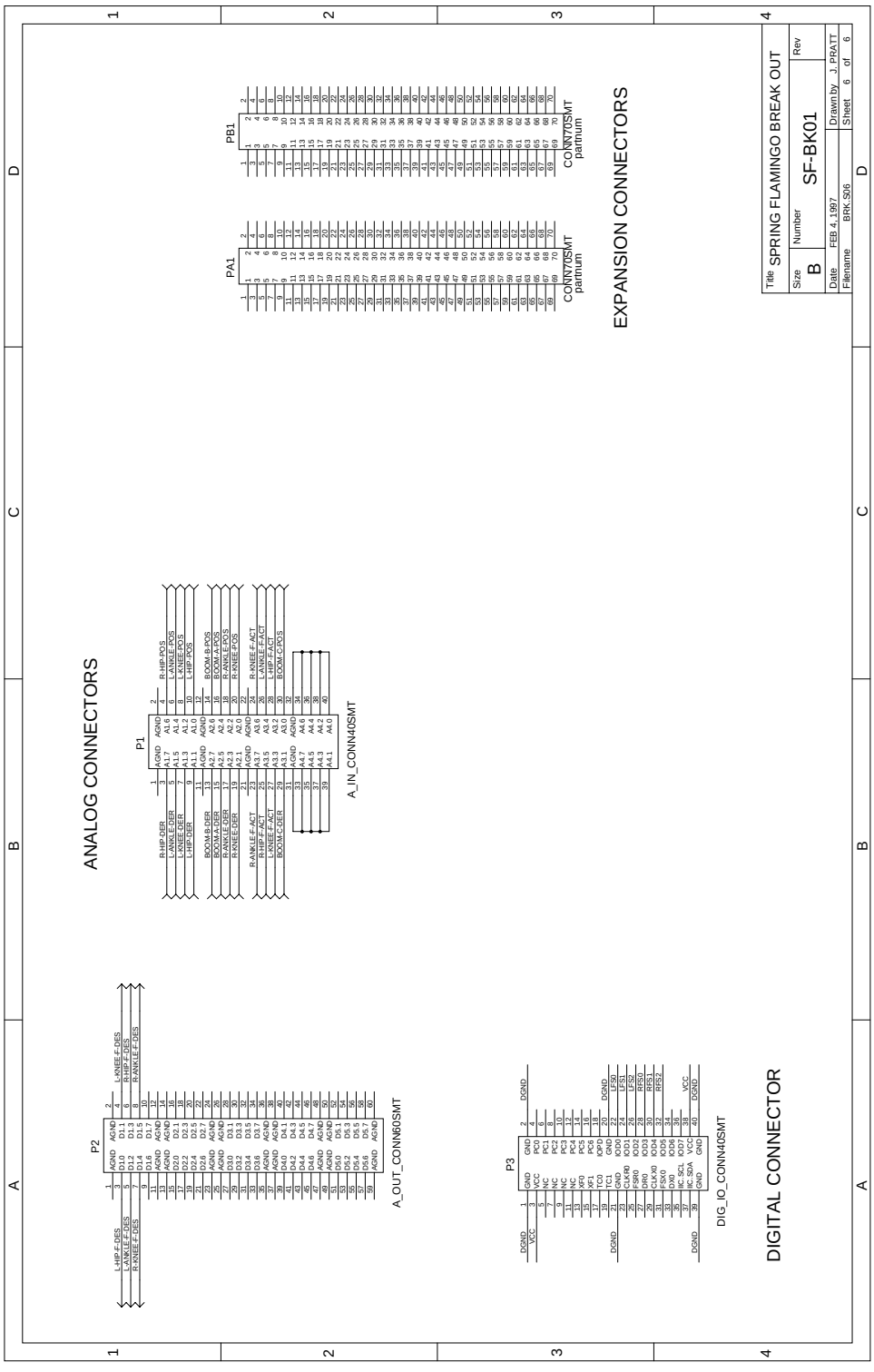


Figure A-22: Breakout Board Schematic Page 6 of 6.

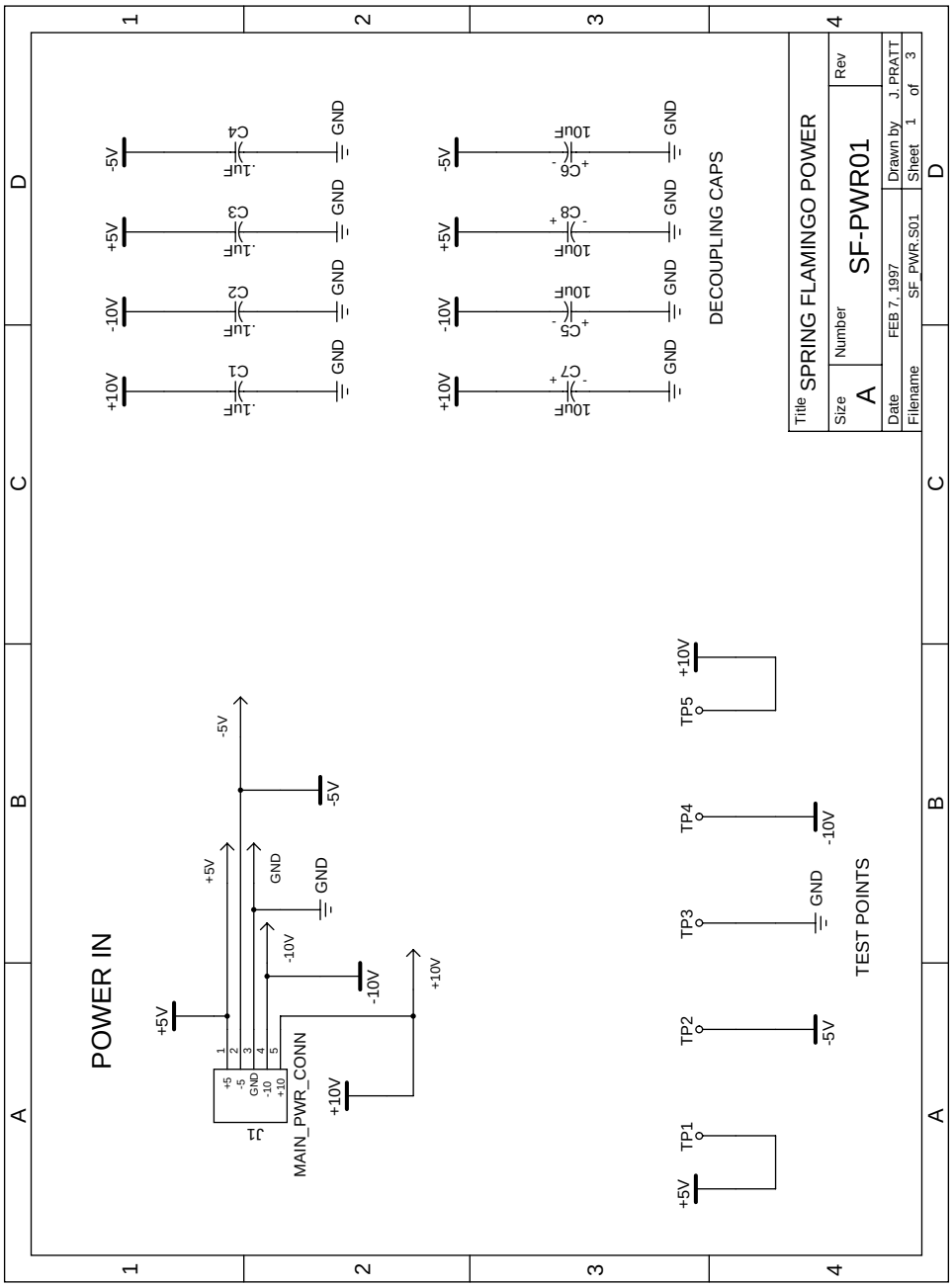


Figure A-23: Power Board Schematic Page 1 of 2.

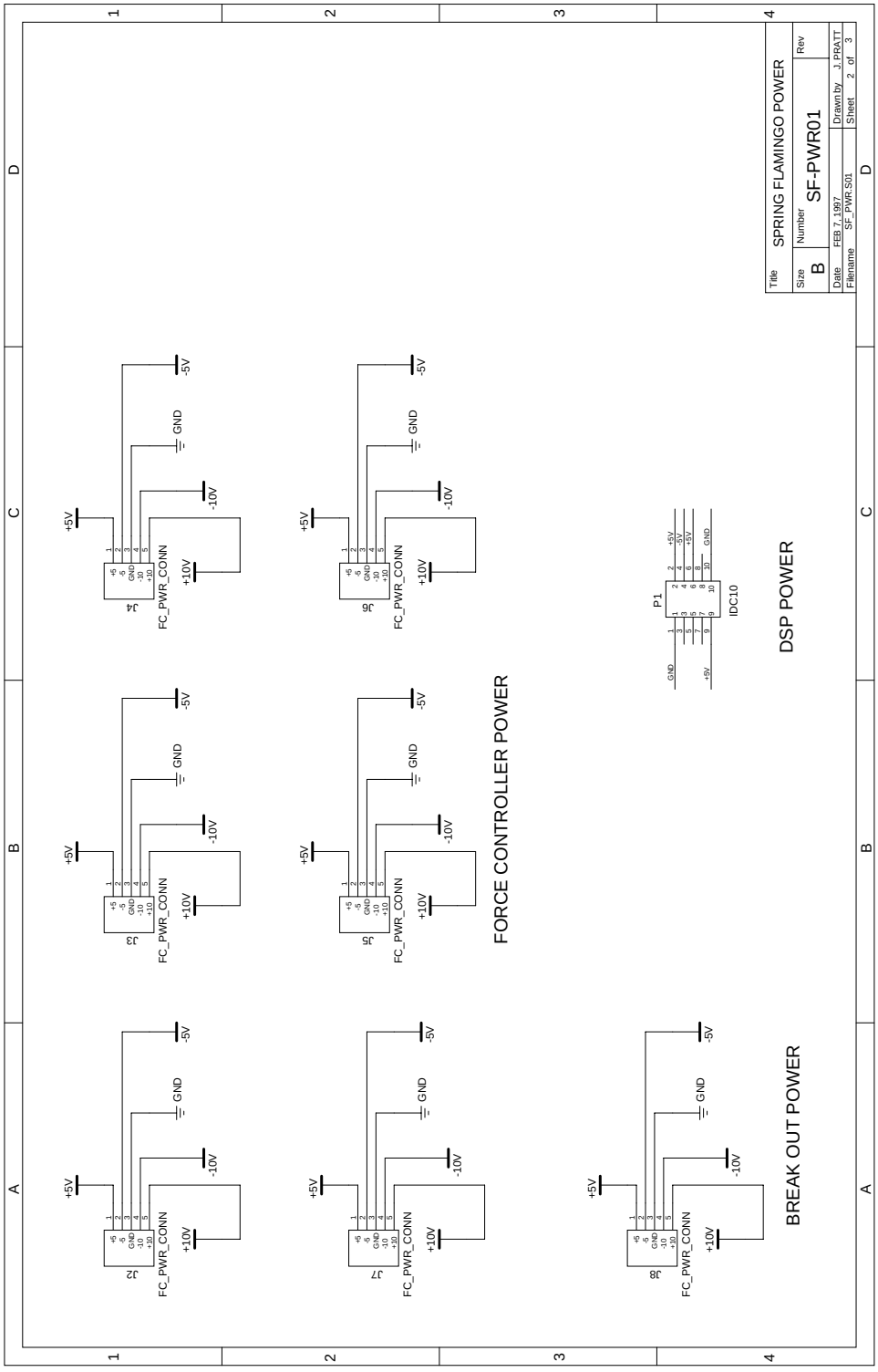


Figure A-24: Power Board Schematic Page 2 of 2.

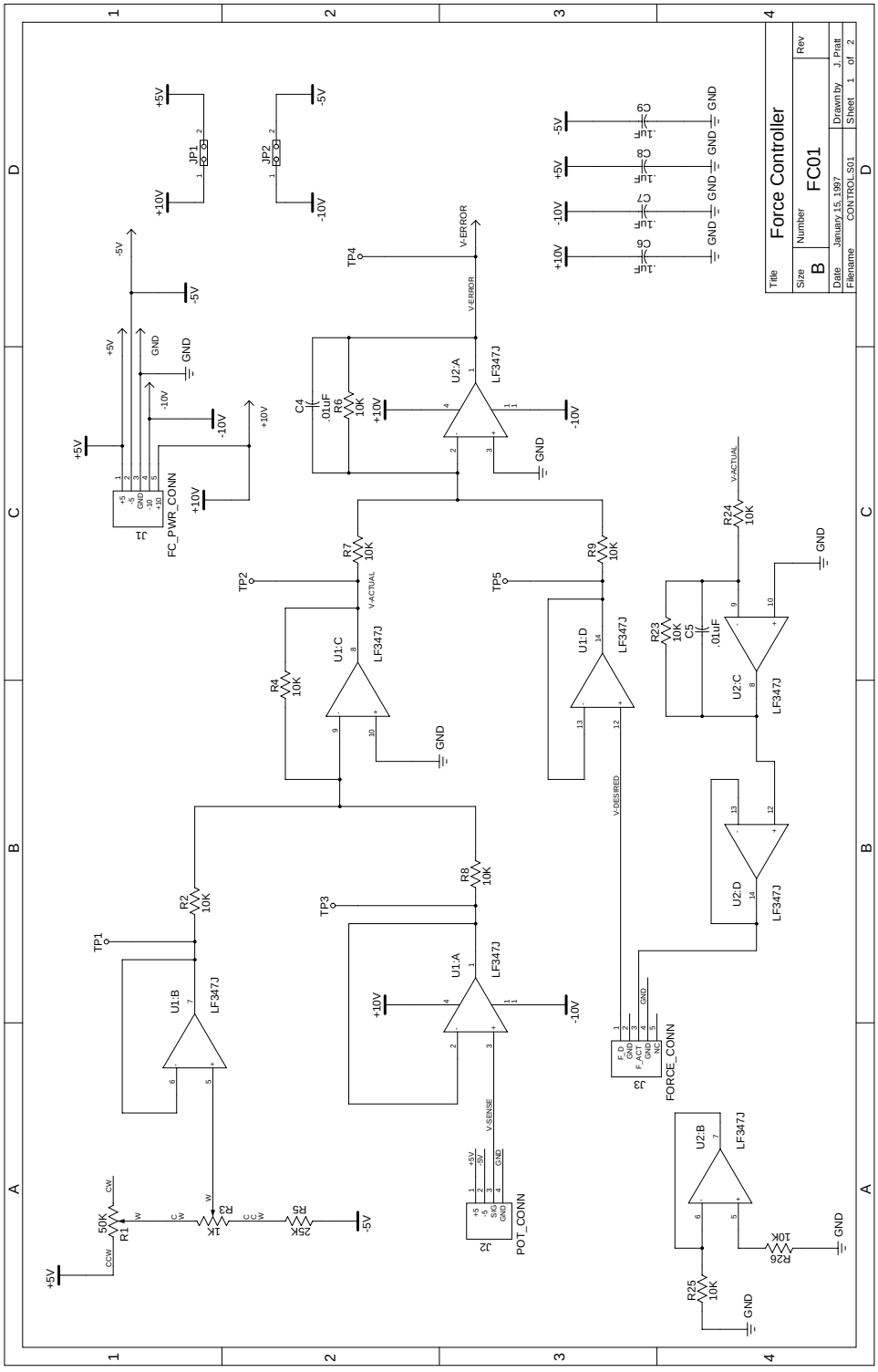


Figure A-25: Force Control Board Schematic Page 1 of 2.

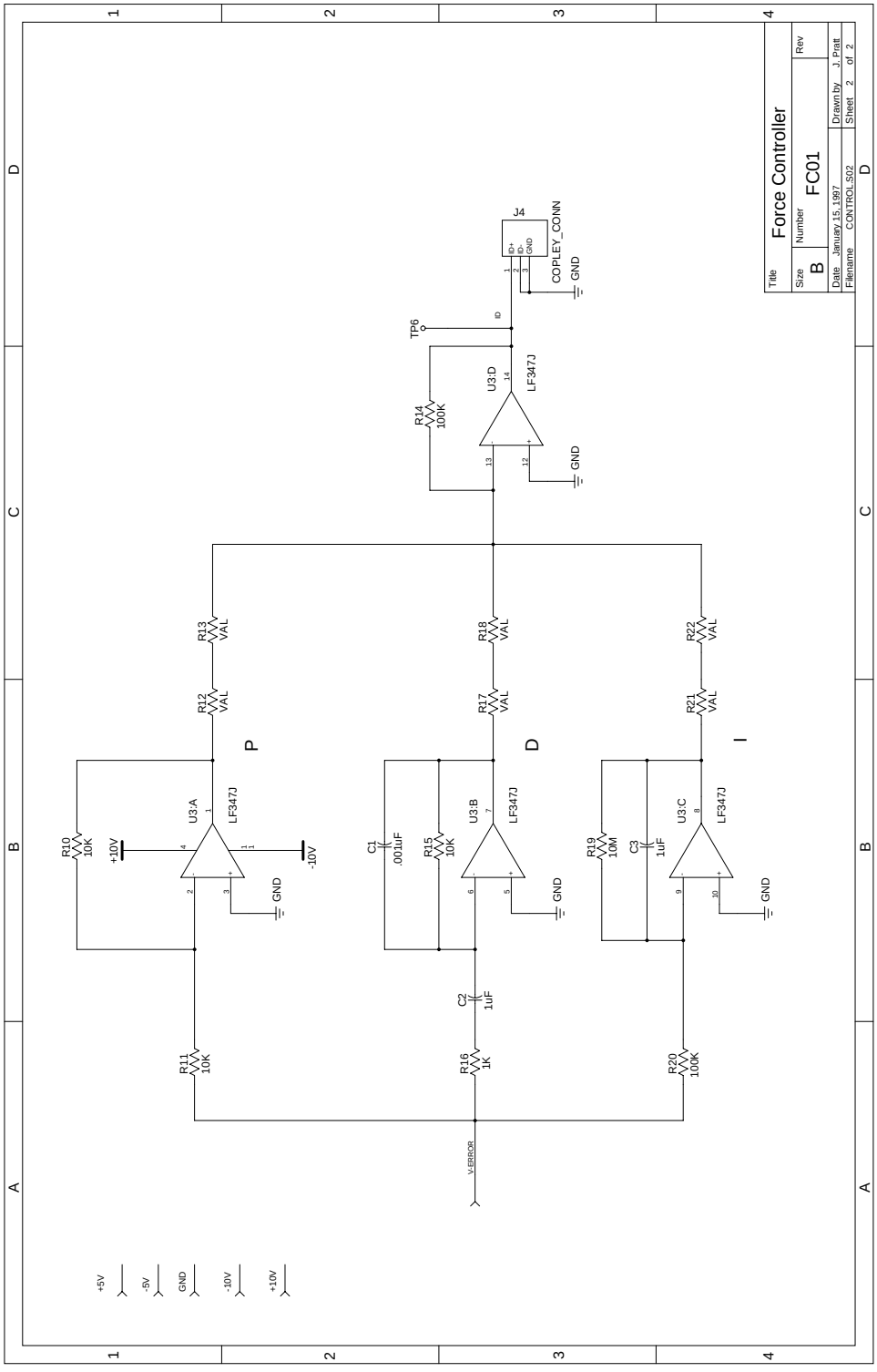


Figure A-26: Force Control Board Schematic Page 2 of 2.

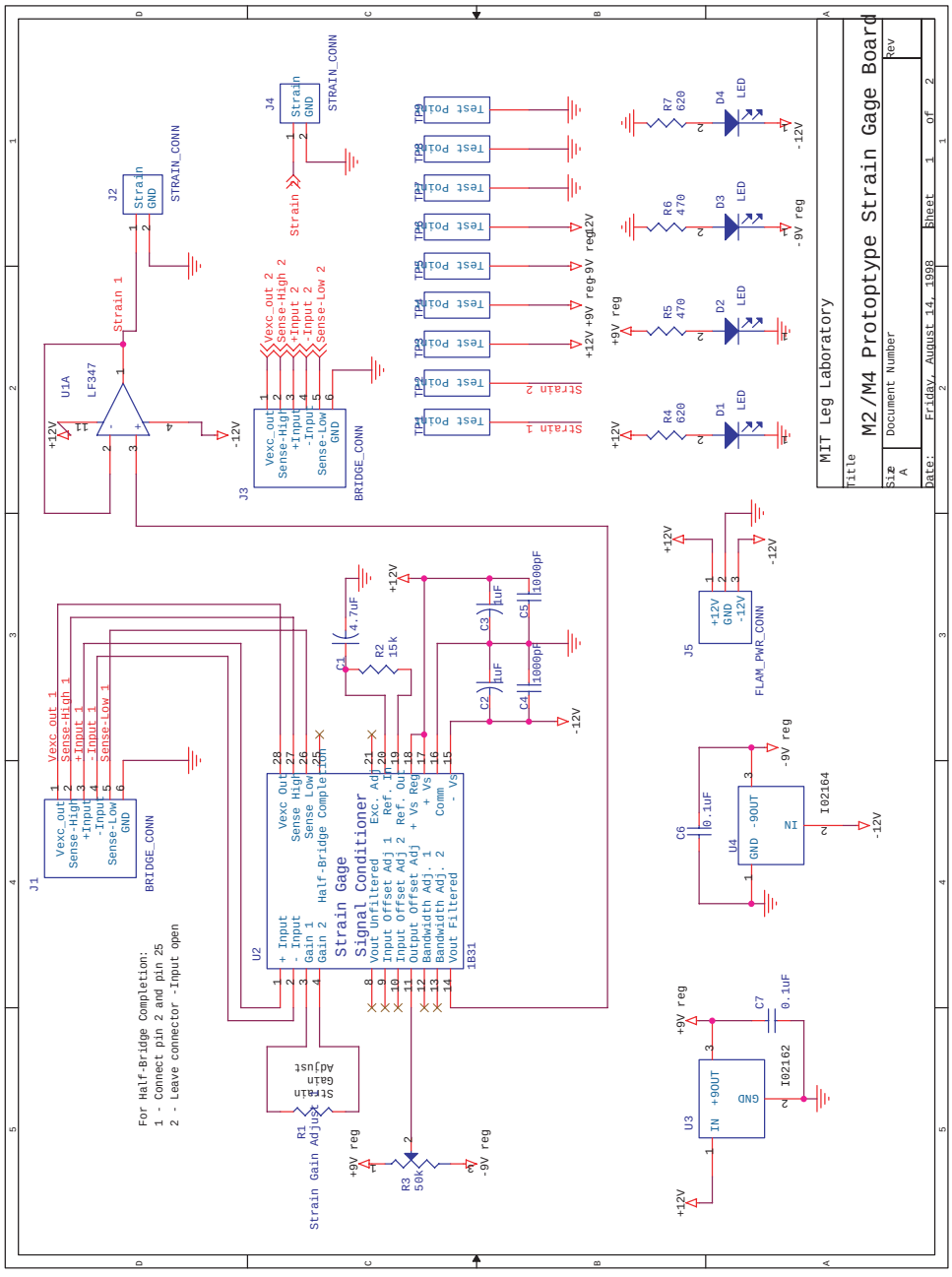


Figure A-27: Strain Gage Conditioning Board Schematic Page 1 of 2.

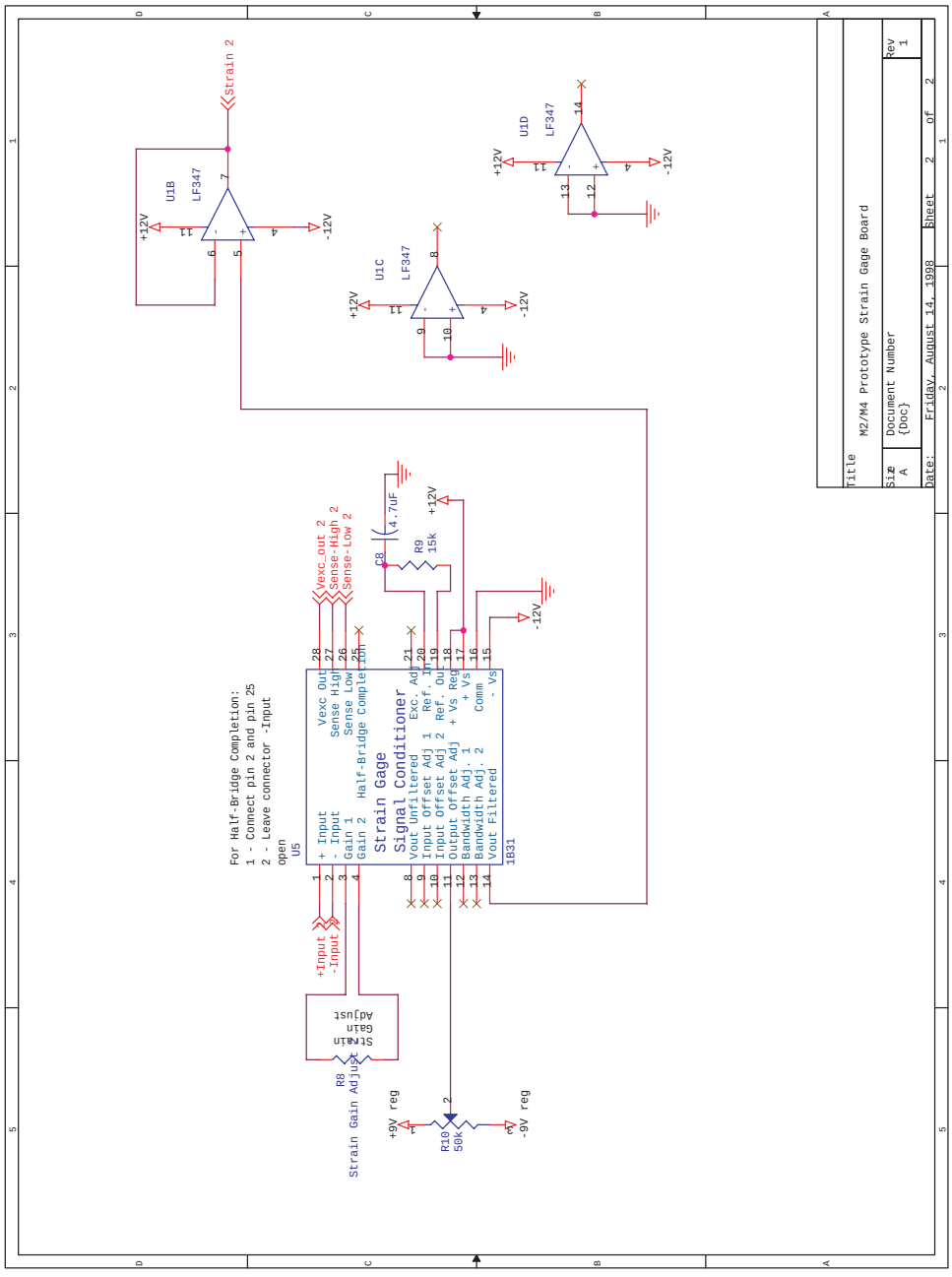


Figure A-28: Strain Gage Conditioning Board Schematic Page 2 of 2.

Table A.3: Parts List of off-the-shelf mechanical and major electrical parts. Only listed are parts required for one robot. Significant spares were also purchased. This list does not include machined parts.

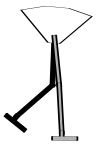
Quantity	Description	Supplier	Price
1	TI C30 DSP System	Digital Designs	3200.00
4	1.0in ID, .06in T, 12in L CFRP Tube	Revolution Enterprises	45.00
12	0.375in OD, 3/16in ID, 14in L CFRP Tube	Revolution Enterprises	140.00
1	2.0in ID, .08in T, 120in L CFRP Tube	Quality Composites	550.00
1	2.0in ID, .08in T, 27in L CFRP Tube	Quality Composites	122.00
6	FAI-035 24V Motors, Specially Wound	Astroflight	600.00
6	85206 Ball Screw Assembly	Motion Systems	400.00
24	D-1123-A Die Compression Springs	Century Spring	120.00
7	Bourns 6637S-1-502 Potentiometer	Newark 12F7102	238.00
1 Case	Dexter-Hysol EA9430 Eng. Adhesive	Industrial Sales	97.50
1 Case	Dexter-Hysol 1C Epoxy Patch	Industrial Sales	100.00
6	Model 303 12A Peak Brushed Motor Amp	Copley Controls	1650.00
250 ft.	2054SN Flexible Cable	Sava Industries	512.50
100	End Crimps	Sava Industries	20.00
8	THA-100-Q Thru Hole Load Cells	Transducer Techniques	3600.00
6	A 6R23M041060 Timing Belt 41 Grooves	Stock Drive Products	25.00
6	A 6A23-025NF0608 Pulley 3mm, 25 grooves	Stock Drive Products	60.00
6	A 6A23-010DF0604 Pulley 3mm, 10 grooves	Stock Drive Products	54.00
60	A 7Y55-FSS5025 Flanged Abec3 Bear. .25 Bore .50 D	Stock Drive Products	310.00
4	A 7Y55-PSS16275 Bear .75d 1.625D .4375W	Stock Drive Products	134.32
4	A 7Y55-FSS8737 Bear .375d .875D .2812W	Stock Drive Products	48.48
20	S435 .25D 3.5L Stainless Ground Shaft	Berg	36.20
250	SS2-32-A Shaft Spacer .25d .375D .063W	Berg	350.00
5	S4-60 .25D 6.0L Stainless Ground Shaft	Berg	13.05
2	PZ-34 Shoulder Screw .3745d .5D .3130L	Berg	9.84
2	PZ-35 Shoulder Screw .3745d .5D .3755L	Berg	10.08
6	PTN-050 Linear Sensor 2.3in. Travel	Novotechnik	120.00
6	S-170 Sensor Slider	Novotechnik	33.00
50	A-SBP-6/6 Rulon-J Bear 3/8ID 9/16OD 3/8L	Small Parts	211.00
30	A-SBP-6/12 Rulon-J Bear 3/8ID 9/16OD 3/4L	Small Parts	127.00
6	A-HNTT-1420 Titanium Nut	Small Parts	8.00
12	SR4A Ang. Cont. Bearing 1/4id 3/4od	Champion Industries	240.00
250ft	AS879 4 Cond. 28 G. Sh. Wire	Cooner Wire	370.00
150ft	NMEF 4/26-6544SJ 4Cond. 26 G. Sh. Wire	Cooner Wire	270.00
10	ASTM4005 Zero Loss Connector Pairs	Tower Hobbies	85.90
500ft	Belden 8760 18 Gauge Pair with Shield	Pacer Electronics	73.00
250ft	9445 Belden 5C. 20G. Unsh. Stranded Wire	Pacer Electronics	45.00
15	FC01 Circuit Boards. 3.20 X 7.75	CFC	640.00
4	SF-BK01 Circuit Boards. 5.575 X 3.875	CFC	1100.00
4	SF-PNA01 Circuit Boards. 4.65 X 3.3	CFC	407.00
10	91590A128 .75 Ext. Ret Ring .704D .046W	McMaster Carr	9.90
10	91580A238 1.625 Int Ring 1.725D .068W	McMaster Carr	27.70

Table A.4: Suppliers, Page 1 of 2.

Name	Description	Address	Web Page
Digital Designs and Systems	Microcontrollers, A/D and D/A conversion, Vision Systems, Ready Built and Custom Designs	3266 North Meridian Street Indianapolis, IN 46208 (317)931-8190	www.dideas.com
Astroflight	Electric Motors	13311 Beach Ave. Marina Del Ray, CA 90292 (310)821-6242	www.astroflight.com
Motion Systems Corporation	Ball Screws	600 Industrial Way West Eatontown, NJ 07724 (908) 222-1800	
Stock Drive Products	Mechanical Parts	2101 Jericho Turnpike, Box 5416 New Hyde Park, NY 11042 (516)328-3300	
Berg	Mechanical Parts	499 Ocean Avenue Box 130 East Rockaway, NY 11518 (516)596-1700	www.wmberg.com
Novotechnik	Linear Potentiometers	237 Cedar Hill Street Marlboro, MA 01752 (508)485-2244	www.novotechnik.com
Small Parts, Inc.	Mechanical Parts	13980 NW 58th Court PO Box 4650 Miami Lakes, FL 33014 (800)220-4242	www.smallparts.com
Century Spring Company	Springs	Box 15287 222 East 16th Street Los Angeles, CA 90015 (800)237-5225	www.centuryspring.com
Newark Electronics	Electronic Parts	41 Pleasant Street Methuen, MA 01844 (800)463-9275	
Industrial Sales Inc.	Distributor: Epoxy, etc.	24 Grata Road Newburyport, MA 01950 (508)851-9494	
REC Engineering	Machining	20 Hopkins Street Wilmington, MA 01887 (508)657-6517	
Digi-Key Corporation	Electronic Parts	701 Brooks Ave. South Thief River Falls, MN 56701 (800)344-4539	www.digikey.com

Table A.5: Suppliers, Page 2 of 2.

Name	Description	Address	Web Page
Copley Controls Corporation	Motor Amplifiers	410 University Avenue Westwood, MA 02090 (617)329-8200	www.copleycontrols.com
Cooner Wire	Flexible Wire	9265 Owensmouth Chatsworth, CA 91311 (818)882-8311	www.coonerwire.com
Tower Hobbies	Model Car and Airplane Hobby Store	PO Box 9078 Champaign, IL 61826 (800)637-6050	www.towerhobbies.com
Pacer Electronics	Electronic Parts, Wire	112 Commerce Way Woburn, MA 01801 (617)935-8330	
CFC, Inc.	Circuit Board Fabrication	179 Bear Hill Road Waltham, MA 02154-1001 (617)890-1878	
McMaster Carr	Mechanical Parts, Supplies	Box 440 New Brunswick, NJ 08903 (908)329-3200	www.mcmaster.com
Quality Composites	Carbon Fiber	8385 S Allen Street #140 Sandy, UT 84070 (801)565-8003	
Sava Industries	Steel Cable	4 N. Corporate Drive, P.O. Box 30 Riverdale, NJ 07457-0030 (201)835-0882	
Revolution Enterprises, Inc.	Carbon Fiber	6335 Nancy Ridge Drive San Diego, CA 92121 (619)554-1106	
Champion Industries, Inc.	Bearings	2233 Milo Dr. Palm Springs, CA 92262 (619)320-0411	
Industrial Aluminum	Stock Aluminum	341 Second Ave. Waltham, MA 02154 (617)890-2410	
Transducer Techniques	Load Cells	43178 Business Park Drive Temecula, CA 92590 (909)676-3965	www.ttloadcells.com



Appendix B

Adaptive Control of Swing Leg

In this appendix, we derive the equations of motion of the swing leg, assuming that it is decoupled from the rest of the robot, and use adaptive control to identify the dynamic parameters. The equations of motion are used in Chapter 7 to control the hip during swing in order to achieve fast walking.

B.1 Equations of Motion

Figure B-1 shows a model of the swing leg. We assume that the rest of the robot is not accelerating and hence plays no role in the dynamics. L_1 and L_2 are the lengths of the upper and lower legs. b_1 and b_2 are the distances to the links center of mass. m_1, m_2, I_1, and I_2 are the mass and inertia parameters. θ_1 is the angle of the upper link from vertical, while θ_2 is the angle of the lower link with respect to the upper link.

The equations of motion can be computed using Lagrange's method. The kinetic co-energy, T^* is

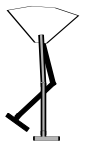
$$
\begin{aligned}
T^* &= T_1^* + T_2^* && \text{(B.1)}\\
T_1^* &= \frac{1}{2} m_1 v_1^2 + \frac{1}{2} I_1 w_1^2 && \text{(B.2)}\\
T_2^* &= \frac{1}{2} m_2 v_2^2 + \frac{1}{2} I_2 w_2^2 && \text{(B.3)}
\end{aligned}
$$

where v_1 and v_2 are the velocities of the center of mass of the upper link (thigh) and lower link (shin). w_1 and w_2 are the angular velocity of the links, all with respect to the fixed world reference frame. v_2^2 is computed using the law of cosines.

$$
\begin{aligned}
v_1^2 &= (b_1\dot{\theta_1})^2 && \text{(B.4)}\\
v_2^2 &= (L_1\dot{\theta_1})^2 + (b_2(\dot{\theta_1}+\dot{\theta_2}))^2 + 2(L_1\dot{\theta_1})(b_2(\dot{\theta_1}+\dot{\theta_2}))\cos(\theta_2) && \text{(B.5)}\\
w_1^2 &= \dot{\theta_1}^2 && \text{(B.6)}\\
w_2^2 &= (\dot{\theta_1}+\dot{\theta_2})^2 && \text{(B.7)}
\end{aligned}
$$

The potential energy, V is

$$
\begin{aligned}
V &= V_1 + V_2 && \text{(B.8)}\\
V_1 &= -m_1 g b_1 \cos(\theta_1) && \text{(B.9)}\\
V_2 &= -m_2 g (L_1 \cos(\theta_1) + b_2 \cos(\theta_1+\theta_2)) && \text{(B.10)}
\end{aligned}
$$



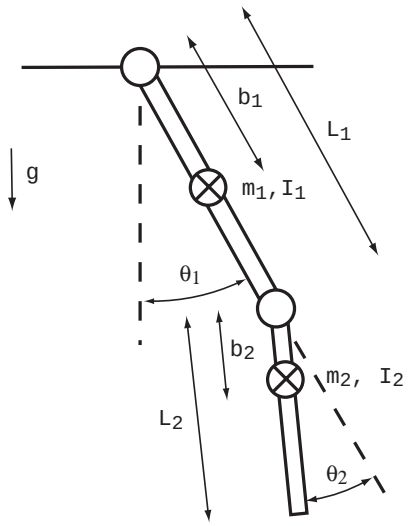


Figure B-1: Dynamic model of swing leg.

We can simplify the equations by substituting the following inertial groups:

$$J_0 = m_1 b_1^2 + I_1 \tag{B.11}$$
$$J_1 = m_2 L_1 b_2 \tag{B.12}$$
$$J_2 = m_2 L_1^2 + m_2 b_2^2 + I_2 \tag{B.13}$$
$$J_3 = m_2 b_2^2 + I_2 \tag{B.14}$$
$$G_1 = (m_1 b_1 + m_2 L_1) g \tag{B.15}$$
$$G_2 = m_2 b_2 g \tag{B.16}$$

J_0 is the inertia of the thigh about the θ_1 pivot. J_3 is the inertia of the shin about θ_2. J_1 and J_2 are the terms associated with the inertia of the shin about the θ_1 pivot. G_1 is associated with the gravitational torques of both links about θ_1 while G_2 is associated with the gravitational torques of the shin about θ_2.

The kinetic co-energy and the gravitational potential energy can now be written as

$$T^* = (\frac{1}{2}J_0 + \frac{1}{2}J_2 + J_1\cos(\theta_2))\dot{\theta_1}^2 + \frac{1}{2}J_3\dot{\theta_2}^2 + (J_3 + J_1\cos(\theta_2))\dot{\theta_1}\dot{\theta_2} \tag{B.17}$$
$$V = -G_1\cos(\theta_1) - G_2\cos(\theta_1 + \theta_2) \tag{B.18}$$

The Lagrangian is

$$\mathcal{L} = T^* - V \tag{B.19}$$

and Lagranges equations are

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{\theta_1}}\right] - \frac{\partial \mathcal{L}}{\partial \theta_1} = \tau_1 \tag{B.20}$$
$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{\theta_2}}\right] - \frac{\partial \mathcal{L}}{\partial \theta_2} = \tau_2 \tag{B.21}$$

Using Lagrange's method results in the equations of motion:

$$\begin{bmatrix} (J_0 + J_2 + 2J_1\cos(\theta_2)) & (J_3 + J_1\cos(\theta_2)) \\ (J_3 + J_1\cos(\theta_2)) & (J_3) \end{bmatrix} \begin{bmatrix} \ddot{\theta_1} \\ \ddot{\theta_2} \end{bmatrix} +$$

$$\begin{bmatrix} -2J_1\sin(\theta_2)\dot{\theta_1}\dot{\theta_2} - J_1\sin(\theta_2)\dot{\theta_2}^2 \\ J_1\sin(\theta_2)\dot{\theta_1}^2 \end{bmatrix} + \begin{bmatrix} G_1\sin(\theta_1) + G_2\sin(\theta_1+\theta_2) \\ G_2\sin(\theta_1+\theta_2) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

On the real robot, we can measure the angle of the body (θ_b), the hip (θ_h), and the knee (θ_k) joints. Also we wish to command the angles of the links with respect to world coordinates. Therefore we define q_1 and q_2 to be the angles of the thigh and shin with respect to vertical:

$$q_1 = \theta_1 = \theta_b + \theta_h \tag{B.22}$$
$$q_2 = \theta_1 + \theta_2 = \theta_b + \theta_h + \theta_k \tag{B.23}$$
$$\tag{B.24}$$

and rewrite the equations of motion in terms of q_1 and q_2:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \vec{g}(q) = \tau \tag{B.25}$$

where H is the mass matrix, C is the matrix of centrifugal and Coriolis terms, and $\vec{g}$ is the gravitational vector:

$$H = \begin{bmatrix} a_1 + a_2\cos(\theta_k) & a_3 + a_2\cos(\theta_k) \\ a_2\cos(\theta_k) & a_3 \end{bmatrix} \tag{B.26}$$
$$C = \begin{bmatrix} a_2\sin(\theta_k)\dot{q_1} & -a_2\sin(\theta_k)\dot{q_2} \\ a_2\sin(\theta_k)\dot{q_1} & 0 \end{bmatrix} \tag{B.27}$$
$$\vec{g} = \begin{bmatrix} a_4\sin(q_1) + a_5\sin(q_2) \\ a_5\sin(q_2) \end{bmatrix} \tag{B.28}$$

where the following dynamic parameters have been substituted:

$$a_1 = J_0 + J_2 - J_3 = m_1 b_1^2 + I_1 + m_2 L_1^2 \tag{B.29}$$
$$a_2 = J_1 = m_2 L_1 b_2 \tag{B.30}$$
$$a_3 = J_3 = m_2 b_2^2 + I_2 \tag{B.31}$$
$$a_4 = G_1 = (m_1 b_1 + m_2 L_1) g \tag{B.32}$$
$$a_5 = G_2 = m_2 b_2 g \tag{B.33}$$

B.2 Adaptive Control

Adaptive control can now be used to identify the parameters a_1 through a_5. We use the approach in [Slotine & Li (1991)]. We will track a desired trajectory, q_d, using the control law

$$\tau = \hat{\tau} - K_d s \tag{B.34}$$

where $\hat{\tau}$ is an inverse dynamic term tracking a reference trajectory, q_r, consisting of both a feedforward desired trajectory, q_d and the integral of a feedback error, $\tilde{q}$. $K_d s$ is a purely feedback error correction term and is equivalent to a PD controller:

$$\hat{\tau} = Y\hat{a} = \hat{H}\ddot{q}_r + \hat{C}\dot{q}_r + \hat{g} \tag{B.35}$$
$$\dot{q}_r = \dot{q}_d - \Lambda\tilde{q} \tag{B.36}$$
$$s = \dot{\tilde{q}} + \Lambda\tilde{q} = \dot{q} - \dot{q}_r \tag{B.37}$$

where $\tilde{q} = q - q_d$. For the swing leg case, Y is:

$$Y = \begin{bmatrix} \ddot{q}_{1r} & (\ddot{q}_1\cos(\theta_k) + \ddot{q}_2\cos(\theta_k) + \dot{q}_1\dot{q}_{1r}\sin(\theta_k) - \dot{q}_2\dot{q}_{2r}\sin(\theta_k)) & \ddot{q}_{2r} & \sin(q_1) & \sin(q_2) \\ 0 & (\ddot{q}_{1r}\cos(\theta_k) + \dot{q}_1\dot{q}_{1r}\sin(\theta_k) & \ddot{q}_{2r} & 0 & sin(q_2) \end{bmatrix} \tag{B.38}$$

The parameters are adapted according to how much they contribute to the current dynamics and how much they can correct for the current error:

$$\dot{\hat{a}} = -\Gamma Y^T s \tag{B.39}$$
$$\tag{B.40}$$

where Γ is a matrix of adaptation weights. To prove convergence of both the error and the dynamic parameters, we use the following Lyapunov function

$$\mathcal{V}(t) = \frac{1}{2}\left[s^T H s + \tilde{a}^T\Gamma^{-1}\tilde{a}\right] \tag{B.41}$$

where $\tilde{a} = \hat{a} - a$. The following simplifications

$$\dot{\mathcal{V}} = s^T H\dot{s} + \frac{1}{2}s^T\dot{H}s + \dot{\tilde{a}}^T\Gamma^{-1}\tilde{a} \tag{B.42}$$
$$\dot{s} = \ddot{q} - \ddot{q}_r \tag{B.43}$$
$$\dot{\mathcal{V}} = s^T H\ddot{q} - s^T H\ddot{q}_r + \frac{1}{2}s^T\dot{H}s + \dot{\tilde{a}}^T\Gamma^{-1}\tilde{a} \tag{B.44}$$
$$H\ddot{q} = \tau - C\dot{q} - g \tag{B.45}$$
$$\dot{\mathcal{V}} = s^T(\tau - H\ddot{q}_r - C\dot{q}_r - g) + \dot{\tilde{a}}^T\Gamma^{-1}\tilde{a} \tag{B.46}$$
$$\dot{\mathcal{V}} = s^T Y\tilde{a} - s^T K_D s + \dot{\hat{a}}\Gamma^{-1}\tilde{a} \tag{B.47}$$
$$\dot{\hat{a}} = -\Gamma Y^T s \tag{B.48}$$
$$\tag{B.49}$$

result in

$$\dot{\mathcal{V}} = -s^T K_D s \leq 0 \tag{B.50}$$

Therefore, by Barbalat's Lemma [Slotine & Li (1991)], if the dynamic model is accurate, tracking will be achieved and the parameters will converge. However, due to unmodeled dynamics, there will be tracking errors. Also, since we are interested in the actual dynamic parameters, it will be important to have sufficient excitation of the dynamics in the desired trajectory.

B.3 Results

Using the above adaptive control law, we found parameter estimates for the swing leg of the robot. To sufficiently excite the dynamics we used a desired trajectory consisting of a large amplitude, low frequency, sine wave and a low amplitude, high frequency, sine wave:

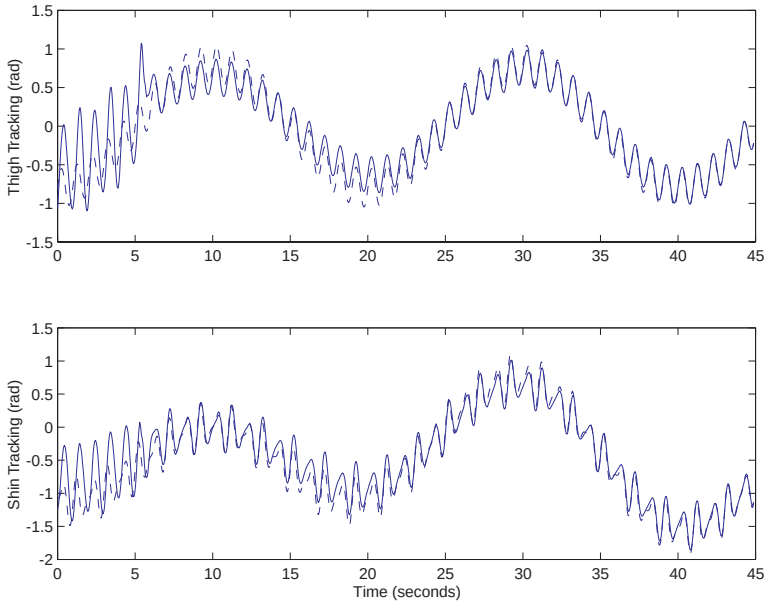


Figure B-2: Swing leg tracking using adaptive control. Adaptive control is turned on at 5 seconds.

$$q_{1d} = 0.8\sin(2\pi(0.05)t) + 0.25\sin(2\pi(1.0)t) \tag{B.51}$$
$$q_{2d} = -0.4 + q_{1d} + 0.4\sin(2\pi(0.03)t) + 0.15\sin(2\pi(1.5)t) \tag{B.52}$$
$$\tag{B.53}$$

The control gains and adaptation gains used are the following:

$$K_{D1} = 1.0 \tag{B.54}$$
$$K_{D2} = 0.7 \tag{B.55}$$
$$\lambda_1 = 16.0 \tag{B.56}$$
$$\lambda_2 = 10.0 \tag{B.57}$$
$$\Gamma_{11} = 0.005 \tag{B.58}$$
$$\Gamma_{22} = 0.001 \tag{B.59}$$
$$\Gamma_{33} = 0.001 \tag{B.60}$$
$$\Gamma_{44} = 0.30 \tag{B.61}$$
$$\Gamma_{55} = 0.08 \tag{B.62}$$

Figure B-2 shows the tracking results and Figure B-3 show the tracking errors. Adaptive control is turned on at $t = 5$. We see that tracking is fairly poor when the dynamic parameters are all at zero. When adaptation is turned on, the tracking becomes significantly better within a few seconds. By 45 seconds, good tracking is achieved.

Figure B-4 shows adaptation of the first 3 parameters which represent the dynamic inertial terms while Figure B-5 show adaptation of the gravitational terms. Within 45 seconds the parameters converge, resulting in the following values

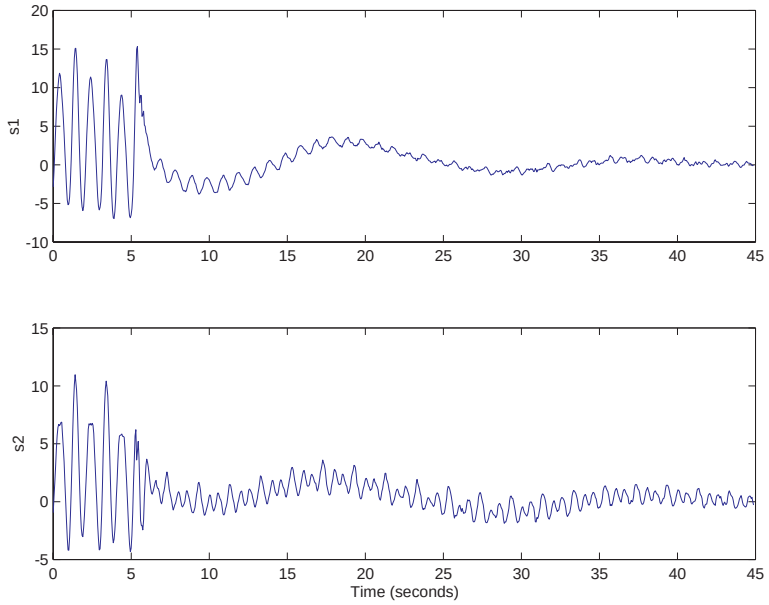


Figure B-3: Tracking errors for adaptive control of swing leg. Adaptive control is turned on at 5 seconds.

$$a1 = 0.21 \tag{B.63}$$
$$a2 = 0.11 \tag{B.64}$$
$$a3 = 0.12 \tag{B.65}$$
$$a4 = 4.5 \tag{B.66}$$
$$a5 = 2.8 \tag{B.67}$$
$$\tag{B.68}$$

This experiment was repeated on the other leg, with similar results. All the parameters were within 10% of those listed above. Therefore we use these parameters for both legs in the fast walking experiments of Chapter 7.

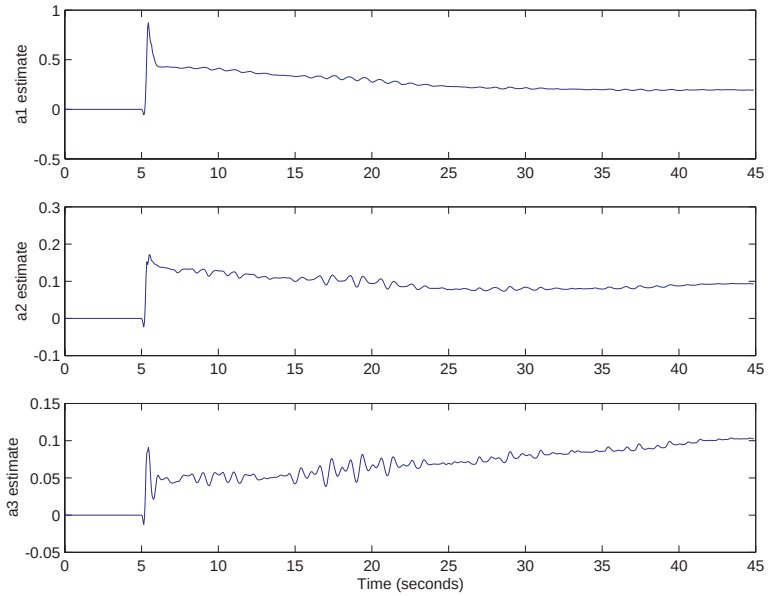


Figure B-4: Dynamic parameter estimates for adaptive control of swing leg. Adaptive control is turned on at 5 seconds.

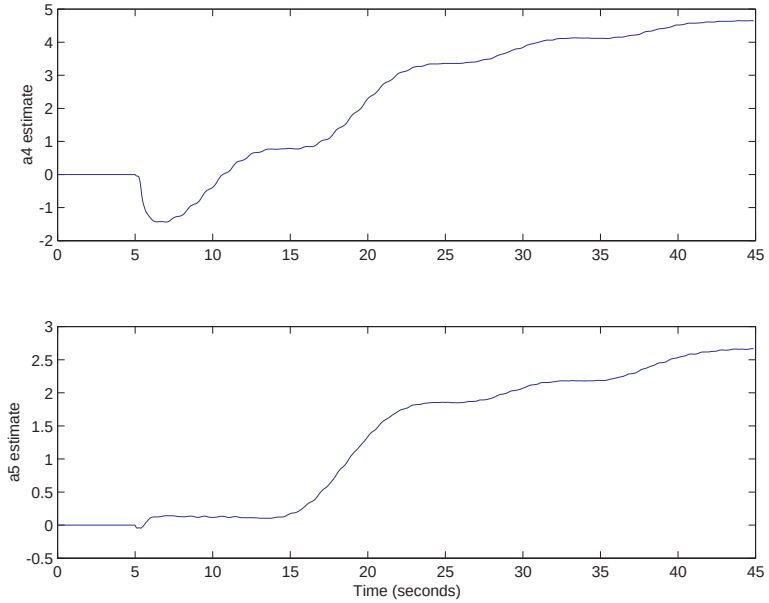


Figure B-5: Gravitational parameter estimates for adaptive control of swing leg. Adaptive control is turned on at 5 seconds.