EFFICIENCY AND SPEED IN LEGGED ROBOTICS

DISSERTATION FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY IN
SYSTEMS ENGINEERING

PAUL L. MUENCH

OAKLAND UNIVERSITY

2011

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
22 MAR 2011	**N/A**	**-**

4. TITLE AND SUBTITLE	
Efficiency and Speed in Legged Robots	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Paul L. Muench**	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **US Army RDECOM-TARDEC 6501 E 11 Mile Rd Warren, MI 48397-5000, USA**	8. PERFORMING ORGANIZATION REPORT NUMBER **21615**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **US Army RDECOM-TARDEC 6501 E 11 Mile Rd Warren, MI 48397-5000, USA**	10. SPONSOR/MONITOR'S ACRONYM(S) **TACOM/TARDEC/RDECOM**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) **21615**

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
Dissertation for the Degree of Doctor of Philosophy in Systems Engineering

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**114**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

EFFICIENCY AND SPEED IN LEGGED ROBOTICS

by

PAUL L. MUENCH

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN SYSTEMS ENGINEERING

2011

Oakland University
Rochester, Michigan

Doctoral Advisory Committee:

Ka Chai Cheok, Ph.D., Chair
Edward Y.L. Gu, Ph.D.
Meir Shillor, Ph.D.
Krzystof Kobus, Ph.D.

The LORD is my rock, and my fortress, and my deliverer;

My God, my strength, in whom I will trust;

Psalm 18:2

ACKNOWLEDGMENTS

I would like to thank Dr. Grant Gerhart, for funding this work through the In-Lab Innovative Research (ILIR) Program at TARDEC. Dr. Marc Raibert, Dr. Jerry Pratt, and Dr. Art Kuo: for making this a subject worth studying. Professor Ka C Cheok, thank you for your patience with my "non-efficiency and speed" in completing this dissertation. I would like to thank Dr. Rob Karlsen, Dr. Mark Brudnak, and Professor Meir Shillor, for their insights into the problem during discussions with them; Adam Tucker, Joseph Alexander, Gregory Czerniak, Dr. Dave Bednarz and Sean Hadley, my co-authors on related conference publications. My committee members Professor C. J. Kobus and Professor Edward Gu: for pushing me beyond where I wanted to go. I would especially like to thank Alex Marecki, Prince Idichandy, Leah Wojciewski, Jamie MacLennan, and Dr. James English for the tremendous help they were as these ideas were taking shape. I would like to thank the late Dr. Emmett Leith for teaching me how research gets done—by having fun. I would like to thank my parents, Walter and Denise Muench, for all their love and support. Finally, and not least, I would like to thank my wife, Colleen, for putting up with me and my PhD over the years—*we are like two gears*.

Paul L. Muench

ABSTRACT

EFFICIENCY AND SPEED IN LEGGED ROBOTICS

by

Paul L. Muench

Advisor: Ka C Cheok, Ph.D.

It may seem preposterous, but walking is actually faster than rolling under certain conditions. Even though bio-inspired robots can have advantageous mobility, they can also be very inefficient. This idea, the balance between efficiency and speed, is the core of this dissertation. Inefficiencies in legged robotics are due to many things, e.g., collision losses, joint friction, etc., but here we address the inefficiencies due to suboptimal control schemes.

We seek a control scheme that will decide: 1) whether it makes sense to walk or not—we seek an energy regime for walking that depends on the energy already in the system, and then 2) how wide we should make our stride, i.e., the optimal stride angle, and finally 3) an optimal switching curve: when should we power, and when should we coast to achieve the greatest possible speed at the least energy cost.

This dissertation presents control schemes for a lossless "rimless wheel" model as well as a two-link serial robot model for walking. Using Pontryagin's Maximum Principle (PMP), we describe the cost function, the state/co-state equations, and the switching conditions for these models. The research results of both models show an

“on-off” control and the “switching curve” between these control extremes. It is not possible to find a complete closed-form solution for some parts of this problem, and numerical methods, such as dynamic programming and Nelder Mead optimization, must be used for simulation and visualization of the results.

TABLE OF CONTENTS

ACKNOWLEDGMENTS iv

ABSTRACT v

LIST OF FIGURES ix

CHAPTER 1
INTRODUCTION 1

1.1. Transportation Problem 1

1.2. Brachistochrone Problem 2

1.3. The Rimless Wheel 3

CHAPTER 2
RESEARCH BACKGROUND 5

2.1. Passive Dynamics 5

2.2. Active Dynamics 5

2.3. Overview of the Research 6

CHAPTER 3
PASSIVE RIMLESS WHEEL 9

3.1. The Simple Pendulum 9

3.2. Optimal Stride Angle 12

3.3. Elliptic Integrals 15

3.4. Cleveland Problem 17

3.5. Position Average 17

3.6. Energy Regime of Walking ($1 < E < 1.5$) 24

TABLE OF CONTENTS—Continued

CHAPTER 4
POWERED RIMLESS WHEEL 27

4.1. Powered Pendulum 27

4.2. Principle of Optimality 29

4.3. Dynamic Programming 29

4.4 Pontryagin's Maximum Principle 34

4.5. Rocket Car 38

4.6. Powered Pendulum 47

CHAPTER 5
A TWO LINK MODEL OF WALKING 56

5.1. Double Pendulum 56

5.2. Trajectory Optimization using PMP 62

5.3. Boundary Conditions for PMP 68

5.4. Nelder Mead Algorithm 69

5.5. Nelder Mead Optimization and the Double Pendulum 72

CHAPTER 6
CONCLUSION 77

APPENDICES 79

A. Dynamic Programming Code 79

B. Nelder Mead Optimization of the Rimless Wheel 83

C. State and Co-State Equations for the Double Pendulum 88

D. Nelder Mead Optimization of the Double Pendulum 96

REFERENCES 101

LIST OF FIGURES

Figure 1.1	The Transportation Problem	1
Figure 1.2	The Brachistochrone Problem	3
Figure 1.3	The Rimless Wheel	4
Figure 3.1	The Simple Pendulum	9
Figure 3.2	The Phase Plane of the Pendulum	11
Figure 3.3	One Stride Length	12
Figure 3.4	Plot of Fastest Stride Angles, a	14
Figure 3.5	Elliptic Integrals	16
Figure 3.6	Pendulum Walker	18
Figure 3.7	Instantaneous forward speed, $v(\Theta)$, plotted in blue with cosine plotted in green for energy levels	20
Figure 3.8	Optimum Stepwidth	26
Figure 4.1	Phase Plane of the Simple Pendulum Discretized into a Directed Network	28
Figure 4.2	Node Numbering Method	28
Figure 4.3	The Principle of Optimality	30
Figure 4.4	The Minimum Path	31
Figure 4.5	Dynamic Programming for the Powered Rimless Wheel	34
Figure 4.6	The PMP Research Outline	39
Figure 4.7	The Positioning Problem	40
Figure 4.8	The Switching Curve	45

Figure 4.9	Switching Curve for the Rocket Car Problem	48
Figure 4.10	Optimal "on-off" Control for the Powered Pendulum	48
Figure 4.11	The Switching Curve for the Powered Pendulum	52
Figure 4.12	Switching Curve, Equation (4.60) is Shown as a Solid Line, Compared with Dynamic Programming Results (shown in circles)	55
Figure 5.1	The DH Double Pendulum	58
Figure 5.2	Geometric Relations	60
Figure 5.3	Trajectory Optimization Problem	63
Figure 5.4	State Trajectory from Initial to Final State	64
Figure 5.5	Nelder-Mead uses function evaluation over the vertices of an *N*-dimensional simplex to optimize a criterion function over an *N*-dimensional space. Shown here is a tetrahedron, which is a simplex for *N*=3. There are *N*+1 vertices in an *N*-dimensional simplex.	70
Figure 5.6	Fourier Series Half Range Expansion	73
Figure 5.7	Nelder Mead for the Powered Pendulum	74
Figure 5.8	Optimal Control Torque u2 (shown in red and the switching conditions s1 and s2 shown in yellow and green)	76

CHAPTER 1

INTRODUCTION

1.1 Transportation Problem

It may seem preposterous, but walking is actually faster than rolling. Normally we think of rolling as the fastest and most practical means of transportation while legged systems have held a somewhat elusive promise of mobility (Raibert, 1986). Until recently, most research in legged robotics has focused on stability. Recent research (Ruina, 1998) has come to the conclusion that stability is a necessary, but not sufficient condition; efficiency is the key to unlocking the true benefits of walking mechanisms. To introduce the problem of transportation, we adopt this useful model shown in Figure 1.1. Given an initial energy E_0, move a mass m from point A to point B in time t expending additional energy E.

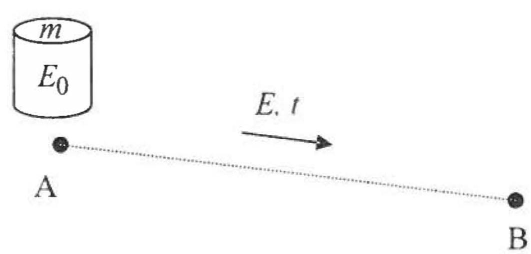


Figure 1.1: The Transportation Problem.

The primary goal of transportation is to "move the mass *m* from point A to B." The most efficient thing would be to leave the mass at A and be done with it! Admittedly, that is not the most useful solution. So if you've resigned yourself to actually move the mass, now what? You are left to balance the cost of how much energy *E* you will expend vs. how much time *t* it will take to do the job. This tradeoff will be the subject of our study as we explore the transportation problem—the optimal control—of two distinct models of walking: 1) the rimless wheel—essentially a simple pendulum; and 2) the compass gait—essentially a double pendulum.

1.1 Brachistochrone Problem

Let's begin the discussion with an example where energy is conserved. If the energy is constant (or conserved), the goal becomes minimizing the travel time *t*—finding the least time path, or brachistochrone. Consider the following problem: We wish to construct a ramp such that our mass *m* starts at rest rolls (or slides without friction) from point A to point B in the least amount of time. The temptation is to draw a straight line from point A to point B, since obviously the shortest geometric distance between two points is a straight line. The only problem with this temptation, of course, is that it's wrong.[1] Looking again at Figure 1.1, imagine that our initial energy E_0 came in two forms: kinetic and potential. If we first start from rest at point A, all of our initial energy E_0 is stored as potential energy; the kinetic bucket is empty. Recall that for conservative systems, we can convert energy from one form to another without penalty; so in order to get things moving, we'd like to convert some of that stored

[1] The solution to the brachistochrone problem is a cycloid. This is a famous problem in mathematics, for the analysis of it led to the formulation of the calculus of variations (Goldstein, 1959).

energy into kinetic energy. Not only that, but we'd like to do it as soon as possible.[2] The fastest way to dump potential energy is to free-fall from position A as shown in Figure 1.2. As we pick up speed, we track our course back towards B. Interesting that the fastest way towards our goal is to head away from it. We will see this idea again as we examine the rimless wheel.

1.3 The Rimless Wheel

Consider the following model for walking, shown in Figure 1.3, originally proposed by Tad McGeer (McGeer, 1990). It consists of a mass *m* with rigid spokes extending outwards, like a wagon wheel without the outer rim. The rimless wheel pivots and collides with the ground on rigid spokes instead of rolling, behaving as an inverted pendulum between intermittent plastic collisions with the ground. From

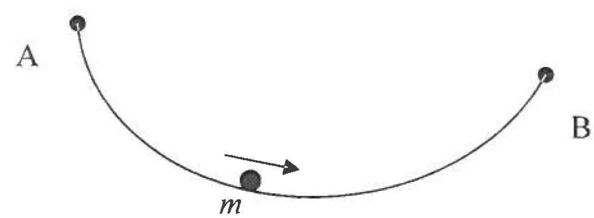


Figure 1.2: The Brachistochrone Problem.

[2] Pick up speed sooner rather than later, so that the time savings start accruing right away. Your financial advisor probably has similar advice.

position A, we pick up speed as in the brachistochrone problem by trading in potential energy for more kinetic energy. If the rim were present, you would be stuck with the kinetic energy you began with, however slow that might be. You would never lose any speed, but you could not gain any either.

What we gain in increased average speed with the rimless wheel model we will pay for by deviating from the straight line path. As we shall soon discover, this tradeoff will have a point of diminishing returns.

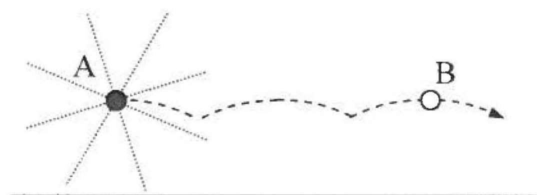


Figure 1.3: The Rimless Wheel

CHAPTER 2

RESEARCH BACKGROUND

2.1 Passive Dynamics

Much of the recent literature on walking mechanisms has focused on passive dynamic walking, or the ability of these mechanisms to walk based only on physics, not on actuation and control. This concept finds its origins in walking toys from the 19th century (Fallis, 1888). These toys wobble back and forth as they periodically swing their legs and "walk" down a shallow slope. They were the inspiration behind Tad McGeer's seminal work (McGeer, 1990), the analysis of a class of walking machines, wherein gravity replaces the energy lost from friction and collision, thus creating a stable walking gait without any external control.

Andy Ruina and his group at Cornell (Garcia, Chatterjee, Ruina, & Coleman, 1998) have simplified these passive dynamic models by introducing the compass gait. Since that time, Art Kuo and others in the community have found biological data suggesting that these methods can extend to human walking gaits as well (Kuo, Donelan, & Ruina, 2005).

2.2 Active Dynamics

While this notion of walking as the passive unfolding of system dynamics is elegant and intriguing, a real world application of these ideas requires actuation and control. Beginning with his hopping robots at the MIT Leg Lab (Raibert, 1986), Marc Raibert has advanced the state of the art in legged robot control for practical application. His most recent creation, the BigDog (Playter, Buehler, & Raibert, 2006), is the most advanced legged robot in the world for outdoor applications. BigDog, based on high-gain, high-bandwidth control, does not incorporate the aforementioned principles of passive dynamics. When actuators are directly coupled to the legs, this is difficult to achieve. Applying a clutching mechanism solves this problem by having both active and passive modalities. Russ Tedrake used optimal control and learning algorithms to his semi-passive walker (Tedrake, 2004). The robot quickly learned how to deal with the uncertainty of its under-actuated control.

Adding a virtual clutch is another way to achieve both active and passive actuation. The series elastic actuator of Gill Pratt (Pratt & Williamson, 1997) acts as a force source during actuation and applies zero force during the passive swing phase. This series elastic actuator has been the central component of several robots built by Jerry Pratt and his co-workers (Pratt & Krupp, 2004). Jerry Pratt combined this actuation method with virtual model control in his doctoral thesis (Pratt J. , 2000). In the research which follows we assume some type of clutching mechanism—of either the virtual or actual variety—in order to take advantage of both active and passive control.

2.3 Overview of the Research

Throughout this dissertation, we examine the rimless wheel without collision losses. Why are we ignoring collisions when they are necessary for limit cycle behavior? First, limit cycle behavior is not a necessary condition of walking—you do not need periodic motion to complete the task of getting from point A to point B. In reality, the necessary condition for walking is simply stability (i.e., not falling down). Second, stability is only a necessary, but not a sufficient condition for walking. As we saw in the brachistochrone problem, what you are really trying to achieve is efficiency and speed! If walking isn't the fastest way to get there, we should pick another mode of transportation, despite this author's preference. Efficiency and speed are the reasons for walking in the first place and thus they are the focus of this dissertation. If we cannot justify walking transport under these ideal conditions (lossless collision), then it's pointless to go further. If, however, we find justification for walking, we bear in mind that this is a theoretical limit which can be approached, but never achieved[1].

In Chapter 3, we will model walking along a constant energy line—the passive rimless wheel. First, we show the stride width with the fastest speed—the optimal stride angle. Next, we find the band of energies where walking makes sense—the energy regime of walking.

In Chapter 4, we examine the problem of optimally powering a pendulum—the active rimless wheel. We show an optimal solution to powering the walking motion with respect to time and energy cost. Using Dynamic Programming and Pontryagin's Maximum Principle (PMP), we show an "on-off" control, and we describe the "switching curve" between these control extremes.

[1] similar to the thermal efficiency of the Carnot Cycle of an engine

Finally, In Chapter 5, we extend these methods to the compass gait, where the dynamics of the state and co-state equations cannot be solved analytically, and we rely on a direct search method, the Nelder Mead Simplex.

CHAPTER 3

PASSIVE RIMLESS WHEEL

3.1 The Simple Pendulum

An energy-based derivation of the simple pendulum equations (Strogatz, 1994) provides a succinct model of the walking behavior between collisions of the rimless wheel. If we ignore friction, this is a conservative system as shown in Figure 3.1. We derive the pendulum equation by adding the kinetic energy $T = \frac{1}{2}mv^2 = \frac{1}{2}ml^2\dot{\theta}^2$ with the potential Energy, $V = -mgl\cos\theta$ to yield the total energy, $E = T + V = \frac{1}{2}ml^2\dot{\theta}^2 - mgl\cos\theta$. We nondimensionalize this result by setting $m = l = g = 1$

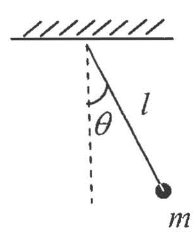


Figure 3.1: The Simple Pendulum

$$E = \frac{1}{2}\dot{\theta}^2 - \cos\theta \tag{3.1}$$

Along constant energy lines we can differentiate both sides to find

$$\frac{dE}{dt} = 0 = \dot{\theta}\left(\ddot{\theta} + \sin\theta\right)$$

Now, either nothing is moving, $\dot{\theta} = 0$, or

$$\ddot{\theta} + \sin\theta = 0 \tag{3.2}$$

which is the pendulum equation. Since this is a conservative system, we can plot the energy contours $E = \frac{1}{2}\dot{\theta}^2 - \cos\theta$ for different values of E as shown Figure 3.2. Note that in equation (3.1), if we set the value of $\dot{\theta} = 0$, and $\theta = \pi$, the pendulum is balanced inverted. You can see that this energy value, E = 1, is the minimum energy for walking, because values of E less than 1 will not swing past $\theta = \pi$, that is, they cannot whirl over the top. The separatrix appears as a red line in this figure at energy level $E = 1$. This contour line is called the separatrix, because it separates two distinct behaviors for the pendulum: whirling over the top or oscillating about the bottom. Energy levels above the separatrix, $E > 1$, have enough energy to whirl over the top, while energy levels below the separatrix, $E < 1$, oscillate about the origin at $\theta = 0$, never quite reaching the top. Figure 3.2 shows a zoomed in view of the phase plane of an inverted pendulum about its unstable node at π. The different energy levels are labeled on the contour lines.

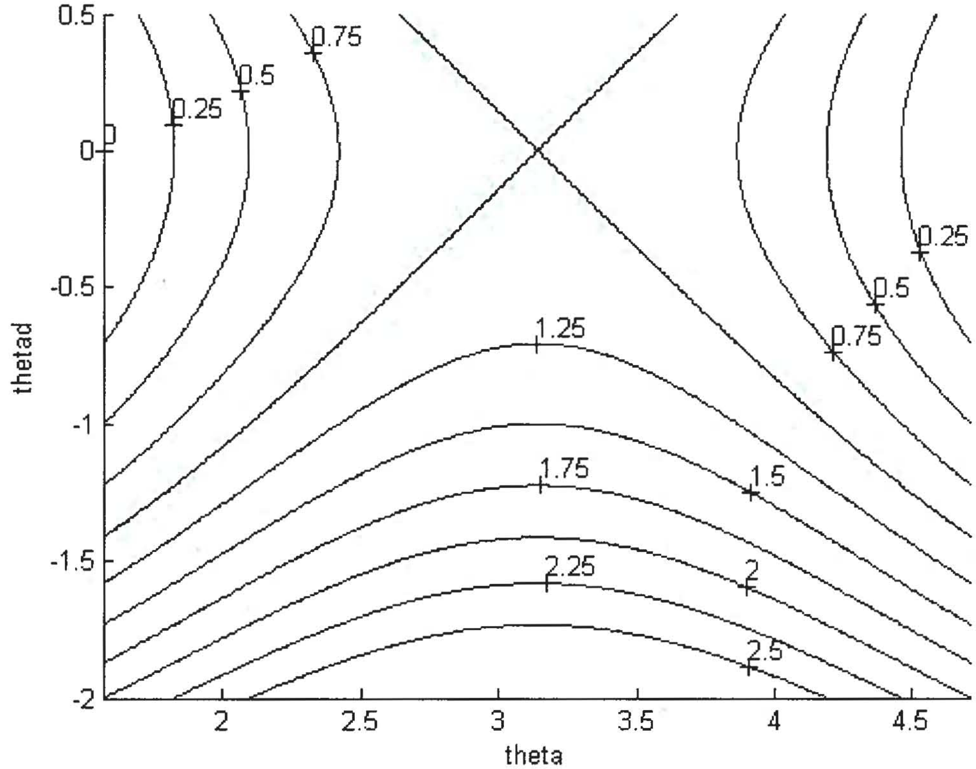


Figure 3.2: The Phase Plane of the Pendulum

We're really not interested in the entire phase plane of the pendulum: for one thing, the rimless wheel cannot dip below the floor, so we are physically limited to

$$\frac{\pi}{2} \leq \theta \leq \frac{3\pi}{2}$$

Practically speaking, we would like to look at an even smaller stride length than that. We can see from Figure 3.3 that the stride angle limits the phase plane even further to the following equation (3.3):

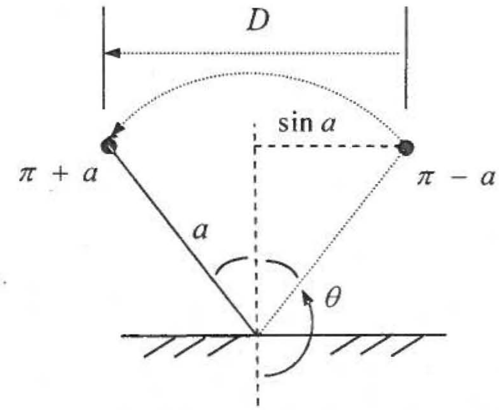


Figure 3.3: One stride length

$$\pi - a \leq \theta \leq \pi + a \tag{3.3}$$

Also note from the figure above that the stride length *D*:

$$D = 2\sin a \tag{3.4}$$

3.2 Optimal Stride Angle

Let's assume, for the moment, that you could create a rimless wheel that had no collision loss. We could argue about what kind of device could possibly have zero collision loss, but if we could create one, the bottom line is this: what step size, *a,* would be the fastest? That is, what step size would give us the fastest average forward speed, $\bar{v}$? This provides a rock-bottom analysis for walking, which should serve as a useful guide for building practical systems. For the simple pendulum, we find the swing time by solving for $\dot{\theta}$ in equation (3.1)

$$\dot{\theta} = \sqrt{2}\sqrt{E + \cos\theta} \tag{3.5}$$

Separation of Variables:

$$T=\int_0^T dt=\int_{\pi-a}^{\pi+a}\frac{d\theta}{\dot{\theta}}=\frac{1}{\sqrt{2}}\int_{\pi-a}^{\pi+a}\frac{d\theta}{\sqrt{E+\cos\theta}} \tag{3.6}$$

The above equation for stride time T can be stated in terms of elliptic integrals, as shown in the next section. If we use equation (3.4) for the stride length D and use equation (3.6) for stride time T, we find the formula for average speed: $\overline{v}=\dfrac{D}{T}$:

$$\overline{v}(E,a)=\frac{2\sin a}{\dfrac{1}{\sqrt{2}}\displaystyle\int_{\pi-a}^{\pi+a}\frac{d\theta}{\sqrt{E+\cos\theta}}} \tag{3.7}$$

After integration is performed, $\overline{v}$ is solely a function of the stride angle, a, as long as the energy E remains constant during the stride. To find the fastest stride-angle a^* for a given energy contour, E, we set the derivative of (3.7) to zero with respect to a, using Leibniz rule, while holding E constant:

$$\frac{2\sin a}{\displaystyle\int_{\pi-a}^{\pi+a}\frac{d\theta}{\sqrt{E+\cos\theta}}}=\cos a\sqrt{E-\cos a} \tag{3.8}$$

Those values of a that satisfy equation (3.8) are the optimal stride angles a^* for a given energy E. Note that the left hand side of the above equation bears remarkable similarity to the average speed $\overline{v}$ in equation (3.7). We will explain the significance of that fact over the course of the chapter. For now, we plot the optimal stride angles a^* over the range of energy contours $1<E<1.5$ as shown in Figure 3.4.

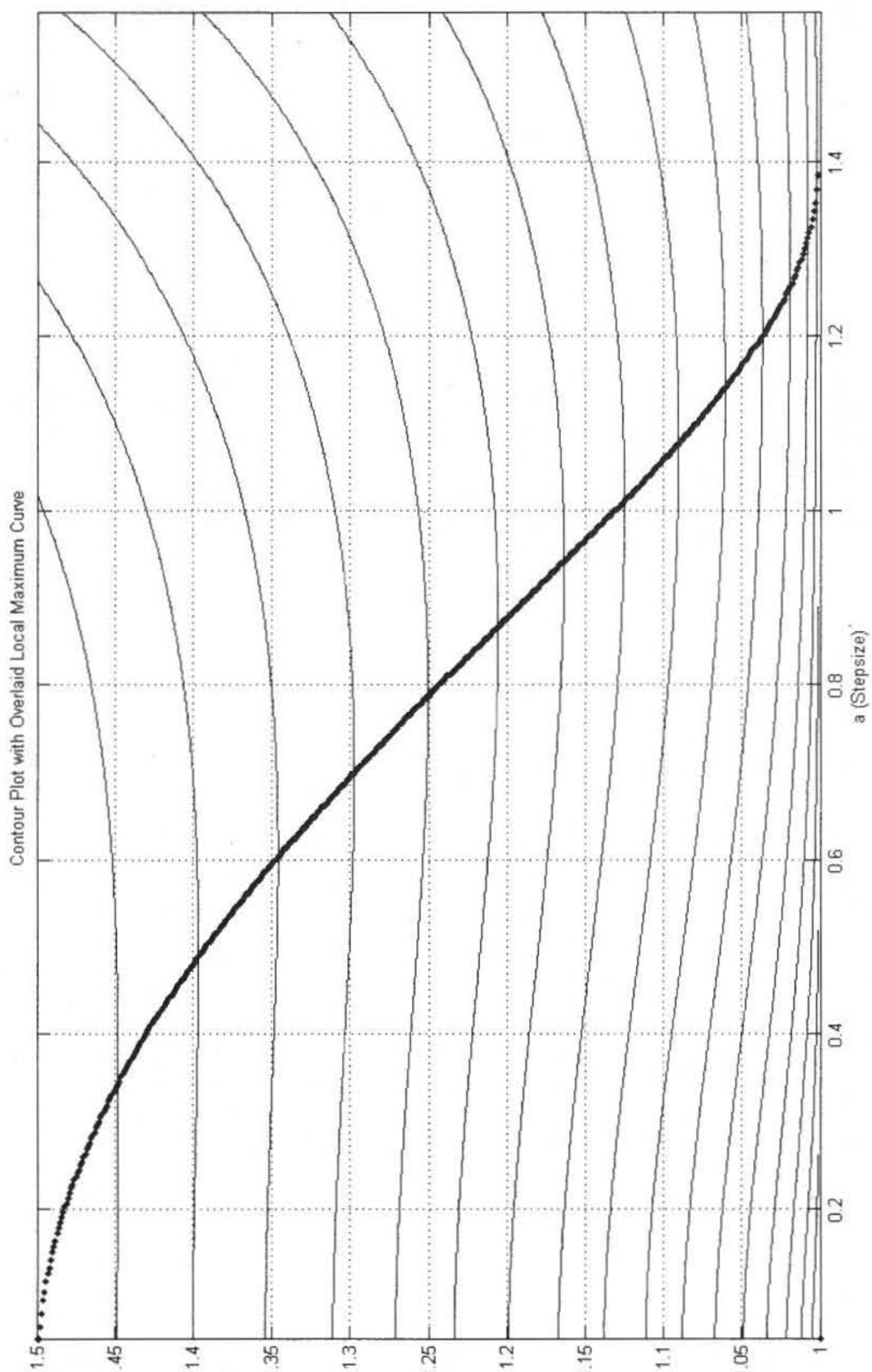


Figure 3.4: Plot of Fastest Stride Angles, *a*

3.3 Elliptic Integrals

We see from Figure 3.4 that the optimal step size goes to zero as we approach E = 1.5. This is the maximum walking energy, a fact which we will explain later in the chapter. In this section, we show that a closed form solution can be found for the swing time. The equation for swing time (3.6) can be described by elliptic integrals (Armitage & Eberlein, 2006). We define the elliptic integral of the first kind as:

$$F[\phi, k] \equiv \int \frac{dz}{\sqrt{1-k \sin z}} \tag{3.9}$$

The complete elliptic integral of the first kind is also defined as:

$$K[k] \equiv \int \frac{dz}{\sqrt{1-k \sin z}} \tag{3.10}$$

This form can be seen in equation (3.6) by substituting the trig identity

$\cos\theta = 1 - 2\sin\dfrac{\theta}{2}$ into the denominator of the integrand:

$$\begin{aligned} \sqrt{E+\cos\theta} &= \sqrt{E+1-2\sin\frac{\theta}{2}} \\ &= \sqrt{E+1}\sqrt{1-\frac{2}{E+1}\sin\frac{\theta}{2}} \end{aligned} \tag{3.11}$$

Substituting $\phi = \dfrac{\theta}{2} \rightarrow d\theta = 2d\phi$ and $k = \dfrac{2}{E+1}$ into equation (3.11), we find a new form of the swing time integral (3.6):

$$T = k \int_{\frac{\pi-a}{2}}^{\frac{\pi+a}{2}} \frac{d\phi}{\sqrt{1-k^2 \sin^2\phi}} \tag{3.12}$$

Now we can write the swing time integral in terms of the standard elliptic integrals defined above. To do this, we take advantage of the periodic nature of $\dfrac{1}{\sqrt{1-k^2\sin^2\phi}}$ as shown in Figure 3.5.

All the information in the integrand is contained in the interval $(0 \rightarrow \frac{\pi}{2})$. Any other interval required can be obtained by symmetry. Since swing time equation (3.12) integrates the section *2A* in the figure, we will recast section *A* in terms of the standard elliptic integrals. *A* can be rewritten as *A+F - F* where *F* is the elliptic integral of the first kind, and *K=A+F* is the complete elliptic integral of the first kind. Thus $T = 2A = 2(K-F)$. Now the swing time is in terms of the standard elliptic integrals:

$$T = 2k\left\{K[k^2] - F\left[\phi, k^2\right]\right\}$$

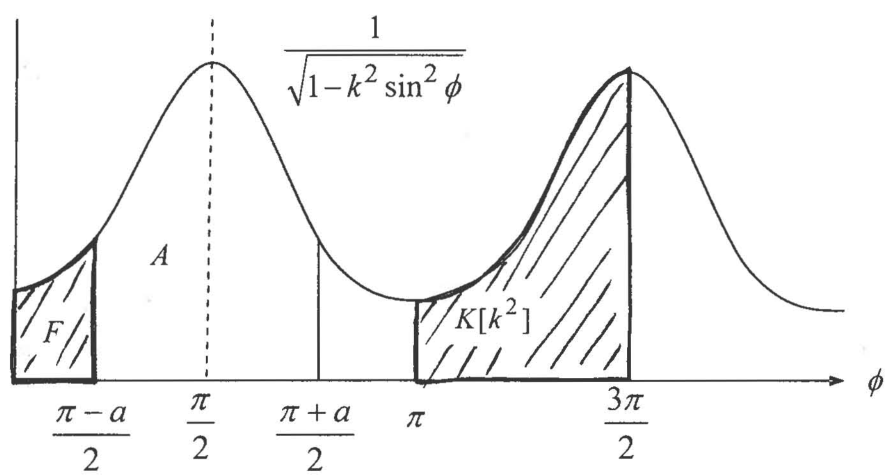


Figure 3.5: Elliptic Integrals

Substituting our values for k and ϕ:

$$T = \frac{2\sqrt{2}}{\sqrt{E+1}}\left\{K\left[\frac{2}{E+1}\right] - F\left[\frac{\pi - a}{2}, \frac{2}{E+1}\right]\right\} \tag{3.13}$$

where F is the elliptic integral of the first kind and K is the complete elliptic integral of the first kind as defined above.

3.4 Cleveland Problem

In the analysis which follows in Sections 3.5-3.6, we examine average walking speed using the position average, $\overline{v}_\theta$. The position average is an approximation of the time average. This can be seen most clearly with what we'll call the "Cleveland Problem": Say you are on a road trip from Detroit to Cleveland. Let's call the mid-point of the trip the Ohio border in Toledo where the speed limit changes from, say, 70mph to 50mph. What is your average speed if you travel the speed limit the entire trip? If we take the position average, the answer is 60mph. However, you are actually spending more time in Ohio, since your speed is slower there, so the time average is somewhat less than 60mph.

3.5 Position Average

To find the position average of speed, $\overline{v}_\theta$, we first need to find speed v as a function of θ. Looking at Figure 3.6, we see the familiar transformation from cylindrical to Cartesian coordinates. The instantaneous speed, v, is along the y-direction in this figure and is given by $v(\theta) = r\dot{\theta}\cos\theta$, if $r = 1$, then

$$v(\theta) = \dot{\theta}\cos\theta \tag{3.14}$$

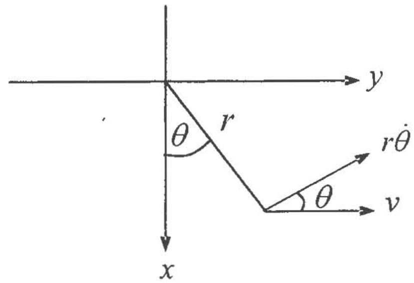


Figure 3.6: Pendulum Walker

We know from past work, equation (3.5), that $\dot{\theta} = \sqrt{2}\sqrt{E + \cos\theta}$. Substituting this into (3.14), we obtain this formula for the instantaneous speed as a function of angle θ:

$$v(\theta) = \sqrt{2}\sqrt{E + \cos\theta}\cos\theta \tag{3.15}$$

The formula for instantaneous speed $v(\theta)$ is not an approximation. However, if we take the position average of (3.15) with respect to θ:

$$\bar{v}_\theta = \frac{1}{2a}\int_{\pi-a}^{\pi+a} \sqrt{2}\sqrt{E + \cos\theta}\cos\theta d\theta \tag{3.16}$$

This is an approximation of the time average. To find optimum stride-angle for a given energy contour, we need to set the derivative of (3.16) to zero with respect to *a*, holding *E* constant (using Leibniz integral rule):

$$\frac{1}{2a}\int_{\pi-a}^{\pi+a} \sqrt{E + \cos\theta}\cos\theta d\theta = -\cos a\sqrt{E - \cos a} \tag{3.17}$$

This is a similar result to the optimal stride angle that we saw in equation (3.8), but here we will show the details. Leibniz' rule for differentiating through the integral sign is:

$$\frac{\partial}{\partial a}\int_{g(a)}^{h(a)} f(\theta,a)\,d\theta = f\left[h(a),a\right]h'(a) - f\left[g(a),a\right]g'(a) + \int_{g(a)}^{h(a)} \frac{\partial f(\theta,a)}{\partial a}\,d\theta \qquad (3.18)$$

Setting the derivative of the average speed to zero requires us to differentiate under the integral sign in equation (3.16):

$$\frac{\partial}{\partial a}\int_{\pi-a}^{\pi+a} \frac{\sqrt{2}}{2a}\sqrt{E+\cos\theta}\,\cos\theta\,d\theta = 0 =$$

$$\frac{\sqrt{2}}{2a}\sqrt{E+\cos(\pi+a)}\cos(\pi+a) + \frac{\sqrt{2}}{2a}\sqrt{E+\cos(\pi-a)}\cos(\pi-a) + \ldots \qquad (3.19)$$

$$\ldots - \frac{1}{2a^2}\int_{\pi-a}^{\pi+a} \sqrt{2}\sqrt{E+\cos\theta}\,\cos\theta\,d\theta$$

Substituting the trigonometric identity $\cos(A+B) = \cos A\cos B - \sin A\sin B$, that is, $\cos(\pi-a) = \cos(\pi+a) = -\cos a$, we find:

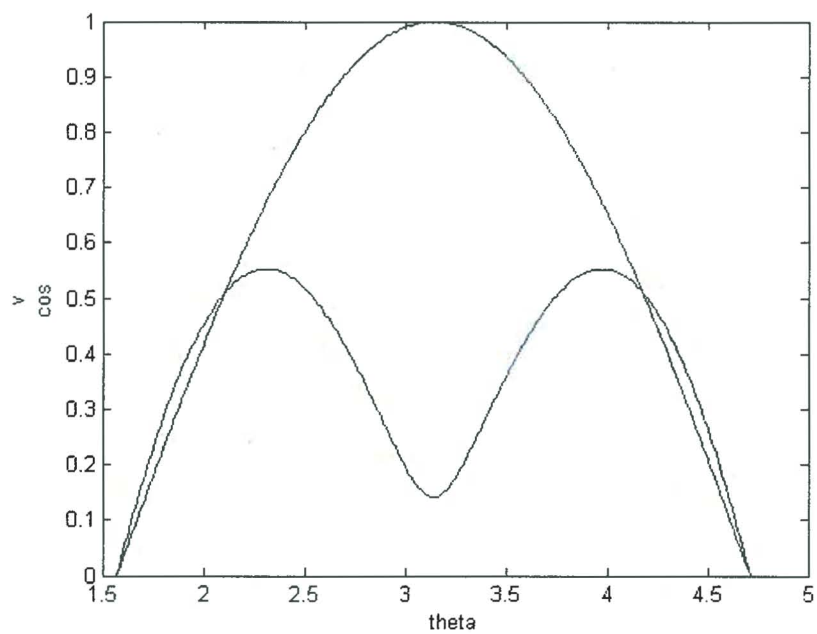
$$\frac{1}{2a}\int_{\pi-a}^{\pi+a} \sqrt{2}\sqrt{E+\cos\theta}\,\cos\theta = -\frac{\sqrt{2}}{2}\sqrt{E-\cos a}\,\cos a - \frac{\sqrt{2}}{2}\sqrt{E-\cos a}\,\cos a$$

$$= -\sqrt{2}\sqrt{E+\cos(\pi+a)}\cos(\pi+a) \qquad (3.20)$$

which is the formula for the fastest stride angle (3.17) that we saw before. The interesting thing to note here is that the left hand side of the equation is the average velocity, as we saw in equation (3.8), and the right hand side of the equation is the instantaneous velocity

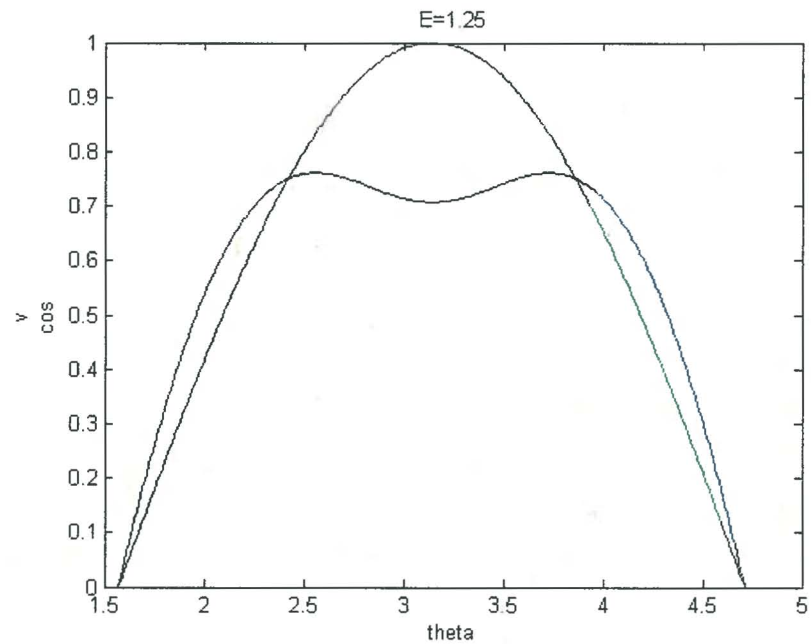
$$\overline{v}_{\theta} = v(\pi + a) \tag{3.21}$$

What this is telling us in words is that we should continue to take a larger step a (which adjusts the average velocity) until the instantaneous speed equals the average speed. We can see this graphically in Figure 3.7.

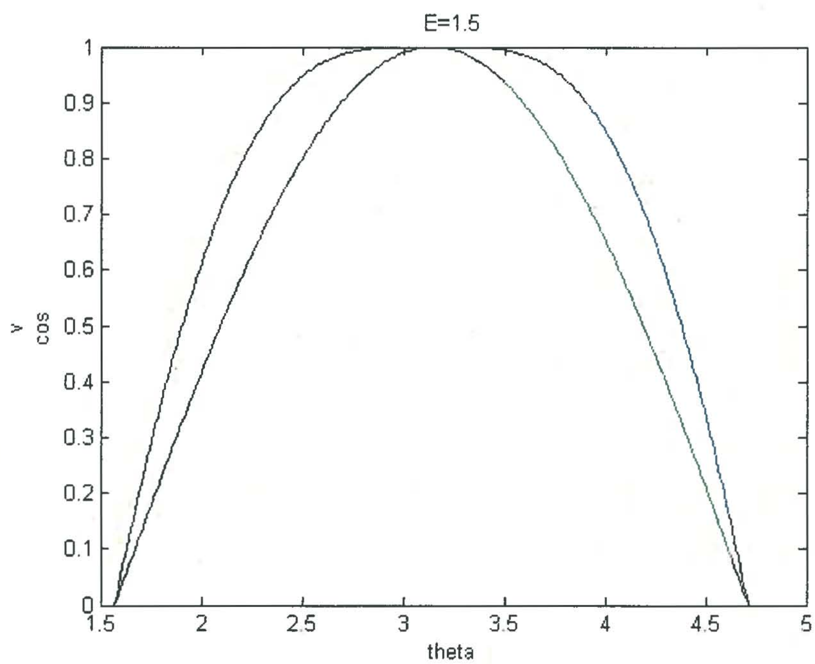


(a)

Figure 3.7: Instantaneous forward speed, $v(\theta)$, plotted in blue with cosine plotted in green for energy levels (a) E=1.01 (b) E=1.25 (c) E=1.5 (*This figure is presented in color; the black and white reproduction may not be an accurate representation*)



(b)



(c)

Figure 3.7-Continued

We plot the instantaneous speed (3.15) as a function of θ. To find the average speed, simply average the height of the blue graph from $\pi - a$ to $\pi + a$. Then vary a to see how this affects the average.

Starting from $a = 0$, to build speed, we fall forwards (increasing a), which takes us away from our desired forward direction—we are taxed with a cosine penalty because of it—but our increased average speed is worth the penalty at first. As we approach $a = \frac{\pi}{2}$ however, the penalty approaches 100% (note the cosine penalty in green) because we are moving towards the ground and not moving forward at all. This creates a tradeoff: take a step (increase a) to increase speed, but don't increase a so large that you nose the velocity into the ground.

In Figure 3.7(a) we see the graph of $v(\theta)$ with $E = 1.01$; the pendulum is barely above the separatrix which means that it slows down a whole lot near the peak at $\theta = \pi$. Here we take a relatively large step to increase our average speed by taking in the higher peaks of the velocity curve, v, to the left and right of $\theta = \pi$. At higher values of E as shown in Figure 3.7(b), we see the local minimum of v at $\theta = \pi$ become less pronounced until finally, in Figure 3.7(c) at $E = 1.5$, the local minimum has become a global maximum and it no longer makes sense to walk—similar to what we saw before in Figure 3.4.

$|\dot{\theta}|$ always has a minimum at $\theta = \pi$, so why does it no longer make sense to walk? At higher energies, the change you get from potential to kinetic energy is chump change compared to the total energy, i.e., E is made up mostly of kinetic energy to begin with, and it is not affected much by changes in θ—note how the contour lines Figure 3.2 become flatter and flatter with respect to θ as E goes from 1 to 1.5. This means that for $E > 1.5$, any deviation from straight line travel from point A to point B is not worth the price. You should not walk, you should roll.

Another way to reason about optimal step width from equation (3.21) is by looking at the sampled average:

$$\overline{v}_n = \frac{1}{n}\left(v_1 + v_2 + \ldots v_n\right) \tag{3.22}$$

where n is the number of samples taken. How would you know if we should be taking a bigger step? We assume that the step is centered around $\theta = \pi$, and we iterate a numerically so that the stepwidth gets incrementally larger. Thus if

$$\overline{v}_{n+1} > \overline{v}_n \tag{3.23}$$

we know that we want to take that incrementally bigger step. From (3.22) we see that

$$\begin{aligned}
\overline{v}_{n+1} &\equiv \frac{1}{n+1}\sum_{1}^{n+1} v_i \\
&= \frac{n}{n+1}\cdot\frac{1}{n}\sum_{1}^{n} v_i + \frac{1}{n+1} v_{n+1} \\
&= \frac{n}{n+1}\overline{v}_n + \frac{1}{n+1} v_{n+1}
\end{aligned}$$

so if we substitute $\overline{v}_{n+1}$ from above into (3.23) we get:

$$\frac{n}{n+1}\overline{v}_n + \frac{1}{n+1} v_{n+1} > \overline{v}_n$$

$$\frac{n-(n+1)}{n+1}\overline{v}_n > -\frac{1}{n+1} v_{n+1}$$

The condition for satisfying (3.23) is

$$v_{n+1} > \overline{v}_n \tag{3.24}$$

In other words, if the next incremental velocity is greater than your average, take it. This is precisely the recipe that we found in equation (3.21) for the best position average and in equation (3.8) for the best time average.

We see this in Figure 3.8, which depicts in blue the right half of the graph of instantaneous velocity that we saw previously in Figure 3.7. Since those graphs were symmetric, we take the average over only the right half side. Look at the plot of the average velocity in red. It increases until it reaches the condition in equation (3.24) where the curves cross, and from then on the average velocity decreases.

3.6 Energy Regime of Walking (1< E < 1.5)

We know how to determine the minimum walking energy, the separatrix where the pendulum just makes it to the top of its swing at $E = 1$. We substitute $\theta = \pi$ and $\dot{\theta} = 0$ into equation (3.1) to find this. We've seen clues from Figure 3.7 that $E = 1.5$ is the maximum walking energy. Is there another way we can find this value? Let's start from equation (3.15) for instantaneous forward velocity. If we take the derivative of v with respect to θ, we see that

$$\frac{dv}{d\theta} = -\frac{\sqrt{2}}{2}\cos\theta\frac{\sin\theta}{\sqrt{E+\cos\theta}} - \sqrt{2}\sin\theta\sqrt{E+\cos\theta}$$

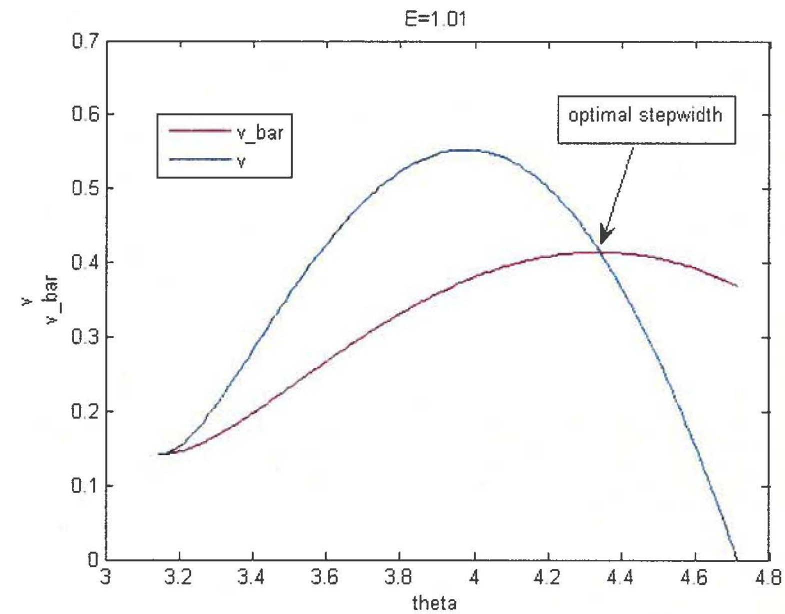
Simplifying:

$$\frac{dv}{d\theta} = -\frac{\sqrt{2}}{2}\sin\theta\left(\frac{2E+3\cos\theta}{\sqrt{E+\cos\theta}}\right) \tag{3.25}$$

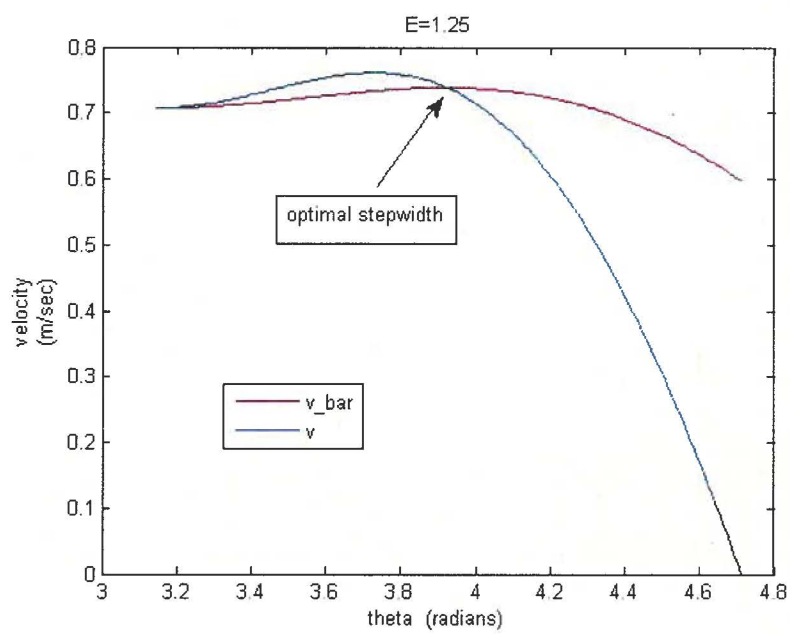
We are interested in the behavior of $\frac{dv}{d\theta}$ at the critical point $\theta = \pi$ because that is the point—evidenced by Figure 3.7—where v changes from a local minimum to a global maximum as E approaches 1.5. There is always an extremum there, because clearly $\sin\pi = 0$. The question is whether that extremum is a maximum or a minimum. If $\frac{dv}{d\theta}$ changes sign from + to – at $\theta = \pi$, then we know v is a local maximum there.

Note that $\sin\theta$ changes from + to – through $\theta = \pi$, so what happens to the sign of $\frac{dv}{d\theta}$ around π depends on the value of E in the numerator and denominator of equation (3.25). $E > 1$ so the sign of the denominator is always positive. We are left with one culprit: the numerator $2E+3\cos\pi$, which changes sign when $E = 1.5$. This is the maximum walking energy—the point where v changes from having a local minimum to having a global maximum.

Figure 3.8 shows the change in the optimal stepwidth for two different energy levels. It also shows that the optimum stepwidth occurs where the instantaneous speed equals the average speed (position average).



(a)



(b)

Figure 3.8: Optimum Stepwidth at (a) $E = 1.01$ (b) $E = 1.25$ (*This figure is presented in color; the black and white reproduction may not be an accurate representation*)

CHAPTER 4

POWERED RIMLESS WHEEL

4.1 Powered Pendulum

We develop a dynamic programming algorithm to calculate the minimum cost in time and energy for a powered simple pendulum given by:

$$\ddot{\theta} + \sin\theta = u \tag{4.1}$$

Powering the pendulum adds energy to the system; it is not a conservative system anymore. However, we can calculate the change in energy by taking the time derivative of the energy equation for the simple pendulum (3.1) and comparing the result to the powered system in (4.1):

$$\frac{dE}{dt} = \frac{d}{dt}\left(\frac{1}{2}\dot{\theta}^2 - \cos\theta\right) = \dot{\theta}\left(\ddot{\theta} + \sin\theta\right) = u\dot{\theta} \tag{4.2}$$

We see from the above equation that at lower speeds, $\dot{\theta}$, a higher torque, u, is required for a given energy change, ΔE. For the dynamic programming analysis which follows, we treat the powered pendulum as a discrete system able to "jump" to a new energy level ΔE, or coast along the constant energy contours of the pendulum shown in Figure 4.1. Figure 4.2 shows this discretization of phase space into a directed network. Starting at the upper right of the graph, we have two choices at node 1: stay on the energy level you are currently at—move to node 6, or increase the energy level by one increment—move to node 7.

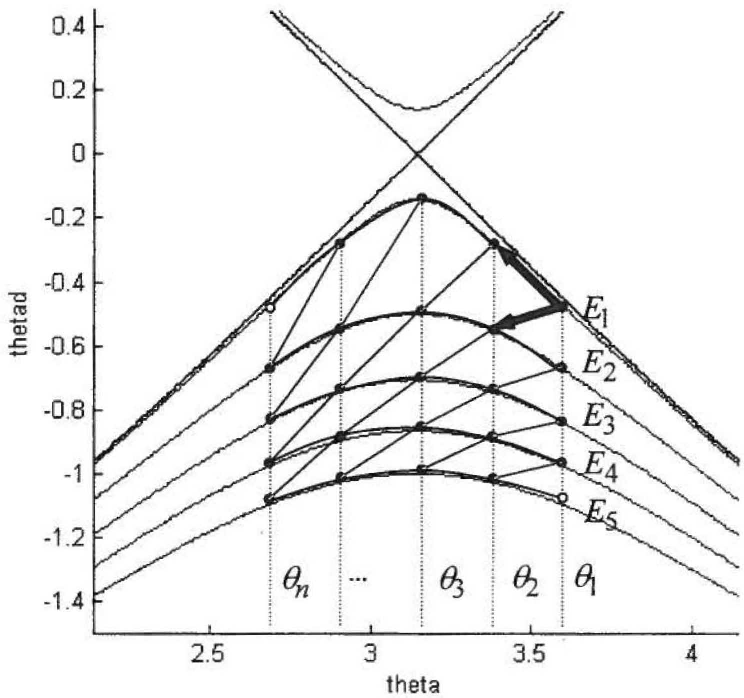


Figure 4.1. Phase Plane of the Simple Pendulum Discretized into a Directed Network.

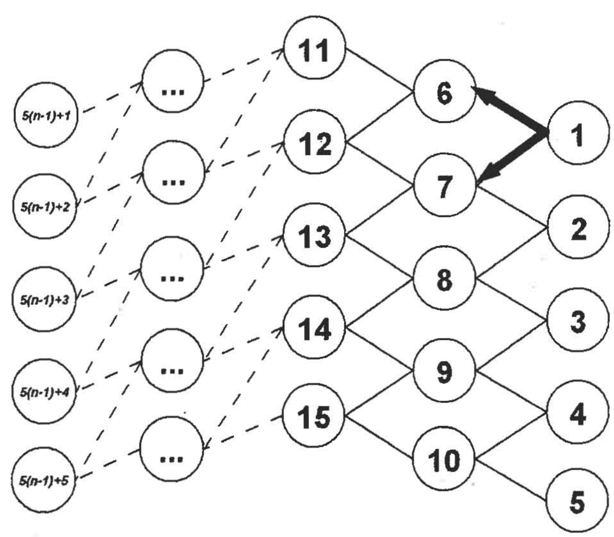


Figure 4.2: Node Numbering Method.

4.2 Principle of Optimality

Both Dynamic Programming and Pontryagin's Maximum Principle are conceptually based on the principle of optimality: If *a-b-d* is the optimal path from *a* to *d*, then *b-d* must be the optimal path from *b* to *d*.

The optimal path for a multistage decision process is shown in Figure 4.3. Suppose that the first decision (made at *a*) results in segment *a-b* with cost J_{ab} and that the remaining decisions yield segment *b-d* at a cost of J_{bd}. The minimum cost J_{ad}^* from *a* to *d* is then:

$$J_{ad}^* = J_{ab} + J_{bd} \tag{4.3}$$

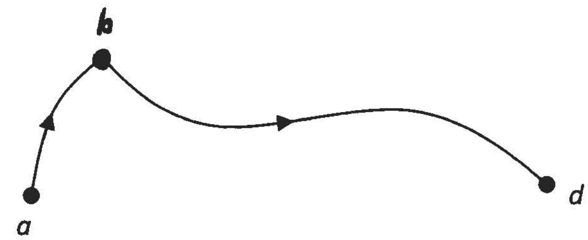
Proof by contradiction (Kirk, 1970): Suppose *b-c-d* in Figure 4.3 is the optimal path from *b* to *d*; then

$$\begin{aligned} &J_{bcd} < J_{bd} \\ &J_{ab} + J_{bcd} < J_{ab} + J_{be} = J_{ad}^* \end{aligned} \tag{4.4}$$

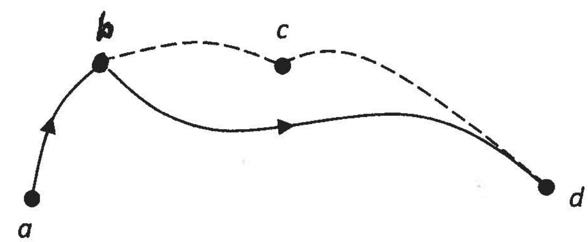
but (4.4) can only be satisfied by violating the condition that *a-b-d* is the optimal path from *a* to *d*. Thus the assertion is proved.

4.3 Dynamic Programming

Dynamic programming is a numerical technique to find the optimal path through a directed network. A directed network consists of a set of nodes and directed arcs between the nodes. A directed arc as we see in Figure 4.3 is simply an ordered pair



(a)



(b)

Figure 4.3: The Principle of Optimality (a) and its Proof (b)

(a,b), where a and b are nodes. The length or cost of a directed arc is denoted as J_{ab}. In our optimization problem (Denardo, 1975), we wish to find the shortest path through the network. Let $f_b \equiv J_{bd}^{*}$ be the minimum travel time from node b to the goal d. We interpret $J_{ab} + f_b$ as the travel time of the path from node a to the goal d that first traverses arc (a,b) and then travels as efficiently as possible from node b to the goal d as we see in Figure 4.4.

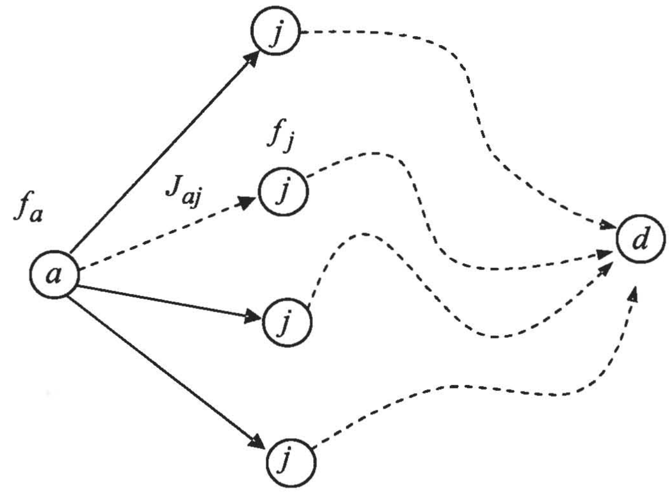


Figure 4.4: The Minimum Path

According to the principle of optimality, in order for $f_a \equiv J_{ad}^*$ to be the optimal path from *a-b-d*, the path $f_b \equiv J_{bd}^*$ must be optimal as well. In Figure 4.4, we use this fact to minimize the path from *a-j-d* over all *j* where (a, j) is an arc. Each of the *j* nodes has a corresponding f_j, which is the best path from *j* forward--shown in dashed lines. The best path from *a* to *d* is the one that minimizes the sum of 1) the immediate cost J_{aj} and 2) the long term cost f_j—the cost of the state you put yourself in—assuming, of course, that we behave optimally as we traverse from *j* to *d*. This is known as the functional equation:

$$f_a = \min_j \{J_{aj} + f_j\} \tag{4.5}$$

We can use this to work our way backwards through the network similar to how you would plan your morning based on your known start time at work: If I have to be at work by 0700 and it takes 30 minutes to drive there, I must leave the house by 0630, and it takes me 15 minutes to eat breakfast so….it's called back timing, you solve the problem of what time to get up for work in the morning by starting from the end. Also note that we solve the problem of finding the minimum path from *a* to *d* by embedding it into the more general problem of finding the minimum path from any point *j* to *d*.

We use the discretization of phase space into E and θ that we saw in Figure 4.2. At each node we have two choices—move up one energy level, or stay on the energy contour you are at. Each choice has an associated time cost Δt. If you stay on the same energy contour, your only cost is the time cost Δt to go from node a to b. Keep in mind that the time cost Δt, varies depending on where you are at on the graph according to the equation $\Delta t = \frac{\Delta\theta}{\dot{\theta}}$ where we use equation (3.5) for $\dot{\theta}$. If you move up an energy level, you incur a cost equal to the time cost plus the energy cost: $\Delta t + \Delta E$

$$Cost(i,j) = \left| \begin{array}{lll} \Delta t & if & j=1 \\ \Delta t + \Delta E & if & j=2 \end{array} \right.$$

This corresponds to an arc length, where there are now only two choices for *j*—staying on the same energy level means that $j=1$ and $j=2$ means we are bumping up one level. To be compatible with the numbering scheme shown in Figure 4.2 we must write $b = a + j + 4$, so that J_{ab} becomes:

$$J_{aj} + f_b = Cost(a,j) + f(a+j+4) \tag{4.6}$$

The cost matrix $Cost(a, j)$ has two columns and as many rows as there are nodes.

With this notation, we can now develop an algorithm for moving backwards through the network to solve the functional equation (4.5):

```
for a = d:-1:1
    for j = 1:2
        Total_Cost(a,j)=Cost(a,j)+f(a+j+4);
    end
    f(a)=min(Total_Cost(a,:));
end
```

For example let's examine Figure 4.2 using the above code for the case where $a = 1$. We are at node 1 and we are trying to see which direction arrow we should take next (go to node 6? or node 7?). The code will find the minimum of $\{Cost(1,1) + f(6)\}$ and $\{Cost(1,2) + f(7)\}$ and set that equal to $f(1)$. This code is thus another statement of equation (4.5). To know what $f(6)$ and $f(7)$ are, we have to work backwards through the code, starting at the end. The nodes are numbered sequentially to allow for the above code to work.

In Figure 4.5, we see the result of this minimum path tree for any point in phase space. Basically, the graph is telling us what should be intuitively obvious: at lower energy levels, it will take so long to get to the other side, you might as well spend the energy and gain the time savings of being on a higher energy level (hence: higher speed) as quickly as possible. What might not be as obvious is that this logic has limits:

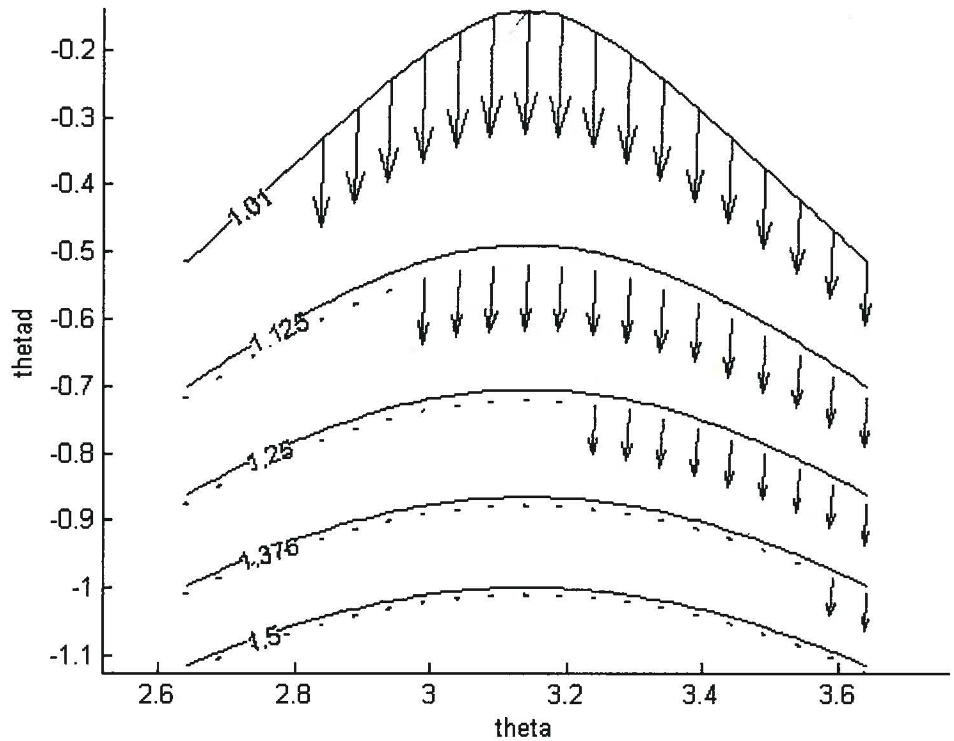


Figure 4.5: Dynamic Programming for the Active Rimless Wheel

there comes a point at which it is no longer worth it to expend the energy. The graph clearly shows the emergence of a switching curve.

4.4 Pontryagin's Maximum Principle

Pontryagin's Maximum Principle (PMP) is the second method we will use to study the active rimless wheel. PMP is a continuous time analog of dynamic programming, and it is a statement of mathematical truth. We introduce PMP with an economic example of maximizing profits.

Consider the decision problem of a firm (Dorfman, 1969) that wishes to maximize its total profits J over some period of time. The firm produces profits at a certain rate ϕ depending on the amount of (income generating) capital stock $x(t)$ at the specified date and decisions $u(t)$ at the specified date—rate of output, price of output, etc… The total profits that will be earned over a period of time are given by summing up this cash flow function over time to yield the value functional (Weber, 2005).

$$J\left[x(0),\vec{u}\right]=\int_0^T \phi(x,u)\,dt \tag{4.7}$$

J is a functional because it depends upon the entire time path $\vec{u}$ of the decision variable, not just the ordinary variable u. If we conjecture that an optimal path u^* of the solution exists, we could follow that path to obtain the optimal value function.

$$J^*\left[x(0)\right]=\max_u J\left[x(0),\vec{u}\right] \tag{4.8}$$

Take a closer look at this expression. This maximum depends only on the starting position of the state variable $x(0)$. It does not depend upon the control variable $u(t)$ because we assume that we are behaving optimally from $x(0)$ on. We are doing the best we can with the hand we were dealt, and thus u has been "maximized out," and J^* is now an ordinary function to which we can apply standard calculus. Since J^*, the optimal value function of the firm, depends only on the initial starting point, we would like to measure the effect of changing that starting point $x(0)$, i.e. we seek its dependence on state. The shadow price of capital or co-state is defined as:

$$z \equiv \frac{\partial J^*}{\partial x} \tag{4.9}$$

The co-state measures the effect of an extra capital unit on the value of the firm[1]. At any time *t*, the value of the firm equals the shadow price of capital multiplied by the firm's capital stock $z(t)x(t)$, At each point in time, the value of the firm is changing by

$$\frac{d}{dt}\left[z(t)x(t)\right] = z\dot{x} + \dot{z}x \tag{4.10}$$

The firm's value changes because the amount of physical capital changes, $z\dot{x}$, and the value of this capital changes, $\dot{z}x$. Let $\phi(x,u)$ be the net revenue at time *t*, expressed in present value terms. The decision maker[2] chooses the time path of output that maximizes the sum of the momentary net revenue and the change in the value of the firm at every instant:

$$\phi(x,u) + z\dot{x} + \dot{z}x \tag{4.11}$$

The time path of the capital stock must also fulfill the state equation.

$$\dot{x} = f(x,u) \tag{4.12}$$

Substituting this into (4.11) yields:

---

[1] For example, the shadow price of a fish is $10 if adding an extra fish to the fish stock increases the value of a fishery by $10. Similarly, extracting a unit of ore reduces the value of a mine by the ore's shadow price. An extra capital unit adds value to an enterprise because it contributes to current and future revenues.

[2] The first extreme might be the short-term manager, who squeezes out every ounce of production, sacrificing the long-term viability of the firm. The other extreme might be the money manager of a trust fund, who in an effort to maximize the endowment, refuses to give out income to the intended beneficiaries.

$$\phi(x,u) + zf(x,u) + \dot{z}x \tag{4.13}$$

Partial differentiation of the equation (4.13), and setting equal to zero for maximization, produces two conditions that the optimal time paths of the decision (control) variable u* and state variable x* must fulfill:

$$\frac{\partial \phi}{\partial u} + z\frac{\partial f}{\partial u} = 0 \tag{4.14}$$

$$\frac{\partial \phi}{\partial x} + z\frac{\partial f}{\partial x} + \dot{z} = 0 \tag{4.15}$$

These equations are recognized as Pontryagin's Maximum Principle. The first equation is the maximum principle and the second equation is the co-state equation.

Another way to motivate PMP following (Dorfman, 1969) is to proceed exactly like we did with Dynamic Programming in Figure 4.4; we maximize the immediate cost of a decision over Δt, and the cost of proceeding optimally from $x(t+\Delta t)$ on

$$\max_u \left\{ \phi \cdot \Delta t + J^*\left[x(t+\Delta t)\right] \right\} \tag{4.16}$$

Since $J^*\left[x(u)\right]$, we use the chain rule to take the derivative of (4.16):

$$\frac{\partial \phi}{\partial u} \cdot \Delta t + \frac{\partial J^*}{\partial x}\frac{\partial}{\partial u}\left[x(t+\Delta t)\right] \tag{4.17}$$

We use the Euler approximation of the state equation $\dot{x} \approx \dfrac{x(t+\Delta t) - x(t)}{\Delta t}$ and the state equation (4.12) to solve for $x(t+\Delta t) \approx x + f(x,u)\Delta t$

$$\frac{\partial \phi}{\partial u} \cdot \Delta t + z \frac{\partial}{\partial u}\left[x + f(x,u) \cdot \Delta t\right] \tag{4.18}$$

where we have substituted the definition of z, the co-state equation(4.9). Setting the derivative equal to zero, we can cancel out the Δt to yield PMP:

$$\frac{\partial \phi}{\partial u} \cdot \Delta t + z \frac{\partial f}{\partial u} \cdot \Delta t = 0$$

To remember these formulas, write the Hamiltonian:

$$H = \phi(x,u) + zf(x,u) \tag{4.19}$$

where

$$\frac{\partial H}{\partial z} = \dot{x} \qquad \text{(state)}$$

$$\frac{\partial H}{\partial x} = -\dot{z} \qquad \text{(co-state)} \tag{4.20}$$

$$\frac{\partial H}{\partial u} = 0 \qquad \text{(PMP)}$$

For the case where $\phi(x,u)$ is a cost we are trying to minimize, and since $\min \phi = \max(-\phi)$, we merely rewrite the Hamiltonian as:

$$H = -\phi(x,u) + zf(x,u) \tag{4.21}$$

4.4 Rocket Car

To begin, we take a textbook optimal control example, the rocket car, with slight modifications to motivate our study: Figure 4.6 shows the outline of the procedure that we will follow for the rest of the chapter—this deliberately simplifies the problem by leaving out potential energy changes in the system. We want to reach the origin with maximum velocity, instead of zero velocity, to simulate the transport of mass as quickly as possible through a single stride.

In the textbook treatment of the rocket car problem, the cost function, ϕ, is assumed to be for a time optimal problem, a fuel optimal problem, an energy optimal, or a mixed case of these.

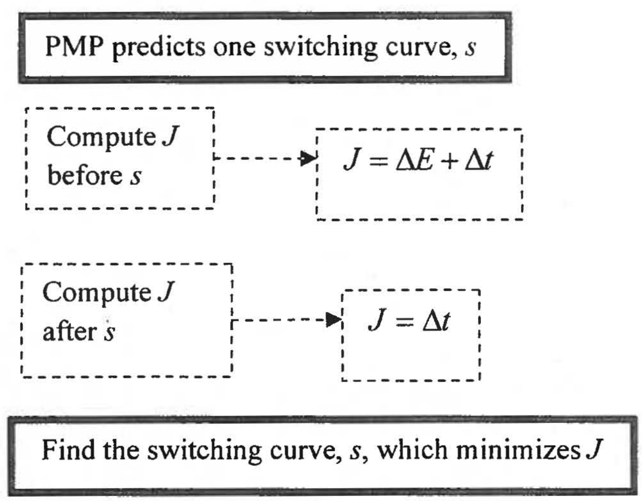


Figure 4.6: The PMP Research Outline

The standard treatment[3] for the energy cost function is $\phi = u^2$. In this section, we derive a treatment for energy where u is a control force instead. We show that the cost function for energy is $\phi = uv$. For the non-continuous cases we will be studying, PMP takes on a slightly different form than (4.20), we now look for supremum of H, rather than its derivative.

Given:

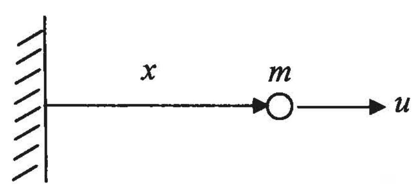


Figure 4.7: The positioning problem

Required:

Drive the mass back to the origin at $x = 0$ with maximum velocity, while minimizing some integral-type cost, J, where:

[3] If u is a current source, then $P = iV = i(iR) = i^2 R$

$$J = \int_0^T \phi(\mathbf{x}, u)\, dt \tag{4.22}$$

In this problem, we choose two components for the cost (time and energy)

$$J = \int dt + \int dE$$

Substituting for power

$$J = \int dt + \int \frac{dE}{dt} dt$$

Substituting for kinetic energy (we have no change in potential energy here)

$$\int \frac{dE}{dt} dt = \int \frac{d}{dt}\left(\frac{1}{2} m v^2\right) dt$$

Noting that $m\dot{v} = ma = u$:

$$\int m v \dot{v} dt = \int u v dt$$

Finally, substitute the above result into (4.22) to yield the cost integral:

$$J = \int_0^T (k + uv)\, dt \tag{4.23}$$

where k is a positive constant to measure the relative importance between time and energy efficiency.

1st step in the Pontryagin Maximum Principle: Treat cost as an additional state. The state equations are thus given by:

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} k+uv \\ v \\ u \end{pmatrix} = \begin{pmatrix} k+ux_2 \\ x_2 \\ u \end{pmatrix} \tag{4.24}$$

Since we want to approach the origin with maximum velocity, u is bounded by zero(Kirk, 1970)

$$-1 \le u \le 0$$

In other words, while you may want to back off the gas, you would never want to apply the brakes.

2nd step in the Pontryagin Maximum Principle: We introduce the co-state variables to form the Hamiltonian:

$$H = z_0\dot{x}_0 + z_1\dot{x}_1 + z_2\dot{x}_2 \tag{4.25}$$

Substituting (4.24) we see that:

$$H = z_0\left(k + ux_2\right) + z_1x_2 + z_2u \tag{4.26}$$

The co-state equations are prescribed by Hamilton's equations:

$$\dot{z}_0 = -\frac{\partial H}{\partial x_0} = 0 \tag{4.27}$$

$$\dot{z}_1 = -\frac{\partial H}{\partial x_1} = 0 \tag{4.28}$$

$$\dot{z}_2 = -\frac{\partial H}{\partial x_2} = -z_1 - z_0u \tag{4.29}$$

3rd step in the Poyntryagin Maximum Principle. Solve the above co-state equations. The equation for $\dot{z}_0$ shows that $z_0 = const$, and the PMP requires that this constant should be negative. Without loss of generality we can choose $z_0 = -1$. The two solutions to (4.27) and (4.28) are thus:

$$z_0 = -1 \qquad z_1 = A \tag{4.30}$$

where *A* is some constant of integration. The solution to equation (4.29) is given by substituting (4.30), which yields

$$\dot{z}_2 = u - A \tag{4.31}$$

and the solution becomes

$$z_2 = \int u dt - At + B \tag{4.32}$$

Noting again that $u = m\dot{v}$, we know that $\int u dt = mv - mv_0$, so that

$$z_2 = mx_2 + B - mv_0 - At \tag{4.33}$$

where *B* is another constant of integration.

4th step in the Poyntryagin Maximum Principle: Find the supremum of *H* as a function of *u*. The Hamiltonian is given by substituting (4.30) and (4.33) into (4.26):

$$H = -k - ux_2 + Ax_2 + u(mx_2 + B - mv_0 - At) \tag{4.34}$$

To maximize *H* as a function of *u,* we must maximize the term *uq* where

$$q = (m-1)x_2 + B - mv_0 - At \tag{4.35}$$

If m=1 then $q' = B' - At$ where $B' = B - mv_0$ and since q' is a linear function of t there is at most one zero crossing for q'. Thus

$$u = \begin{vmatrix} -1 & if & q' < 0 \\ 0 & if & q' > 0 \end{vmatrix} \tag{4.36}$$

PMP has thus shown that we have "on-off" control with a single switch. This makes intuitive sense, since we would want to apply the force early on in order to gain the time savings accrued throughout the stride. However, we would want to shut the force off after some time period in order to save energy.

We desire to find the optimal control u^*. From the get go, we assume that the optimal solution is "on-off" as described in the previous section. First, we proceed with a bang, i.e., from the initial position to the switching position s we apply full reverse force $(u = -1)$, and then at s we cut the engines off $(u = 0)$ and coast through the origin with a cruising speed of v_c as shown in the phase plot in Figure 4.8. Here's the key: we can find the switching position s by re-writing the cost integral (4.23) as a function of x instead of t:

$$J = \int_0^T (k + uv)\,dt = \int_{x1}^{x2} k\frac{dx}{v} + \int_{x1}^{x2} udx \tag{4.37}$$

where we have used the the definition of work $E = \int F \cdot ds = \int udx$.

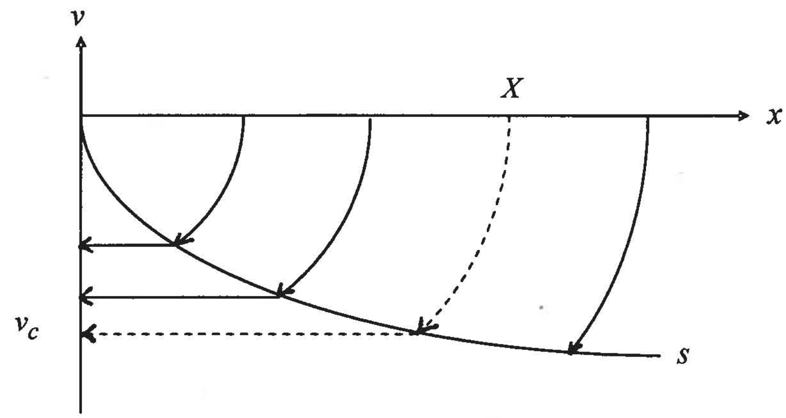


Figure 4.8: The Switching Curve

For the "bang" section of travel, we incur both time and energy cost. Working backwards from the initial position X at zero velocity until the switch at position s and velocity v_c, the time cost is:

$$\int_X^s k\frac{dx}{v} = km\int_0^{v_c}\frac{dv}{u} = km\int_0^{v_c}\frac{dv}{-1} = -kmv_c \tag{4.38}$$

which uses the physics result $ads = vdv$ recast as $udx = mvdv$. Recall that the velocity is negative (so the above cost is positive). The energy cost for this period of travel is:

$$\int_X^s udx = m\int_0^{v_c} vdv = \frac{1}{2}mv_c^{\,2} \tag{4.39}$$

For the "off" section of travel, there is only time cost. We cruise at velocity v_c from position s to the origin, so the time cost is:

$$k\int_s^0 \frac{dx}{v_c} = -\frac{ks}{v_c} \tag{4.40}$$

By adding together the above three equations we get the optimal cost:

$$J^* = -kmv_c + \frac{1}{2}mv_c^2 - k\frac{s}{v_c} \tag{4.41}$$

We desire a result that minimizes $J^*(v_c)$:

$$\frac{dJ^*}{dv_c} = 0 = -km + mv_c - k\left(\frac{s'}{v_c} - \frac{s}{v_c^2}\right) \tag{4.42}$$

where we must use implicit differentiation, since s and v_c are related. We find that relation by integrating $udx = mvdv$:

$$\int_X^s dx = \int_0^{v_c} \frac{mv}{-1} dv \tag{4.43}$$

we find the relation:

$$s = X - \frac{1}{2}mv_c^2 \tag{4.44}$$

and the derivative of s with respect to v_c:

$$s' = -mv_c \tag{4.45}$$

which we substitute into (4.42) :

$$0 = -km + mv_c - k\left(\frac{-mv_c}{v_c} - \frac{s}{v_c^2}\right) \tag{4.46}$$

to find the switching curve

$$s = -\frac{m}{k} v_c^{\,3} \tag{4.47}$$

This equation is plotted in Figure 4.9, where we compare it to dynamic programming results:

4.6 Powered Pendulum

As stated in the beginning of Chapter 3, between foot collisions, the simple pendulum provides a succinct model of the behavior of the rimless wheel, and as we saw in Section 4.3 on dynamic programming, there is evidence that powering the rimless wheel, i.e., the powered pendulum, has a switching curve. Proceeding along the lines of Section 4.4 on the rocket car, we will now examine the switching curve of the powered pendulum shown in Figure 4.10. *Required:* Drive the pendulum from $\pi - a$ to $\pi + a$ with maximum speed, while minimizing the energy expenditure. In other words, we would like to minimize the integral cost $J = \int_0^T \phi(\mathbf{x}, u)\, dt$ where $\mathbf{x}$ is the state vector and ϕ is the cost function. As before, J is broken up into two parts: time cost and energy cost:

$$J = \int dt + \int \frac{dE}{dt} dt$$

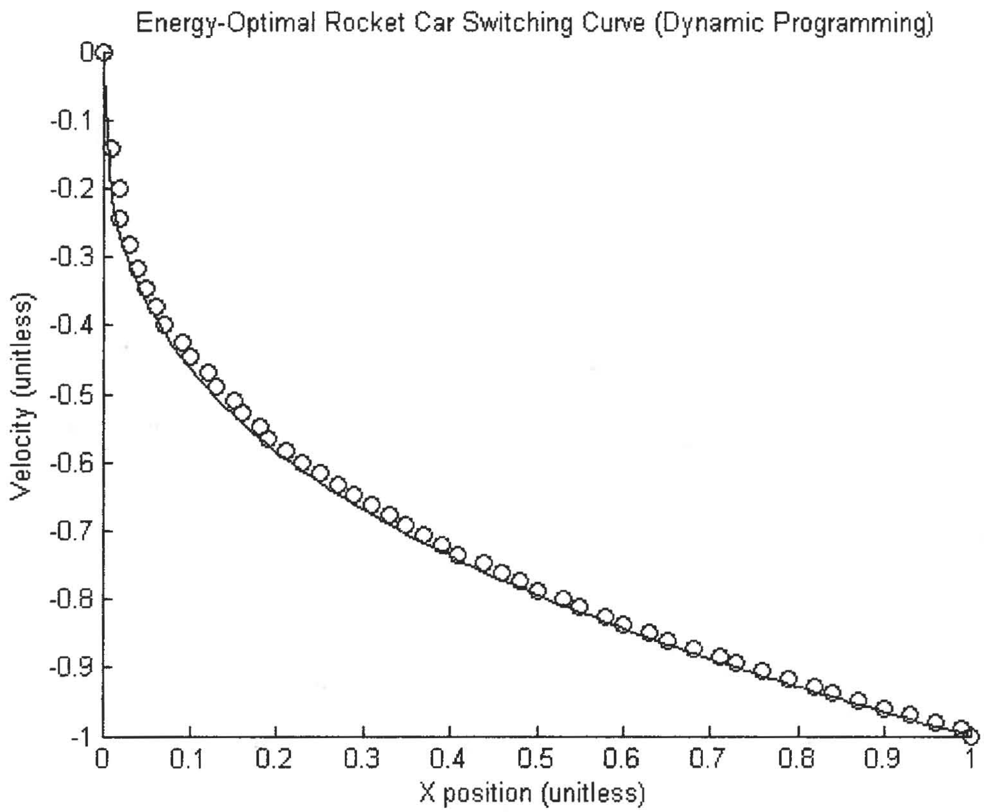


Figure 4.9: Switching Curve for the Rocket Car Problem

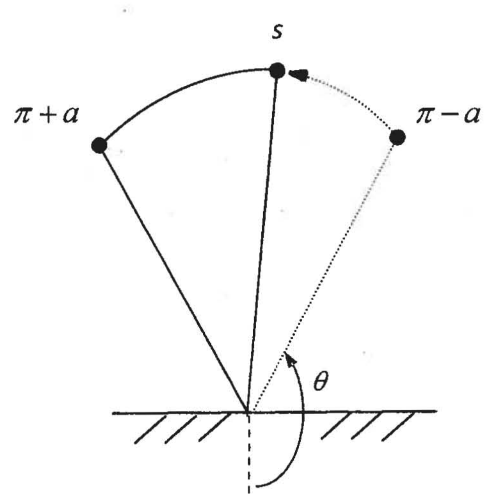


Figure 4.10: Optimal "on-off" control for the powered pendulum

Substituting (4.2) into the above, with a positive constant k as a measure of the relative importance between time and energy cost :

$$J = \int_0^T \left(k + u\dot{\theta}\right) dt \tag{4.48}$$

1[st] step in the Poyntryagin Maximum Principle: Treat the cost J as an additional state (Hocking, 1991)

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} J \\ \theta \\ \dot{\theta} \end{pmatrix}$$

The state equations are thus given by:

$$\begin{pmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} k + u\dot{\theta} \\ \dot{\theta} \\ u - \sin\theta \end{pmatrix} = \begin{pmatrix} k + ux_2 \\ x_2 \\ u - \sin x_1 \end{pmatrix} \tag{4.49}$$

Since we want to reach our goal with maximum velocity, the control is bounded by zero:

$$0 \le u \le 1$$

In other words, you may want to back off the gas, but you would never want to apply the brakes.

2[nd] step in the Poyntryagin Maximum Principle: We introduce the co-state variables to form the Hamiltonian:

$$H = z_0\dot{x}_0 + z_1\dot{x}_1 + z_2\dot{x}_2$$

Substituting (4.49) we see that:

$$H = z_0 (k + u x_2) + z_1 x_2 + z_2 (u - \sin x_1) \tag{4.50}$$

The co-state equations are prescribed by Hamilton's equations:

$$\dot{z}_0 = -\frac{\partial H}{\partial x_0} = 0 \tag{4.51}$$

$$\dot{z}_1 = -\frac{\partial H}{\partial x_1} = z_2 \cos x_1 \tag{4.52}$$

$$\dot{z}_2 = -\frac{\partial H}{\partial x_2} = -z_1 - z_0 u \tag{4.53}$$

3rd step in the Poyntryagin Maximum Principle

Solve the above co-state equations. The equation for $\dot{z}_0$ shows that $z_0 = const$, and the PMP requires that this constant should be *negative*. Without loss of generality we can choose $z_0 = -1$. The solution to (4.52) is coupled with(4.53), and without solving both equations, we cannot guarantee that the switching function is unique.

4th step in the Poyntryagin Maximum Principle: Find the supremum of H as a function of u.

$$H^* \geq H$$

$$-\left(k + \dot{\theta} u^*\right) + z_1 \dot{\theta} + z_2 \left(u^* - \sin\theta\right) \geq -\left(k + \dot{\theta} u\right) + z_1 \dot{\theta} + z_2 (u - \sin\theta)$$

$$u^* \left(z_2 - \dot{\theta}\right) \geq u \left(z_2 - \dot{\theta}\right)$$

To maximize H, we note the sign of term in parenthesis. If it is positive, we multiply it by the maximum positive control torque. If it is negative we multiply it by the minimum control torque:

$$u = \begin{vmatrix} 1 & if & z_2 - \dot{\theta} > 0 \\ 0 & if & z_2 - \dot{\theta} < 0 \end{vmatrix} \tag{4.54}$$

PMP has thus shown that we have "on-off" control switch. This again makes intuitive sense, since we would want to apply the control torque early on in order to gain the time savings accrued throughout the stride, but we would want to shut the torque off after some time period in order to save energy.

We desire to find the optimal control u^*. From the get go, we assume that the optimal solution is "on-off" as described in the previous section. First, we proceed with a bang, i.e., from the initial position $\pi - a$ to the switching position s we apply full torque $(u = 1)$, and then at s we cut the engines off $(u = 0)$ and coast through the $\pi + a$ as shown in Figure 4.11.

Here's the key: we can find the switching position s by re-writing the cost integral (4.48) as a function of θ instead of t:

$$J(s) = \int_0^T \left(k + u\dot{\theta}\right) dt = k \int_{\pi-a}^{s} \frac{d\theta}{\dot{\theta}} + \int_{\pi-a}^{s} u d\theta + k \int_{s}^{\pi+a} \frac{d\theta}{\dot{\theta}} \tag{4.55}$$

The first term is the time cost before the switch, the second term is the energy cost before the switch using the definition of work of a couple. $E = \int u d\theta$. The third term is the time cost after the switch, and there is no energy cost after the switch.

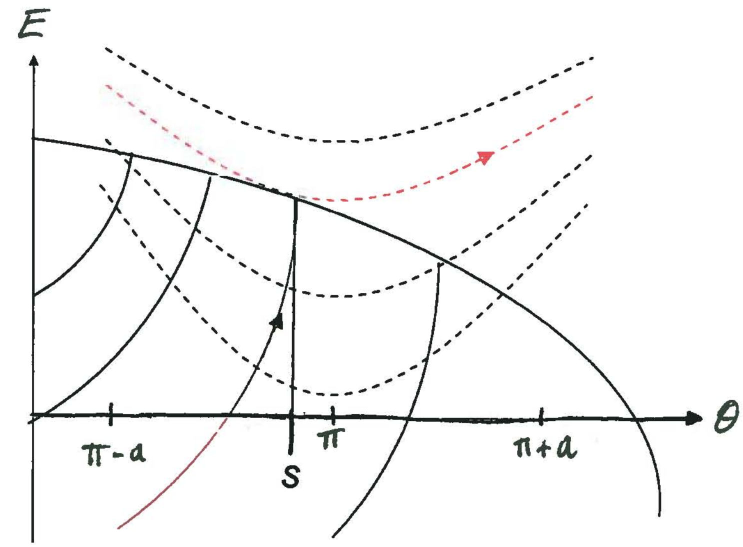


Figure 4.11: The Switching Curve for the Powered Pendulum

To determine $\dot{\theta}$, we must find an equation for the energy, $E(\theta)$, as a function of θ. At the initial position $\theta = \pi - a$, our initial energy is E_0, and we add energy according to the definition of work above until we reach s:

$$E(\theta) = E_0 + \int_{\pi-a}^{\theta} u d\theta = E_0 + \theta - \pi + a \quad \text{for} \quad \pi - a < \theta < s \tag{4.56}$$

We substitute this result into equation (3.5), $\dot{\theta} = \sqrt{2}\sqrt{E + \cos\theta}$:

$$\dot{\theta} = \sqrt{2}\sqrt{E_0 + \theta - \pi + a + \cos\theta} \tag{4.57}$$

Plugging this into (4.55), we find that

$$J(s) = \frac{k}{\sqrt{2}} \int_{\pi-a}^{s} \frac{d\theta}{\sqrt{E_0 + \theta - \pi + a + \cos\theta}} + s - \pi + a + \ldots$$
$$\ldots + \frac{k}{\sqrt{2}} \int_{s}^{\pi+a} \frac{d\theta}{\sqrt{E_0 + s - \pi + a + \cos\theta}} \tag{4.58}$$

To find the minimum of $J(s)$, we take the derivative with respect to s, and set it equal to zero: $\frac{dJ}{ds} = 0 = \ldots$ to find this derivative, we must use Leibniz' Rule for differentiating through the integral sign:

$$\frac{d}{ds} \int_{a(s)}^{b(s)} g(\theta,s)d\theta = g\left[b(s),s\right]b'(s) - g\left[a(s),s\right]a'(s) + \int_{a(s)}^{b(s)} \frac{\partial g(\theta,s)}{\partial s} d\theta \tag{4.59}$$

Using Leibniz Rule, $\frac{dJ}{ds} = 0$ becomes:

$$\cancel{\frac{k}{\sqrt{2}} \frac{1}{\sqrt{E_0 + s - \pi + a + \cos s}}} + 1 - \cancel{\frac{k}{\sqrt{2}} \frac{1}{\sqrt{E_0 + s - \pi + a + \cos s}}} + \ldots$$
$$\ldots + \frac{k}{2\sqrt{2}} \int_{s}^{\pi+a} \left(E_0 + s - \pi + a + \cos\theta\right)^{-3/2} d\theta = 0$$

Thus, we find that the switching curve must satisfy the integral equation:

$$\frac{k}{2\sqrt{2}} \int_{s}^{\pi+a} \left(E_0 + s - \pi + a + \cos\theta\right)^{-3/2} d\theta = 1 \tag{4.60}$$

To graph this on a phase plot, as shown in Figure 4.12, we find s that satisfies the above relation, and then we plug it into (4.56) then (4.57) to find the $\dot{\theta}$ that corresponds to the switching curve, $s(\theta,\dot{\theta})$. If we just used E_0 to find $\dot{\theta}$, we would not take into account the gain in energy that occurs during the powered phase before the switch.

In this chapter, we have found a switching curve for the powered rimless wheel, enabling the control of a rimless wheel walker to efficiently and quickly move from $\pi - a$ to $\pi + a$ during the stance phase. Figure 4.12 assumes a given step width, but this control is optimal no matter where you are in phase space. If you happen to move a little off the optimum path, you merely look up your position on the map and either power or coast depending on which side of the swithing curve you are on. In the next chapter, we will extend these concepts to a multiple link walking model where analytic methods do not suffice, and numerical methods must be used.

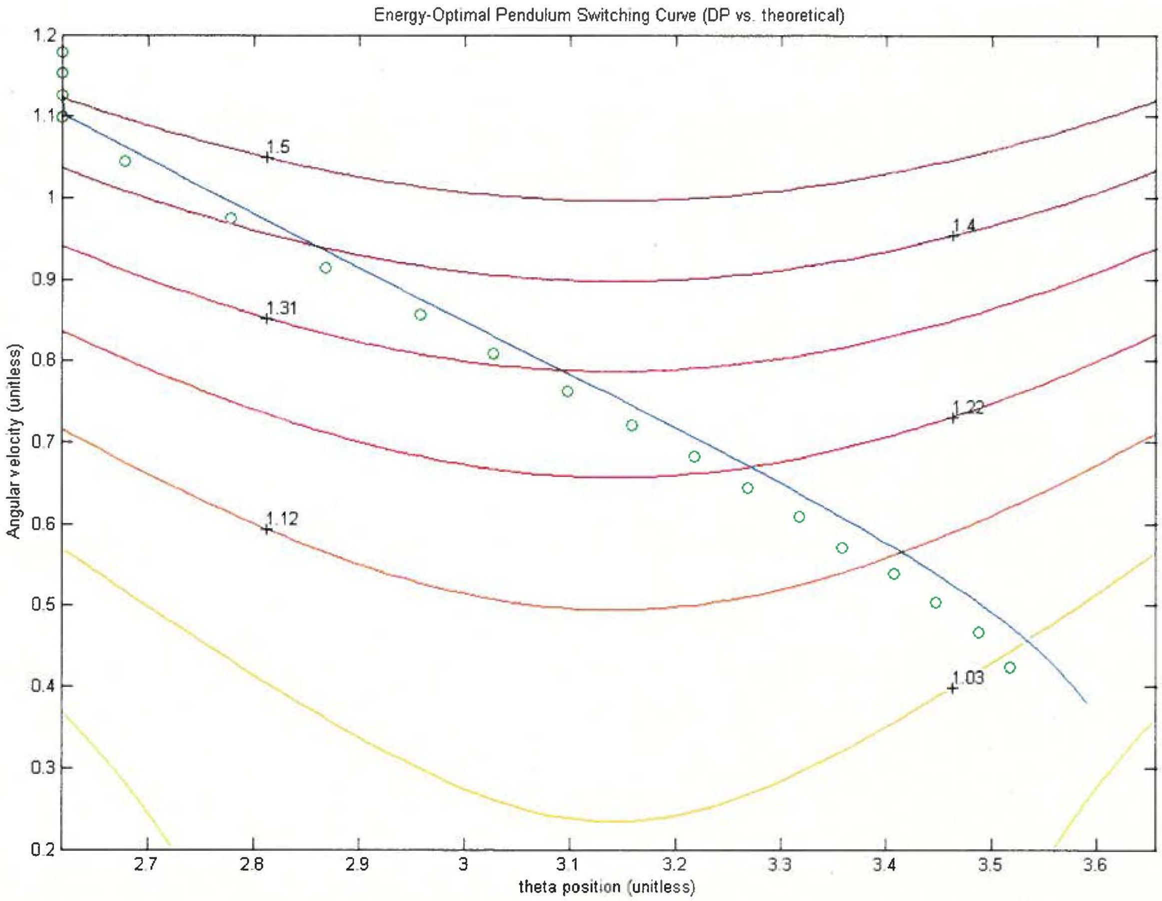


Figure 4.12: Switching Curve, Equation (4.60) is shown as a solid line, compared with dynamic programming results (shown in circles) (*This figure is presented in color; the black and white reproduction may not be an accurate representation*)

CHAPTER 5

TWO-LINK MODEL OF WALKING

5.1 Double Pendulum

To complete our study of walking behavior, we will examine the extension of the principles outlined in Chapters 1 through 4 to more complicated models of walking. While we have used simple models to grasp the underlying principles of walking, clearly, in the design of a practical walking machine, we will need a more complete model of the physical hardware to validate our ideas. We seek to verify the extension of these principles into more complete models of the dynamics.

In this chapter, we study a two-link model of walking—the double pendulum. This model is called the compass gait in the literature (Garcia, Chatterjee, Ruina, & Coleman, 1998). Again, we describe the swing phase of the gait before foot collision. Looking at Figure 5.1, we can derive the equations of motion using the Denavit-Hartenberg (DH) convention (J. Denavit, 1955). The new x-axis must be perpendicular to (DH1) and intersect (DH2), the old z-axis. For planar systems, DH2 suffices. This convention defines the homogeneous transform (translation plus rotation) from coordinate frame (x_0, y_0) to (x_1, y_1) as follows following (Mark W. Spong, 1989):

$$\mathbf{A}_0^1 = \begin{bmatrix} c_1 & -s_1 & a_1 c_1 \\ s_1 & c_1 & a_1 s_1 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

The superscript 1 on $\mathbf{A}_0^1$ refers to coordinate frame 1 and the subscript 0 refers to frame 0. We are using the shorthand notation $c_1 = \cos(\theta_1)$ and $s_1 = \sin(\theta_1)$. The link lengths are a_1 and a_2 respectively. This notation has the form: $\mathbf{A}_0^1 = \begin{bmatrix} \mathbf{R}_0^1 & \mathbf{d}_0^1 \\ \mathbf{0} & 1 \end{bmatrix}$ where $\mathbf{R}_{2\times2}$ is the rotation matrix and we note that $\mathbf{R}_0^1 = R_{z,\theta_1}$ is a rotation about the z-axis; $\mathbf{d}_0^1 = \begin{pmatrix} x_0^1 \\ y_0^1 \end{pmatrix}$ is the position vector; and $\mathbf{0}_{1\times2}$ is the zero vector—the last row of the matrix $\mathbf{A}_0^1$—it is for computational compatibility only. In our case, we are limited to the plane, so **A** is a 3x3 matrix instead of a 4x4 matrix. The homogeneous transform from (x_1, y_1) to (x_2, y_2) looks similar to equation (5.1):

$$\mathbf{A}_1^2 = \begin{bmatrix} c_2 & -s_2 & a_2c_2 \\ s_2 & c_2 & a_2s_2 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

We must matrix multiply the transforms together to get the composite transform from (x_0, y_0) to (x_2, y_2):

$$\mathbf{A}_0^2 = \mathbf{A}_0^1\mathbf{A}_1^2 = \begin{bmatrix} c_{12} & -s_{12} & a_2c_{12} + a_1c_1 \\ s_{12} & c_{12} & a_2s_{12} + a_1s_1 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

Here we are using a shorthand notation for two trigonometric identities:

$$\begin{aligned} c_{12} &= \cos(\theta_1 + \theta_2) = c_1c_2 - s_1s_2 \\ s_{12} &= \sin(\theta_1 + \theta_2) = s_1c_2 + c_1s_2 \end{aligned} \tag{5.4}$$

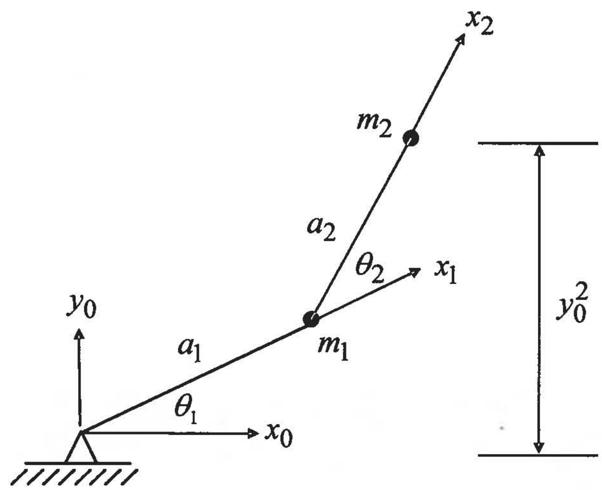


Figure 5.1: The DH Double Pendulum

To find the equations of motion for this system, we will use Lagrange's equation (Goldstein, 1950):

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right)-\frac{\partial L}{\partial \theta_i}=u_i \qquad \text{for} \qquad i=1,2 \tag{5.5}$$

The Lagrangian is defined as, $L=T-V$ where T is the kinetic energy and V is the potential energy and u is the torque about each respective joint. The kinetic energy is given by

$$T=\frac{1}{2}m_1 v_1^{\,2}+\frac{1}{2}m_2 v_2^{\,2} \tag{5.6}$$

where v_i is the velocity of mass *i*. Finding the velocities requires a bit of work. The velocity of frame 1 with respect to frame 0 is found by taking the time derivative of $\mathbf{d}_0^1 = \begin{pmatrix} a_1c_1 \\ a_1s_1 \end{pmatrix}$ as follows:

$$ {}_0^1 = \dot{\mathbf{d}}_0^1 = \begin{pmatrix} \dot{x}_1 \\ \dot{y}_1 \end{pmatrix} = \begin{pmatrix} -a_1s_1\dot{\theta}_1 \\ a_1c_1\dot{\theta}_1 \end{pmatrix} \tag{5.7} $$

Speed is the vector length of velocity given by: $v^2 = \dot{x}^2 + \dot{y}^2$. To find v_1^2 in equation (5.6), we sum the squares of the terms in (5.7), seeing that:

$$ v_1^2 = a_1^2\left(s_1^2 + c_1^2\right)\dot{\theta}_1^2 = a_1^2\dot{\theta}_1^2 \tag{5.8} $$

This agrees with the result $v = r\omega$ from Physics. We go through a similar procedure to find the velocity of frame 2 with respect to frame 0: We note from (5.3) that $d_0^2 = \begin{pmatrix} x_0^2 \\ y_0^2 \end{pmatrix} = \begin{pmatrix} a_2c_{12} + a_1c_1 \\ a_2s_{12} + a_1s_1 \end{pmatrix}$ which we can see more clearly from Figure 5.2. As in (5.7), we find the vector relation for velocity of the second mass:

$$ {}_0^2 = \dot{\mathbf{d}}_0^2 = \begin{pmatrix} \dot{x}_2 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} -a_2s_{12}\dot{\theta}_{12} - a_1s_1\dot{\theta}_1 \\ a_2c_{12}\dot{\theta}_{12} + a_1c_1\dot{\theta}_1 \end{pmatrix} \tag{5.9} $$

where $\dot{\theta}_{12} = \dot{\theta}_1 + \dot{\theta}_2$, and squaring the terms above:

$$ \begin{aligned} \dot{x}_2^2 &= a_2^2 s_{12}^2\dot{\theta}_{12}^2 + 2a_1a_2s_1s_{12}\dot{\theta}_1\dot{\theta}_{12} + a_1^2s_1^2\dot{\theta}_1^2 \\ \dot{y}_2^2 &= a_2^2 c_{12}^2\dot{\theta}_{12}^2 + 2a_1a_2c_1c_{12}\dot{\theta}_1\dot{\theta}_{12} + a_1^2c_1^2\dot{\theta}_1^2 \end{aligned} \tag{5.10} $$

Sum the squares:

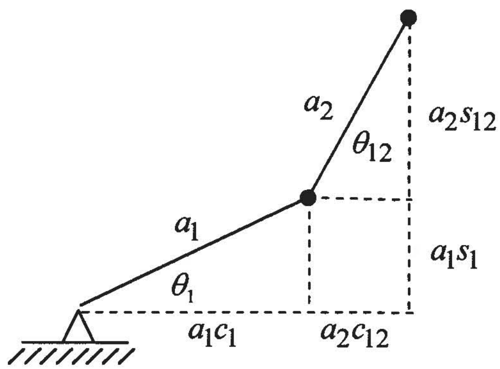


Figure 5.2: Geometric Relations

$$v_2^2 = x_2^2 + y_2^2 = a_1^2\dot{\theta}_1^2 + a_2^2\dot{\theta}_{12}^2 + 2a_1a_2c_2\dot{\theta}_1\dot{\theta}_{12} \tag{5.11}$$

where we have used the trigonometric identity:

$$\begin{aligned} s_1 s_{12} + c_1 c_{12} &= s_1\left(s_1c_2 + c_1s_2\right) + c_1\left(c_1c_2 - s_1s_2\right) \\ &= s_1^2 c_2 + \cancel{s_1c_1s_2} + c_1^2 c_2 - \cancel{s_1c_1s_2} \\ &= c_2 \end{aligned} \tag{5.12}$$

Substituting (5.11) and (5.8) into (5.6), the kinetic energy is:

$$T = \frac{1}{2}m_1a_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2\left[a_1^2\dot{\theta}_1^2 + a_2^2\dot{\theta}_{12}^2 + 2a_1a_2c_2\left(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2\right)\right] \tag{5.13}$$

The potential energy is:

$$\begin{aligned} V &= m_1 g y_0^1 + m_2 g y_0^2 \\ &= m_1 g\left(a_1s_1\right) + m_2 g\left(a_1s_1 + a_2s_{12}\right) \end{aligned} \tag{5.14}$$

We now find the Lagrangian:

$$L = \frac{1}{2}m_{12}a_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2a_2^2\dot{\theta}_{12}^2 + m_2a_1a_2c_2\left(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2\right) - m_{12}ga_1s_1 - m_2ga_2s_{12} \tag{5.15}$$

where $m_{12} = m_1 + m_2$. Taking its derivative with respect to the generalized coordinate θ_1:

$$\frac{\partial L}{\partial \theta_1} = -m_{12} g a_1 c_1 - m_2 g a_2 s_{12} \tag{5.16}$$

Take the derivative of L with respect to $\dot{\theta}_1$:

$$\frac{\partial L}{\partial \dot{\theta}_1} = m_{12} a_1^2 \dot{\theta}_1 + m_2 a_2^2 \dot{\theta}_{12} + m_2 a_1 a_2 c_2 \left(2\dot{\theta}_1 + \dot{\theta}_2\right) \tag{5.17}$$

Now take the time derivative of (5.17):

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = m_{12} a_1^2 \ddot{\theta}_1 + m_2 a_2^2 \ddot{\theta}_{12} + m_2 a_1 a_2 c_2 \left(2\ddot{\theta}_1 + \ddot{\theta}_2\right) - m_2 a_1 a_2 s_2 \left(2\dot{\theta}_1 + \dot{\theta}_2\right)\dot{\theta}_2 \tag{5.18}$$

Combining equations (5.18) and (5.16) we find the first equation of motion (5.5):

$$\begin{aligned}&\left[m_{12} a_1^2 + m_2 a_2^2 + 2 m_2 a_1 a_2 c_2\right]\ddot{\theta}_1 + \left[m_2 a_2^2 + m_2 a_1 a_2 c_2\right]\ddot{\theta}_2 \\ &+ \left[-m_2 a_1 a_2 s_2 \left(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2\right)\right] + \left[m_{12} g a_1 c_1 + m_2 g a_2 c_{12}\right] = u_1\end{aligned} \tag{5.19}$$

We can put the above into standard form:

$$M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + V_1 + G_1 = u_1 \tag{5.20}$$

where the terms in brackets in (5.19) are the respective terms above.

Now we must take the derivative of L with respect to the generalized coordinate θ_2:

$$\frac{\partial L}{\partial \theta_2} = -m_2 a_1 a_2 s_2 \left(\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2\right) - m_2 g a_2 c_{12} \tag{5.21}$$

Take the derivative of L with respect to $\dot{\theta}_2$ then the time derivative of that result:

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 a_2^2 \dot{\theta}_{12} + m_2 a_1 a_2 c_2 \dot{\theta}_1$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) = m_2 a_2^2 \ddot{\theta}_{12} + m_2 a_1 a_2 c_2 \ddot{\theta}_1 - m_2 a_1 a_2 s_2 \dot{\theta}_1 \dot{\theta}_2 \tag{5.22}$$

Combining equations (5.21) and (5.22) we find the second equation of motion (5.5):

$$\begin{aligned} &\left[m_2 a_2^2 + m_2 a_1 a_2 c_2 \right] \ddot{\theta}_1 + \left[m_2 a_2^2 \right] \ddot{\theta}_2 \\ &+ \left[m_2 a_1 a_2 s_2 \dot{\theta}_1^2 \right] + \left[m_2 g a_2 c_{12} \right] = u_2 \end{aligned} \tag{5.23}$$

Again, we put the above into the standard form:

$$M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + V_2 + G_2 = u_2 \tag{5.24}$$

where the terms in brackets in (5.23) are the respective terms above. Finally we can put the equations in matrix form

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} + \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} + \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{5.25}$$

5.2 Trajectory Optimization using PMP

As we see in Figure 5.3, we re-cast our equations in state form, where:

$$\mathbf{x} = \begin{bmatrix} J \\ \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{5.26}$$

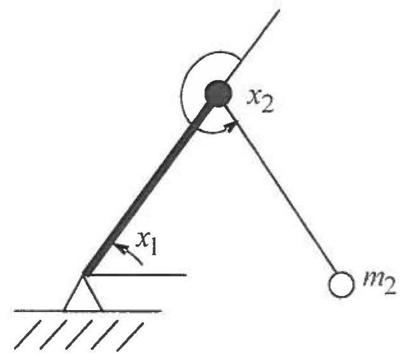


Figure 5.3: Trajectory Optimization Problem

Required: Drive the second mass back to the position $x_2 = x_{2d}$, while minimizing the total cost, J, where:

$$J = h(\mathbf{x}_f) + \int_0^T \phi(\mathbf{x}, \mathbf{u})dt$$

Substituting values for the final cost, h, and the integral cost, ϕ:

$$J = M\left(x_{2f} - x_{2df}\right)^2 + \int_0^T (k + u_1\dot{x}_1 + u_2\dot{x}_2)dt \tag{5.27}$$

In this problem, we chose the cost J to be for the time and energy-optimal case, with the relative importance between the two given by k. The penalty for error in the final state is given by h where M is a constant multiplier to weight the importance of reaching the desired final state, x_{2df}. We assume that we will reach $x_1 = \frac{\pi}{2} - a$ no matter what, so that the state x_1 is not included in the final cost, h.

1st step in the Poyntryagin Maximum Principle: How do we deal with this cost? Clearly it adds a new dimension to the problem. We'll treat cost as an *additional state*, i.e., $\mathbf{x}$ = (*cost, position, velocity*) is our new state vector as shown in Figure 5.4.

The state equations are now taking shape: our first state equation flows directly from the cost integral

$$\dot{x}_0 = \phi = k + u_1\dot{x}_1 + u_2\dot{x}_2 \tag{5.28}$$

Since the derivative of position is velocity, our second and third state equations are easy:

$$\begin{aligned} \dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \end{aligned} \tag{5.29}$$

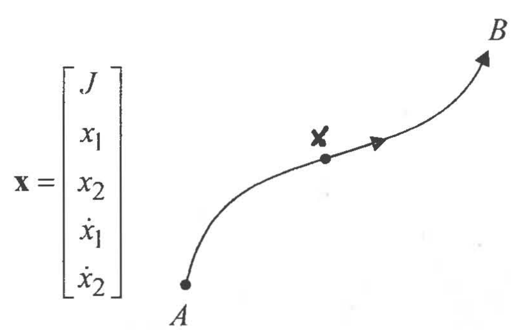


Figure 5.4: State Trajectory from initial to final state

The fourth and fifth state equations come from the double pendulum equations of motion. These equations as well as the co-state equations below are quite long; reproducing them here is rather uninstructive, so the co-state equations will be relegated to the appendix. Here are the equations of motion:

$$\ddot{\theta}_1 = \frac{\left\{\begin{array}{l} a_2^2 m_2[a_2(-g)m_2\cos(\theta_{12}) - a_1 g(m_{12})\cos(\theta_1) + \\ a_1 a_2 m_2(\dot{\theta}_2^2 + 2\dot{\theta}_1\dot{\theta}_2)\sin(\theta_2) + u_1] + \\ (a_2^2(-m_2) - a_1 a_2 m_2\cos(\theta_2))[-a_2 g m_2\cos(\theta_{12}) + \\ a_1 a_2 m_2 \dot{\theta}_1^{\,2}(-\sin(\theta_2)) + u_2] \end{array}\right\}}{a_1^2 a_2^2 m_2^2(-\cos^2(\theta_2)) + a_1^2 a_2^2 m_2^2 + a_1^2 a_2^2 m_1 m_2} \tag{5.30}$$

$$\ddot{\theta}_2 = \frac{\left\{\begin{array}{l} [2a_2 a_1 m_2\cos(\theta_2) + a_1^2(m_{12}) + a_2^2 m_2]\times \\ \left[-a_2 g m_2\cos(\theta_{12}) + a_1 a_2 m_2\dot{\theta}_1^{\,2}(-\sin(\theta_2)) + u_2\right] + \\ [a_2^2(-m_2) - a_1 a_2 m_2\cos(\theta_2)]\times \\ \left[\begin{array}{l} a_2(-g)m_2\cos(\theta_{12}) - a_1 g(m_{12})\cos(\theta_1) + \\ a_1 a_2 m_2(\dot{\theta}_2^{\,2} + 2\dot{\theta}_1\dot{\theta}_2)\sin(\theta_2) + u_1 \end{array}\right] \end{array}\right\}}{a_1^2 a_2^2 m_2^2(-\cos^2(\theta_2)) + a_1^2 a_2^2 m_2^2 + a_1^2 a_2^2 m_1 m_2} \tag{5.31}$$

The state equations are thus:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \phi \\ x_3 \\ x_4 \\ \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} \tag{5.32}$$

2^{nd} step in the Poyntryagin Maximum Principle: Introduce the co-state variables $\mathbf{z}^T = [-1 \quad z_1 \quad z_2 \quad z_3 \quad z_4]$ and form the Hamiltonian, $H = -\phi + \mathbf{z}^T\dot{\mathbf{x}}$, so we are maximizing the negative of a positive number, i.e., $\min a = \max(-a)$ if $a > 0$

$$H = z_0\dot{x}_0 + z_1\dot{x}_1 + z_2\dot{x}_2 + z_3\dot{x}_3 + z_4\dot{x}_4 \tag{5.33}$$

Substituting (5.33) and (5.28), we see that

$$H = -k - u_1x_3 - u_2x_4 + z_1x_3 + z_2x_4 + z_3\ddot{x}_1 + z_4\ddot{x}_2 \tag{5.34}$$

From this Hamiltonian (Hocking, 1991) the state and co-state equations are prescribed by Hamilton's equations:

$$\dot{z}_0 = -\frac{\partial H}{\partial x_0} = 0 \quad \Rightarrow \quad z_0 = -1$$

$$\dot{z}_1 = -\frac{\partial H}{\partial x_1}$$

$$\dot{z}_2 = -\frac{\partial H}{\partial x_2}$$

$$\dot{z}_3 = -\frac{\partial H}{\partial x_3}$$

$$\dot{z}_4 = -\frac{\partial H}{\partial x_4}$$

The other set of Hamilton's equations are the state equations:

$$\dot{x}_0 = \frac{\partial H}{\partial z_0} = \phi \qquad \dot{x}_1 = \frac{\partial H}{\partial z_1} = x_3 \qquad \dot{x}_2 = \frac{\partial H}{\partial z_2} = x_4 \quad \ldots$$

3rd step in the Poyntryagin Maximum Principle: Solve the co-state equations above. We cannot do this symbolically, so we will solve the state and co-state equations numerically after we incorporate the switching conditions on u_1 and u_2.

4th step in the Poyntryagin Maximum Principle: This is the crucial step in applying the PMP. We choose the control functions u_1 and u_2 to give the greatest possible value for the Hamiltonian H, considered as a function of u_1 and u_2 respectively. To perform this we take a partial difference, like a partial derivative or gradient $s_1 \equiv \dfrac{\Delta H}{\Delta u_1} = \dfrac{H_1^* - H}{u_1^* - u_1}$.

This is like the MIT rule, a.k.a. the delta rule: for maximizing, if the slope is positive, keep going in that direction; if it is negative, back up. Looking at only u_1 for now: If $H_1^* > H$ then $H_1^* - H > 0$ and we find a switching condition s_1 such that $\left(u_1^* - u_1\right)s_1 > 0$ so that if $s_1 > 0$ then $u_1^* \rightarrow \max$, and if $s_1 < 0$ the control $u_1^* \rightarrow \min$.

The control for u_1 is bounded by +1 and 0, so the supremum of H is attained when u_1 takes the value +1 when s_1 is positive and the value 0 when s_1 is negative.

$$u_1 = \begin{cases} +1 & for \quad s_1 > 0 \\ 0 & for \quad s_1 < 0 \end{cases} \tag{5.35}$$

Thus we assume "on-off" control, with the control switching between its extreme values when s_1 passes through zero. The same procedure applies for the second switching condition, s_2. Since these conditions are based on the values of the state and co-state, we will numerically solve the state and co-state equations, monitoring the value of the switching condition(s).

$$u_2 = \begin{cases} 0 & for \quad s_2 > 0 \\ -1 & for \quad s_2 < 0 \end{cases} \tag{5.36}$$

5.3 Boundary Conditions for PMP

The final value of the state and co-state are specified by the following formula for the boundary conditions (Kirk, 1970)

$$\left[\left.\frac{\partial h}{\partial \mathbf{x}}\right|_{t=T} - \mathbf{z}(T)\right]^T \delta \mathbf{x}_f + \left[H(T) + \left.\frac{\partial h}{\partial t}\right|_{t=T}\right]\delta T = 0 \qquad (5.37)$$

In our problem, the final time is free, so δT is arbitrary, and therefore its coefficient in brackets must go to zero:

$$\begin{aligned} &H^*(T) + \left.\frac{\partial h}{\partial t}\right|_{t=T} = 0 \\ &H^*(T) + \left.\frac{d}{dt}\left(M\left(x_2^* - x_{2desired}\right)^2\right)\right|_{t=T} = 0 \\ &H^*(T) + 2M\left(x_2^*(T) - x_{2desired}\right)\dot{x}_2^*(T) = 0 \end{aligned} \qquad (5.38)$$

The final state in our problem is mixed between the fixed state: $(\theta_1) \rightarrow (x_1)$ and free states $(\theta_2, \dot{\theta}_1, \dot{\theta}_2) \rightarrow (x_2, x_3, x_4)$: For the fixed final state $\delta x_1 = 0$ and the final value of the co-state z_1 is arbitrary. For the free final states, x_2 , x_3 and x_4 , the variation δx_i is arbitrary, and therefore the coefficient of δx_i in (5.37) must go to zero:

$$\begin{aligned} &\left.\frac{\partial h}{\partial x_2}\right|_{t=T} - z_3(T) = 0 \\ &\left.\frac{\partial h}{\partial x_3}\right|_{t=T} - z_3(T) = 0 \\ &\left.\frac{\partial h}{\partial x_4}\right|_{t=T} - z_4(T) = 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} &2M\left(x_2^*(T) - x_{2desired}\right) - z_2^*(T) = 0 \\ &z_3^*(T) = 0 \\ &z_4^*(T) = 0 \end{aligned} \qquad (5.39)$$

h is only dependent on x_2. This tells us that the final values for the co-states z_3 and z_4 must go to zero at the final time.

Taken together, the boundary conditions are

$$\begin{aligned}
&H^*(T)+2M\left(x_2^*(T)-x_{2desired}\right)\dot{x}_2(T)=0 \\
&2M\left(x_2^*(T)-x_{2desired}\right)-z_2^*(T)=0 \\
&z_3^*(T)=0 \\
&z_4^*(T)=0
\end{aligned} \tag{5.40}$$

The trouble with these boundary conditions is that they tell us what the final values should be, not what the initial values should be. Therefore, to perform numeric integration forwards, we must guess at the initial values of the co-states (fortunately, we already know the initial values of the states). After numeric integration, we compare the final values to those given in (5.40). This error is the metric that we will minimize using the Nelder Mead algorithm.

5.4 Nelder Mead Algorithm

Nelder and Mead's simplex algorithm (Nelder & Mead, 1965) is a popular, time-proven technique for optimizing general (nonsmooth) multivariable functions. Optimization algorithms of this type are called direct methods as opposed to gradient methods that rely on derivatives and, hence, smoothness. They are robust and general and applicable to the problem of gait determination by parameterizing the gait and defining a metric over these parameters for optimization.

For function $f(\bullet)$ defined on an n-dimensional domain space, Nelder and Mead's method finds a minimum using a simplex formed with $n+1$ points. The process of finding a minimum proceeds through a process of transforming the simplex by 1) shifting points in the direction of a minimum, 2) shrinking, and 3) expanding. In addition, a restarting criterion can be used. This restarting criterion uses the concept of a simplex gradient. The Nelder-Mead algorithm continually updates a simplex in the *N*-dimensional space of the problem. For *N*=3, this simplex is a tetrahedron as illustrated in Figure 5.4. There are *N*+1 vertices and *N*+1 corresponding function values.

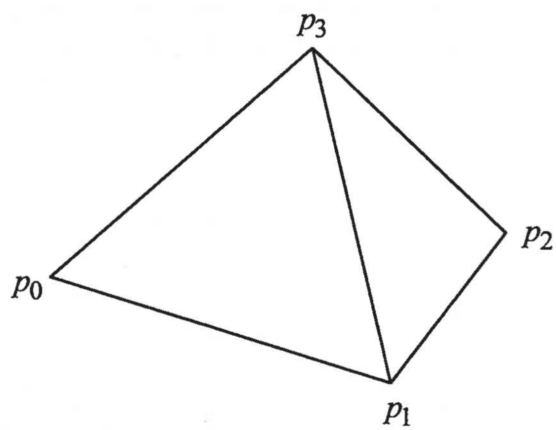


Figure 5.5: Nelder-Mead uses function evaluation over the vertices of an *N*-dimensional simplex to optimize a criterion function over an *N*-dimensional space. Shown here is a tetrahedron, which is a simplex for *N*=3. There are *N*+1 vertices in an *N*-dimensional simplex.

The Nelder Mead Algorithm consists of the following steps:

1) Order the points from lowest cost to greatest cost.
2) Calculate the values of each simplex point.
3) Try a new point to replace the worst point. The algorithm attempts each step below in order until the cost of the proposed point is less than that of the worst point:
   - Reflection – reflects worst point about the center of the simplex;
   - Expansion – expands in the direction of the best point;
   - Contraction – shifts worst point towards the best point;
   - Reduction – if no other options are available, each simplex point except the best is shrunk towards the center;
4) Accept the point once a step successfully reduces the cost.
5) Repeat process until a pre-specified tolerance is met.

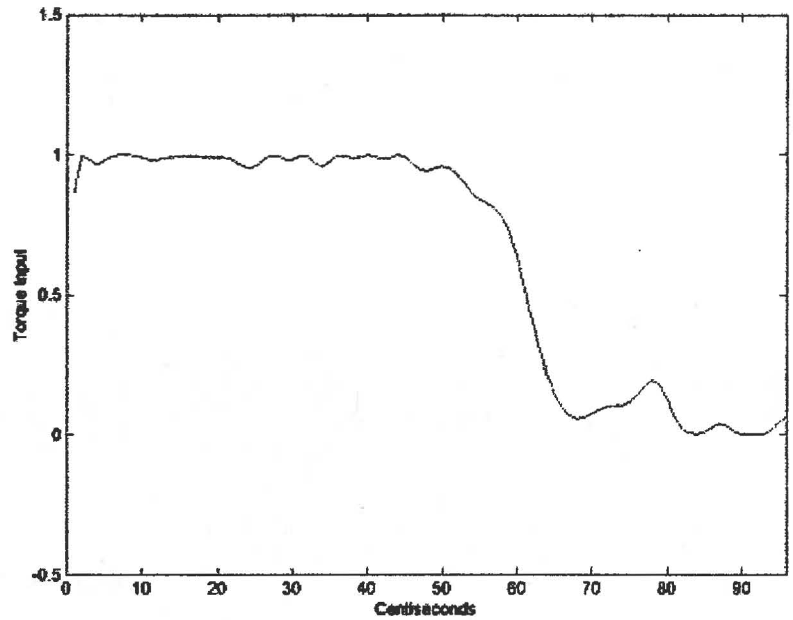
Using Nelder Mead, in Figure 5.5 (a), we minimize the functional, J for the powered rimless wheel of Chapter 4. Since $u(t)$ can take on any shape, we describe it using a cubic spline. Splines give a continuous and continuously differentiable function that is well behaved and practical to implement. These spline values are equally distributed over time (20-40 points). We feed the spline points into Nelder Mead and optimize them to minimize the cost functional, J. As we saw in previous results using Pontryagin's Maximum Principle, the correct answer to this problem at $k = 5$ is a switching curve at $t = 0.6$s with a cost of 5.34. Nelder-Mead confirms this result.

This result was found after adjusting the initial simplex parameters and adding a sufficient number of components. One point to take away: when using Nelder Mead, you must put an upper limit on the input torque: merely penalizing it with the square error of the torque amplitude will not suffice. In Figure 5.5 (b) we used Nelder-Mead to optimize the coefficients of a Fourier series (half-range expansion of the torque function) which found a similar result for the switching curve.

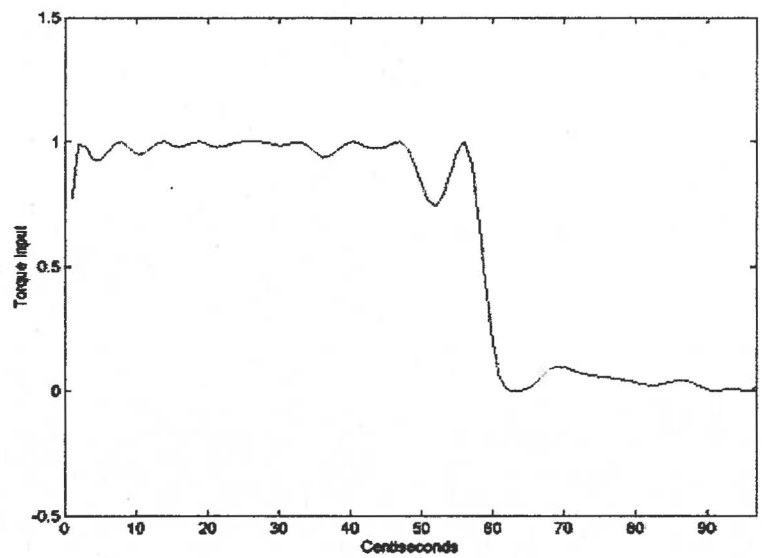
Finally, in Figure 5.6, we used Nelder Mead on spline points where we clipped the output between $0 < u(t) < 1$. The code for this example is shown in the appendix.

5.5 Nelder Mead Optimization and the Double Pendulum

Taking what we learned about optimizing the powered pendulum, we now tackle the powered double pendulum with Nelder Mead. We must stress here the importance of giving Nelder Mead several different starting values. Otherwise the algorithm can get caught in one of several local minima. In this section, we show an example of Nelder Mead optimization of the initial conditions on the co-state variables with a metric as outlined in equation (5.40). As an example, we set the masses equal to $m_1 = 1$ and $m_2 = 0.01$. The link lengths and gravity were both set to one and k was set to 5. The initial value for the joint positions was $\theta_1 = \frac{\pi}{2} - \frac{\pi}{6}$ and $\theta_2 = -\frac{\pi}{2} - \frac{\pi}{6}$. The initial conditions on the joint velocities were $\dot{\theta}_1 = 1$ and $\dot{\theta}_2 = -1$. This is meant to be just one example of a selection that could be used. With the high weight $M = 100$ of metric selection we used in equation (5.40), it effectively sought to drive the point-placement error to near zero. This metric selection was used with randomized Nelder-Mead

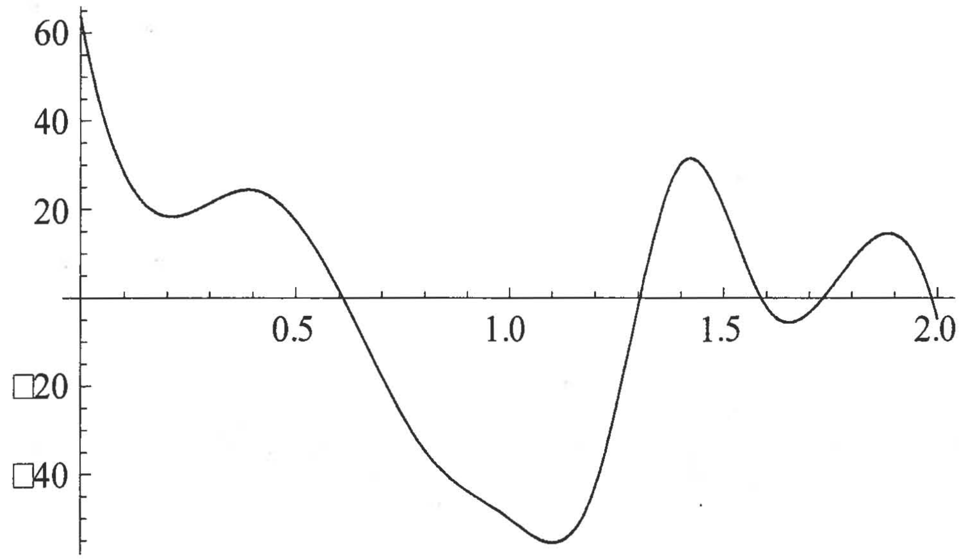


(a)

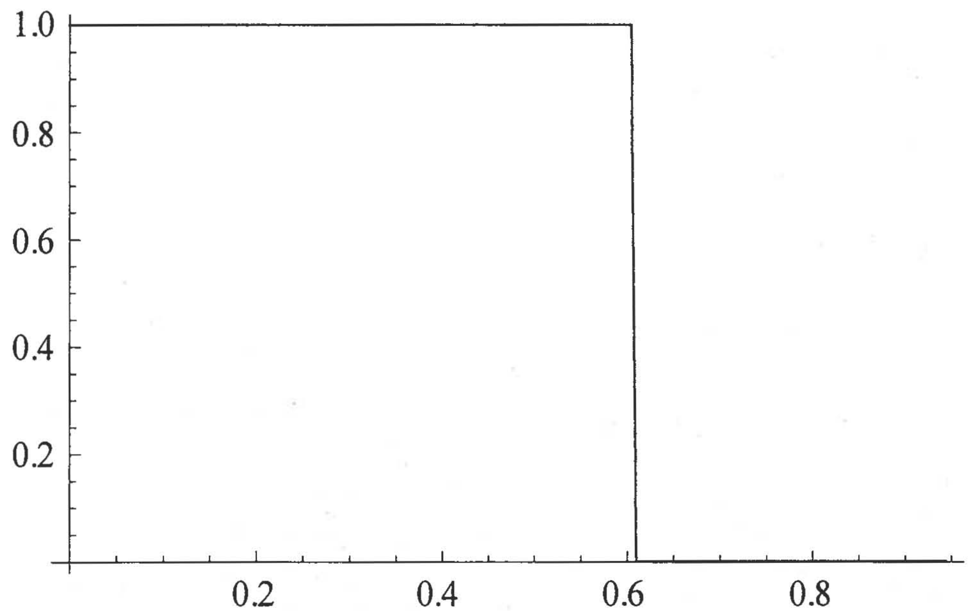


(b)

Figure 5.5: Nelder-Mead Optimization using (a) Fourier Series half range expansion and (b) Spline Components



(a)

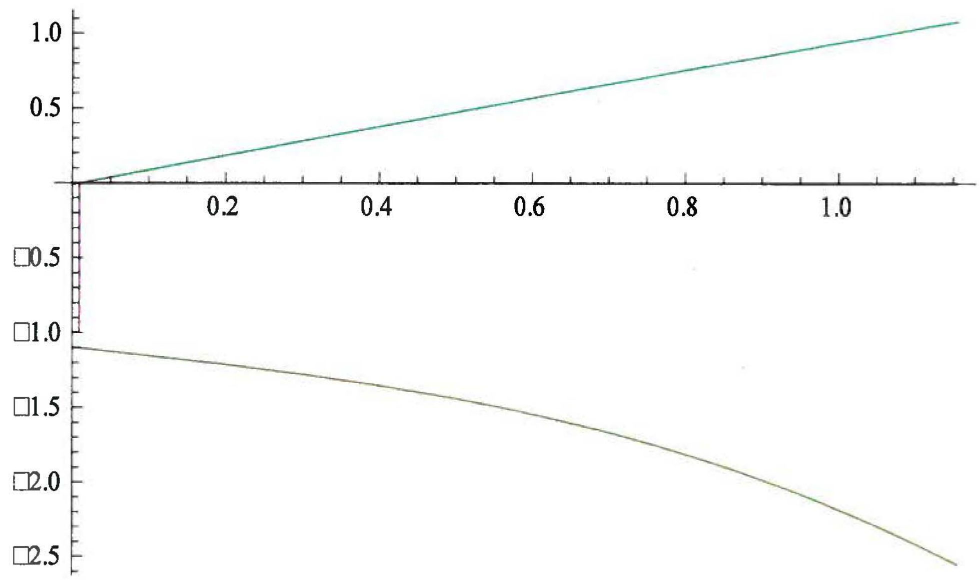


(b)

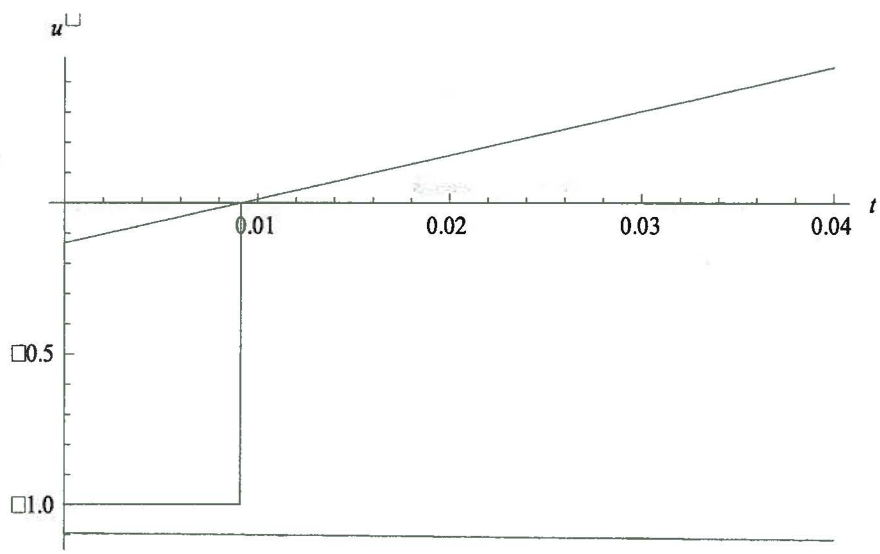
Figure 5.6: Nelder Mead for the powered pendulum showing (a) the raw spline curve (b) control torque after applying a hard clip $0 < u < 1$.

optimization to find an optimum stride by calculating 20 results and choosing the results with the smallest metric values. The results fell into a few local minima. The first had no control torque whatsoever, which resulted in a large metric (over 80K) and thus it was not considered further. The second local minimum actually had the best metric (1.407), meaning that it was the best fit to equation (5.40); however, it had a worse overall cost $J = 1.857$. The third minimum is what is presented here in Figure 5.7, with a metric of 2.804, but a cost of only $J = 0.064$.

This chapter has shown that we can extend the results of Chapters 3 and 4 to solve more complicated walking problems. Using the simple rimless wheel model helped us gain insight into problem, now as we move towards solving practical problems, we need more realistic models of walking. Increasing the degrees of freedom as shown here makes the problem more realistic, but we must be aware of the potholes along the way. As the dimension of the problem increases, local minimum can trap the Nelder-Mead optimization algorithm.



(a)



(b)

Figure 5.8: Optimal control torque u_2 shown in red and the switching conditions s_1 and s_2 shown in yellow and green for (a) the entire time period and (b) a close-up of the initial time. *(This figure is presented in color; the black and white reproduction may not be an accurate representation)*

CHAPTER 6

CONCLUSION

In this dissertation, we showed a control scheme for walking that decided 1) whether it made sense to walk or not—an energy regime for walking 2) the optimal stride angle to take when it made sense to walk and 3) an optimal switching curve: when we should power, and when we should coast. We used simple models of walking behavior: the passive rimless wheel and the powered rimless wheel, and we showed how this work could be extended to higher dimensional models of walking.

Several results from this work, most notably the optimal control using Pontryagin's Maximum Principle, required numeric methods to implement. In this dissertation, we showed how to use Nelder Mead optimization to reduce the number of variables for optimization down to the initial conditions of the co-state equations. This approach is highly parallelizable because you need different random starting conditions for Nelder-Mead.

For future work, we would like to compare these results to various classic descriptions of inverted pendulum walking (Alexander, 2003; Kuo, 2002; Kuo et al. 2005; Ruina et al. 2005)). These models incorporate nondimensional parameters that would be well suited to comparison here.

Another avenue for future work is to incorporate collision costs in these results. (Srinivasan 2006) shows that, for a simple model, inverted pendulum walking is preferred over all other possible gaits at low speed even when collision is included.

A key result of this thesis was the optimal switching curve in phase space for the powered rimless wheel. For this simple model, phase space was a plane, and the switching curve was a one-dimensional curve separating phase space into distinct regions (power and coast). What would this "switching curve" look like in higher dimensional systems? Even if the full "switching curve" could not be described analytically, could you describe it for a local region of phase space? This result would be useful to an off-road robot, where the initial conditions for the gait are constantly changing.

APPENDIX A

DYNAMIC PROGRAMMING CODE

```
%This program figures out the energy cost for a powered pendulum
%Where the goal is to reach a specific ANGLE
%Nodes are in terms of (Energy,theta) which is then used to find thetad
%All nodes are below the separatrix (cw rotations--whirling)

%we first build an energy contour map

delta_theta=-0.01;
delta_E=0.125;
n=4;%n+1 is the number of energy levels
%the walker is moving ccw (thetad is negative)
%We plot the different energy contours of the simple pendulum
[theta,E] = meshgrid(pi+0.5:delta_theta:pi-0.5, 1.01:delta_E:1.01+n*delta_E);

%thetad as a function of (Energy, theta)
thetad=-sqrt(2)*((E+cos(theta)).^0.5); %Strogatz p. 170

%and now we plot the energy contour map on theta,thetad state-space
%mesh(theta,thetad,E);
hold on;
v=[1.01 1.125 1.25 1.375 1.5]; %other Energies
[C,h]=contour(theta,thetad,E,v,'k-');
clabel(C,h,'manual'); %in-line manual labeling
xlabel('theta');
ylabel('thetad');
axis equal


%Now we need an energy and time cost matrix

%find the time cost per iteration
delta_t=delta_theta ./thetad;

%Goal is Energy level 5
%Start at Energy level 1



%Each angle has 5 energy levels (5 nodes)
%At any given time we have two choices:
%We can stay on the same energy contour (denoted by 2 in the tree)
%or move up to the next energy contour (denoted by 3 in the tree)
%The final option is where there is no path (denoted by 1 in the tree)
```

```
%First column is the time cost alone (no E input)
%Second column is the cost of energy input (0.25) + time cost

for i = 1:5*length(delta_t) %5 x column length
   Cost(i,1) = delta_t(i);
   Cost(i,2) = delta_E + delta_t(i);
end

%Every 5th node is at the max energy level
%NaN indicates a non-existant path
%NaN is ignored by the min() function

for i=1:length(delta_t)
   Cost(5*i,2)=NaN; %Max Energy level is 5
end



%t(i,j) from Cost(m,n)? i=m, j=m+n+4
%f(i)=min_j{t(i,j) + f(j)}  %see Denardo

f(501)=0; %The entire column of theta=pi-0.5 is the goal
f(502)=0; %hence we need five zero cost nodes
f(503)=0;
f(504)=0;
f(505)=0;
f(506)=NaN; %for computational compatibility
f(507)=NaN;
f(508)=NaN;
f(509)=NaN;
f(510)=NaN;


%Tree of Shortest Paths

clear Total_Cost;
for m = 500:-1:1   %decrement
   for n = 1:2  %there are two choices per node
      Total_Cost(m,n)=Cost(m,n)+f(m+n+4); %total cost (NaN + f = NaN)
   end
   f(m)=min(Total_Cost(m,:)); %need to update f
end
```

```
Na=NaN(500,1); %NaN is used to denote a nonexistant path
Total_Cost=cat(2,Na,Total_Cost); %concatenate Total Cost with a row of NaNs
[f2,k]=min(Total_Cost'); %there is a min path choice/node
k(505)=1;

%min path tree

for n=1:length(delta_t) %column length
   for m=1:5  %rows then columns
    K(m,n)=k(5*(n-1)+m); %5*(n-1) is the # of the cells already filled
   end
end

%Now, from the min path tree, I would like to build a vector map
%that shows the direction of the preferred flow for the powered pendulum
%We need to calculate v,u the rise and run of the vectors at each point
%the run, u, is easy it is simply delta_theta
%we must make a matrix of delta_thetas

u=ones(5,101)*delta_theta; %vector data for the run
for n=1:length(delta_t)
   for m=1:5
      if K(m,n)==3 %moving up to the next energy contour
         v(m,n)=thetad(m+1,n+1)-thetad(m,n);
      end
      if K(m,n)==2 %staying on the same energy contour
         v(m,n)=thetad(m,n+1)-thetad(m,n);
      end
      if K(m,n)==1 %no path
         u(m,n)=0;
         v(m,n)=0;
      end
   end
end

%plot the vector map--every fifth element so not too many arrows
%quiver(theta,thetad,u,v);
quiver(theta(:,1:5:101),thetad(:,1:5:101),u(:,1:5:101),v(:,1:5:101));
```

APPENDIX B

NELDER MEAD OPTIMIZATION OF THE POWERED RIMLESS WHEEL

(* This program uses Nelder Mead to optimize the torque profile to move a pendulum from pi-a to pi+a with free final time *)

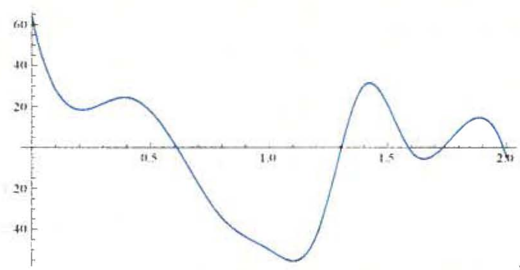
```
(* number of torque spline points *)
numPoints = 11;

(* torque values *)
torques=.;

(* time increment of the spline points *)
deltaT = 0.2;

(*time values *)
times = Table[deltaT*(ii-1),{ii,1,numPoints}]


{0,0.2,0.4,0.6,0.8,1.,1.2,1.4,1.6,1.8,2.}


(* define the input function as spline values over the data *)
dataTable[xTimes_,xTorques_]:=Table[{xTimes[[ii]],
xTorques[[ii]]},{ii,1,Length[xTorques]}];

(* u is the interpolation of the spline points *)
u[t_,xTimes_,xTorques_]:=Interpolation[dataTable[xTimes,
xTorques],Method->"Spline"][t];

(* u is the interpolation of the raw spline data that we feed into Nelder Mead *)
(* Clip will limit u between 0<u<1 *)

(* define the pendulum equation with CLIPPED u (t) between 0 and 1 *)
thetaDD[t_,xTimes_,xTorques_]:= -Sin[y[t]] + Clip[u[t,xTimes,xTorques],{0,1}];

(* define the metric to be minimized, i.e. J = E+kT *)
a = Pi/6;  (* half the step width *)
thetaFinal= Pi+a;
k=5; (* the relative importance of time, where 0<k<inf *)
```

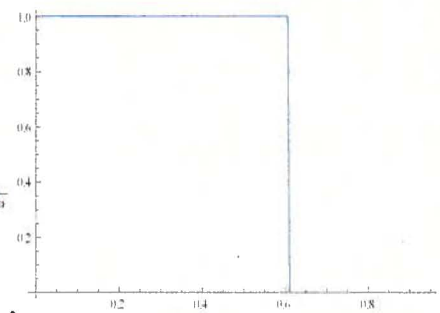
```
metricW[torques_?(VectorQ[#,NumericQ]&),times_,theta0_,
thetadot0_,tf_]:=
 Block[{},
       Off[NDSolve::"mxst"];
       (* msoln finds the solution for more time than we need, i.e., tf_ is our guess at
       the final time.  We then use findRoot to find where theta (tfinal) = thetaFinal.
       We then truncate the integration for the pathMetric between 0 and tfinal *)

       msoln:=NDSolve[{y"[t]==thetaDD[t,times,torques],y[0]==theta0,
       y'[0]==thetadot0},y,{t,0,tf}];

       (* find the final time where theta = thetaFinal, start the guess at t = 0.5 *)
       finalTime=FindRoot[Evaluate[y[t]/.msoln]-thetaFinal,{t,0.5}] ;

       (* finalTime is not a number, it is something like {t->2.33455} *)
       tfinal=t/.finalTime ;

       (* Here was the hard part of the program...finding the right pathMetric that
       Mathematica could handle. We wanted to use pathMetric=NIntegrate[u[t]*y'[t]],
       which took a while to figure out the right format *)
       (* After a lot of guessing, we finally found something that works *)
       (*  The following yielded a cost of 5.32899 with a pronounced switch at around
       t = 0.6, and tfinal = 0.944 *)

       pathMet=NIntegrate[Clip[u[t,times,torques],{0,1}]*
       Evaluate[y'[t]/.msoln],{t,0.001,tfinal-0.001}];

       (* when using Evaluate, we need to define pathMetric = pathMet[[1]], otherwise
       use pathMetric. pathMet is not a number, it is something like {4.32345} *)

       pathMetric=pathMet[[1]];
       (* u*thetadot where u is clipped b/t 0 and 1 *)
       (* this is the metric for Nelder Mead to evaluate *)
       (pathMetric+k*tfinal)
  ]; (* end of Block *)


(* perform Nelder Mead optimization on spline points *)

xx = {x0,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10};
initialPosition = Pi-Pi/6;
initialVelocity = 0.8;
 (* E0 = 1.186 *)
finalTimeGuess = 2; (* overshoot the final time to include underpowered guesses *)
```

```
sln=NMinimize[
  {metricW[xx,times,initialPosition,initialVelocity,finalTimeGuess]}, xx,
  Method->{"NelderMead","RandomSeed"->1}]
```

{5.32899,{x0->63.5233,x1->18.3641,x2->24.3306,x3->1.79457,x4->-34.5851,x5->-50.0206,x6->-42.955,x7->30.234,x8->-2.19082,x9->8.39748,x10->-4.77845}}

```
Row[Plot[u[t,times,xx/.sln[[2]]],{t,0,finalTimeGuess}],Plot[Clip[u[t,times,xx/.sln[[2]]],
{0,1}],{t,0,tfinal}]]
tfinal (* Clip will limit u between -1 and 1 *)
```

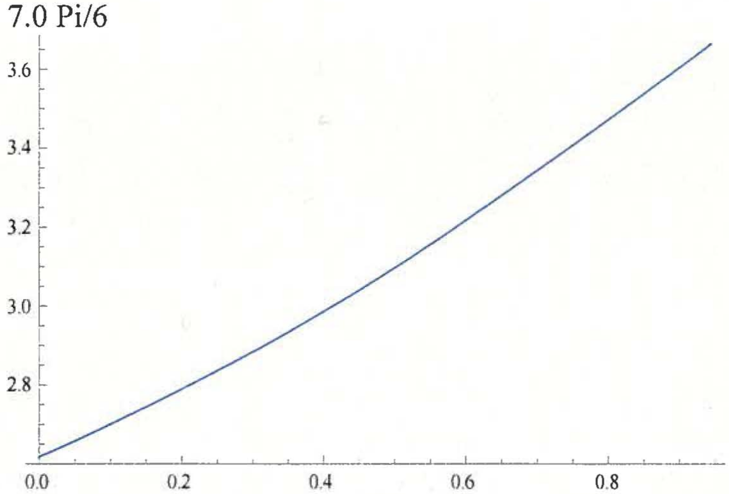




0.944315

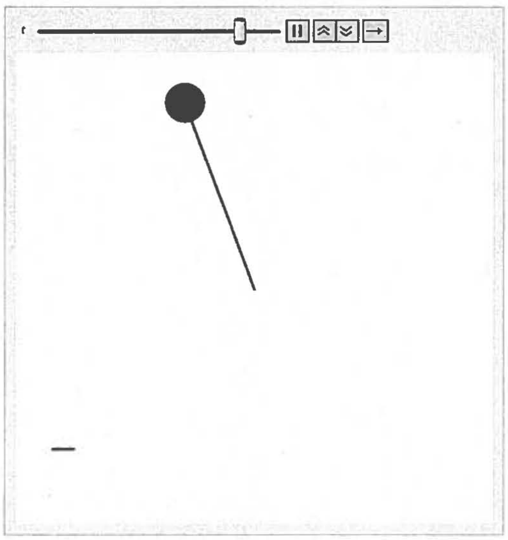
```
thetasoln=NDSolve[{y''[t]==thetaDD[t,times,xx/.sln[[2]]],
y[0]==initialPosition,y'[0]==initialVelocity},y,
{t,0,tfinal}];
Plot[Evaluate[y[t]]/.thetasoln,{t,0,tfinal}]
```

7.0 Pi/6



3.66519

```
gg[t_]:=Graphics[{
    {
     {Gray,Rectangle[{-.9,-.8},{-1,-.8 + Clip[u[t,times,xx/.sln[[2]]],{0,1}]}]},
     {Thick,Red,Line[{{-.9,-.8},{-1,-.8}}]},
     Thick,Line[{{0,0},{Sin[th],-Cos[th]}}]},
    Disk[{Sin[th],-Cos[th]},0.1]
    },PlotRange->{{-1.1,1.1},{-1.1,1.1}}]/.th->
Evaluate[y[t]]/.thetasoln[[1]];Animate[gg[t],{t,0,tfinal}]
```



APPENDIX C

STATE AND CO-STATE EQUATIONS FOR THE DOUBLE PENDULUM

(* Mathematica notebook file *)

$M11 = (m_1 + m_2)a_1 \wedge 2 + m_2 a_2 \wedge 2 + 2 m_2 a_1 a_2 Cos[\theta_2]$

$2 a_2 a_1 m_2 \cos(\theta_2) + a_1^2 (m_1 + m_2) + a_2^2 m_2$

$M12 = m_2 a_2 \wedge 2 + m_2 a_1 a_2 Cos[\theta_2]$

$a_1 a_2 m_2 \cos(\theta_2) + a_2^2 m_2$

$M21 = m_2 a_2 \wedge 2 + m_2 a_1 a_2 Cos[\theta_2]$

$a_1 a_2 m_2 \cos(\theta_2) + a_2^2 m_2$

$M22 = m_2 a_2 \wedge 2$

$a_2^2 m_2$

$MM = \{\{M11, M12\}, \{M21, M22\}\}$

$$\begin{matrix} (m_1 + m_2)a_1^2 + 2\cos(\theta_2) a_2 m_2 a_1 + a_2^2 m_2 & m_2 a_2^2 + \cos(\theta_2) a_1 m_2 a_2 \\ m_2 a_2^2 + \cos(\theta_2) a_1 m_2 a_2 & a_2^2 m_2 \end{matrix}$$

$MMI = Inverse[MM]$

$$\begin{matrix} \dfrac{a_2^2 m_2}{-\cos^2(\theta_2) a_1^2 m_2^2 a_2^2 + a_1^2 m_2^2 a_2^2 + a_1^2 m_1 m_2 a_2^2} & \dfrac{-m_2 a_2^2 - \cos(\theta_2) a_1 m_2 a_2}{-\cos^2(\theta_2) a_1^2 m_2^2 a_2^2 + a_1^2 m_2^2 a_2^2 + a_1^2 m_1 m_2 a_2^2} \\ \dfrac{-m_2 a_2^2 - \cos(\theta_2) a_1 m_2 a_2}{-\cos^2(\theta_2) a_1^2 m_2^2 a_2^2 + a_1^2 m_2^2 a_2^2 + a_1^2 m_1 m_2 a_2^2} & \dfrac{(m_1 + m_2)a_1^2 + 2\cos(\theta_2) a_2 m_2 a_1 + a_2^2 m_2}{-\cos^2(\theta_2) a_1^2 m_2^2 a_2^2 + a_1^2 m_2^2 a_2^2 + a_1^2 m_1 m_2 a_2^2} \end{matrix}$$

$V1 = -m_2 a_1 a_2 Sin[\theta_2](2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2 \wedge 2)$

$a_1 a_2 m_2 (\dot{\theta}_2^2 + 2\dot{\theta}_1 \dot{\theta}_2)(-\sin(\theta_2))$

$G1 = (m_1 + m_2) g a_1 Cos[\theta_1] + m_2 g a_2 Cos[\theta_1 + \theta_2]$

$$a_2 g m_2 \cos(\theta_1+\theta_2)+a_1 g(m_1+m_2)\cos(\theta_1)$$

$$V2 = m_2 a_1 a_2 Sin[\theta_2]\dot{\theta}_1\verb|^|2$$

$$a_1 a_2 m_2 \dot{\theta}_1^2 \sin(\theta_2)$$

$$G2 = m_2 g a_2 Cos[\theta_1+\theta_2]$$

$$a_2 g m_2 \cos(\theta_1+\theta_2)$$

(* Coriolis and Gravitational Torques *)

$$TT = \{V1+G1, V2+G2\}$$

$$\{a_2 g m_2 \cos(\theta_1+\theta_2)+a_1 g(m_1+m_2)\cos(\theta_1)-a_1 a_2 m_2(\dot{\theta}_2^2+2\dot{\theta}_1\dot{\theta}_2)\sin(\theta_2),$$
$$a_2 g m_2 \cos(\theta_1+\theta_2)+a_1 a_2 m_2 \dot{\theta}_1^2 \sin(\theta_2)\}$$

(* State Eqn M.qdd+V+G=tau
qdd=MMI.(tau-V-G) *)

$$thetaDD = MMI.(-TT+\{u_1,u_2\})$$

$$\frac{a_2^2 m_2(a_2(-g)m_2\cos(\theta_1+\theta_2)-a_1 g(m_1+m_2)\cos(\theta_1)}{a_1^2 a_2^2 m_2^2(-\cos^2(\theta_2))+a_1^2 a_2^2 m_2^2+a_1^2 a_2^2 m_1 m_2}+$$
$$\frac{(a_2^2(-m_2)-a_1 a_2 m_2\cos(\theta_2))(-a_2 g m_2\cos(\theta_1+\theta_2)+a_1 a_2 m_2\dot{\theta}_1^2(-\sin(\theta_2))+u_2)}{a_1^2 a_2^2 m_2^2(-\cos^2(\theta_2))+a_1^2 a_2^2 m_2^2+a_1^2 a_2^2 m_1 m_2}+$$
$$\frac{a_1 a_2 m_2(\dot{\theta}_2^2+2\dot{\theta}_1\dot{\theta}_2)\sin(\theta_2)+u_1)}{a_1^2 a_2^2 m_2^2(-\cos^2(\theta_2))+a_1^2 a_2^2 m_2^2+a_1^2 a_2^2 m_1 m_2}$$

$$\frac{(2a_2 a_1 m_2\cos(\theta_2)+a_1^2(m_1+m_2)+a_2^2 m_2)(-a_2 g m_2\cos(\theta_1+\theta_2)}{a_1^2 a_2^2 m_2^2(-\cos^2(\theta_2))+a_1^2 a_2^2 m_2^2+a_1^2 a_2^2 m_1 m_2}+$$
$$\frac{(a_2^2(-m_2)-a_1 a_2 m_2\cos(\theta_2))(a_2(-g)m_2\cos(\theta_1+\theta_2)-a_1 g(m_1+m_2)\cos(\theta_1)}{a_1^2 a_2^2 m_2^2(-\cos^2(\theta_2))+a_1^2 a_2^2 m_2^2+a_1^2 a_2^2 m_1 m_2}+$$
$$\frac{a_1 a_2 m_2(\dot{\theta}_2^2+2\dot{\theta}_1\dot{\theta}_2)\sin(\theta_2)+u_1)+a_1 a_2 m_2\dot{\theta}_1^2(-\sin(\theta_2))+u_2)}{a_1^2 a_2^2 m_2^2(-\cos^2(\theta_2))+a_1^2 a_2^2 m_2^2+a_1^2 a_2^2 m_1 m_2}$$

(* write thetaDD in state space form in preparation for symbolic calculation of the Hamiltonian *)

$$thetaDDS = thetaDD\,/.\,\{\theta_1 \to x_1, \theta_2 \to x_2, \dot{\theta}_1 \to x_3, \dot{\theta}_2 \to x_4\}$$

(*

The cost is given by : $J = (x_{final} - x_{desired})^2 + \int_0^T k + u_1\,\dot{\theta}_1 + u_2\,\dot{\theta}_2$ ât

thus $h = (x_2 - x_d)^2$ and

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} J \\ \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

The Hamiltonian is given by:

$$H = z_0\,\dot{x}_0 + z_1\,\dot{x}_1 + z_2\,\dot{x}_2 + z_3\,\dot{x}_3 + z_4\,\dot{x}_4$$

$$H = -\phi + z_1\,\dot{\theta}_1 + z_2\,\dot{\theta}_2 + z_3\,\ddot{\theta}_1 + z_4\,\ddot{\theta}_2$$

$$H = -k - u_1 x_3 - u_2 x_4 + z_1 x_3 + z_2 x_4 + z_3\,\dot{x}_3 + z_4\,\dot{x}_4$$

*)

$$H = -k - u_1 x_3 - u_2 x_4 + z_1 x_3 + z_2 x_4 + z_3 thetaDDS[[1]] + z_4 thetaDDS[[2]]$$

(* CoState Equations *)

$$z1dot = -D[H, x_1]$$
$$z2dot = -D[H, x_2]$$
$$z3dot = -D[H, x_3]$$
$$z4dot = -D[H, x_4]$$

(* above output suppressed for brevity *)

$H1^* = H/.\{u_1 \to u1^*\}$

$H2^* = H/.\{u_2 \to u2^*\}$

(*
switching condition is the PMP, i.e. the supremum for H
if $H^* > H$ then $H^* - H > 0$
and $(u^* - u)a > 0$
thus if $a > 0$
then $u^* \to \max$
and if $a < 0$
then $u^* \to \min$
*)

$$switch1 = Simplify[\frac{Simplify[H1^* - H]}{u1^* - u_1}]$$

$$\frac{-a_2 a_1^2 x_3 (m_2 \sin^2(x_2) + m_1) - a_1 z_4 \cos(x_2) + a_2(z_3 - z_4)}{a_1^2 a_2 (m_2 \sin^2(x_2) + m_1)}$$

$$switch2 = Simplify[\frac{Simplify[H2^* - H]}{u2^* - u_2}]$$

$$\frac{a_1^2((m_1 + m_2)z_4 - a_2^2 m_2 x_4 (m_2 \sin^2(x_2) + m_1)) - a_2 a_1 m_2 (z_3 - 2z_4)\cos(x_2) + \ldots \ldots + a_2^2 m_2 (z_4 - z_3)}{a_1^2 a_2^2 m_2 (m_2 \sin^2(x_2) + m_1)}$$

(*
the switching condition given above defines where the control torque u_1 switches. We now define the switching condition in terms of numerical values in preparation for numerical integration of the state and costate eqns. If you change numeric values, don't forget to change:
1) switch condition s[t_] here
2) stateDDN and costateDDN
3) graphics
*)

$HN := H/.\{x_1 \to x1[t], x_2 \to x2[t], x_3 \to x1'[t], x_4 \to x2'[t],$
$z_1 \to z1[t], z_2 \to z2[t], z_3 \to z3[t], z_4 \to z4[t],$

$k \to 5,$

$a_1 \to 1,$

$a_2 \to 1,$

$m_1 \to 1,$

$m_2 \to 0.01,$

$g \to 1,$

$u_1 \to u1star,$

$u_2 \to u2star\}$

$s1 = switch1 /. \{x_1 \to x1[t], x_2 \to x2[t], x_3 \to x1'[t], x_4 \to x2'[t],$

$z_1 \to z1[t], z_2 \to z2[t], z_3 \to z3[t], z_4 \to z4[t],$

$a_1 \to 1,$

$a_2 \to 1,$

$m_1 \to 1,$

$m_2 \to 0.01,$

$g \to 1\}$

$s2 = switch2 /. \{x_1 \to x1[t], x_2 \to x2[t], x_3 \to x1'[t], x_4 \to x2'[t],$

$z_1 \to z1[t], z_2 \to z2[t], z_3 \to z3[t], z_4 \to z4[t],$

$a_1 \to 1,$

$a_2 \to 1,$

$m_1 \to 1,$

$m_2 \to 0.01,$

$g \to 1\}$

(* The control torque ustar[t_] will switch from "high" to "low" based on the conditional statement s[t] shown above; if you would like to use zero control torque, try 0. instead of 0

Because Simplify used $u - u^*$ we must switch like this
if $s1 > 0$

then $u1^* \to \max$
if $s1 < 0$

then $u1^* \to \min$ *)

$u1star := Piecewise[\{\{0., s1 <= 0.\}, \{1., s1 > 0.\}\}]$

$u2star := Piecewise[\{\{-1., s2 <= 0.\}, \{0., s2 > 0.\}\}]$

(* Substitute numerical values in the state eqn preparation for numerical integration *)

$stateDDN = thetaDD /. \{\theta_1 \to x1[t], \theta_2 \to x2[t], \dot{\theta}_1 \to x1'[t], \dot{\theta}_2 \to x2'[t],$

$a_1 \to 1,$

$a_2 \to 1,$

$m_1 \to 1,$

$m_2 \to 0.01,$

$g \to 1,$

$u_1 \to u1star,$

$u_2 \to u2star$

$\}$

(* include numbers for the different masses, lengths, etc.. in preparation for numeric integration if you change numeric values, don't forget to change for the
1)switch
2) stateDDN and costateDDN, and
3)Graphics! *)

$costateDD = \{z1dot, z2dot, z3dot, z4dot\};$

$costateDDN = costateDD /. \{x_1 \to x1[t], x_2 \to x2[t], x_3 \to x1'[t], x_4 \to x2'[t],$

$z_1 \to z1[t], z_2 \to z2[t], z_3 \to z3[t], z_4 \to z4[t],$

$a_1 \to 1,$
$a_2 \to 1,$
$m_1 \to 1,$
$m_2 \to 0.01,$
$g \to 1,$
$u_1 \to u1star,$
$u_2 \to u2star$
$\}$

APPENDIX D

NELDER MEAD OPTIMIZATION OF THE DOUBLE PENDULUM

```
(* Mathematica notebook file *)
(* This code is a Nelder Mead optimization of the initial co-state values for a double
pendulum.  It uses the symbolic results for state and co-state equations as well as the
Hamiltonian and switching conditions from Appendix C *)
x2desired = -3Pi/2+Pi/6;
metricZ[zz_?(VectorQ[#,NumericQ]&)] :=
    Block[{},
        Off[NDSolve::"mxst"];
        msoln := NDSolve[
            {
            x1''[t] == stateDDN[[1]], x2''[t] ==stateDDN[[2]],
            z1'[t] == costateDDN[[1]],
            z2'[t] == costateDDN[[2]],
            z3'[t] == costateDDN[[3]],
            z4'[t] == costateDDN[[4]],
            (* initial conditions—if you change here, change in soln2 *)
            x1[0] ==Pi/2-Pi/6, x2[0] == -Pi/2-Pi/6, x1'[0] == 1, x2'[0] == -1,
            (* Nelder Mead must adjust these co-stae initial conditions *)
            z1[0] == zz[[1]],
            z2[0] == zz[[2]],
            z3[0] == zz[[3]],
            z4[0] == zz[[4]] },
```

(* it is critical to cover the entire range for interpolation, rather than extrapolation *)

{x1,x2,z1,z2,z3,z4},{t,0,2}

]; (* end of NDSolve *)

(* find the final time, i.e., the time where theta1 = Pi + a *)

T = FindRoot[Evaluate[x1[t]/.msoln] – (Pi/2 + Pi/6), {t,1}];

tfinal = t/.T;

(*
the metric for Nelder Mead to minimize is the boundary conditions that PMP must satisfy, final time is free:

$$H(T) = -\frac{\partial h}{\partial t}(T) = -2M\dot{x}_2(x_{2f} - x_{2d}) /. \{t \to t_f\}$$

final states x3 and x4 are free
z3(T) = 0
z4(T) = 0
and final state x2 is free (with penalty):

$$z2(T) = \frac{\partial h}{\partial x_2} = 2M(x_{2f} - x_{2d}) /. \{t \to t_f\}$$

the only final state that is fixed is x1, thus z1(T) is arbitrary.
*)

z2f = (z2[tfinal] /. msoln)[[1]];

z3f = (z3[tfinal] /. msoln)[[1]];

z4f = (z4[tfinal] /. msoln)[[1]];

HNf = (HN /. msoln /. t -> tfinal)[[1]];

x2df = (x2'[tfinal] /. msoln)[[1]];

x2f = (x2[tfinal] /. msoln)[[1]];

(* weighing the final error a little higher *)

dhf = 200 x2df (x2f – x2desired);

(* x1 fixed, we force it to be, but x2 is free: we allow it to vary (at a cost, h) *)

```
dh2f = 200 (x2f – x2desired);

(z3f^2 + (z2f – dh2f)^2 + (HNf + dhf)^2 + z4f^2)

]; (* end of Block *)

(* Perform Nelder Mead optimization on the guesses at the initial values for the co-
states *)

zz = {zz1, zz2, zz3, zz4};

sln = NMinimize[

        {metricZ[zz]}, zz, Method -> {"NelderMead", "RandomSeed" -> 8}]

(* change the random seed to get different results *)

    {2.80382,{zz1->0.788431,zz2->-0.583146,zz3->-0.104222,zz4->-0.0157834}}

(* display the results of the Nelder Mead optimization *)

tfinal

soln2 = NDSolve[

        {

        x1''[t] == stateDDN[[1]], x2''[t] == stateDDN[[2]],

        z1'[t] == costateDDN[[1]],

        z2'[t] == costateDDN[[2]],

        z3'[t] == costateDDN[[3]],

        z4'[t] == costateDDN[[4]],

        (* initial conditions must match those in msoln *)

        x1[0] == Pi/2 – Pi/6, x2[0] == -Pi/2 – Pi/6, x1'[0] == 1,

        x2'[0] == -1,
```

```
(* Nelder Mead must adjust the co-state initial conditions *)

        z1[0] == zz1 /. sln[[2]],

        z2[0] == zz2 /. sln[[2]],

        z3[0] == zz3 /. sln[[2]],

        z4[0] == zz4 /. sln[[2]],

        },

{x1, x2, z1, z2, z3, z4}, {t, 0, 2}

]
```

REFERENCES

Alexander, R. (2003) *Principles of Animal Locomotion.* Princeton University Press, Princeton, NJ.

Armitage, and Eberlein. (2006). *Elliptic Functions.* Cambridge University Press, Cambridge, UK.

Bekker, M. (1956). *Theory of Land Locomotion.* University of Michigan Press, Ann Arbor, MI.

Bertram, J. and Ruina, A. (2001). Multiple walking speed-frequency relations are predicted by constrained optimization. *Journal of Theoretical Biology,* 209(4): 445-453.

Bowman. (1961). *Introduction to Elliptic Functions with Applications.* Dover, Mineola, NY.

Coleman, M. (1998). *A Stability Study of a Three-Dimensional Passive-Dynamic Model of Human Gait.* PhD Thesis: Cornell University, Ithaca, NY.

Denardo, D. (1975). *Dynamic Programming.* Dover, Mineola, NY.

Denavit, J., Hartenberg, R. (1955). A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *ASME Journal of Applied Mechanics* , 23:215-221.

Donelan, J., Kram, R., and Kuo, A. (2001) Mechanical and metabolic determinants of the preferred step width in human walking. *Proceedings of the Royal Society of London B*, 268:1985-1992.

Donelan, J., Kram, R., and Kuo, A. (2002) Mechanical and metabolic costs of step-to-step transitions in human walking. *Journal of Experimental Biology*, 205:3717-3727.

Dorfman, R. (1969). An Economic Interpretation of Optimal Control Theory. The Economic Review , Vol. 59, No. 5, 817-831.

Fallis, G. T. (1888). *Patent No. 376,588.* United States of America.

Garcia, M., Chatterjee, A., Ruina, A., & Coleman, M. (1998). The Simplest Walking Model: Stability, Complexity, and Scaling. *ASME Journal of Biomechanical Engineering.*

Goldstein, H. (1950). *Classical Mechanics*. Addison-Wesley, Reading, MA.

Hartog, J. D. (1961). *Mechanics*. Dover, Mineola, NY.

Hocking, L. (1991). *Optimal Control, An Introduction to the Theory with Applications*. Oxford University Press.

Kajita, S. (1991). Study of Dynamic Biped Locomotion on Rugged Terrain. *IEEE*.

Kirk, D. (1970). *Optimal Control Theory*. Dover, Mineola, NY.

Kuo, A. (2001). A simple model predicts the step length-speed relationship in human walking. *Journal of Biomechanical Engineering*, 123:264-269.

Kuo, A. D., Donelan, J. M., & Ruina, A. (2005). Energetic Consequences of Walking Like an Inverted Pendulum: Step to Step Transitions. *Exercise and Sport Sciences Reviews* , 33:88-97.

L. Landau, E. L. (2001). *Mechanics*. Butterworth Heinemann, Oxford.

Mark W. Spong, M. V. (1989). *Robot Dynamics and Control*. Wiley, Hoboken, NJ.

McGeer, T. (1990). Passive Dynamic Walking. *International Journal of Robotics Research* , 9:62-82.

Muench, P., and Tucker, A. (2003). Passive Dynamics. *GMTV*. Houghton, MI.

Muench, P., Haueisen, B., Overholt, J. (2004) A Brief History of Legged Mechanisms. *IVSS*. Traverse City, MI

Muench, P., Alexander, J., Quinn, R., & Aschenbeck, K. (2005) Pneumatic Spring for Legged Walker. *SPIE*. Orlando, FL.

Muench, P., and Tucker, A. (2005). Slide Walker. *GMTV*. Houghton, MI

Muench, P. (2006) Slider-Crank Mechanism: A Simple Monopod *Dynamic Walking* Ann Arbor, MI.

Muench, P., and Marecki, A. (2007). Pendulum Walker. *SPIE*. Orlando, FL.

Muench, P., Alexander, J., Hadley, S., Starkey, S. (2008) Bipedal Walkiing *Army Science Conference*. Orlando, FL.

Muench, P., Cheok, K. C., Czerniak, G., & Bednarz, D. (2009). Optimal Time and Energy Efficiency in Legged Robotics. *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS): NDIA*. Troy, MI.
Muench, P., Cheok, K.C., Czerniak, G., (2010) Optimal Powering Schemes for Legged Robotics. *SPIE.* Orlando, FL.

Nelder, J. A., and Mead, R. (1965). A Simplex Method for Function Minimization. *Computer Journal* , 7:308-313.

Playter, R., Buehler, M., & Raibert, M. (2006). BigDog. Orlando, FL: *SPIE.*

Pratt, G. A., and Williamson, M. W. (1997). *Patent No. 5,650,704*. United States of America.

Pratt, J. (2000). *Exploiting inherent robustness and natural dynamics in the control of bipedal walking robots*. PhD Thesis, MIT, Cambridge, MA.

Pratt, J., and Krupp, B. (2004). Series Elastic Actuators for legged robots. (pp. 5422:135-144). Orlando, FL: *SPIE.*

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

Srinivasan, M (2006) *Why Walk and Run: Energetic Costs and Energetic Optimality in Simple Mechanics-Based Models of a Bipedal Animal.* PhD thesis, Cornell Univeristy, Ithaca, NY.

Srinivasan, M. (2010). Trajectory Optimization, a brief introduction. *Dynamic Walking 2010*. Cambridge, MA.

Strogatz, S. (1994). *Nonlinear Dynamics and Chaos*. Perseus. Cambridge, MA.

Tedrake, R. L. (2004). *Applied Optimal Control for Dynamically Stable Legged Locomotion*. PhD thesis, MIT, Cambridge, MA.

Weber, E. (2005). Optimal Control Theory for Undergraduates Using the Microsoft Excel Solver Tool. *CHEER* , Vol. 19.