TUNING A DISCRETE LINEAR QUADRATIC REGULATOR CONTROLLER USING THE REGION OF ATTRACTION FOR THE SIMPLEST WALKER MODEL

by

JUSTIN ERNST, B.S.

THESIS
Presented to the Graduate Faculty of
The University of Texas at San Antonio
In Partial Fulfillment
Of the Requirements
For the Degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

COMMITTEE MEMBERS:
Pranav Bhounsule, Ph.D., Chair
Hung-da Wan, Ph.D.
Adel Alaeddini, Ph.D.

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
College of Engineering
Department of Mechanical Engineering
August 2019

DEDICATION

I would like to dedicate this thesis to my loving wife and daughter, without whom, this would not be possible. Thank you to my parents who have supported me through my education and have always pushed me to be the best that I can be.

ACKNOWLEDGEMENTS

To begin, I would like to thank Dr. Pranav Bhounsule for his constant support and guidance. The simulation used within this thesis to test the control technique and tuning method is a modified version of what he supplies on his personal web-page for beginning with passive-dynamic walking. Without him and his support this thesis would not be possible. Additionally, through several conversations with Ali Zamani I was able to develop a better understanding of passive dynamic walking. Friendships built during my time in the College of Engineering at The University of Texas at San Antonio have made the graduate school experience less stressful and more enjoyable.

August 2019

TUNING A DISCRETE LINEAR QUADRATIC REGULATOR CONTROLLER USING THE REGION OF ATTRACTION FOR THE SIMPLEST WALKER MODEL

Justin Ernst, M.S.
The University of Texas at San Antonio, 2019

Supervising Professor: Pranav Bhounsule, Ph.D.

The work done within this thesis strives to expand the capabilities of a designed Discrete Linear Quadratic Regulator (DLQR) controller. The controller is implemented into The Simplest Walker Model, and after perturbing the initial state of the system the goal is to cause the system trajectory to converge back onto the limit cycle. Every initial state that converges to the limit cycle lies within the region of attraction, depicting the controller strength. An approach to visualizing the region of attraction is used to not only depict the robustness of the controller but also assist in tuning the input parameters of the DLQR controller. The region of attraction is measured by the percentage of successful initial states out of the total attempted initial states. The percentage produced using other input parameters is used to determine the effects they have on the system. This approach is used to tune the input parameters so that a larger percentage is produced by the region of attraction. The region of attraction is calculated for the uncontrolled Simplest Walker and for the walker controlled by a basic DLQR controller using the identity matrices as inputs. The approach using the region of attraction to tune the DLQR controller is then compared with the uncontrolled and the basic DLQR-controlled Simplest Walker. The comparison displays that the tuning approach is valid in expanding the region of attraction for the Simplest Walker.

TABLE OF CONTENTS

Acknowledgements . . . . . **iii**

Abstract . . . . . **iv**

List of Tables . . . . . **vii**

List of Figures . . . . . **viii**

Chapter 1: Introduction **1**

1.1 Thesis Contribution 2

Chapter 2: Literature Review **3**

2.1 Walking machines 3

2.2 The design of The Simplest Walker 4

2.3 Process of taking a step 4

2.4 Important terms to know 7

2.4.1 Limit Cycle 7

2.4.2 Fixed Point 9

2.4.3 Poincaré Map 10

2.4.4 Poincaré Section 11

2.4.5 Eigenvalues 11

2.4.6 Linearized Stability 12

2.4.7 Stability of Discrete Time Systems 13

2.5 Overview of Related work 13

Chapter 3: Methodology **16**

3.1 Achieved by the simulation 16

3.2 Exploration of the simulation 17

3.3 Original simulation results 19
3.4 Un-perturbed and Uncontrolled system 20
3.5 Perturbing the System 21
3.6 Controlling perturbed walker 23
3.7 Region of Attraction 30

Chapter 4: Results 32
4.1 Region of attraction exploration 32
4.2 Controllable Percentage 43
4.3 Region of Attraction 43

Chapter 5: Conclusion 47
5.1 Results summary 47
5.2 Future work 48

Bibliography 49

Vita

LIST OF TABLES

3.1 A description of the system for the use in the original simulation. 19

4.1 A summary of the preferable $q1$, $q2$, and $q3$ values discovered from the trend-line of using a diagonal of the Q matrix less than one. 36

4.2 A summary of the $q1$, $q2$, and $q3$ values discovered from the trend-line of using a diagonal of the Q matrix greater than one. 39

4.3 A summary of the different controllers tested on The Simplest Walker Model. 43

LIST OF FIGURES

2.1 Descriptic graphic of The Simplest Walker Model. 5

2.2 The sequence of taking one step for The Simplest Walker Model beginning at the Mid-stance position. 6

2.3 An illustration of the sequence of taking one step for The Simplest Walker Model begining at the Mid-stance position. 8

2.4 A graphic depicting a Limit Cycle with the Fixed Point labeled, with implementation of a controller represented by u_0. 10

2.5 A visual interpretation of a Poincaré Map and Poincaré Section for The Simplest Walker. In this case the Poincaré Section is defined at the Mid-stance position. 12

2.6 An arrangement of various Passive-Dynamic Walkers. 14

3.1 Initial position of the original walker 20

3.2 Graph of θ and ϕ angles versus time for the original walker 21

3.3 Initial position of the mid-stance position walker 22

3.4 Graph of θ and ϕ angles versus time for the mid-stance position walker . . . 23

3.5 Initial position of the perturbed walker 24

3.6 Graph of θ and ϕ angles versus time for perturbed walker 25

3.7 Graph depicting the perturbed walker starting at mid-stance and converging to the limit cycle. 27

4.1 The Region of Attraction calculated for the Uncontrolled Simplest Walker . 33

4.2 The Region of Attraction calculated for the Simplest Walker controller tuned using the identity matrix 33

4.3 The trend-line depicting the effects of the smaller Q matrix. 34

4.4 The trend-line depicting the effects of the smaller Q matrix, focusing on the first fourth. 35
4.5 The trend-line depicting the effects of the smaller Q matrix, focusing on the section capable of the highest controllable percentage. 35
4.6 The same trend-line using color values to depict the values for $q1$. 36
4.7 The same trend-line using color values to depict the values for $q2$. 37
4.8 The same trend-line using color values to depict the values for $q3$. 37
4.9 The trend-line produced to show the effects of adjusting ρ. 38
4.10 The trend-line produced to show the effects of adjusting ρ, now using the color scale. 39
4.11 The trend-line produced to show the effects of a Q matrix with the diagonal greater than one. 40
4.12 The same trend-line using color values to depict the values for $q1$. 40
4.13 The same trend-line using color values to depict the values for $q2$. 41
4.14 The same trend-line using color values to depict the values for $q3$. 41
4.15 The trend-line showing the effects of changing the value of ρ for the Q matrix with the diagonal greater than one. 42
4.16 The same trend-line showing the effects of changing the value of ρ for the Q matrix with the diagonal greater than one, now using a color scale. 42
4.17 The slice of the region of attraction at the layer of ϕ equal to 0.1 for the uncontrolled Simplest Walker. The fixed point is represented by the red star. 44
4.18 The slice of the region of attraction at the layer of ϕ equal to 0.1 for the controller tuned using the identity. The fixed point is represented by the red star. 45
4.19 The slice of the region of attraction at the layer of ϕ equal to 0.1 for the controller tuned using the Lower Q matrix. The fixed point is represented by the red star. 45

4.20 A side-by-side comparison of the region of attraction for the uncontrolled Simplest Walker and two different controllers tuned using different parameters. 46

CHAPTER 1: INTRODUCTION

Bipedal locomotion has been a topic of study for many years. The motion of walking has been demonstrated by highly complex humanoid robots and even simple, non-powered, children's toys. Humans are capable of producing a very natural gait when they walk, allowing for low energy consumption, superior versatility, and exceptional stability. These characteristics of a natural gait are the reason for the elevated interest and research in bipedal locomotion. Highly sophisticated humanoid robots are capable of producing a stable and versatile walking gait but come at a cost. These humanoid robots have central control schemes known as the Zero Moment Point (ZMP) method. [1] The ZMP method always strives to keep the system at a static equilibrium point. This constant effort to maintain the equilibrium point leads to every joint needing to be controlled at all times, leading to high energy consumption and very unnatural walking gaits. The drawbacks that come from walking machines that use a ZMP control scheme can be avoided if the approach to walking is modified. McGeer proposed a different approach, known as passive-dynamic walking. [2] The main motivation for studying passive walking is because the mechanics of the system are simplified. The work done by McGeer leads to the physical dimensions of the human body being responsible for the quality of the gait. The main take-away from his work is that the control of the legs alone is not responsible for effective bipedal walking. Passive-dynamic walking is well demonstrated by toy walkers. These walkers are able to travel down a ramp with minimal slope without the use of motors or controllers. Passive-dynamic walkers led to the development of passive dynamic based walkers, which are essentially passive walkers capable of walking on a level surface thanks to the use of applied hip torque from a controller. Passive-dynamic based walkers are highly energy efficient and can open the gateway for better humanoid robots. Energy efficient bipedal locomotion is important for the development of practical bipedal robots. The work done by Zamani et al. focused on expanding the capabilities of bipedal walkers on different terrains, adding practicality. [3]

1.1 Thesis Contribution

Control of passive-dynamic based walkers can be accomplished through a variety of different techniques, though the focus of this thesis will be on using a Discrete Linear Quadratic Regulator (DLQR). With the work completed in this thesis, my goal is to effectively implement a DLQR controller and establish a better understanding of how to effectively tune it using the region of attraction (ROA). Primary techniques for tuning a DLQR controller rely on a trial-and-error approach with the goal of achieving a certain output. The approach I will be presenting will use the trial-and-error approach but, in addition, offer a technique for effectively tuning the DLQR input parameters. To help display the results of how effective the controller is, the region of attraction for that controller will be generated, thus showing its capability. This thesis will focus on mapping the region of attraction of a controller within a certain range and then calculate a percentage of how many points are controllable. The calculated percentage will describe the strength of the controller, and thus provide a metric to determine if using different controllers is improving the control or hindering it. The goal is to see the size of the region of attraction for The Simplest Walker increase, inferring that the controller has increased the capabilities of the walker. The region of attraction will be calculated for several controllers and careful inspection of all regions of attraction will help to tune the weighing matrices of the DLQR controller. The goal is to contribute a better approach to visualizing the region of attraction, a method of tuning the weighing matrices via the use of the region of attraction, and also a technique for expanding the region of attraction for the Simplest Walker.

CHAPTER 2: LITERATURE REVIEW

Within this section, important terms and concepts will be discussed to provide a basic understanding to aid in comprehending what this thesis is attempting to accomplish. Concepts and terms discussed will build a foundation to develop a method to test and understand the results of different controllers. Prior work done using passive dynamics will also be discussed and can be used as steppingstones to reach the approach done within this thesis.

2.1 Walking machines

There are two main classes of waking robots: static walkers and dynamic walkers. The main goal of static walkers is maintaining a certain static equilibrium throughout their motion. The equilibrium point is best maintained by adding more legs to the machine, therefore, it is more common to find four or even six legs on a static walking robot. With an equilibrium point to maintain, inertial forces are required to remain minimal to prevent the machine from being thrown off balance. To ensure minimal inertial force, the cyclic accelerations are limited. The accelerations of walking robots are defined as being cyclic because the motion of taking a step is done with acceleration values that occur in a cycle. These regularly repeating acceleration values depict a curve with a closed path. The basic example of a closed path is a circle. The other main class of walking robots is dynamic walkers. Dynamic walkers have fewer legs than what is typically found on static walkers and more closely resemble a person. Dynamic walkers have the potential for going faster than static walkers because they are not restricted by the cyclic accelerations. A passive dynamic walker is capable of traversing a ramp at a downward slope without the use of powered components and can be traced back to the machines created by McGeer in 1990 [2], and even earlier to walking toys. Passive-dynamic walkers achieve their locomotion through mechanics alone instead of the use of motors. The mechanics that empower the movement of the walker come from the mechanical design of the walker and the force of gravity. The slope of the ramp adds a gravity potential that is transformed into kinetic energy. Additionally, the mass distribution of the walker, the leg length, and the foot

shape determine the capability of walking. Since passive-dynamic walkers require no processor to control motors and sensors, they have been described as being brainless. [4]

2.2 The design of The Simplest Walker

The walker studied in this thesis, The Simplest Walker, is comprised of two thin legs connected at the hip by a frictionless joint. The mass distribution is only in the hip and the feet, with the mass in the hip being much higher than the mass in the feet. The larger amount of mass in the hip prevents the hips from being affected from the swinging motion of the feet. This walker is capable of moving on its own when placed on a ramp of a specific angle of decline. It is a simplified two-dimensional model with straight legs, and some assumptions are made to allow for simplicity in calculations. To return to the mid-stance position, it is assumed that the swing leg passes through the floor momentarily before continuing to swing with the step. The brief passage into the ramp surface allows for scuffing of the feet to be avoided, whereas in the physical world different methods would be needed to prevent this, i.e. knees or a swaying motion. There are four main parameters of interest when studying a bipedal walker. The first parameter is the angle θ, which is the angle between the stance leg and the normal to the ramp surface. The second parameter is the time derivative of θ, i.e. the rate of change of θ. The third parameter is ϕ, representing the angle between the stance leg and the swing leg. The fourth parameter is, as expected, the time derivative of ϕ also referred to as the rate of change of ϕ. A Graphic of the simplest walker can be seen in Figure 2.1. The mass at the hip is labeled as M, the mass at each foot is labeled as m, the leg length defined as l, the angle of the ramp defined as γ, and lastly gravity as g.

2.3 Process of taking a step

In order to understand the process of a passive-dynamic walker taking a step, it is important to know the role the legs are playing. The walker has two legs, one being the stance leg and the other being the swing leg. The stance leg is always in contact with the ground and the swing leg is lurching forward in the motion of the step. The walker has a cycle of events that take place

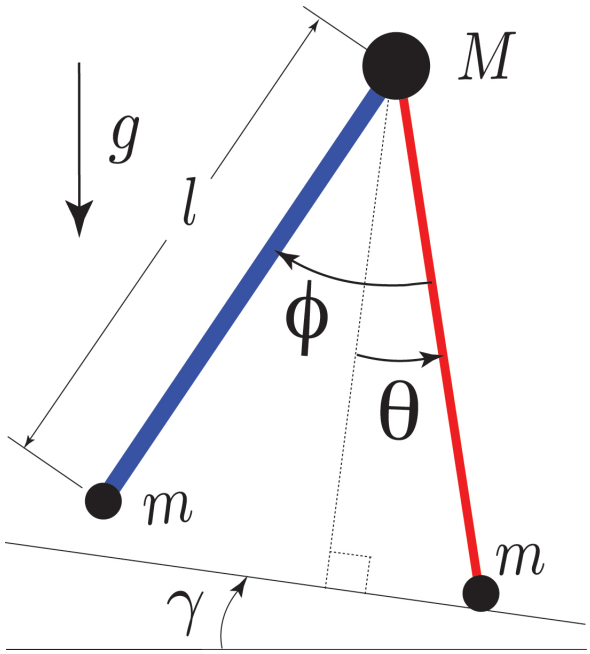


Figure 2.1: Descriptic graphic of The Simplest Walker Model.

for each step. In order for the walker to complete one step it must begin at the initial position, which for the study of this thesis, was set to be mid-stance. Mid-stance represents the moment in time when the stance leg is perpendicular to the surface of the ramp with the swing leg preceding it. Once the walker is released from the initial stance it is in a phase known as single stance, where the walker will roll forward until the swing leg reaches the ramp, creating the first contact point. The instant at which the swing leg makes contact with the ground is known as heel-strike and has a no-slip, no-bounce collision, known as a plastic collision. Once the swing leg reaches heel-strike, also know as the foot-ground contact point, the legs switch roles in a phase referred to as foot-strike. In the foot-strike phase the stance leg becomes the new swing leg, and the previous swing leg is now the new stance leg. The new swing leg begins to move forward, in the single stance phase once more, taking its next step. It is along this swing where the walker returns to the mid-stance position, completing its first step. The walker will continue to oscillate through these steps until it reaches a state where it becomes unstable and falls, such as coming in contact with the level floor. A simple breakdown of the order of events using the technical phase names is as follows: mid-stance, single stance, heel-strike or foot-ground contact event, foot-strike, single

stance, and finally back to mid-stance. The sequence of a single step can be seen in Figure 2.2. The process of taking a step can be condensed into two phases: a single stance phase and a foot-strike phase. [5] [6] The two phases can be looked at as containing check points. For the single stance phase the check point would be the mid-stance position. Alternatively, the check point for the foot-strike phase is the collision with the ground at heel-strike. Each phase of a single step is represented by its own equations. The single-stance phase, containing the mid-stance position, is

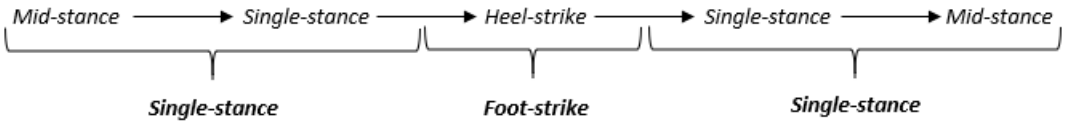


Figure 2.2: The sequence of taking one step for The Simplest Walker Model beginning at the Mid-stance position.

assumed to have no hinge friction in the hip, no slipping between the foot of the stance leg and the ground, and scuffing of the feet is ignored entirely. The single-stance phase is represented by Equation 2.1 and Equation 2.2. Angular momentum balancing is used to obtain the two equations. Equation 2.1 is balanced around the foot contact point and Equation 2.2 is balanced around the hip. In addition to angular momentum balancing, the time units were removed from the equations in a process known as non-dimensionalization. The time units were removed by substituting $\sqrt{l/g}$ in for t. Lastly, the limit of $m/M \to 0$ was applied to further simplify the equations.

$$\ddot{\theta} = \sin(\theta - \gamma) \tag{2.1}$$

$$\ddot{\phi} = \sin(\theta - \gamma) + \{\dot{\theta}^2 - \cos(\theta - \gamma)\} \sin(\phi) \tag{2.2}$$

The heel-strike or foot-ground contact occurs when Equation 2.3 is true.

$$\phi = 2\theta. \tag{2.3}$$

The foot-strike phase is assumed to have a plastic collision with the ground and occur briefly so that double support from both legs does not occur.

$$\theta^+ = -\theta^- \tag{2.4}$$

$$\phi^+ = -\phi^- = -2\theta^- \tag{2.5}$$

$$\dot{\theta}^+ = \cos(2\theta^-)\dot{\theta}^- \tag{2.6}$$

$$\dot{\phi}^+ = \Big(1 - \cos(2\theta^-)\Big)\cos(2\theta^-)\dot{\theta}^- \tag{2.7}$$

The foot-strike equations use superscripts to denote when the values of θ and ϕ are taken. The superscripts represent the instance right before foot-strike, $+$, and right after foot-strike, $-$. The moment the legs switch roles is depicted in Equation 2.4 and Equation 2.5. Equation 2.6 and Equation 2.7 represent the angular rates of the legs, calculated using conservation of angular momentum about the foot-strike location and the hip of the walker, respectively. Once again, time was non-dimensionalized using $\sqrt{l/g}$ and simplified using the limit $m/M \to 0$. A series of pictures displaying a typical step of the simplest walker can be seen in Figure 2.3.

2.4 Important terms to know

In order to understand the ideas and reasoning behind techniques explored, it is important to understand a few key concepts. This section will provide an overview, and the intention is to instill enough knowledge to comprehend the experiments and results.

2.4.1 Limit Cycle

A limit cycle refers to a closed path trajectory that is isolated from neighboring trajectories, meaning the surrounding trajectories are not closed. These surrounding trajectories can circle inward towards the limit cycle or they can circle outwards away. The path of the surrounding trajectories helps determine stability of the limit cycle. If the path begins to circle inwards and approach the limit cycle, the limit cycle is defined as being attracting, i.e. stable. Likewise, if the path travels

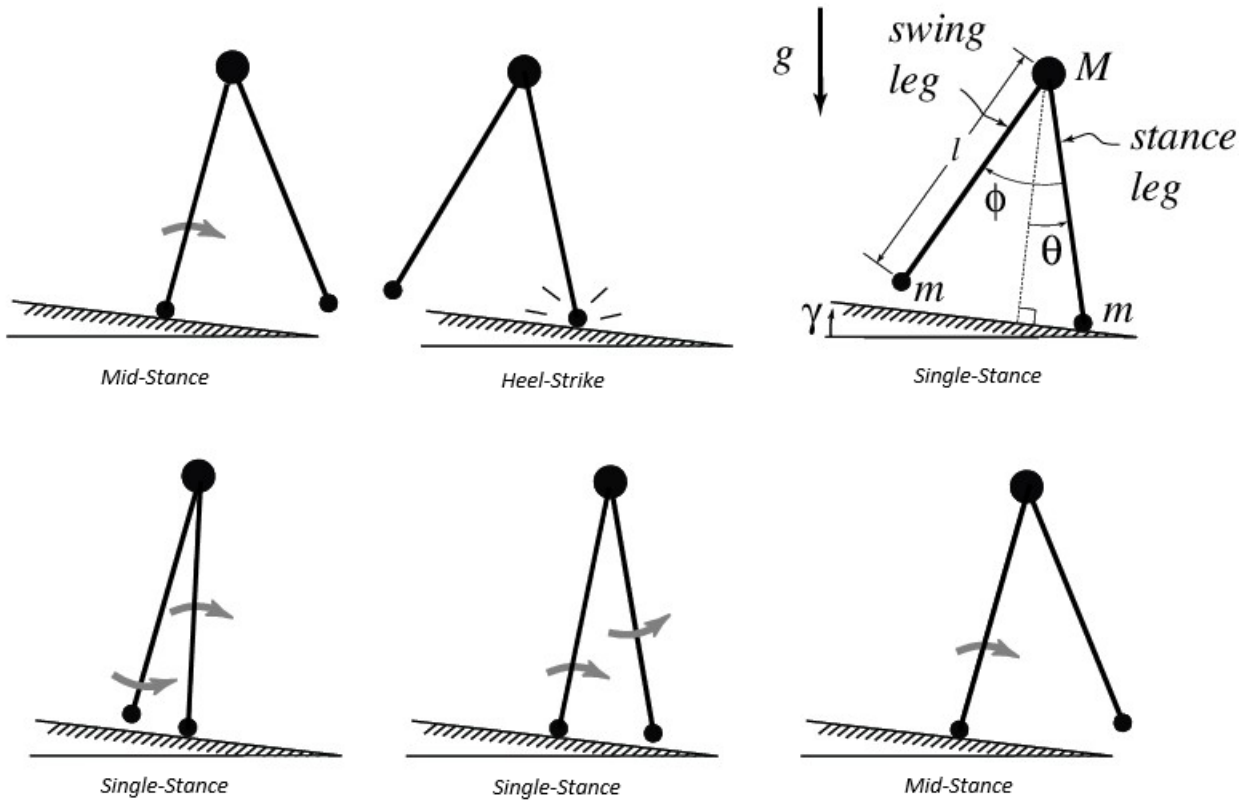


Figure 2.3: An illustration of the sequence of taking one step for The Simplest Walker Model begining at the Mid-stance position.

away from the limit cycle, the limit cycle is defined as being unstable. There is a third case that may arise in certain situations: the half-stable limit cycle. A half-stable limit cycle occurs when not all the paths approach or move away from the limit cycle. An example of a half-stable limit cycle would be a limit cycle with two neighboring open trajectories with one approaching the limit cycle and the other moving away. Stable limit cycles help to model systems capable of maintaining oscillations without an outside force. A couple of common examples of limit cycles include the beating of a heart and the firing of a pacemaker. These examples are relevant in that their oscillations maintain a preferred period, waveform, and amplitude. [7] The characteristic of interest is if a limit cycle is perturbed slightly it will inevitably return to the original cycle. The important thing to remember is that limit cycles cannot occur in linear systems. A linear system can be observed to have a closed orbit, though the trajectories will not be isolated. A linear system is defined as

$\dot{x} = Ax$ with $x(t)$ being a solution, so since it is linear, $cx(t)$ is also a solution for any c not equal to zero ($c \neq 0$). The solution $x(t)$ is surrounded by a family of closed paths, with an amplitude defined by the initial conditions. It is also important to note that any disturbance to the system will be in effect endlessly. Determining if a system has a limit cycle can be difficult, because at first glance, the equations of a system do not necessarily depict oscillations of a limit cycle. There are different techniques to determining if closed trajectories exist and approximating their shape and period length. When studying the passive-dynamic based walker in this thesis, it is clearly seen, graphically, that the system has a limit cycle.

2.4.2 Fixed Point

Mathematically, a fixed point is defined as a point x such that a function $f(x)$ is equal to x. In simpler terms, a fixed point can be described as a point that can undergo any application of a map and remain the original point. Fixed points can be discovered with repeated iteration of a function f, as described by Seidel. [8] By iterating a function, you take the output from one iteration and use it as the input for the next iteration, and repeat this process as many times as required. This idea is represented in Equations 2.8 through 2.10 and demonstrate only three iterations. Simplification of the process leads to Equation 2.11.

$$x_1 = f(x_0) \tag{2.8}$$

$$x_2 = f(f(x_0)) = f(x_1) \tag{2.9}$$

$$x_3 = f(f(f(x_0))) = f(x_2) \tag{2.10}$$

$$x_n = f(x_{n-1}) \tag{2.11}$$

Within a limit cycle, the fixed point is reached once every lap of the cycle. A fixed point is defined as attracting if, for an initial point slightly away from the fixed point, it converges back to the fixed point. A fixed point is considered repelling if, for an initial point slightly away from the fixed point, it does not converge back to the fixed point. A visual representation of a Limit Cycle with a

Fixed Point can be seen in Figure 2.4

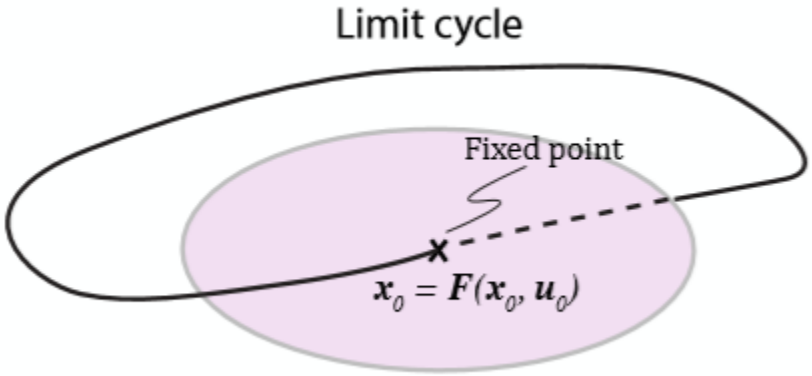


Figure 2.4: A graphic depicting a Limit Cycle with the Fixed Point labeled, with implementation of a controller represented by u_0.

2.4.3 Poincaré Map

The continuous periodic flow of a system can be represented by a different discrete dynamic system known as a Poincaré Map. [9] Poincaré Maps are beneficial when studying trajectories flowing in a circular motion, specifically the flow around a limit cycle. When a system clearly has a periodic cycle, a Poincaré Map for that system is said to be well defined. [10] A formal definition of a Poincaré Map described by Strogatz, in his book discussing non-linearity, is briefly described. [7] Imagine a system, represented by $\dot{x} = f(x)$, as being an n-dimensional size. Now define a surface, represented by S, as being $(n-1)$-dimensional where the initial trajectories pass through the surface instead of running parallel to it. Mapping S from the initial crossing of the surface to the next intersection of the surface creates the Poincaré Map P. A Poincaré Map is defined as $x_{k+1} = P(x_k)$, where x_k represents the k^{th} intersection with the surface S. Now, consider a fixed point of the Poincaré Map P, i.e. $P(x^*) = x^*$, the beginning trajectory at x^* will return to x^* after one period of time T, thus creating a closed curve. By observing the trajectories around the fixed point of the Poincaré Map, the stability of the cycle can be concluded. The classification of the system stability is described within the Limit Cycle section of this thesis. In passive-dynamic

walking, the Poincaré Map is crucial in determining the motions of steady state walking and the motion stability.

2.4.4 Poincaré Section

A Poincaré Section has already been mentioned within this thesis, but not by its technical name, Poincaré Section. The definition provided within the last section discussing Poincaré Maps defines a surface transverse to the trajectory; that surface is classified as the Poincaré Section. A Poincaré Map can only be constructed using a Poincaré Section. Poincaré Maps display trajectories in an n-dimensional phase space, while Poincaré Sections present the data in an $(n-1)$-dimensional space. Strogatz helps describe the Poincaré Section by relating it to research techniques used by Biologists. [7] When Biologists study three-dimensional structures, they use a strategy of breaking up a complex formation into thin slices that are then used to create slides. A Biologist can then study the slides to determine what is taking place at each layer of the structure. A Poincaré Section is a single slice from the Poincaré Map. A Poincaré Section will capture the trajectories in the same phase for each cycle rotation. The Poincaré Section used within the scope of this thesis is defined at the event of mid-stance. In Figure 2.5, a Poincaré Map is displayed using a period-one limit cycle of The Simplest Walker. The Poincaré Section is labeled as well as the fixed point. Studying the Poincaré Map and Poincaré Section is done by perturbing the initial state and observing if the system will return to the limit cycle. The red line depicts the system trajectory and demonstrates its ability to converge back onto the limit cycle from its initial starting point.

2.4.5 Eigenvalues

Eigenvalues determine the characteristics of the system using the equations of motion of the linearized system. The equations of motion for the Simplest Walker are linearized around the fixed point and are represented in matrix form using the state space representation, $\mathbf{A}$. Eigenvalues, represented by λ, have a corresponding Eigenvector, $\mathbf{X}$, that can be represented by Equation 2.12. The Eigenvalue and Eigenvectors help to characterize the system. Solving for the Eigenvalues is done

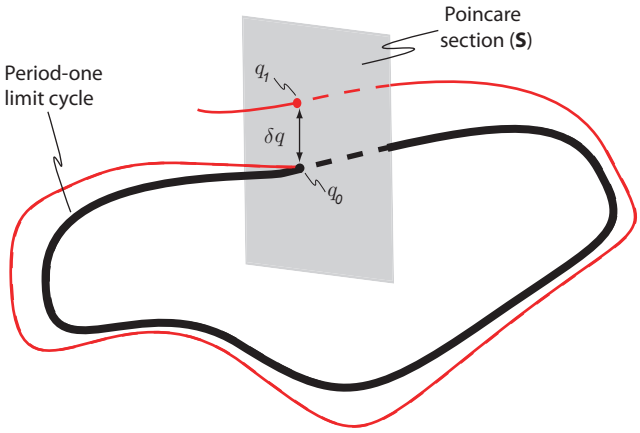


Figure 2.5: A visual interpretation of a Poincaré Map and Poincaré Section for The Simplest Walker. In this case the Poincaré Section is defined at the Mid-stance position.

by finding the roots of the system when setting the determinant equal to zero, shown in Equation 2.13.

$$(\mathbf{A} - \lambda I)\mathbf{X} = 0 \tag{2.12}$$

$$\det(\mathbf{A} - \lambda I) = 0 \tag{2.13}$$

2.4.6 Linearized Stability

When determining the stability of fixed points, there is a simple test that can be conducted thanks to Liapunov, the Eigenvalue Criteria for Stability. [11] This method for determining the stability begins with calculating the Eigenvalues of the state matrix. When the equations are written in the State Space representation in the form $\dot{x} = Ax + Bu$, the state matrix is represented by the A matrix. In addition, the A matrix is also known as the Jacobian of the system linearized about a fixed point. Once the Eigenvalues of the Jacobian have been calculated, they will be in the complex form, i.e. a real and an imaginary part. For determining the stability, the real part of the Eigenvalue is observed. If the real part for all the Eigenvalues are less than zero, then the system is said to be asymptotically stable. If the real part of all the Eigenvalues is greater than zero, then the

system is said to unstable. Unfortunately, if the real part of the Eigenvalues is equal to zero, this method cannot describe its stability, therefore a further analysis is required to accurately determine stability. This stability analysis described above is commonly referred to as Liapunov's Theorem. The real part of the Eigenvalues is classified as the characteristic, or Floquet, multipliers. [7]

2.4.7 Stability of Discrete Time Systems

To discuss stability of discrete time systems, it is important to first understand the difference between discrete time and continuous time systems. To begin, lets develop a simple understanding of the difference between continuous and discrete time systems. A continuous time system is defined by the generic differential equation $\dot{x} = f(x(t))$, showing that the state vector x is evolving in time. Continuous systems contain state variables that can change continuously over time, i.e. the amount of water flowing through a pipe. A discrete time system is represented by the formula $x_{k+1} = F(x_k)$, where k is the time increment being used. State variables of a discrete time system are only calculated at certain moments in time. The passive walker is a discrete time system, where the state of the system is calculated at the Poincaré Section of the limit cycle. Stability of discrete time systems relies on the Eigenvalues of the state matrix as well. If one or more have a magnitude larger than one, then the system will increase exponentially and be unstable. If at least one Eigenvalue has a magnitude equal to one, then a component in the state response is generated as an under-damped oscillatory component, causing the system to be marginally stable. The third option of stability is known as asymptotically stable, and it occurs when all magnitudes of the Eigenvalues are less than one.

2.5 Overview of Related work

The concept of passive-dynamic walking has been around for over a century, with the first walking toy patent published in 1888 [12], followed by others in 1909 [13], 1912 [14], and in 1938 The Wilson Walker, shown in Figure 2.6 (a) [15]. McGeer, inspired by The Wilson Walker, introduced the first passive-dynamic walking machine, Dynamite. [2] McGeer's Dynamite consisted of four

legs and knees with the inner and outer legs alternating during the walking process. With the correct initial conditions, both the Wilson Walker and Dynamite were able to traverse downhill. Another model is a three-dimensional passive-dynamic robot comprised of two legs with knees and two arms designed to swing side-to-side to help reduce scuffing of the feet. This model also included specially designed feet to stabilize the side-to-side motion and turning of the legs, shown in a photo sequence in Figure 2.6 part (d). [16] Parts (b) and (c) of Figure 2.6 come from Collins. [17] The first

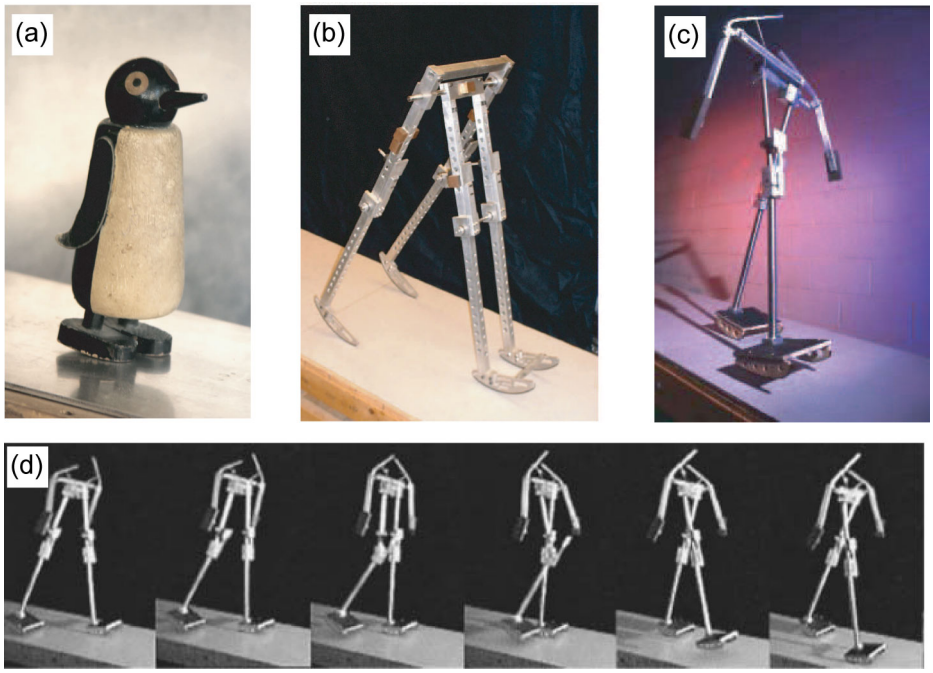


Figure 2.6: An arrangement of various Passive-Dynamic Walkers.

passive-dynamic-running robot used a four-legged design with knees, like McGeer's Dynamite, with an axial spring in each leg to cushion collisions, an additional spring in the hip to aid in the swinging motion, and had arc shaped feet. [18] This running robot could run 36 steps down a ramp with a slope of 0.22 radians. Another popular walker, referred to as the compass-gait walker, resembled the drawing tool known as a compass. [19] A highly simplified model, called The Simplest Walker, consists of massless legs and a mass in the hip much greater than the mass in the feet. [20] The Simplest Walker Model is favored when beginning to study passive dynamics due to the simplicity. The important feature that all of these robots have in common is that they

use their natural dynamics to walk or run downhill. The highly energy-efficient motion is natural and cyclic, and resembles a human walking. A study on the leg swing in human walking has shown that the swing is greatly influenced by the natural dynamics and requires little control. [21] This theory demonstrated in passive walking implies that humans might take advantage of the natural dynamics and expend very minimal amounts of energy to walk. This natural characteristic of passive-dynamic walking is most intriguing for developing legged robots. Stability of bipedal robots is represented in two ways: the robot not falling down and the robot's ability to remain on a defined trajectory. For this thesis, the approach of remaining on the defined path is what will determine the success of the controller. The ability to remain on the defined path can be measured in several ways, including evaluation of the largest Eigenvalue of the Poincaré Map and the region of attraction size. When evaluating the Poincaré Map's largest Eigenvalue, the closer it is to zero, the faster it will converge back to the cycle, and if the Eigenvalue is larger than one it will never converge back and will become unstable. When using the region of attraction method, the larger the region is, the more capable the controller is, whereas the smaller the region, the less capable the controller is. The region of attraction is the set of all initial points that are controllable by the controller. The initial point attempted by the controller is the fixed point of the system with a perturbation amount added to it. The difference between the two techniques discussed are their ranges of controllable initial points. The Eigenvalue approach is only useful when the perturbation amounts are very small, and the region of attraction method is useful when exploring larger perturbation amounts. The region of attraction is used to compare stability of controlled and uncontrolled cases when using a Discrete Control Lyapunov Function (DCLF) for The Simplest Walker. [5] For that study, the region of attraction was calculated by perturbing the initial velocity of the walker and simulating fifty steps. The maximum and minimum controllable initial velocities depict the bounds of the region of attraction. The region of attraction can also be calculated via the cell mapping method, where a cell represents several controllable points within the region of attraction. [22] In addition to using the DCLF approach for controlling The Simplest Walker, a dead-beat controller can be used to fully recover from the disturbance in a single step. [23]

CHAPTER 3: METHODOLOGY

The research of this thesis is centered around the control of a simple passive bipedal walker. In order to control a passive walker, it first must be shown that the walker can be perturbed enough to fall over. Once the walker has effectively been disrupted sufficiently enough to fall over, a controller can be designed to assist with the leg movement to keep the walker upright. The implementation of a controller will supply a torque to the hip when needed to maintain the stability of the walker. Due to the added torque, the walker is no longer considered to be passive. The resulting powered walker is described as a passive-dynamic based walker. [4] To successfully study the effects of a controller, it is important to determine the robustness of the implemented controller. For this thesis, an applied torque at the hip is added to the simplest walker model [20], with the goal of implementing a controller to keep it stable upon a disruption.

3.1 Achieved by the simulation

The focus of this thesis is not to create a simulation of The Simplest Walker, but instead to simulate the control needed to keep the walker moving. The use of a simulator accompanied by an animation will aid in the process of adequately constructing a controller capable of preventing the walker from falling. The simulation used for testing the controller discussed in this thesis is initially uncontrolled and unperturbed. It simulates the walker traversing a ramp that, for the ease of demonstration, has been rotated to appear as if it is walking on level ground, though it is at a designated angle of γ. Following the animation depicting the successful trek of the two-legged walker, a two-dimensional graph displays the θ angles and ϕ angles versus time. This graph can easily show if any hiccups happen during the journey of the walker because it will disrupt the typically cyclic path of motion.

3.2 Exploration of the simulation

The simulation begins with defining the physical parameters of the walker and the initial state at which the legs begin motion. Upon initializing the model, root finding is run to determine the period-one gait and establish the fixed point of the system. Using the newly determined fixed point, the Jacobian is solved for numerically. The Jacobian is used to determine stability by evaluating the Eigenvalues of the Poincaré Map. As previously mentioned, the magnitude of the Eigenvalues should return as less than one, depicting stability of the system. The Jacobian is calculated numerically using the central difference technique. The central difference technique is derived using Taylor series expansions, and evaluates a function of x, i.e. $f(x)$, from points on the left and right side of the point x. To produce quality results, the points will have to be chosen to be an equal distance from x on either side. This distance is typically a low value, in the proximity of one hundredth (0.01) of a unit or smaller, that is added and subtracted to x in the function. The central difference technique is true to its name, and calculates the difference between the function with the distance added and the function with the distance subtracted. This difference is then divided by a factor of two times the distance to produce an approximation of the Jacobian. The formula for the central distance numerical approximation is shown in Equation 3.1. [24]

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \tag{3.1}$$

The function used within the central difference calculations is the process used to calculate the data for taking one step, and the x value is represented by the fixed point position. The distance away from the fixed point used to approximate the Jacobian is set at one hundred thousandth (0.00001) of a unit to ensure an accurate approximation. The function used for processing the action of taking a step goes through a series of different events. Processing each step requires collision detection between the feet and the ground to determine the moment of heel-strike, the original starting and stopping position. Actions of the walker during foot-strike and single-stance are also calculated to determine the final state of the walker after one step. This step function is used in the central

difference calculations to determine the Jacobian of the linearized system. Then the Eigenvalues of the Jacobian are calculated to help determine the stability of the system. Now that the stability is determined, the data for all the steps can be collected. Collecting this data is done using the same equations used in the Jacobian calculations for taking one step, but is now done over the course of ten steps. The initial state of the bipedal walker is fed to the function as initial conditions and produces the final state after one step. The newly determined final state becomes the initial state for step two, which is then used to produce the final state after step two. This process of the final state becoming the initial state for the next step is continued for the entirety of all ten steps. Each time the simulation calculates the next step, nine variables are collected to help depict what the walker is experiencing through that step. These nine variables include θ, $\dot{\theta}$, ϕ, $\dot{\phi}$, the total energy of the walker, the distance in the X direction, the velocity in the X direction, the height in the Y direction, and the velocity in the Y direction. These nine ever-changing values are sent to the function responsible for calculating data points from one step, and uses the output as the next input. The goal of the walker is to return to the same fixed point at the same instance in a step as before, therefore, if the fixed point is displayed for each step, it should never change. With the completion of calculations for the data of taking ten steps, the process of animating the walker can begin. Animating the walker is done by initializing the frame the walker will be drawn in and then creating objects for the hip, legs, and feet. With the created objects in place, the initial location can be set and displayed to the frame. The animation function updates the location every iteration and then redraws them to the frame. Redrawing to the frame will, by default, erase the old data and incorporate the new data; in other words, overwrite the old information. Once the animation of taking ten steps is complete, the same data is used to create a two-dimensional plot of the θ angle versus time, and the ϕ angle versus time. Both graphs are displayed in the same frame, allowing for easy exposure to the limit cycle. The produced graph demonstrates the oscillatory motion of a limit cycle with a period length equivalent to one step.

Table 3.1: A description of the system for the use in the original simulation.

(a) System Parameters

Description	Variable	Value
Hip mass	M	1000
Leg mass	m	1.0
Leg length	l	1.0
Leg inertia	I	0.0
Foot radius	r	0.0

(b) Initial State

Variable	Value
θ	0.2
$\dot{\theta}$	-0.2
ϕ	0.4
$\dot{\phi}$	-0.3

3.3 Original simulation results

The original code initializes the model with the parameters and initial state shown in Table 3.1. The initial state represents the initial values for θ, $\dot{\theta}$, ϕ, and $\dot{\phi}$, and the parameters are described in the table. The radius of the foot is set to zero to eliminate the feet of the walker; this is done to create a model similar to one in literature. To help provide validation for the results from the simulation, the model used comes from The Simplest Walker model. [20] The fixed point of the system is calculated and is equivalent to what was solved for by Garcia et al., shown below. Following the fixed point is the Jacobian of the linearized system with its calculated Eigenvalues depicting stability.

The fixed point:

$$zstar = \begin{bmatrix} 0.200161 & -0.199906 & 0.400322 & -0.015805 \end{bmatrix}$$

The Jacobian:

$$J = \begin{bmatrix} -6.865237 & -6.870654 & 0.403433 & 0.328762 \\ 5.690368 & 5.734742 & -0.287708 & -0.233780 \\ -13.730474 & -13.741309 & 0.806867 & 0.657524 \\ 1.519596 & 1.523949 & -0.085607 & -0.069709 \end{bmatrix}$$

The Eigenvalues of the Jacobian:

$$eig(J) = \begin{bmatrix} -0.196667 + 0.544727i \\ -0.196667 - 0.544727i \\ 0.259 * 10^{-9} + 0.000000i \\ -0.261 * 10^{-9} + 0.000000i \end{bmatrix}$$

The simulation is to run for ten steps, and the data of those ten steps is computed with an initial position at the heel-strike phase, shown in Figure 3.1. Upon completion of the animation, the graph displaying the θ angle versus time, and ϕ angles versus time, is produced. The graph clearly depicts a limit cycle with a period length of one step, showing a total of ten periods. The angles versus time graph can be seen in Figure 3.2.

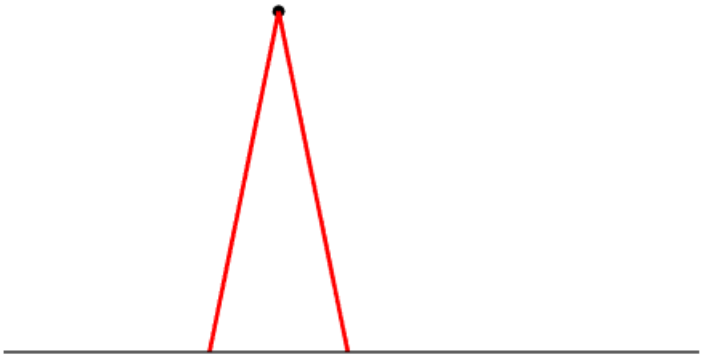


Figure 3.1: Initial position of the original walker

3.4 Un-perturbed and Uncontrolled system

The initial position of this thesis was defined as being at the mid-stance position, so a few changes were needed in the passive walker simulation. To accomplish the change in initial position, the

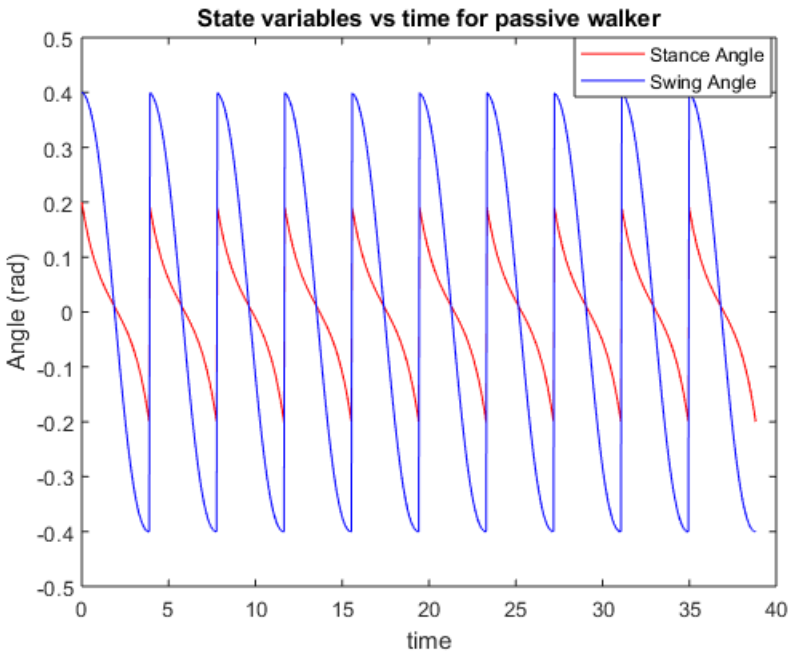


Figure 3.2: Graph of θ and ϕ angles versus time for the original walker

initial state was changed such that θ was equal to zero. A check for the mid-stance position was also implemented to provide a stopping point for the simulation after one step. Incorporating the necessary changes into the simulation produces the starting position shown in Figure 3.3. The leg angles versus time graph is still produced, now depicting the new initial starting stance, shown in Figure 3.4. The slightly modified simulation still produces the desired results of a walker taking ten steps, but for future experiments a few more modifications are required. To better illustrate the fixed point and the values at the mid-stance position and the end of each step, the simulation was changed to take one step ten times versus ten consecutive steps. This change will allow us to display the initial position at the start of each step to see that the initial position is converging back to the fixed point.

3.5 Perturbing the System

Perturbing the system is a critical part in studying the control of the walker to keep it upright; without perturbing the system no controller would be needed. To perturb the walker, the initial

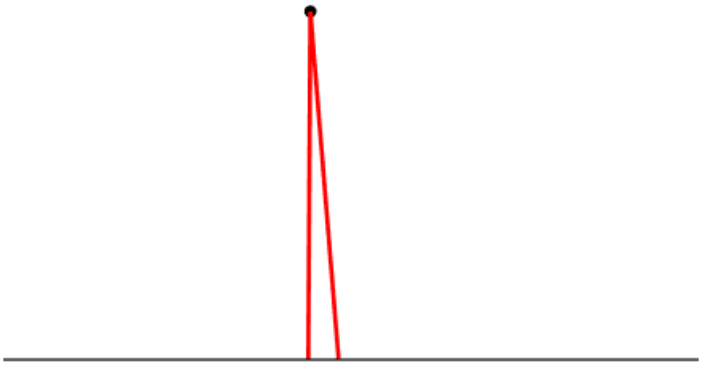


Figure 3.3: Initial position of the mid-stance position walker

position must be changed. The initial position of the walker is typically the fixed point of the system, so by moving the starting point away from the fixed point we want to see if the system is capable of correcting its motion to divert its path back to the limit cycle. The fixed point in the simulation is referred to as $zstar$, so to perturb $zstar$ a perturbation vector is added, shown by Equation 3.2.

$$perturbation = \begin{bmatrix} element1 & element2 & element3 & element4 \end{bmatrix}$$

$$z0 = zstar + perturbation \tag{3.2}$$

In order to maintain the starting position at mid-stance, the first element of the perturbation vector must remain zero. If the first element in the perturbation vector were not zero, the walker would be starting at a different position besides mid-stance. In order to begin controlling the walker, the walker must first fall over from being perturbed. The walker falling over can be seen in the animation and depicted in the graph of the leg angles versus time. The perturbed walker can be seen with the initial position in Figure 3.5 and depicted falling over in Figure 3.6. The Perturbation

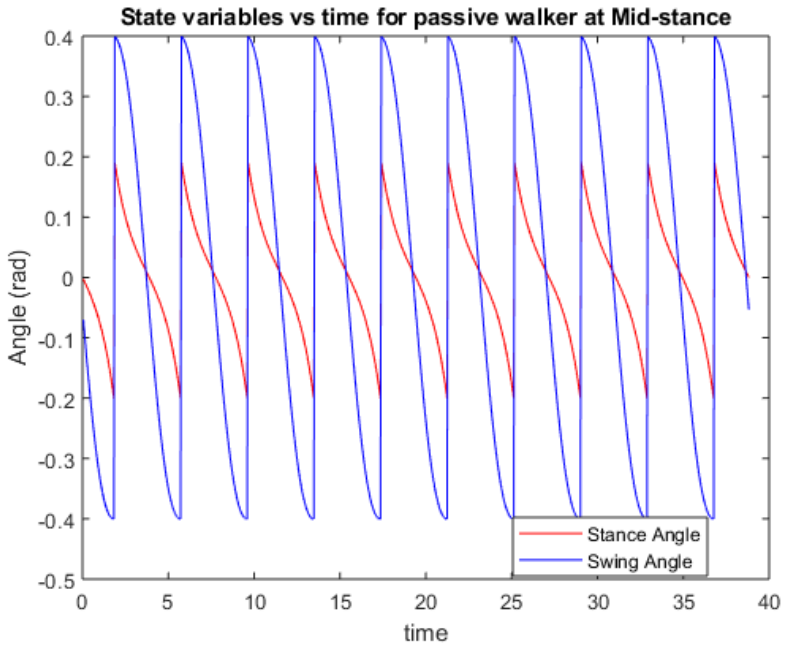


Figure 3.4: Graph of θ and ϕ angles versus time for the mid-stance position walker

vector used to perturb the system can be seen below.

$$perturbation = \begin{bmatrix} 0 & 0.055 & 0.041 & 0.035 \end{bmatrix}$$

As seen in Figure 3.5, the initial position is almost identical to the un-perturbed walker initial position. The reason the perturbed walker becomes unstable and falls is largely due to the small $\dot{\theta}$ value. The size of the value indicates that the stance leg of the walker is slowly rotating forward, forcing the walker to begin falling backwards. The movement of the swing leg prevents the walker from falling backwards, but upon rolling forward, the walker goes completely unstable and falls.

3.6 Controlling perturbed walker

Now that the walker displays the capability of being perturbed to the point of falling, a controller can be designed to keep the walker stable and upright. The controller of choice for this thesis is a Discrete Linear Quadratic Regulator (DLQR). The goal of a DLQR is to optimize the gain matrix K such that the feedback law δ_{u_n} minimizes the quadratic cost function J for the discrete time

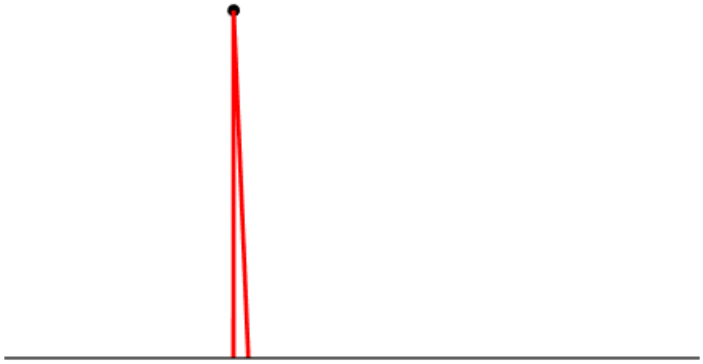


Figure 3.5: Initial position of the perturbed walker

state space mode represented by $\delta_{x_{n+1}}$. The discrete time state space model is shown by Equation 3.3, the feedback law is shown in Equation 3.4, and finally the quadratic cost function is shown in Equation 3.5.

$$\delta_{x_{n+1}} = A\delta_{x_n} + B\delta_{u_n} \tag{3.3}$$

$$\delta_{u_n} = -K\delta_{x_n} \tag{3.4}$$

$$J(u) = \sum_{n=1}^{\infty}(\delta_{x_n}^T Q\delta_{x_n} + \delta_{u_n}^T R\delta_{u_n} + 2\delta_{x_n}^T N\delta_{u_n}) \tag{3.5}$$

The parameters A, B, Q, R, and N in the DLQR formulation above represent the system and penalties for different vectors in the cost function. The parameter A specifies the state matrix of the system of size nxn, where n is the number of states in the system. B is the input matrix of size nxm, where m is the number of inputs. Q is a symmetric positive semi-definite matrix that penalizes the state vector δ_{x_n} in the cost function J. R is a symmetric positive definite matrix responsible for penalizing the input vector δ_{u_n} in the cost function J. The parameter N represents a matrix penalizing the cross product between the state vector and the input vector. By default, the parameter N is set to be a matrix of zeros the same size as B, and for the control of the

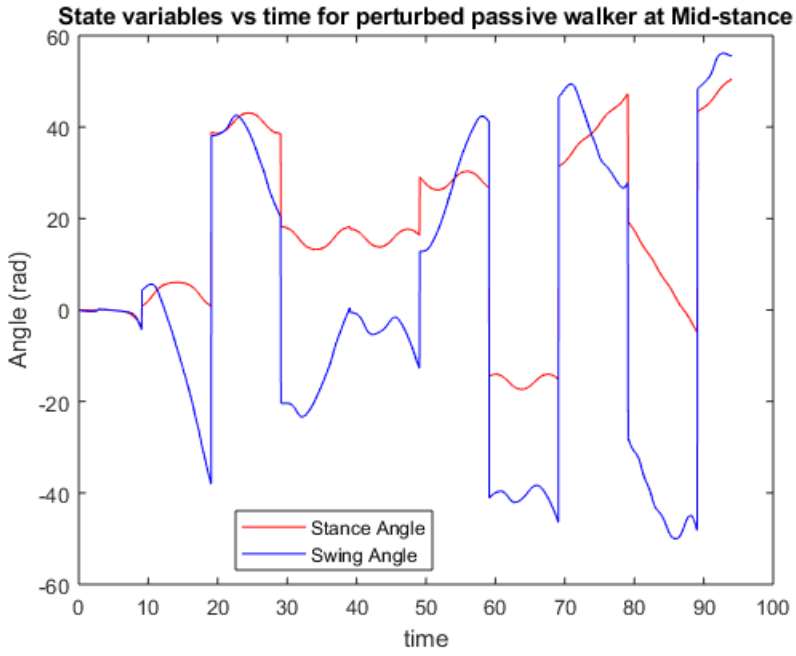


Figure 3.6: Graph of θ and ϕ angles versus time for perturbed walker

walker the default is used. Defining the cross product weighing matrix to be zero means it can be removed entirely from the cost function. The reason for using a matrix of zeros for N is to test the effects of the state weighing matrix Q and the input weighing matrix R only. The original cost function in Equation 3.5 can be simplified to create a new cost function shown in Equation 3.6. The simplification of the cost function is to study the effects of just the state weighing and input weighing matrices.

$$J(u) = \sum_{n=1}^{\infty} (\delta_{x_n}^T Q \delta_{x_n} + \delta_{u_n}^T R \delta_{u_n}) \tag{3.6}$$

With the required input parameters defined, let's start identifying them in the simple passive walker system. The state matrix of the system is the Jacobian, calculated via the central difference technique. The state matrix is defined as $A = \frac{\partial \delta_{x_{n+1}}}{\partial \delta_{x_n}}$, where using the central difference technique produces a numerical derivative. To produce the input matrix for the walker, the central difference technique is used again. The input matrix is slightly different, defined as $B = \frac{\partial \delta_{x_{n+1}}}{\partial \delta_{u_n}}$ with δ_{u_n} defined as the control vector or feedback law. The method of using the numerical finite difference

technique for calculating the sensitivity matrices A and B is classified as a brute force method. [25] Once the state matrix and input matrix have been numerically solved for, DLQR calculations can begin. δx_{n+1} is the state's final position of that step, and δx_n is the initial state position for that step. The optimal values for the gain matrix K, solved for in the DLQR calculations, can be changed by tuning the inputs Q and R to meet any specified criteria. Implementing the controller is done by applying a certain torque at the hip. Luckily, the simulation used for these experiments allows for easy implementation of a torque at the hip. The original simulation has a defined torque set to zero, so there was no applied torque, but there was the possibility of applying it. The easy implementation of a torque at the hip allows for easier testing of control techniques on The Simplest Walker Model. The torque applied was specified by the controller architecture, which is the gain matrix K multiplied by the deviations in the state δx_n. With the controller implemented into the simulation, tuning of the state and input weighing matrices can begin. To begin, the state weighing matrix Q is defined as the identity matrix of size nxn. The variable n represents the number of states of the system, leading to the identity matrix of size 3x3. The state matrix is of size 3x3 because setting the Poincaré section at mid-stance reduces the number of states by one. The size reduction is due to θ being set to zero to meet the mid-stance starting position, therefore, to maintain the correct starting position θ must never be perturbed. The input weighing matrix R is of the size 1xm with m being the number of control variables. The number of control variables for The Simplest Walker is only one, the angle between the legs right before the heel-strike phase, ϕ^-. The starting point for defining the R matrix is the identity matrix of size mxm multiplied by a constant ρ, which can be varied to achieve a good response. For the first test of the designed controller, ρ was defined as 1. Graphing the leg angles versus time will demonstrate how the walker is acting during the ten attempted steps, producing Figure 3.7. The simply designed controller is capable of supplying an adequate amount of torque to the hip to keep the walker upright. The designed controller was able to lead the walker back into its natural cyclic walking pattern, as shown in Figure 3.7. The same perturbation amounts that were experimented with in causing the walker to fall over were also used in this experiment. Using the same perturbation amounts guarantees that the walker will fall

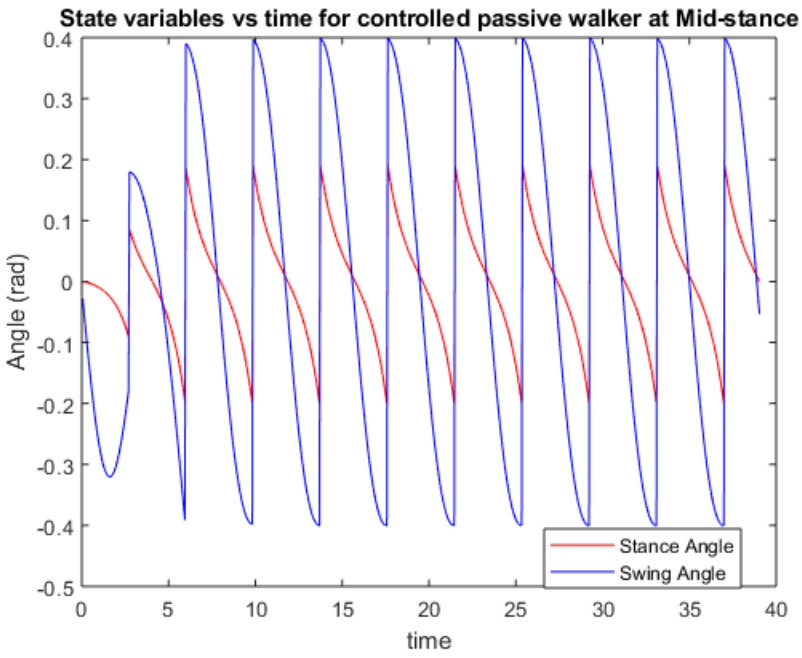


Figure 3.7: Graph depicting the perturbed walker starting at mid-stance and converging to the limit cycle.

over unless it is controlled, so the ability to accomplish reentry onto the defined trajectory shows that a successful controller was designed. Now that a working controller can be implemented and demonstrate its ability to drive the walker back to its natural path, lets determine its strength. The method of examining the region of attraction will be used to determine the range of the controller's capabilities. The region of attraction is the set of initial starting points that can be controlled and brought to the defined trajectory of the walking gait. To accurately define the region of attraction, the last three elements of the perturbation vector must be changed and attempted for a wide range of values. Remember, only the last three elements of the perturbation vector can be changed in order to maintain the mid-stance starting position. Proper setup will allow for elements two, three, and four to be changed for each iteration of running the simulation. This method of testing the strength of a controller is computationally expensive, but the tradeoff is that it is able to detect larger perturbation amounts that can be controlled. In order to reduce simulation time, the initial range of the region of attraction was set to begin at -0.15 and increase to 0.15 with a step size of 0.01, allowing for a total of 29,791 perturbation vectors to be tried. The simulation needs to

be configured to not allow the perturbation vector of all zeros to be tested. The vector of zeros is caught right after defining the perturbation vector and immediately breaks and begins with the next perturbation amount. Once the vector of zeros is ignored, the simulation must be adjusted to catch perturbation amounts that lead to falling. Detecting if the walker is falling is done by reading certain data values: the height of the hip in the Y direction, the position of the hip in the X direction, and the first value of $\dot{\theta}$. The check point for reading these values is done within the simulation of taking ten steps, whereas the zero vector is caught before the simulation of ten steps begins. After calculating what data points the walker hits during taking one step, the simulation observes the three criteria to see if the walker falls during that step. When looking at the height of the hip in the Y direction, the walker is defined as falling if at any point within the step the hip falls below the floor, i.e. is less than zero. If the hip position in the X direction ever falls below zero, it is falling backwards. The $\dot{\theta}$ value is only observed at the beginning of a step; if the value is greater than zero, then the walker is rolling backwards, thus it will not be able to be driven to the defined trajectory of the Poincaré Map. If any of the three conditions is detected then the simulation will stop calculating the ten steps, and will move to the next perturbation amount to be tested. Once incorporating the designed controller into the newly modified simulation, its region of attraction can be determined. The designed controller is the basic starting point for tuning a DLQR controller; it is tuned using the identity matrix as weighing matrices. Now, let's explore a better way of tuning the Q and R weighing matrices. As stated before, the primary method of tuning a DLQR controller is done by a trial-and-error approach. Tuning the weights is an iterative process done by analyzing the results and then making certain adjustments. Designing a different controller for the system requires some more thought than the original identity matrices used. It's important to recall that the Q and R matrices must be symmetric positive semi-definitive, and symmetric positive definite, respectively. In the case of The Simplest Walker, there is only one control input, so the R matrix will always be symmetric, therefore, as long as the number is positive, it is acceptable. The easiest way to ensure a symmetric matrix is to use a diagonal matrix, thus Q will be defined as a diagonal matrix with values greater than or equal to zero. There are many possibilities when defining the

diagonal matrix Q, but for this thesis the capabilities of the Q defined below are explored. Defining the diagonal of Q begins with trial-and-error, yet the incorporation of the fixed point elements creates a dependency on the fixed point. In addition to using the values of the fixed point, x^*, in the state weighing matrix, they each will also be multiplied by a different constant, represented by $q1$, $q2$, and $q3$. The Q matrix can be seen below, and Equation 3.7 helps to explain the construction of Q.

$$Q = \begin{bmatrix} \frac{1}{d1} & 0 & 0 \\ 0 & \frac{1}{d2} & 0 \\ 0 & 0 & \frac{1}{d3} \end{bmatrix}$$

$$di = qi * x_i^* \qquad i = 1, 2, 3 \tag{3.7}$$

The capabilities of the Q matrix are explored to determine how it alone can affect the region of attraction. For the study of the Q matrix, the R matrix was defined as being ρ, where ρ is equal to 1. Once implementation of the new Q matrix is successful, the limits of the variables $q1$, $q2$, and $q3$ are explored. Every combination of $q1$, $q2$, and $q3$ represents a different controller, and for each controller the walker simulation will need to be run. To cut back on computational time, the number of different controllers was limited to one hundred. The values of $q1$, $q2$, and $q3$ increase over the course of creating the one hundred different controllers. There are four possibilities for $q1$, five possibilities for $q2$, and five possibilities for $q3$. The range for $q1$ was set to be from one to thirteen, with a step size of four. The range for $q2$ and $q3$ was set to be from one to sixty-one with a step size of fifteen. To run all 100 different controllers required a large amount of time, but once all results were calculated, the values for $q1$, $q2$, and $q3$ could be better represented. For determining the strength of the different controllers, a percentage of how many of the test points were controllable was calculated for each controller. Using these results, a graph can be produced to create a trend-line of the controller's strength. This trend-line can be used to determine what type of $q1$, $q2$, and $q3$ values to use for testing different ρ values. Another trend line can be drawn

to help determine what type of ρ value will help the controller. For testing the effects from ρ, four different values were used. To test the effects of the input weighing matrix, the simulation was configured to begin with ρ defined as 0.01 and increase to 0.61 in steps of 0.2. The range of ρ is minimal, but this is intentional to guarantee the calculations within the DLQR controller are successful. The previous set of controllers consisted of the diagonal of Q being less than one; a similar simulation can be run to determine the effects of values on the diagonal being greater than one. The new Q matrix definition can be seen below, with Equation 3.8 describing the definition of the diagonal.

$$Q = \begin{bmatrix} d1 & 0 & 0 \\ 0 & d2 & 0 \\ 0 & 0 & d3 \end{bmatrix}$$

$$di = qi * x_i^* \qquad i = 1, 2, 3 \tag{3.8}$$

The new values of $q1$, $q2$, and $q3$ can be determined by using a similar style as before. The new range for $q1$, $q2$, and $q3$ is set to be from one to sixty-one with a step size of thirty. The new ranges allow for only twenty-seven controller possibilities, but also the larger step size allows for the affects to the weighing matrices to be easily shown. Using the same tactic as before, a percentage of the total controllable points can be calculated. Once all the results from the different controllers are produced, another trend-line can be mapped, and the results of the different values in Q can be interpreted. Once the controller with the largest controllable percentage is determined, the state weighing matrix configuration can be used to determine what value of ρ is better suited for maximum control. The effects of ρ were tested using the same technique used before.

3.7 Region of Attraction

The region of attraction is a key part to interpreting the results of the different controllers created for this thesis; it demonstrates how robust the design of the controller is. The region of attraction

consists of all the points that can be controlled by the controller, so it typically takes quite some time to generate and can be difficult to interpret. The simplest walker being studied starts at the mid-stance position, which reduces the region of attraction by one dimension. The reduction is due to the first element of the state, θ, being restricted to remain at zero. The produced region of attraction is plotted as a three-dimensional shape and, due to lack of shading, the exact shape can be difficult to see. The technical region of attraction is a three-dimensional shape, but because of all the initial points being tested, the computational time is long. It has been mentioned before that the region of attraction method is computationally expensive, so this approach to receive a three-dimensional plot that can be difficult to interpret is often not appealing. Viewing the region of attraction in two dimensions is more aesthetically pleasing, so the three-dimensional shape can be viewed by layer. This can be done be setting the perturbation amount of either $\dot{\theta}$, ϕ, or $\dot{\phi}$ as a constant while the other two are increasing by the predetermined step size. Calculating the region of attraction with a single value set as a constant will also allow for a wider inspection. This technique can be used to examine certain layers, allowing the region of attraction to be viewed at the outer, middle, or even every tenth layer.

CHAPTER 4: RESULTS

In this section, I will be discussing the results collected from various simulations conducted to gain insight of what effect the weighing matrices have on the system and how they can be interpreted. The results of the unperturbed and uncontrolled system have already been shown, as well as the results from the perturbed and uncontrolled system. In addition, the system has already demonstrated the capabilities of being controlled by a simple controller comprised of the identity matrix, but the focus was on tuning a non-identity weighing matrix. This section will focus on the results from that non-identity weighing matrix and the exploration of expanding the region of attraction.

4.1 Region of attraction exploration

To begin, the region of attraction of the uncontrolled walker is calculated, shown in Figure 4.1. The uncontrolled walker is capable of remaining upright and converges to the limit cycle for 6.48% of the perturbation amounts attempted. It makes sense that the value is lower because the controller is turned off, yet it also displays that the simplest walker has some leeway if the exact initial conditions are not met at takeoff. The perturbation amounts used to prove the walker can be perturbed enough to fall over were also used with the first controller design. Implementation of the basic controller demonstrates the walker being controlled to the point of converging back to the limit cycle. Building the region of attraction will provide a metric to determine the capabilities of the controller. The region of attraction for the identity controller can be seen in Figure 4.2. The identity controller is capable of controlling 31.07% of the total initial points attempted, providing a standard for the other controllers. The next controller design defined the diagonal of the state weighing matrix to be less than one. Several controllers were tested with this technique, and for each one the region of attraction was calculated. Once all the points were calculated, a percentage of how many of those points were controllable was generated. A trend-line was created by graphing the controllable percentage versus the controller number, shown in Figure 4.3. At first glance this constructed graph doesn't appear to provide any helpful information, but upon further inspection,

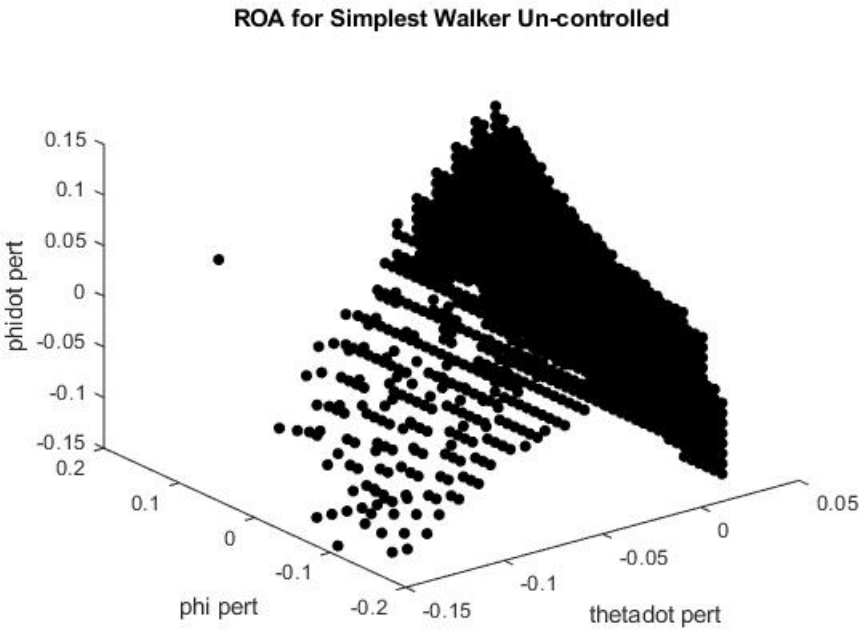


Figure 4.1: The Region of Attraction calculated for the Uncontrolled Simplest Walker

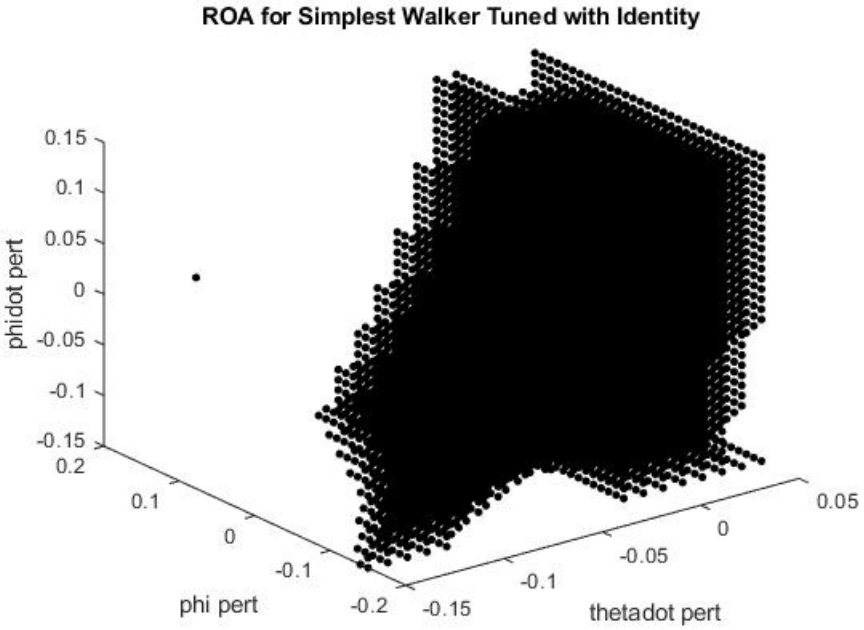


Figure 4.2: The Region of Attraction calculated for the Simplest Walker controller tuned using the identity matrix

the values of $q1$, $q2$, and $q3$ can be seen. The graph contains four main sections, each containing four peaks. These four main sections represent the four different possibilities for the value of $q1$. Examining the graph, it can be seen that the lower the value of $q1$ is, the higher the controllable percentage is. Zooming in on one of the four main sections, the effects of the $q2$ values are discovered. Figure 4.4 displays the first fourth of the overall graph. Within this section there exists five subsections, each containing a spike towards the beginning. The five subsections depict the five possibilities of the $q2$ value. From this observation, it can be concluded that $q2$ provides a higher percentage of controllable points when its value is higher. Zooming in once more, the values of $q3$ can be studied. Figure 4.5 focuses in on one fifth of the last figure to help visualize the values of $q3$. The new subsections are not as easy to distinguish, but with the help of grid-lines, they can be emphasized. The closer look at the values of $q3$ indicates that the lower the value, the higher the percentage of controllable points. A summary of the discoveries made from this approach can be seen in Table 4.1.

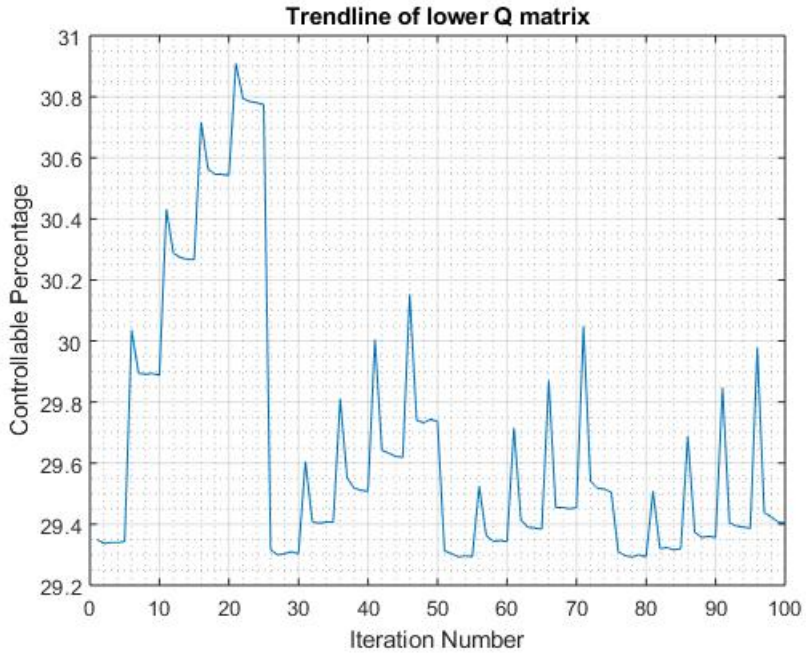


Figure 4.3: The trend-line depicting the effects of the smaller Q matrix.

The values for $q1$, $q2$, and $q3$ are not explicitly shown in the trend-line graph, so instead color

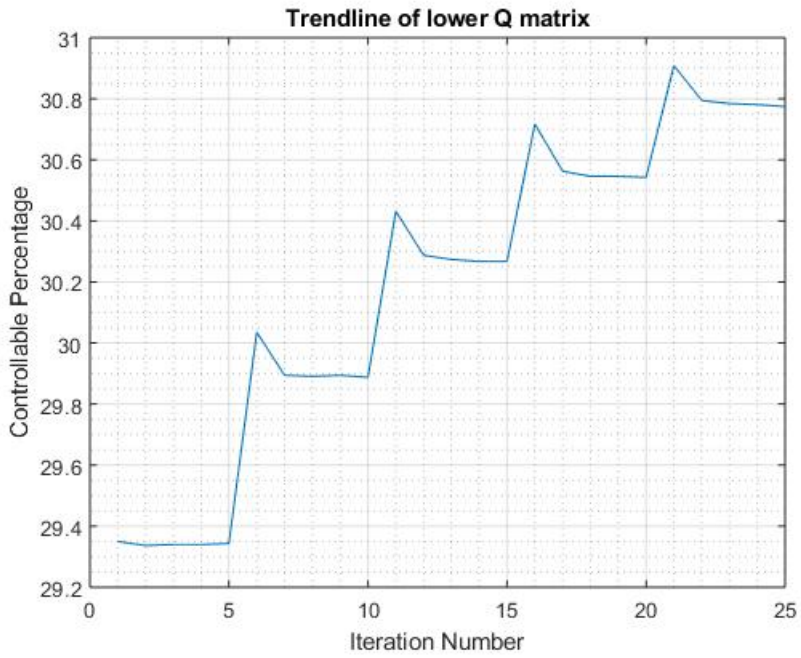


Figure 4.4: The trend-line depicting the effects of the smaller Q matrix, focusing on the first fourth.

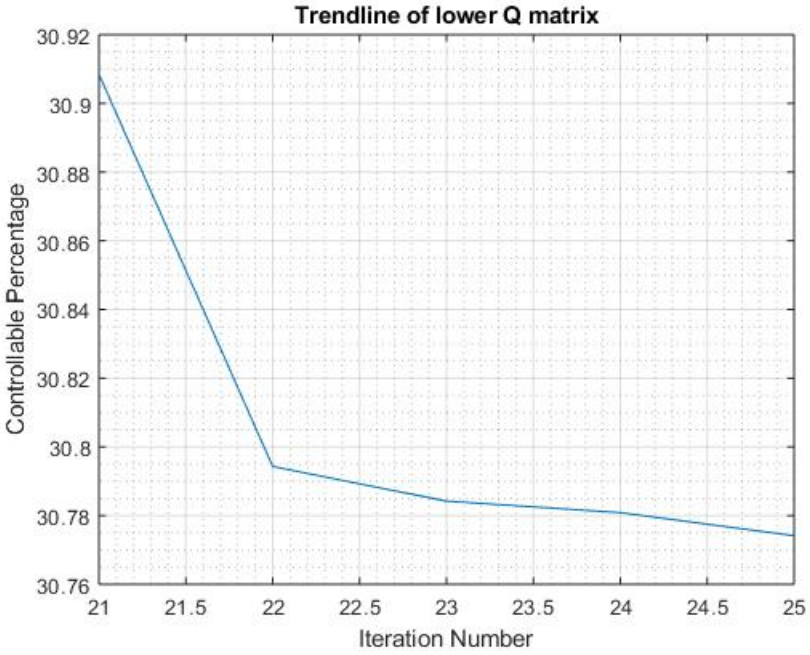


Figure 4.5: The trend-line depicting the effects of the smaller Q matrix, focusing on the section capable of the highest controllable percentage.

Table 4.1: A summary of the preferable $q1$, $q2$, and $q3$ values discovered from the trend-line of using a diagonal of the Q matrix less than one.

Variable	Type of Value
$q1$	Lower Value
$q2$	Higher Value
$q3$	Lower Value

scale graphs are produced to better illustrate the results. Figure 4.6 displays the color scale graph for $q1$ values. There is a corresponding color bar at the bottom of the graph, providing a numeric value associated with the $q1$ variable. Figure 4.7 and Figure 4.8 are the color scale graphs associated with $q2$ and $q3$ values respectively. Each color scale graph uses the same trend-line, but uses color values associated with the corresponding $q1$, $q2$, or $q3$ values. Using color to illustrate the values of $q1$, $q2$, and $q3$ adds another dimension to the results, essentially viewing the results on a two-dimensional plane from a three-dimensional perspective. This approach to viewing the results helps to better uncover the values of $q1$, $q2$, and $q3$ that are affecting the state weighing matrix.

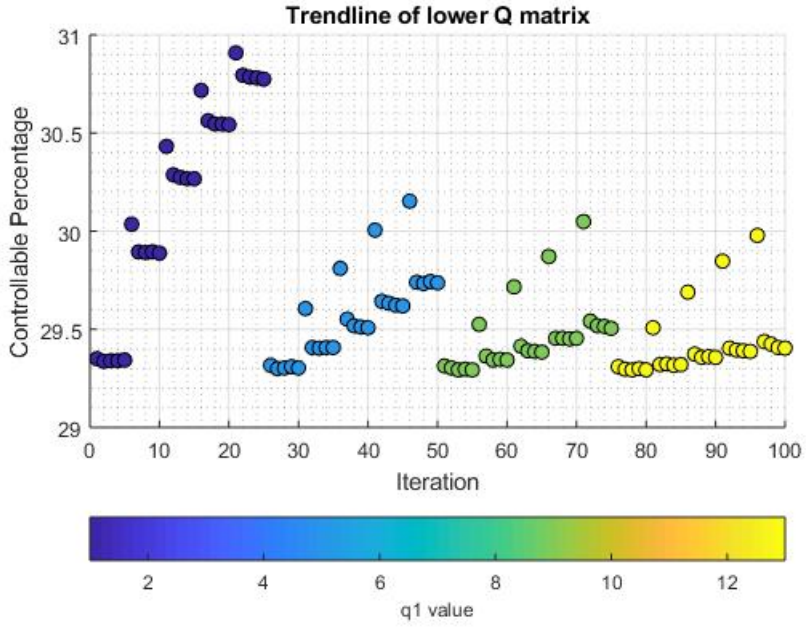


Figure 4.6: The same trend-line using color values to depict the values for $q1$.

Now that the effects of the state weighing matrix are known, the effects from the input weighing

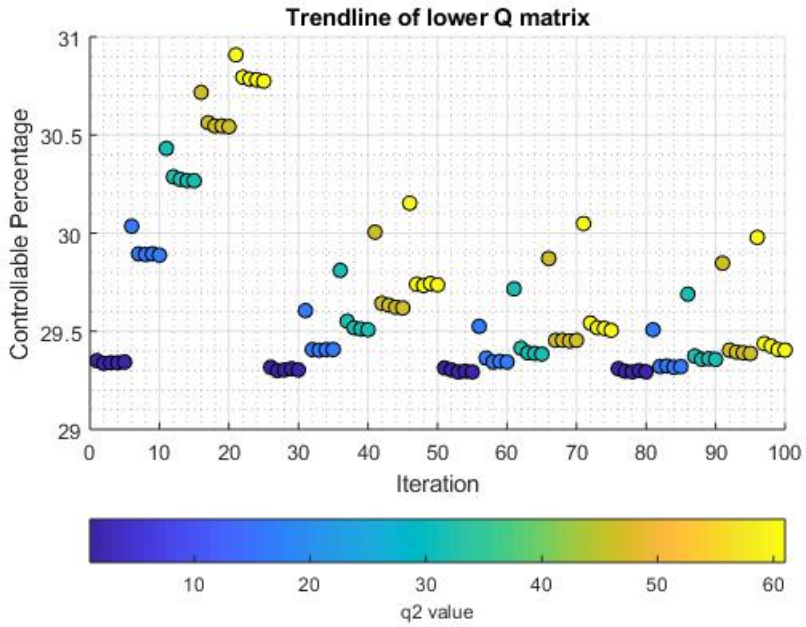


Figure 4.7: The same trend-line using color values to depict the values for $q2$.

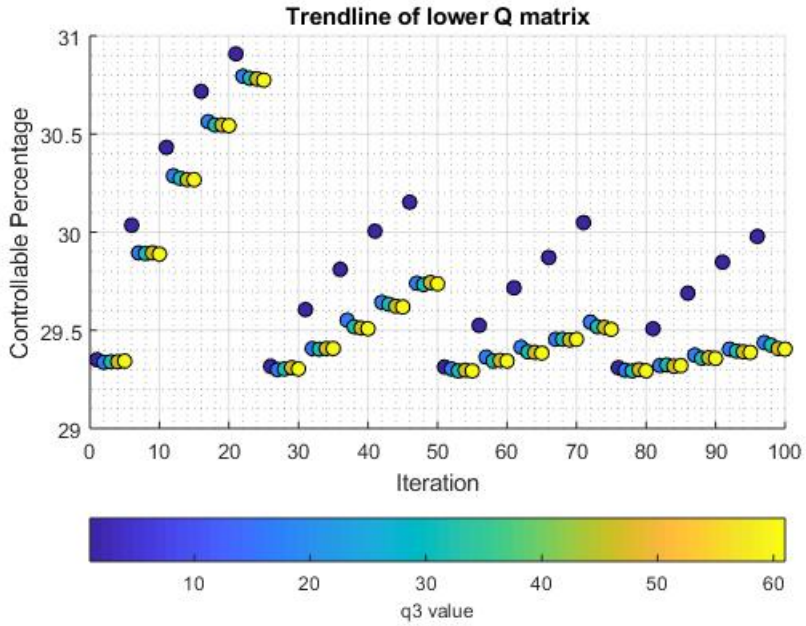


Figure 4.8: The same trend-line using color values to depict the values for $q3$.

matrix can be tested. To test the effects, the value of ρ is varied and tested with the arrangement of $q1$, $q2$, and $q3$ that provide that highest controllable percentage. To visualize the results, a trend-

line was also created to help observe the effects, as seen in Figure 4.9. Once again, the effects could be difficult for some to distinguish, so the color scale is implemented. The results can be seen in Figure 4.10, and display that the higher the value of ρ, the lower the percentage of controllable points, and the lower the value of ρ, the higher the percentage. Using the color scale approach to visualize the results helps to validate the observations from viewing the original trend-line. When

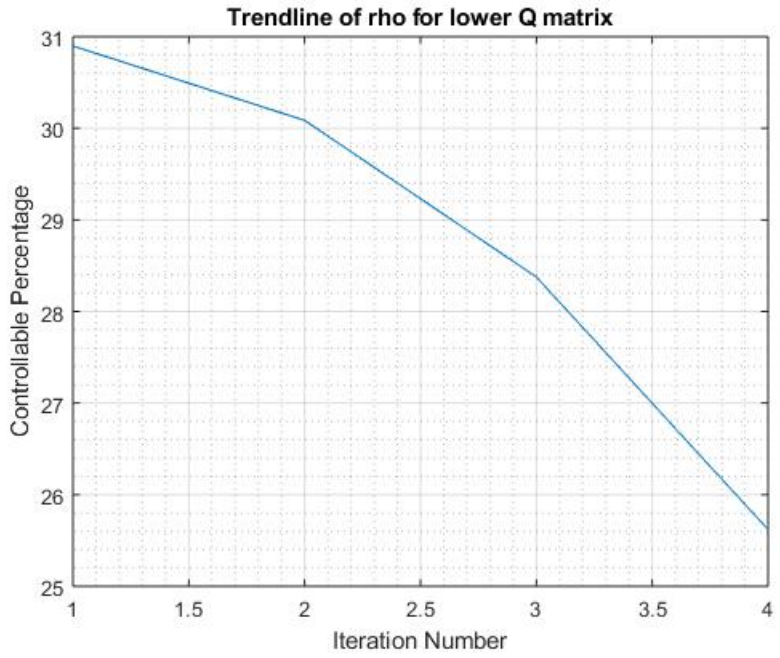


Figure 4.9: The trend-line produced to show the effects of adjusting ρ.

the Q matrix has diagonal values larger than one, the region of attraction is again calculated for several varying controllers. The same method of calculating a percentage of controllable points is conducted, and leads to the creation of the trend-line shown in Figure 4.11. The produced trend-line can be viewed by section once again to determine the values for $q1$, $q2$, and $q3$. This trend-line demonstrates different sections, but the difference between points is very small. To add the $q1$, $q2$, and $q3$ values to the graph, the color scale method is used once more. The color scale method is shown in Figure 4.12 for $q1$ values, Figure 4.13 for $q2$ values, and Figure 4.14 for $q3$ values. The respective figure indicates that if the value of $q1$ is higher, then the controllable percentage will be higher. In addition, if the value of $q2$ is defined as a lower value, then the controllable percentage

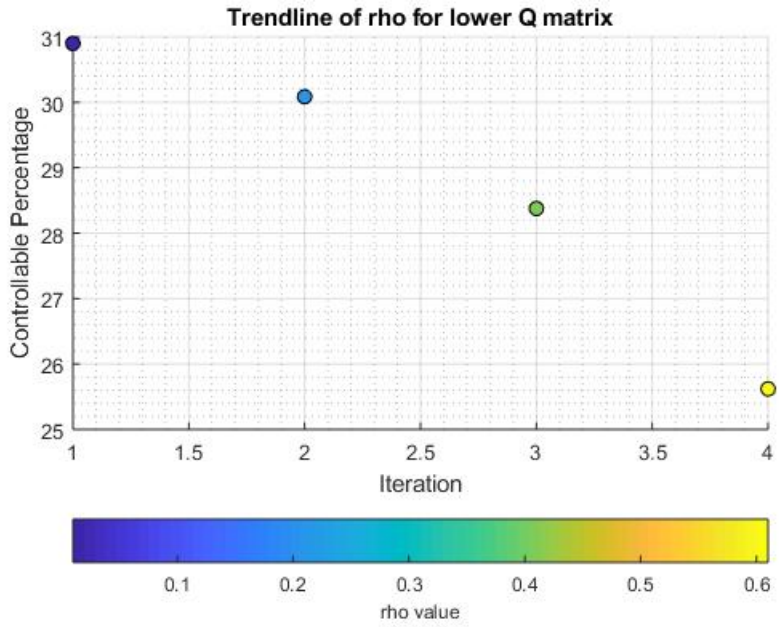


Figure 4.10: The trend-line produced to show the effects of adjusting ρ, now using the color scale.

Table 4.2: A summary of the $q1$, $q2$, and $q3$ values discovered from the trend-line of using a diagonal of the Q matrix greater than one.

Variable	Type of Value
$q1$	Higher Value
$q2$	Lower Value
$q3$	Higher Value

will be higher. Finally, looking at the $q3$ values in the same way shows that if its value is higher, then the controllable percentage will be higher. These conclusions are summarized in Table 4.2.

Once more, exploring the effects of ρ on the combination of $q1$, $q2$, and $q3$ values that produce the highest percentage of controllable points, a new trend-line can be drawn then inspected. The produced trend-line can be seen in Figure 4.15. Inspection of the trend-line leads to the discovery that the lower the value of ρ, the higher the percentage of controllable points will be. The color scale approach can confirm this finding, shown in Figure 4.16.

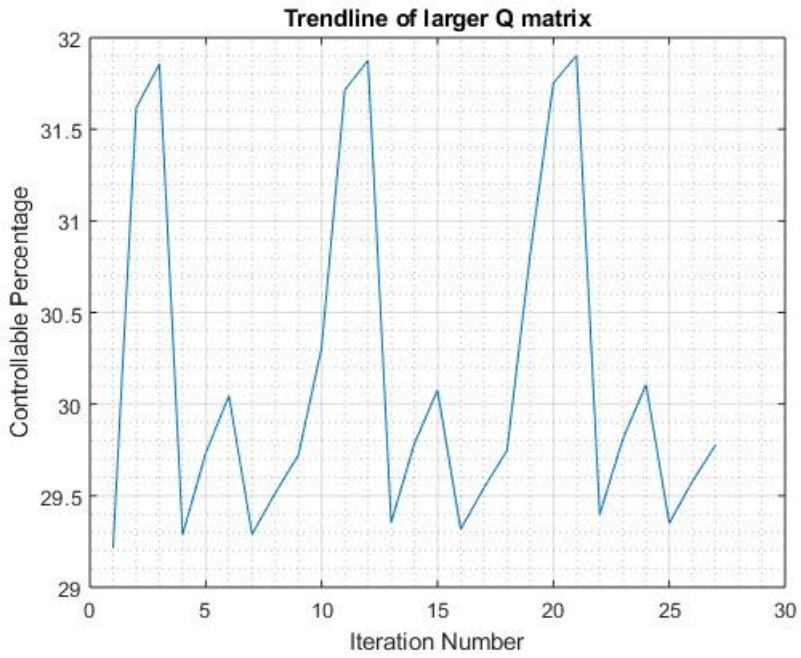


Figure 4.11: The trend-line produced to show the effects of a Q matrix with the diagonal greater than one.

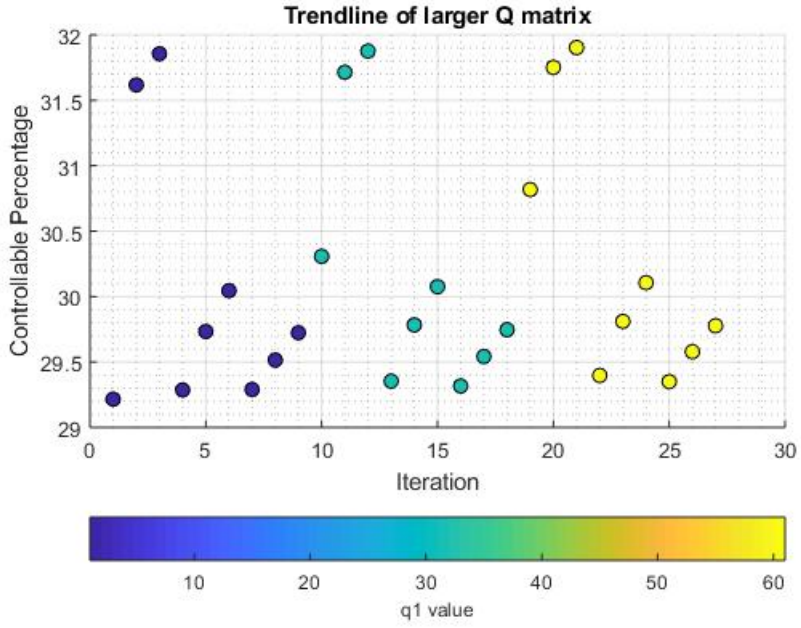


Figure 4.12: The same trend-line using color values to depict the values for $q1$.

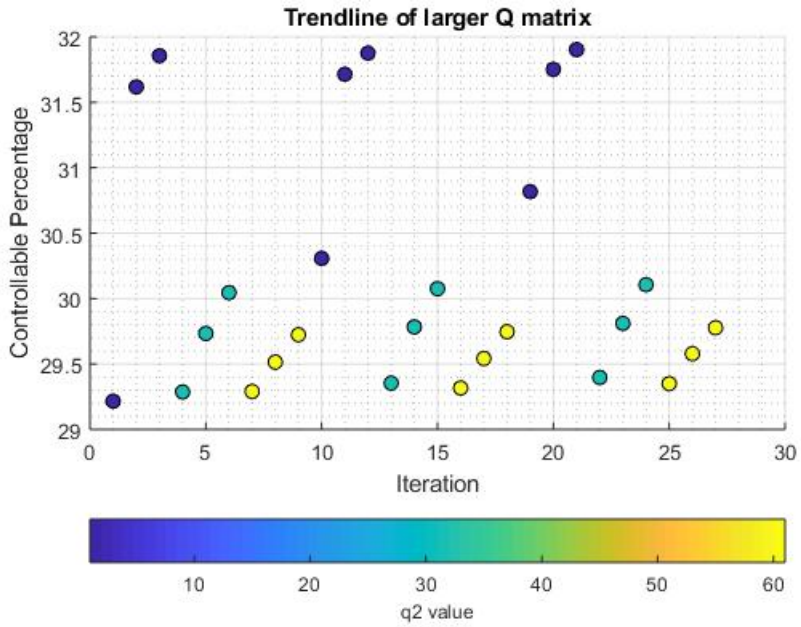


Figure 4.13: The same trend-line using color values to depict the values for $q2$.

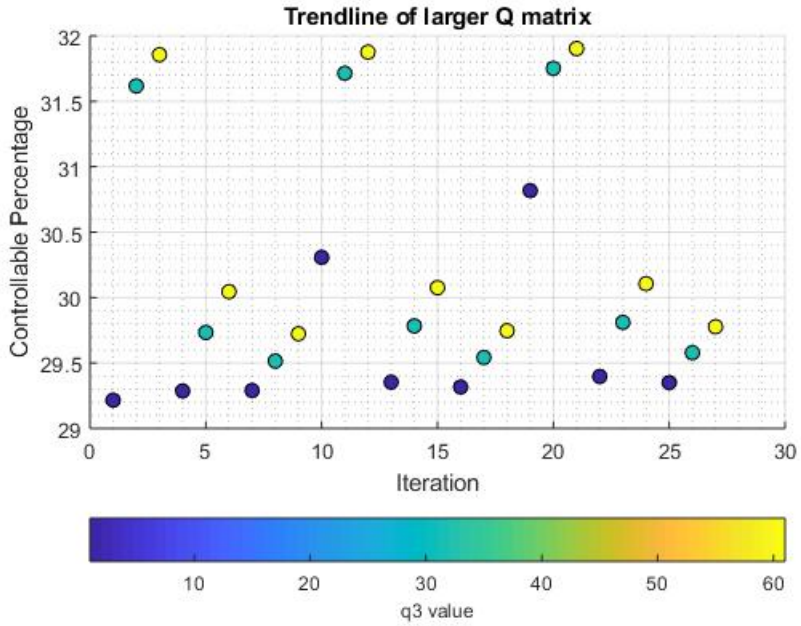


Figure 4.14: The same trend-line using color values to depict the values for $q3$.

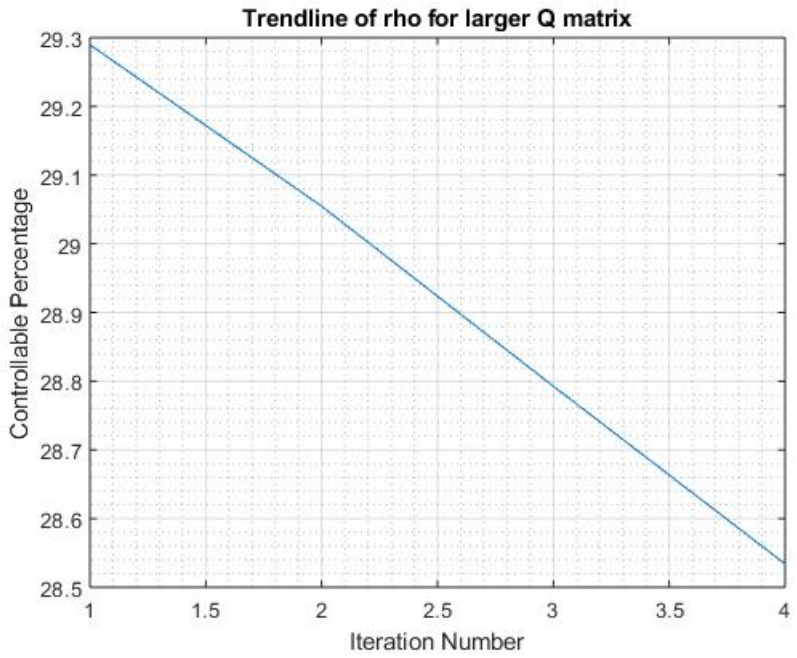


Figure 4.15: The trend-line showing the effects of changing the value of ρ for the Q matrix with the diagonal greater than one.

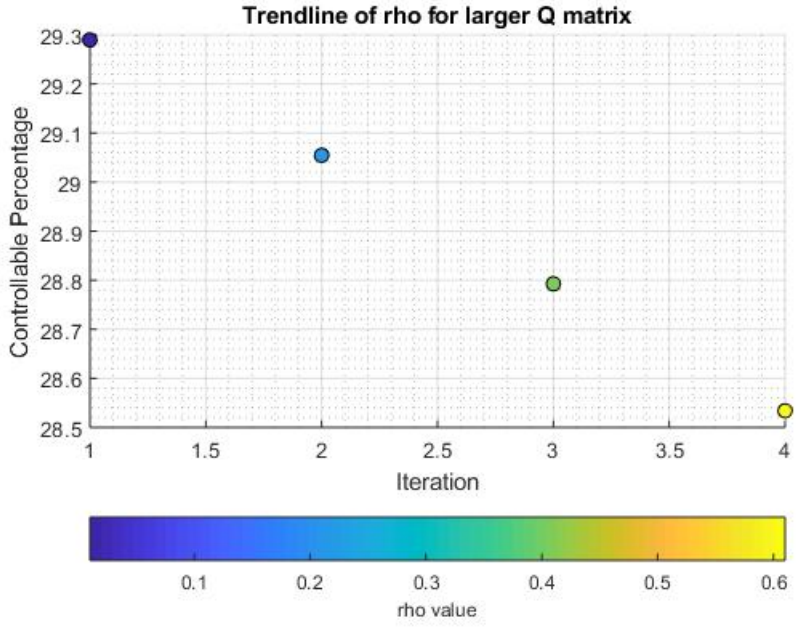


Figure 4.16: The same trend-line showing the effects of changing the value of ρ for the Q matrix with the diagonal greater than one, now using a color scale.

4.2 Controllable Percentage

The metric used for determining the capabilities of the different controllers was the calculated controllable percentage. The method used for tuning the controller using the region of attraction demonstrated what values of the state weighing matrix and the input weighing matrix were capable of producing the highest controllable percentage. Table 4.3 is a summary of the designed controllers accompanied by their capabilities. The controller tuned with the Q matrix diagonal less than one is represented by Lower Q, and the diagonal greater than one is represented by Higher Q. Naturally, the uncontrolled Simplest Walker has no tuned parameters. The metric of controllable percentage is represented by CP in the table.

Table 4.3: A summary of the different controllers tested on The Simplest Walker Model.

Un-controlled		Identity Tuned		Lower Q		Higher Q	
$q1$	-	$q1$	1	$q1$	1	$q1$	61
$q2$	-	$q2$	1	$q2$	61	$q2$	1
$q3$	-	$q3$	1	$q3$	1	$q3$	61
ρ	-	ρ	0.01	ρ	0.01	ρ	0.01
CP	6.48%	CP	31.07%	CP	30.91%	CP	29.23%

4.3 Region of Attraction

The region of attraction calculated for each controller is limited by its boundary of the initial positions tested. This boundary was set to reduce computational time, but it does not include all of the region of attraction. In an attempt to see more of the region of attraction, the method of setting a perturbation amount for either $\dot{\theta}$, ϕ, or $\dot{\phi}$ was used. ϕ was set to remain a constant value, so the region of attraction was mapped using $\dot{\theta}$ versus $\dot{\phi}$. This method was used with the three different systems tested. Figure 4.17 shows the wider view of the region of attraction of the uncontrolled system. Figure 4.18 shows the wider view of the region of attraction for the implemented controller tuned using the identity. Lastly, the region of attraction slice for the controller tuned using the region of attraction method can be seen in Figure 4.19. The controller used in Figure 4.19 is the

controller tuned using the Q matrix with the diagonal less than one. The reason this Q matrix was used is because of the two different styles of Q matrices explored, the diagonal consisting of values below one is capable of a higher controllable percentage. Please note the cutoff of the plotted points on the right side of the graphs, where $\dot{\theta}$ is approximately 0.05. This limit represents the maximum value of $\dot{\theta}$ before the walker falls backwards. The goal of this thesis is to expand the region of

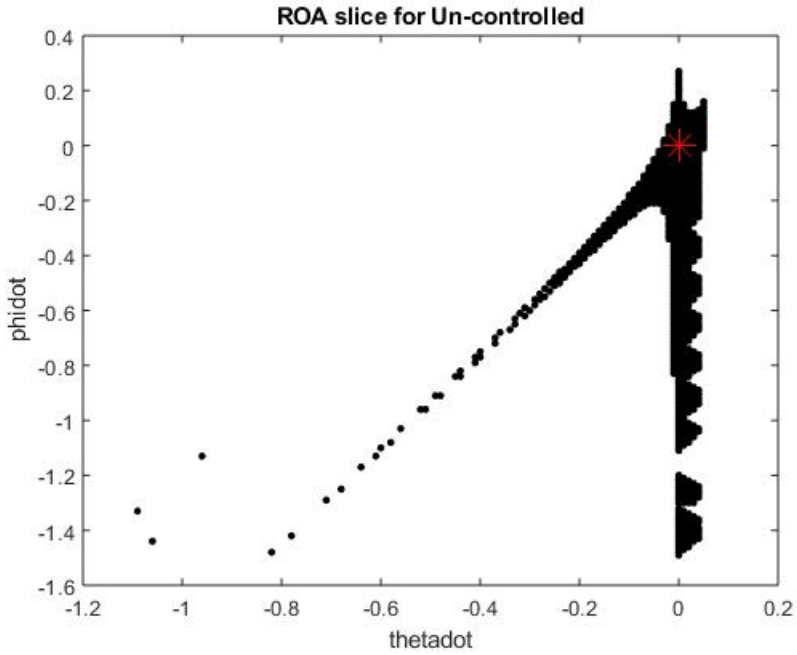


Figure 4.17: The slice of the region of attraction at the layer of ϕ equal to 0.1 for the uncontrolled Simplest Walker. The fixed point is represented by the red star.

attraction of the simplest walker, and viewing the region at the same layer will provide a clear and simple conclusion. The wider view of the region of attraction of both controllers shows that each controller produces a different region of attraction, but it is difficult to tell if the controllable percentage is increasing. To provide a better comparison, the region of attraction must be viewed at a closer resolution. It has already been shown that viewing a slice of the region of attraction allows for better interpretation, so that same method is used to visualize the size increase. The region of attraction was calculated within a specific range, so by viewing the outermost layer, we can see the size increase. The reason for viewing the outermost layer is because the more capable the Simplest Walker is, the further away from the fixed point it can be and still be controlled. Figure

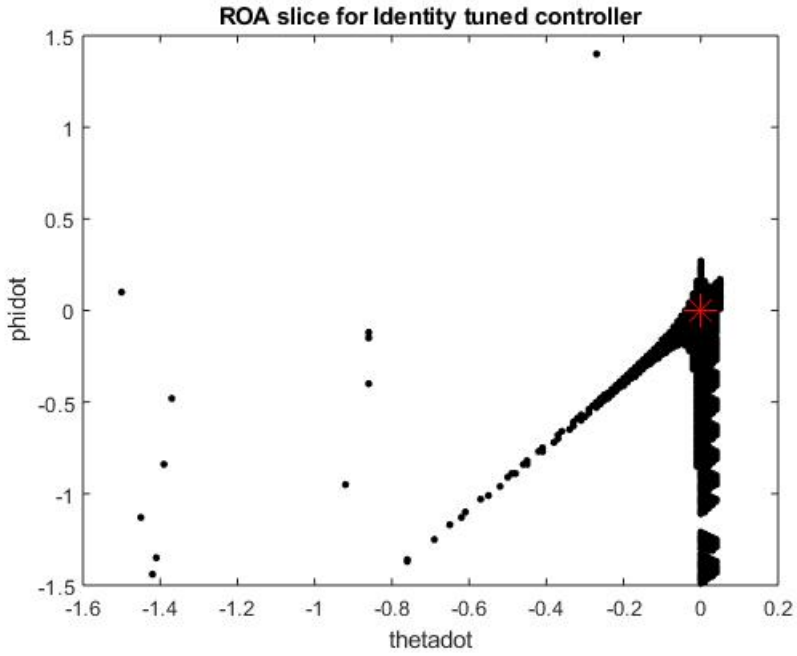


Figure 4.18: The slice of the region of attraction at the layer of ϕ equal to 0.1 for the controller tuned using the identity. The fixed point is represented by the red star.

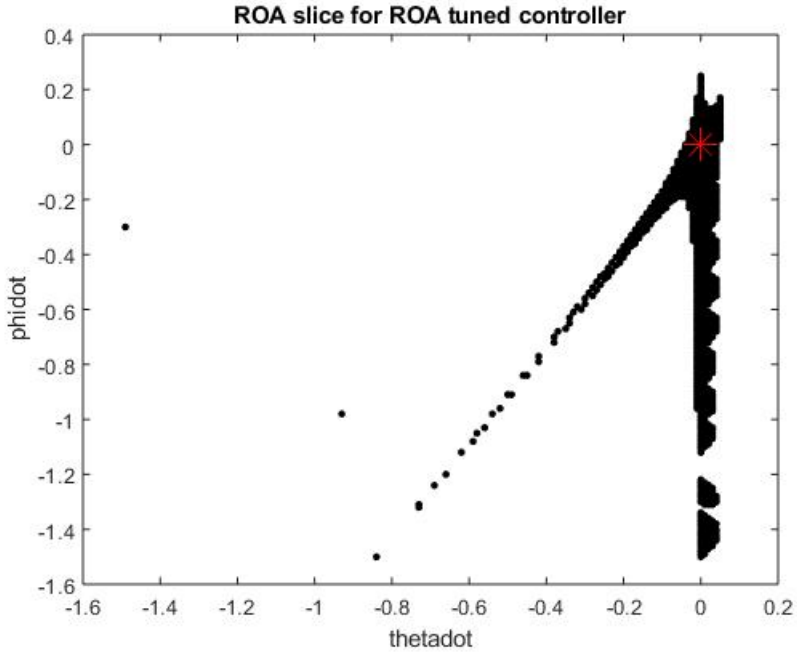
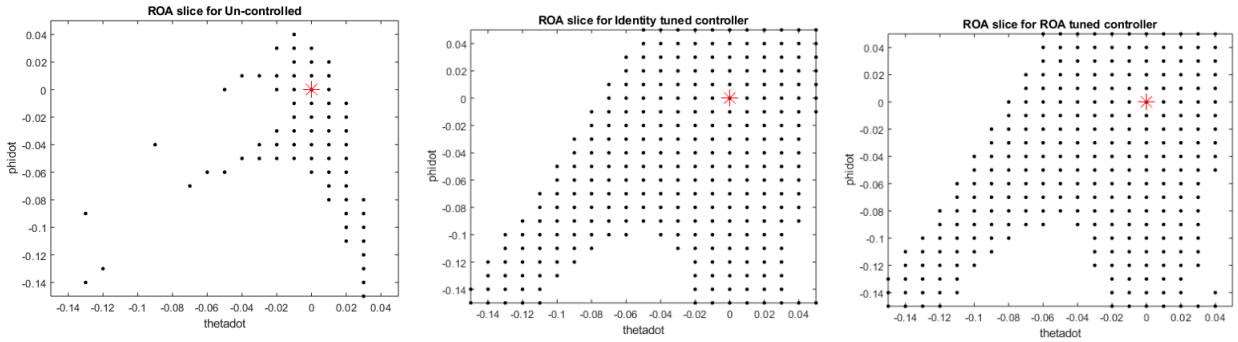


Figure 4.19: The slice of the region of attraction at the layer of ϕ equal to 0.1 for the controller tuned using the Lower Q matrix. The fixed point is represented by the red star.

4.20 displays a side-by-side comparison of the regions of attraction and it can be seen that with the implementation of the DLQR controller the region of attraction has increased.



(a) Uncontrolled ROA (b) Identity Tuned ROA (c) ROA Tuned ROA

Figure 4.20: A side-by-side comparison of the region of attraction for the uncontrolled Simplest Walker and two different controllers tuned using different parameters.

CHAPTER 5: CONCLUSION

In conclusion, the results will be summarized and discussed to provide answers to any additional questions that may exist. The purpose is to clearly define how the values of the diagonal state weighing matrix relate to each other, and the relationship the input weighing matrix has with them. Once a brief summary is finished, consideration of future work related to this method of tuning using the region of attraction is discussed.

5.1 Results summary

To summarize, the work done in this thesis explores a technique of using the region of attraction to tune the input parameters of a DLQR controller. The goal was to begin with the region of attraction of the un-perturbed Simplest Walker and expand the region of attraction with the implementation of a designed controller. The controller design was determined from calculating the region of attraction for a wide variety of different DLQR controllers. For each controller, a percentage of controllable initial points was calculated based off all the initial points attempted. Using the controllable percentage, a DLQR controller was designed to obtain a larger value for the controllable percentage. From the experiments conducted, the controller capable of producing a larger controllable percentage consists of a smaller $q1$ value, a larger $q2$ value, and a smaller $q3$ value in the Q matrix with the diagonal less than one, accompanied by a smaller ρ value. This was determined by testing different diagonal Q matrices with values less than one and greater than one, while maintaining the requirement of being positive semi-definite. Additionally, the different values of ρ were chosen to increase in size to determine the effect while meeting its criteria of being positive definite. The diagonal values of the Q matrix less than one was able to produce a larger controllable percentage of initial points tested. Testing the diagonal values of the Q matrix greater than one revealed that the best value for $q1$ would be a higher value, the best value for $q2$ would be a lower value, and the best value for $q3$ would be a higher value. The two conclusions agree with each other, because the diagonal of the first style of Q studied is the reciprocal of the

second diagonal of Q. The approach used with the diagonal of Q allowed for diagonal values less than one and greater than one to be tested. The technique of displaying the region of attraction as layers presented a user-friendly method; this was done to supply accurate and robust visualization. The simpler visualization shows that the region of attraction of the simplest walker was increased thanks to accurate implementation of a DLQR controller.

5.2 Future work

In addition to the brute force method of using the region of attraction to tune the input parameters of the DLQR controller, optimization techniques could be implemented to determine the optimal values of $q1$, $q2$, $q3$, and ρ. Through optimization, the results from the region of attraction method can be confirmed, providing additional support for the results. The cost metric for The Simplest Walker could be the distance between the state and the fixed point on the Poincaré Section after three steps. The optimization parameters would be the values of $q1$, $q2$, $q3$, and ρ. An initial guess will need to be defined, and then starting at that point, the function will try to find the minimum distance between the state and the fixed point. The cost function that is trying to be minimized should be classified as a black-box function because the form of the equation is analytically unknown. Several initial values of $q1$, $q2$, $q3$, and ρ should be tested, but it is important to use the same perturbation amounts for each set of optimization parameters. The addition of this tuning technique would add additional validation to the tuning method used within this thesis.

BIBLIOGRAPHY

[1] Y. Zhu Y. Liu X. Liu, X. Zang and J. Zhao. System overview and walking dynamics of a passive dynamic walking robot with flat feet. *Advances in Mechanical Engineering*, 7(12):1–10, 2015.

[2] T. McGeer. Passive dynamic walking. *The International Jornal of Robotics Research*, 9(2):62–82, 1990.

[3] A. Zamani, P. A. Bhounsule, and A. Taha. Planning energy-efficient bipedal locomotion on patterned terrain. In *Unmanned Systems Technology XVIII*, volume 9837 of , page 98370A, May 2016.

[4] M. Hoffmann and R. Pfeifer. The implications of embodiment for behavior and cognition: Animal and robotic case studies. In *The Implications of Embodiment: Cognition and Communication*, pages 31–58. Exeter: Imprint Academic, 2012.

[5] P. A. Bhounsule and A. Zamani. A discrete control lyapunov function for exponential orbital stabilization of the simplest walker. *ASME Jornal of Mechanism and Robotics*, 9(5), 2017.

[6] A. Zamani and P. A. Bhounsule. Foot placement and ankle push-off control for the orbital stabilization of bipedal robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4883–4888, Sep. 2017.

[7] S. H. Strogatz. *Nonlinear Dynamics and Chaos*. Perseus Books Publishing, L.L.C, 1994.

[8] P. Seidel. Iteration, fixed points, Fall 2011. 18.01 Lecture Notes.

[9] J. C. Albahaca. Analytical and numerical study of the poincaré map with applications on the computation of periodic orbits, May 2015. U.U.D.M. Project Report.

[10] W. Tucker. Computing accurate poincaré maps. *Physica D*, 171:127–137, 2002.

[11] Unknown. Stability and linearization. World Wide Web electronic publication, Unknown.

[12] G.T. Fallis. Walking toy, 1888. U.S. Patent No. 376588.

[13] J. J. Mahan and J. F. Moran. Toy, 1909. U.S. Patent No. 1007316.

[14] B.U. Bechstein. Improvements in and relating to toys, 1912. UK Patent 7453.

[15] J. E. Wilson. Walking toy, 1938. US Patent 2140275.

[16] S.H. Collins, M. Wisse, and A. Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607, 2001.

[17] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082, 2005.

[18] Dai Owaki, Masatoshi Koyama, Shinichi Yamaguchi, Shota Kubo, and Akio Ishiguro. A 2-d passive-dynamic-running biped with elastic elements. *Robotics, IEEE Transactions on*, 27(1):156–162, 2011.

[19] Ambarish Goswami, Benoit Thuilot, and Bernard Espiau. A study of the passive gait of a compass-like biped robot: symmetry and chaos. *International Journal of Robotics Research*, 17(12):1282–1301, 1998.

[20] A. Ruina M. Garcia, A. Chatterjee and M. Coleman. The simplest walking model: Stability, complexity, and scaling. *ASME Journal of Biomechanical Engineering*, 120(2):281–288, 1998.

[21] Simon Mochon and Thomas A McMahon. Ballistic walking. *Journal of biomechanics*, 13(1):49–57, 1980.

[22] A.l. Schwab and M. Wisse. Basin of attraction of the simplest walking model. In *Proceedings of DETC 2001*, 2001.

[23] Pranav A. Bhounsule and Ali Zamani. Stable bipedal walking with a swing-leg protraction strategy. *Journal of Biomechanics*, 51:123 – 127, 2017.

[24] Unknown. Approximating the derivative. World Wide Web electronic publication, 2003, http://mathfaculty.fullerton.edu/mathews/n2003/differentiation/NumericalDiffProof.pdf.

[25] Pranav A Bhounsule, Andy Ruina, and Gregg Stiesberg. Discrete-decision continuous-actuation control: balance of an inverted pendulum and pumping a pendulum swing. *Journal of Dynamic Systems, Measurement, and Control*, 137(5):9, 2014.

VITA

Justin Ernst is from Lubbock, Texas. He studied mathematics and earned a Bachelor's of Science degree in Mathematics from Lubbock Christian University. He then decided to pursue a more applied math approach and began studying mechanical engineering. He will graduate from The University of Texas at San Antonio with a Master's of Science degree in Mechanical Engineering. Upon completion of school, Justin plans to work in industry, where he wants to help build a better future through discovery and improvement in the field of mechanical engineering.