VERTICAL CONTROL FOR AN ONE-LEGGED HOPPING ROBOT

by

LIJUN LI

Submitted in partial fulfillment of the requirements

For the degree of Master of Science

Thesis Advisor: Dr. Wei Lin

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

January, 2008

CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the thesis/dissertation of

Lijun Li

candidate for the M.S. degree *.

(signed) Wei Lin
(chair of the committee)

M. Cenk Cavusoglu

Vira Chankong

(date) 10/25/2007

*We also certify that written approval has been obtained for any proprietary material contained therein.

To my parents
with
Love and Gratitude

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Acknowledgement

I would like to thank my supervisor, Dr. Wei Lin, for his support and review of this thesis. His wide knowledge and rigorous way of thinking have been of great value to me. His understanding and guidance have provide a good basis for this thesis.

My appreciation as well to Graham Alldredge for his help in understanding the problems at the beginning of the project and his help in debugging the simulation code.

Special thanks to Hao Lei and Zongtao Lu, who provide their thoughts and discussions which help me to design the controllers and conduct the simulation.

Vertical Control for an One-Legged Hopping Robot

Abstract

by

LIJUN LI

The vertical hopping control for an one-legged hopping robot is an essential problem in the study of balanced legged machines that walk and run. The field of dynamically stable legged locomotion has made great strides in 1980s, led primarily by Marc Raibert. The study of one-legged hopping problem has fascinated many scholars and researchers since then.

Following a review of the literature, a generic one-legged robot model is developed. Two models are presented one is the full model and the other one is only for the vertical hopping. The two phases and four events in a hopping cycle is defined. Control can only be applied during the stance period.

Four control algorithms that regulate the one-legged robot's hopping height are presented and simulation results are shown. They are energy based control (EBC), near inverse method (NIM), linear quadratic regulator (LQR) and PD control.

Chapter 1. **INTRODUCTION**

In this thesis a special type of robot named one-legged robot is studied. The research on this subject has a long history and a great success is achieved in the 1980's led by Marc Raibert. In this chapter the importance of this research and how the humans can benefit from it are presented.

1.1 Background

There are several reasons to conduct this type of research.

The first reason is to study the mobility on rough terrain. "The mobility of a legged system is generally limited by the best foothold in the reachable terrain and a wheeled system is limited by the worst terrain," point out Raibert in (27). When considering displacements on rugged terrains, perturbations resulting from the system's interaction with the ground are intermittent and discretized in space in the case of legged vehicles, while their effects on the vehicle are permanent in the case of wheeled locomotion. It is commonly acknowledged that legged locomotion is superior to wheeled locomotion when the terrain is soft or uneven and wheeled vehicles excel on prepared surfaces such as rails and roads. It is also considered that legged vehicle can easily overcome obstacles by utilizing the flight period (i.e. the period that the robot has no contact with ground). Due to various limitations, only about half the earth landmass is accessible to existing wheeled and tracked vehicles, whereas a much greater area can be reached by animals on foot. It should be possible to build legged vehicles that can go to the places that animals can now reach.

The second reason to study legged machines is to understand biological legged locomotion. The principles of control which is used in human and animal locomotion is still not understood. Humans and animal enjoy high mobility and efficiency of

locomotion due to their naturally designed legged system. It is of great interest to the researchers to build mechanical machines which replicates human and animal motions.

Among the various legged robots, the planar one-legged robots have attracted many researchers due to the simplicity of its mechanical design. It was thought that the analysis and experiments of the one-legged robots would enlighten the designing of biped, quadruped, and multi-legged robots.

The first one-legged robot was built by Raibert. It had a pneumatic cylinder installed in its leg and hence moved as a springy inverted pendulum while on the ground. Raibert decomposed the control problem into body attitude control, hopping height control and horizontal velocity control and showed that separate designs of the three controllers may be robust enough to allow decoupled operation (24). After Raibert's works, one legged running robots filled with a leg spring have been widely studied both experimentally and theoretically. Most of the one-legged robots proposed during the past two decades were similar to Raibert's design since they had some parts functioning like a spring to restore energy and provide force for take-off. These robots were also called one legged hoppers since their locomotion were series of cyclic hops.

1.2 Objectives

In this thesis, a generic model of the planner one-legged hopping robot is developed from the system dynamics. Based on this model various control techniques are studied, A new controllers is proposed and the soundness of the proposed controller is illustrated via simulations. As mentioned above, the control problem is divided into three subproblems, body attitude control, hopping height control and horizontal velocity control and only the hopping height control, which is also called vertical hopping control is studied in this thesis.

The specific objectives are as follows

- A vertical hopping model will be build based on the vertical motion which is extracted from the system dynamics. Apart from forward speed control, stable

control of the vertical oscillation is central to any successful implementation of dynamically stable legged locomotion.

- A brief review of previous work on legged machines will be presented. There are various techniques to control the hopping height in a sustained periodic hopping motion and two of them will be studied in details.

- A novel approach for the vertical control will be proposed. In the new approach, the characteristics of hopping motion and system dynamics are utilized and a PD controller is applied to control the vertical hopping height.

1.3 Thesis Outline

Chapter 2 presents a literature review of the one-legged vertical hopping problem. It provides insight into legged machine research in general, but refers the reader to other materials for specific details on legged machine research not relevant to this work.

Chapter 3 describes the system dynamics and the model which is going to be used in simulation. The model shown in this chapter can take both the vertical and horizontal motion into consideration. Characteristics of the one-legged hopping robot locomotion is introduced. The stance and flight phases are defined together with four key events of a hopping cycle. A summarization of parameters used to build the model is provided.

Chapter 4 shows two methods developed in the past two decades to control the vertical motion. One is energy based control and the other one is the near inverse method. The control problem for the vertical hopping motion is complicated due to the fact that control can only be applied during the stance period which is very short.

- The energy based control proposed by Raibert (24) calculates the difference between the desired energy of the system and the energy the system currently has and determines how much energy should be injected or removed from the system. The energy calculation is done once per hopping cycle during the stance period.

- The near inverse method relies on a simplified discrete model of the plant dynamics which relates the control input, the height of previous hop and the height of the next hop.

Chapter 5 presents a controller which use the linear quadratic regulator technique to control the vertical hopping motion. Linear quadratic regulator is a powerful technique which can be used to design controllers for complex systems with stringent requirements. This technique is used in order to achieve robust performance and energy efficiency. This is our first trail to apply optimum control technique in the field of dynamic balanced robot.

Chapter 6 proposes a PD controller. By using the appropriate coefficient and reference point for the position and speed of the body (i.e. the upper part of the one-legged robot), the system shows robust performance and meanwhile it is optimized in some sense. Compared with previous controllers, it is simple and has excellent performance.

Chapter 7 provides a summary, conclusions and areas for future work.

Chapter 2. **LITERATURE REVIEW**

2.1 Types of Legged Robots

Legged robots fall into two classes, *statically stable robots* and *dynamically stable robots*.

The static stability is a simple concept. There is a contact polygon formed by connecting all the neighboring footholds. It is called statically stable when the projection of the center of mass of the body lies in the convex hull of the contact polygon. A statically stable robot can stand still without falling over. Static stability is a useful feature and it can be achieved by requiring enough legs on the robot to provide sufficient static points of support.

On the other hand, dynamic stability allows a robot to be stable while moving. For example, one-legged hopping robots are dynamically stable and they can hop in place or to various places without falling over. It is enough to say a robot is dynamically stable when it maintains balance in the overall locomotion cycle even if the robot is not balanced statically at any time.

A statically stable system could be dynamically unstable and a statically unstable system could be dynamically stable. For example, a chair is statically stable, but if accelerated quickly enough it can tip over. On the contrary, an inverted pendulum can be stabilized by applying the proper acceleration to the base.

Statically stable machines are only simple successful solutions for low-speed locomotion, where dynamic forces are small compared to statical ones. As a result,statically stable legged machines suffer not only from a large number of legs but also from a low cruising speed. On the contrary, dynamically stable robots enjoy less design complexity, achieve higher speeds and can be more energy efficient. The main disadvantage of dynamic balance is the lack of a general control methodology.

2.2 Research on Legged Machines

The scientific study of legged locomotion began over a century ago when Muybridge tried to find out whether or not a trotting horse left the ground with all four feet at the same time. After that Muybridge went on to document the walking and running behavior over 40 animals including humans (19)(20). His photographic data are still of considerable value and survive as a landmark in locomotion research. The studies of walking machines also has its origin in Muybridge's time. An early walking model appeared in about 1870 (10). It used a linkage originally designed by the famous Russian mathematician Chebyshev some years earlier to move the body along a straight horizontal path while the feet moved up and down to exchange support.

During the 90 years that followed, people viewed that building walking machines as the task of designing kinematic linkages that would generate suitable stepping motion. Many designs were proposed (e.g. (3)(17)(29)(30)(32)), but the performance of such machines was limited by their fixed patterns of motion since they could not adjust to the terrain's variations. By late 1950s it had become clear that a linkage providing fixed motion would not do the trick of walking or running, and useful walking machines would need control (9).

One approach to control was harness human. Ralph Mosher used this approach in building four legged walking truck at General Electric in mid 1960s (18). Despite its dependence on a well-trained human for control, this walking machine was a landmark in legged technology.

Computer control became alternative to human control of legged vehicles in 1970s. McGhee's group was the first to use this approach successfully (13). They built an insect-like hexapod that could walk with a number of standard gaits and negotiate simple obstacles. The computer's major task is to solve kinematic equations in order to coordinate the 18 electric motors driving the legs. Gurfinkel and his group build a machine with quite similar performances to McGhee's at about the same time. It used a hybrid computer for control (4). Hirose realized the linkage design

and computer control are not exclusive and his experience with clever and unusual mechanisms led to a simplified control of locomotion and improved their efficiency (5) (6). McGhee, Gurfinkel, and Hirose's walking machine groups represent a class called *Static Crawlers*. Each differs in the details of construction and computing technology used for control. Several other machines that fall into this class have been studied in the intervening years.

Another class of legged systems are dynamic machines that balance actively. The legged machines fall into this class operate in a regime where the velocity and kinetic energies are important determinants of behavior. The exchange of energy among its various forms is also important in the dynamics of legged locomotion. Shannon was probably the first to build a machine that balanced an inverted pendulum atop of a small powered truck. The truck drove back and forth in response to the tipping movements of the pendulum. This study was forwarded by his students to demonstrate controllers for two pendulums at once, and finally the case that one pendulum were mounted on top of the other. Later, they extended these techniques to provide balance for a flexible inverted pendulum.

Miura and Shimoyama (16) built the first walking machine that really balanced actively. The control of their biped relied on an inverted pendulum model. Each time a foot was placed on the floor, its position was chosen according to the tipping behavior expected from an inverted pendulum.

Matsuoka (11) was the first to build a machine that was able to run, where running is defined by the presence of intervals of ballistic flight when all feet are off the ground. Matsuoka's goal was to model repetitive hopping in humans. He formulated a model with a body and one massless leg and also simplified the problem by assuming that the duration of the support phase was short compared with the flight phase.

The field of dynamically stable legged locomotion has made great strides in 1980s, led primarily by Marc Raibert. He build a variety of running robots, starting with a planner one-legged machine, followed by a 3D one-legged, a two-legged planar robot and a four-legged quadruped. Among his works, Raibert (24) introduced a scheme for exciting the leg spring from rest in order to regulate the energy in the spring-mass.

Raibert has also proposed a tabular control algorithm (26) which uses a large table of pre-computed data and calculates the control signal by interpolation. This algorithm was applied experimentally to 2-D and 3-D physical prototypes.

After Raibert's work, many researchers were attracted to study and conduct research on one-legged running robots either experimentally or theoretically.

Based on their established approach to analyze the intermittent dynamics of a juggling robot (1), Koditschek and B*ü*hler (2) derived an analytical discrete map for a simplified vertical hopper, modelled after Raibert's robots. Lapshin (7) proved the asymptotic global stability of the vertical motion of a one-legged robot with linear leg spring. Vakakis and Burdick (33) observed chaotic behavior when they changed the parameters of a model similar to that used by Koditschek and B*ü*hler. Later M'Closky and Burdick used a plannar hopper model with a body placed on top of a massless and an inertia-less leg similar to Raibert's, (14). They showed by perturbation and numerical integration methods that even the planner hopper can show chaotic behavior in a certain set of parameters. There exist very few related results that use the continuous-time framework for stability analysis of hopping robots due to their intermittent dynamics. Li and He used a perturbation approach to study the existence and stability of limit cycles for a vertical hopper (34).

Sznair and Damborg (1989) (31) used an adaptive control algorithm for both vertical and horizontal motion of a two dimensional hopping robot, deriving a rather simple analytical solution for vertical motion control. Prosser and Kam (21)(22), suggested an approach which involved a near inverse of the system dynamics (based on off-line synthesis and inverse model) on an electrically actuated hopper. The preliminary control strategy in their electrically actuated hopping robot was obtained by a least-square fit of data to a multivariable polynomial. This method was later enhanced by applying on-line estimation of Controllers parameters (Lebaudy, 1993) (8). Rad et al (1993) (23) employed an open loop control approach on an electrically actuated hopper. A high gain PD controller was used to return the actuator. Some of the vertical controllers mentioned above will be reviewed in details in Chapter 4.

Chapter 3. **MODEL AND SYSTEM DYNAMICS**

3.1 The Model for Vertical and Horizontal Motion

In this chapter, a one-legged hopping machine is modelled by a springy leg with nonzero mass, a simple body, and an actuated hinge-type hip. This model can take both vertical and horizontal motion into account. The most important characteristic of the body is that it forms an elevated mass that must be balanced atop the leg and that it forms a structure from which torques can be applied to the leg. Legs typically do two things during locomotion: they change length and orientation with respect to the body.

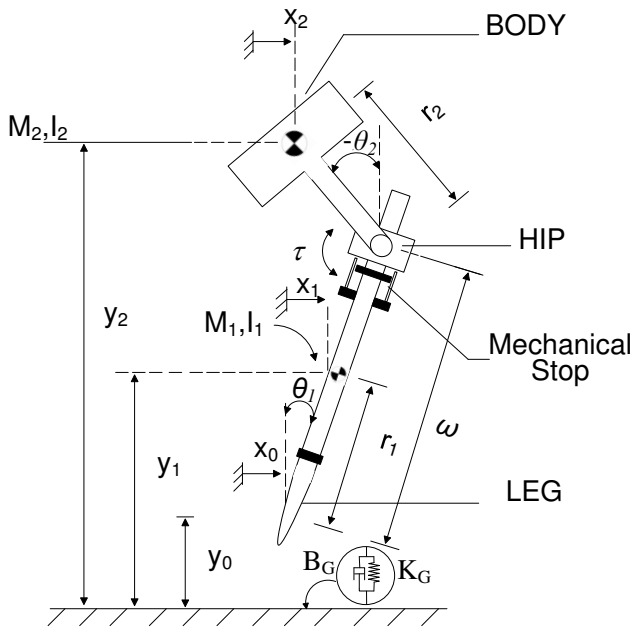

Fig. 3.1: Planar one-legged model

Figure 3.1 illustrates the model used for analysis and simulation. The essential features of the model are a body of mass M_2, a compliant leg of mass M_1, and the ground. The overall length of the leg is influenced by a spring, a position actuator in

series with the spring, and a mechanical stop.

The position actuator, length χ, is arranged in series with the leg spring, acting between the leg spring and the hip (see Figure 4.1 for illustration). When the actuator changes length, it does work on the leg spring to increase or decrease the energy stored in the spring.

The leg spring, which has length k_0 at rest and stiffness K_L, is modelled as though one end is rigidly connected to the foot, with the other end fastened to one side of the position actuator. The mechanical stop, modelled as a very stiff spring with damping, acts to prevent the spring from expanding beyond its rest length. The spring and mechanical stop are arranged so that only the spring generates forces when $(w-\chi) < k_0$, and only the mechanical stop generates forces when $(w-\chi) > k_0$. See Figure 3.1. The stiffness and damping of the mechanical stop, K_{L2} and B_{L2}, are chosen so that vibrations between the body and leg at lift-off decay quickly.

The ground is modelled as a stiff, damped spring which has stiffness K_G and damping B_G. The damping coefficient is chosen to keep the foot from bouncing on the ground during touchdown and lift-off. This compliance in the ground represents the compliances of both the ground and the foot. We assume that the stiffness of the ground is much greater than the stiffness of the leg, $K_G \gg K_L$.

3.1.1 Body and leg dynamics

The following equations are derived using geometric relationships shown in Figure 3.1, then differentiating twice with respect to time. First, x_0 is the horizontal displacement of the foot and x_1 is the horizontal displacement of the leg:

$$\begin{aligned} x_1 &= x_0 + r_1 sin\theta_1 \\ \dot{x}_1 &= \dot{x}_0 + r_1 cos\theta_1 \dot{\theta}_1 \\ \ddot{x}_1 &= \ddot{x}_0 + r_1 \ddot{\theta}_1 cos\theta_1 - r_1 (\dot{\theta}_1)^2 sin\theta_1 \end{aligned} \tag{3.1.1}$$

Next, y_0 is the vertical displacement of the foot and y_1 is the vertical displacement of the leg:

$$y_1 = y_0 + r_1 cos\theta_1$$

$$\begin{aligned} \dot{y}_1 &= \dot{y}_0 + r_1(-sin\theta_1)\dot{\theta}_1 \\ \ddot{y}_1 &= \ddot{y}_0 - r_1(\ddot{\theta}_1 sin\theta_1 + cos\theta_1(\dot{\theta}_1)^2) \end{aligned} \tag{3.1.2}$$

Third, x_2 is the horizontal displacement of the body:

$$\begin{aligned} x_2 &= x_0 + wsin\theta_1 - r_2 sin(-\theta_2) \\ \dot{x}_2 &= \dot{x}_0 + \dot{w}sin\theta_1 + w\dot{\theta}_1 cos\theta_1 + r_2\dot{\theta}_2 cos\theta_2 \\ \ddot{x}_2 &= \ddot{x}_0 + \ddot{w}sin\theta_1 + w\ddot{\theta}_1 cos\theta_1 - w(\dot{\theta}_1)^2 sin\theta_1 \\ &\quad + r_2(\ddot{\theta}_2 cos\theta_2 - (\dot{\theta}_2)^2 sin\theta_2) + 2\dot{w}\dot{\theta}_1 cos\theta_1 \end{aligned} \tag{3.1.3}$$

Finally, y_2 is the vertical displacement of the body:

$$\begin{aligned} y_2 &= y_0 + wcos\theta_1 + r_2 cos(-\theta_2) \\ \dot{y}_2 &= \dot{y}_0 + \dot{w}cos\theta_1 + w(-sin\theta_1)\dot{\theta}_1 - r_2 sin\theta_2\dot{\theta}_2 \\ \ddot{y}_2 &= \ddot{y}_0 + \ddot{w}cos\theta_1 + \dot{w}(-sin\theta_1)(\dot{\theta}_1)^2 \\ &\quad + r_2(\ddot{\theta}_2 cos\theta_2 - (\dot{\theta}_2)^2 sin\theta_2) + 2\dot{w}\theta_1 cos\theta_1 \end{aligned} \tag{3.1.4}$$

The above twelve equations are the main equations used to model the horizontal and vertical motions of the one-legged hopping robot. For a technical convenience, a summary of parameters and notations is presented as follows:

Symbol	Parameter
M_1	Leg mass
M_2	Body mass
r_1	Leg center of mass
r_2	Body center of mass
I_1	Leg moment of inertia
I_2	Body moment of inertia
k_0	Leg spring rest length
K_L	Leg spring stiffness
K_{L2}	Mechanical stop stiffness
B_{L2}	Mechanical stop damping
K_G	Ground stiffness
B_G	Ground damping
χ	Position actuator length

Tab. 3.1: Parameters and Notations

3.1.2 Force and torque analysis of leg and body

There are external and internal forces posed on the one-legged robot. The forces are decomposed and studied into details in this section.

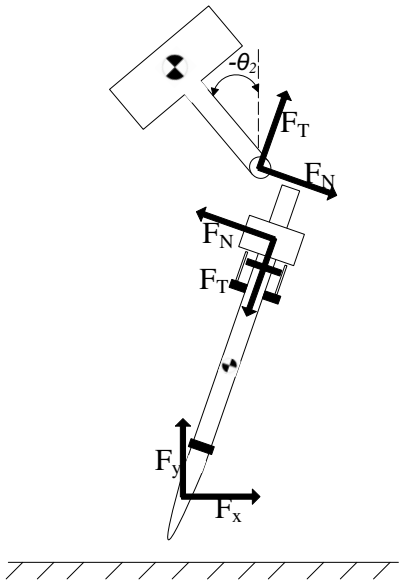

Fig. 3.2: Force illustration for full model

Figure 3.2 shows a force diagram of the external and internal forces in the model. From this diagram, force balance equations for the model can be derived. The first two equations are derived by summing the forces in the x and y directions for the leg. The third equation sums the moments of inertia for the leg:

$$M_1\ddot{x}_1 = F_x - F_t sin\theta_1 - F_N cos\theta_1 \tag{3.1.5}$$

$$M_1\ddot{y}_1 = F_y - F_T cos\theta_1 + F_N sin\theta_1 - M_1 g \tag{3.1.6}$$

$$I_1\ddot{\theta}_1 = -F_x r_1 cos\theta_1 + F_y r_1 sin\theta_1 - F_N(w - r_1) - \tau(t) \tag{3.1.7}$$

The next three equations are similarly derived for the body:

$$M_2\ddot{y}_2 = F_T cos\theta_1 - F_N sin\theta_1 - M_2 g \tag{3.1.8}$$

$$M_2\ddot{x}_2 = F_T sin\theta_1 + F_N cos\theta_1 \tag{3.1.9}$$

$$I_2\ddot{\theta}_2 = F_T r_2 sin(\theta_2 - \theta_1) - F_N r_2 cos(\theta_2 - \theta_1) + \tau(t) \tag{3.1.10}$$

where

F_x, F_y horizontal and vertical forces on the foot

F_T, F_N forces acting at the hip between the leg and body. F_T acts tangent to the leg, and F_N acts perpendicular to the leg.

3.2 The Hop Cycle

3.2.1 Phases and Events

Hopping is a cycle which has two phases. During a hopping cycle the phase in which the foot is not touching the ground is called flight. During flight, the trajectory of the center of gravity of the system is ballistic. The other phase when the foot is touching the ground is called stance. During stance, the leg provides support and the behavior of the system is like that of an inverted pendulum.

In addition to the two phases, there are four well-defined events in the hopping cycle (27):

LIFT-OFF:	The moment at which the foot loses contact with the ground
TOP:	The moment in flight when the body has peak altitude and vertical motion changes from upward to downward
TOUCHDOWN:	The moment the foot makes contact with the ground
BOTTOM:	The moment in stance when the body has minimum altitude and vertical motion of the body changes from downward to upward

Tab. 3.2: Events in One Hopping Cycle

These events can each be detected from the behavior of the the state variables.

3.2.2 Period of A Hopping Cycle

For the case of repetitive hopping in which periods of support alternate with periods of flight, the following values can be calculated. During the stance phase, the system can be viewed as a spring-mass oscillator with natural frequency ω_n:

$$\omega_n = \sqrt{\frac{K_L}{M_2}} \tag{3.2.11}$$

During repetitive hopping, each stance interval has duration equal to one half of the period associated with ω_n:

$$T_{STANCE} = \frac{1}{2} \cdot \frac{2\pi}{\omega_n} = \frac{\pi}{\omega_n} = \pi\sqrt{\frac{M_2}{K_L}} \tag{3.2.12}$$

During flight the system moves along a parabolic trajectory determined by the acceleration of gravity. The period of flight is

$$T_{FLIGHT} = \sqrt{\frac{8H}{g}} \tag{3.2.13}$$

where

g is the acceleration of gravity and

H is the hopping height measured at the foot.

The period of a full hopping cycle is just the sum of T_{STANCE} and T_{FLIGHT}:

$$T = \pi\sqrt{\frac{M_2}{K_L}} + \sqrt{\frac{8H}{g}} \tag{3.2.14}$$

Chapter 4. VERTICAL HOPPING AND CONTROL STRATEGIES

We are only interested in the vertical motion control in this thesis and a generic vertical hopping model is shown in this chapter. Energy based control and near inverse method are studied in detail. Simulation results are shown for each method.

4.1 Simplified Vertical Model

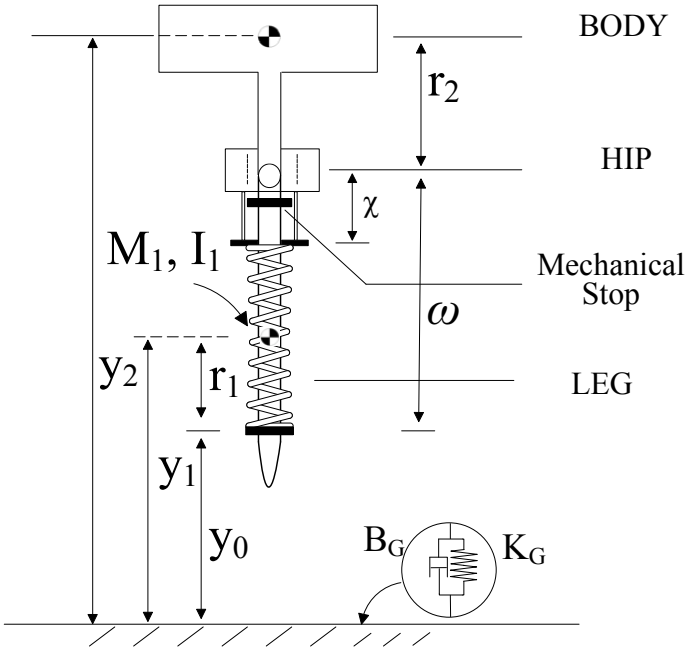

Fig. 4.1: One legged hopping robot vertical only

Figure 4.1 shows the simplified vertical model and based on this model new vertical controllers are designed and simulations are performed. Equations (3.1.5) - (3.1.10) can be reduced to the following for vertical motion:

$$M_1\ddot{y}_1 = F_y - F_T - M_1 g \tag{4.1.1}$$

$$M_2\ddot{y}_2 = F_T - M_2 g \tag{4.1.2}$$

$$y_0 = y_1 - r_1 \tag{4.1.3}$$

$$w = y_2 - y_1 + r_1 - r_2 \tag{4.1.4}$$

where

$$F_T = \begin{cases} K_L(k_0 - w + \chi) & \text{if} \quad (k_0 - w + \chi) > 0 \\ K_{L2}(k_0 - w + \chi)) - B_{L2}\dot{w} & \text{otherwise} \end{cases} \tag{4.1.5}$$

$$F_y = \begin{cases} -K_G y_0 - B_G \dot{y}_0 & \text{if} \quad y_0 < 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.1.6}$$

4.2 Energy Based Vertical Control

The objective of the vertical controller is to initiate and terminate hopping, as well as switch between hopping heights quickly. The control problem is complicated by the fact that the hopping machine can only be controlled during the stance phase.

This implementation of the vertical controller calculates the desired energy of the system (which is based on the desired hopping height, H), then approximates the current energy in the system. The difference between these two energies determines how much energy should be injected into (or removed from) the system (24). These energy calculations are done once every hopping cycle, while the robot is on the ground.

4.2.1 Controller Overview

Energy is injected into (or removed from) the system using the position actuator. The position actuator changes only once per cycle, while the hopping robot is on the ground. The position of the position actuator is determined by the desired energy change (denoted ΔE_H) through the following control law:

$$\Delta\chi = -(\chi - w + k_0) + \sqrt{(\chi - w + k_0)^2 + \frac{2\Delta E_H}{K_L}} \tag{4.2.7}$$

where $\Delta E_H = E_H - E_{FLIGHT}$ is the difference between the energy of the desired hopping height, E_H, and the energy of the next flight, E_{FLIGHT}. Both E_H and E_{FLIGHT} are given in Section 4.2.2. A detailed derivations of the controller (4.2.7) can also be found in Section 4.2.2.

4.2.2 Detailed Derivations

Total energy of the system

There are three kinds of energy present in the system: kinetic, gravitational potential, and elastic potential. The total energy in the system, then, can be written as

$$\begin{aligned} E &= PE_g(M_1) + PE_g(M_2) + KE(M_1) + KE(M_2) + PE_e(M_1) + PE_e(M_2) \\ &= M_1 g y_1 + M_2 g y_2 + \frac{1}{2} M_1 \dot{y}_1^2 + \frac{1}{2} M_2 \dot{y}_2^2 + \frac{1}{2} K_L (k_0 - w + \chi)^2 + \frac{1}{2} K_G y_0^2 \end{aligned} \quad (4.2.8)$$

Note that the expressions for gravitational potential energy are chosen such that the gravitational potential energy would be zero if both masses were on the ground, that is, at zero height.

Energy losses

The two energy losses mentioned above can be quantified. At touchdown, the velocity of M_1 is instantly forced to zero. The energy loss, then, is all the kinetic energy of M_1 at the moment before touchdown:

$$E_{TD-LOSS} = KE(M_1) = \frac{1}{2} M_1 \dot{y}_{1,TD^-} \quad (4.2.9)$$

Note that the subscript TD^- is used to denote the moment before touchdown. Because the controller measures the energy of the system after touchdown, this energy loss does not need to be taken into account by the controller.

The second energy loss, which occurs at lift-off, is more important because the controller will need to predict and "pre-compensate" for this loss. To calculate this energy loss, the change in velocity should be calculated first by equating linear momentum just before and just after lift-off:

$$M_2 \dot{y}_{2,LO^-} = (M_1 + M_2) \dot{y}_{2,LO^+} \quad (4.2.10)$$

$$\dot{y}_{2,LO^+} = \frac{M_2}{M_1 + M_2} \dot{y}_{2,LO^-} \quad (4.2.11)$$

The subscripts LO^- and LO^+ denote the instants just before and just after lift-off, respectively. The energy loss can then be calculated by taking into account this

change in velocity:

$$\begin{aligned} E_{LO-LOSS} &= \frac{1}{2}M_2\dot{y}_{2,LO^-}^2 - \frac{1}{2}(M_1+M_2)\dot{y}_{2,LO^+}^2 \\ &= \frac{1}{2}M_2\dot{y}_{2,LO^-}^2 - \frac{1}{2}(M_1+M_2)(\frac{M_2}{M_1+M_2}\dot{y}_{2,LO^-})^2 \\ &= \frac{M_1M_2}{2(M_1+M_2)}\dot{y}_{2,LO^-}^2 \\ &= \frac{M_1}{M_1+M_2}KE_{LO^-} \end{aligned} \tag{4.2.12}$$

The controller's energy calculations

The controller must first calculate the energy associated with the desired height. This is easily calculated by finding the gravitational potential energy of the system at the desired height:

$$E_H = M_1g(H+r_1) + M_2g(H+k_0+r_2) \tag{4.2.13}$$

Recall that at the maximum height, the velocity of the system is zero and the spring is at its relaxed length. Therefore, all the energy of the system will be in the form of gravitational potential energy.

Secondly, the controller must predict the energy of the next flight. This is easily done using Eq. (4.2.8) to find the current energy in the system, then adjusting that by the amount of energy which will be lost at lift-off (using Eq. (4.2.12)):

$$\begin{aligned} E_{FLIGHT} = \frac{M_2}{M_1+M_2}(&M_1gy_1 + M_2gy_2 + \frac{1}{2}M_1\dot{y}_1^2 + \frac{1}{2}M_2\dot{y}_2^2 + \\ &\frac{1}{2}K_L(k_0-w+\chi)^2 + \frac{1}{2}K_Gy_0^2) \end{aligned} \tag{4.2.14}$$

Note that this calculation assumes that no energy will be lost during flight. The difference between these two, denoted as ΔE_H, then is:

$$\Delta E_H = E_H - E_{FLIGHT} \tag{4.2.15}$$

Now, the controller must calculate the change in the actuator which will inject (or remove) the desired amount of energy into (or from) the system. The change in energy when the actuator moves from χ to $\chi + \Delta\chi$ is:

$$\Delta E_\chi = \frac{1}{2}K_L(\chi+\Delta\chi-w+k_0)^2 - \frac{1}{2}K_L(\chi-w+k_0)^2$$

$$
\begin{aligned}
&= \frac{1}{2}K_L(2(\chi - w + k_0)\Delta\chi + \Delta\chi^2) \\
&= K_L[\frac{1}{2}\Delta\chi^2 + \Delta\chi(\chi - w + k_0)] \qquad (4.2.16)
\end{aligned}
$$

Then, the value of $\Delta\chi$ for the controller (4.2.7) can be obtained by setting $\Delta E_\chi = \Delta E_H$ and completing the square.

Remark: The implementation of the vertical controller (4.2.7) requires the states y_1, $\dot{y}_1$, y_2, and $\dot{y}_2$ to be measurable. This point can be seen clearly from (4.2.15) and (4.2.13)-(4.2.14).

4.2.3 Simulation

In this section, MATLAB was used to simulate the vertical control of the system. The simplified vertical model from Section 4.1 was used with the following parameters:

M_1	1 kg	K_L	1e3 N/m	K_{L2}	1e4 N/m
M_2	10 kg	k_0	1 m	B_{L2}	135 N·s/m
r_1	0.5 m	χ_{min}	0 m	K_G	1e4 N/m
r_2	0.4 m	χ_{max}	0.2 m	B_G	75 N·s/m

The following results represent the case when the initial height is 0.2 m and the desired height $H = 2$. In this simulation, the control input χ is calculated and actuated at the bottom event so that the maximum amount of energy is injected. After lift-off, χ is retracted back to an initial position. For the first simulations, that initial position is 0.

Figure 4.2 shows the response of the system in the time domain. The height of the bottom of the foot, y_0, is shown in the top plot. The bottom plot shows w, the distance from the hip to the foot. Recall that the spring is compressed when $\chi - w < k_0$. It is easy to notice that the robot hops to the desired height quickly and then maintain the hopping height at the desired level. It is assumed that the actuator starts to compress the spring once the bottom event occurs. By doing so, the actuator can inject the maximum energy to the hopping machine. That explains the quick convergence to the desired hopping height. In reality, it is difficult to push the spring at the bottom event. The reason is as follows: the stance period is very short

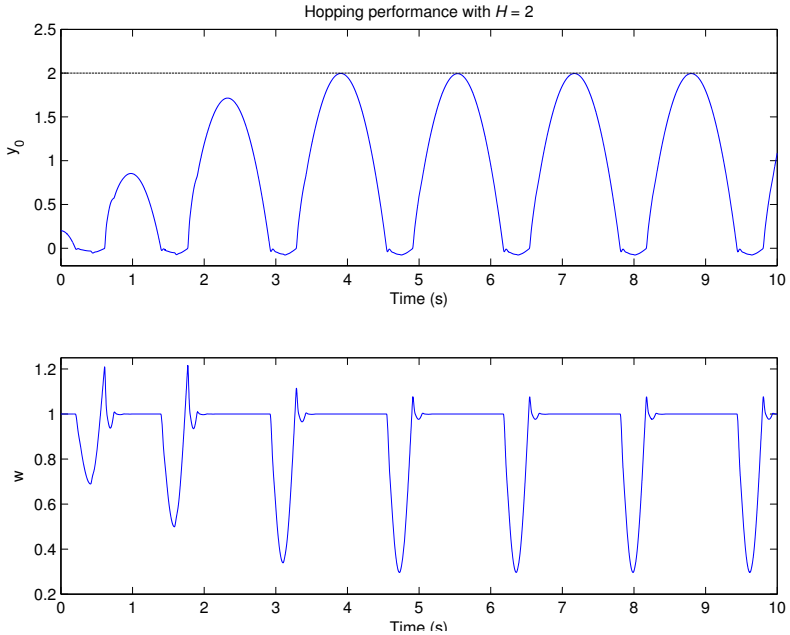

Fig. 4.2: EBC-vertical hopping start from rest

and the control has to be applied during that short period. It is even more difficult if one choose to start to control the actuator to compress the leg spring starting from the bottom event. Here we are trying to demonstrate how the energy based control works and we assume that the control is only applied at the bottom event.

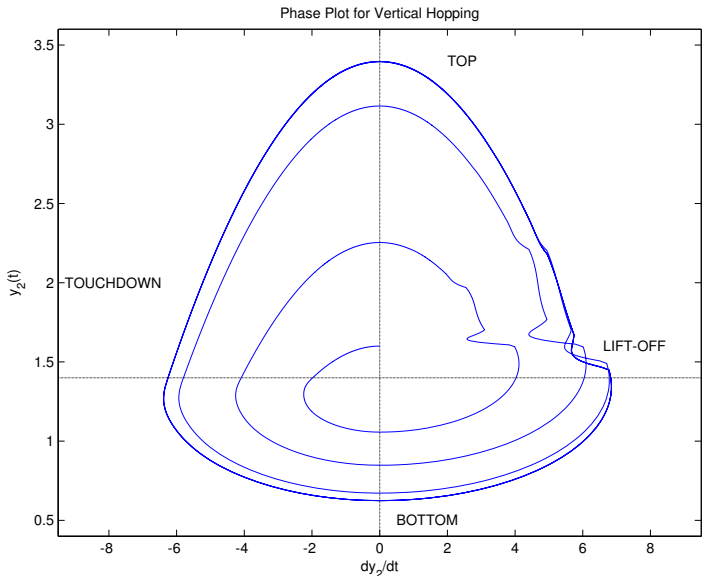

Fig. 4.3: EBC-phase plot for vertical hopping

Figure 4.3 is a phase plot of the $\dot{y}_2$ and y_2. The system starts off at $(\dot{y}_2, y_2) = (0, 1.6)$, in the middle of the phase plot. The system then approaches a periodic orbit.

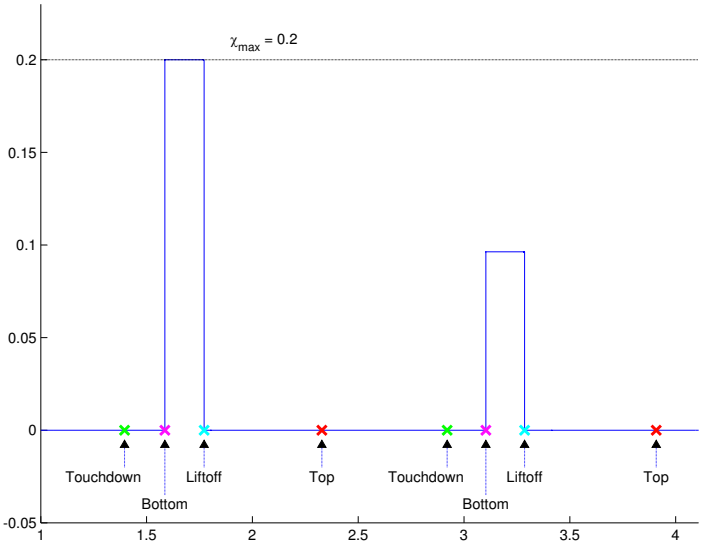

Fig. 4.4: EBC-position actuator χ for two cycles

Figure 4.4 is a plot of the actuator length for the second and third cycles. Note that the actuator doesn't start to compress the spring until the bottom event so that it injects the maximum amount of energy. Also note that we assumed the time it takes the position actuator to move is negligible.

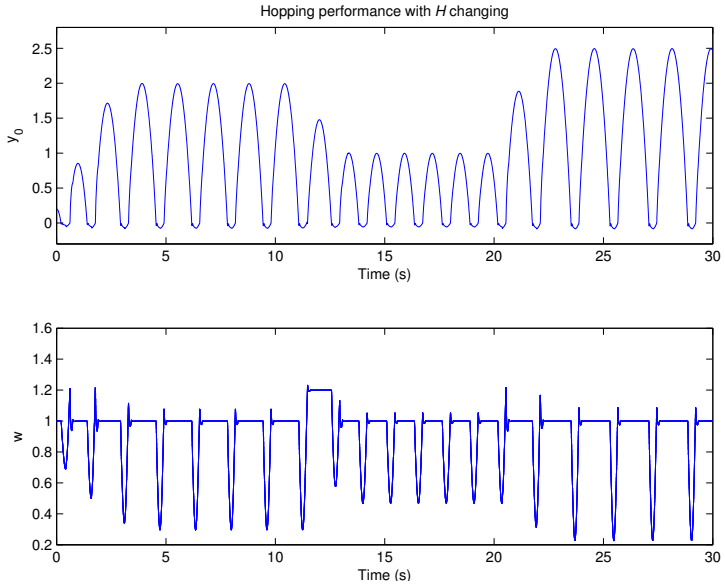

Fig. 4.5: EBC-vertical hopping with changing values of H_d

Figure 4.5 shows the trajectory of the system when the desired height, H is changed during hopping. H is equal to 2.0 initially, then decreased to 1.0 at $t = 10$ s, then increased to 2.5 at $t = 20$ s.

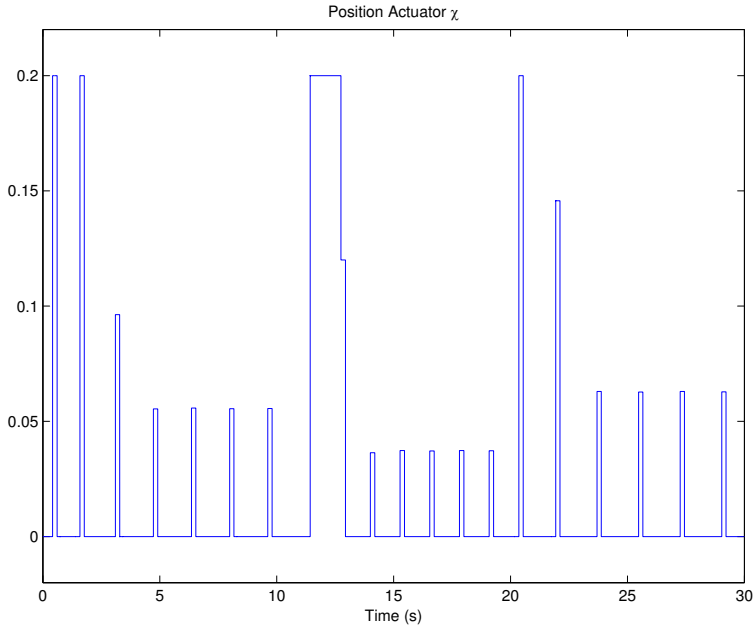

Fig. 4.6: EBC-position actuator for the hopping in Figure 4.5

Figure 4.6 shows how the actuator changes to adapt to the changing value of H. Note that around $t = 10$, the actuator retracts to the max position ($\chi_{max} = 0.2$) so that energy can be withdrawn during the next stance period.

4.3 Near Inverse Method design

The Near Inverse Method relies on a simplified discrete model of the plant dynamics which relates the control input, the height of previous hop and the height of the next hop. The Near Inverse Method can be used to reduce the steady-state error in hopping height and sensitivity to parameter variations (21).

The hop height of the i^{th} hop, H_i, is a function of the height of the previous hop, H_{i-1}, and the pseudo-control signal, $\Delta\chi_i$, i.e.,

$$H_i = f(\Delta\chi_i, H_{i-1}) \tag{4.3.17}$$

If f were available, we would be interested in solving for the control input, $\Delta\chi_i$, as a function of the measured height of the previous hop H_{i-1}, and the desired hopping height, H_i. It might be possible to obtain an explicit expression for f analytically, but g, the inverse of f, is difficult to find. In order to perform control, an approximation

to (4.3.17) is used to generate a control signal. The solution provides a 'near-inverse' relation which exhibits reduced sensitivity to parameters of the leg and ground.

4.3.1 Near Inverse Control

The following model is proposed based on a stated objective of balancing low approximation error with manageable computational efficiency:

$$H_i = H_d = \hat{H}_i + e(\Delta\chi_i, H_{i-1}) \tag{4.3.18}$$

where $e(\Delta\chi_i, H_{i-1})$ represents the approximation error, and

$$\begin{aligned} \hat{H}_i &= a_0 + a_1\Delta\chi_i + a_2 H_{i-1} + a_3\Delta\chi_i H_{i-1} + a_4(\Delta\chi_i)^2 + \\ &\quad a_5\Delta\chi_i H_{i-1}^2 + a_6\Delta\chi_i\sqrt{H_{i-1}} \end{aligned} \tag{4.3.19}$$

is the approximation function.

In order to find the coefficients $a_1, \cdots, a_6$ in equation (4.3.19), the mechanical system was simulated off-line for several hopping cycles with different values used for $\Delta\chi$ and initial height. The hopping machine was placed at an initial height, allowed to fall, and actuated by $\Delta\chi$, during stance period. The resulting height of the next hop is recorded, along with the initial height and the control parameter value which produced it. After obtaining sufficient data, $a_1, \cdots, a_6$ were found using a least-squares error curve fit.

Now, in order to calculate the actuator length $\Delta\chi_i$, the desired hop height H_d is substituted for $\hat{H}_i$ in equation (4.3.19), which is then solved in terms of $\Delta\chi_i$ to give:

$$\Delta\chi_i = \frac{-B \pm \sqrt{B^2 - 4a_4(a_0 + a_2 H_{i-1} - H_d)}}{2a_4} \tag{4.3.20}$$

where

$$B = a_1 + a_3 H_{i-1} + a_5 H_{i-1}^2 + a_6\sqrt{H_{i-1}} \tag{4.3.21}$$

The accuracy of Equation (4.3.20) may then be further improved through the use of an estimator for the approximation error for the i^{th} hop,

$$e(\Delta\chi_i, H_{i-1}) \approx e(\Delta\chi_{i-1}, H_{i-2}) = H_{i-1} - \hat{H}_{i-1} \tag{4.3.22}$$

which, when substituted into Equation (4.3.20) creates a more accurate control signal expression:

$$\Delta\chi_i = \frac{-B \pm \sqrt{B^2 - 4a_4(a_0 + a_2 H_{i-1} - (H_d - H_{i-1} + \hat{H}_{i-1}))}}{2a_4} \tag{4.3.23}$$

Equation (4.3.23) is claimed to be more accurate than Equation (4.3.20), but it requires the storage of two most recent hops, instead of the single most recent. Note that each equation has two solutions due to the square root function. In each case, one such solution will present a physical impossibility for the actuator and will be ignored. SSE increases with the deviation of system parameters from the nominal case. Integral-error feedback control, which is based on previous hop heights, is added in order to compensate for these errors. Integral control action reduced SSE from order of 10^{-2}m to order of 10^{-3} m in experimental results presented in (8).

In (21) and (24), the actuator length was considered to be the control input to the system. In (8) and (22), the mechanical model includes the actuator dynamics and the voltage applied to the DC motor is the control input.

4.3.2 Simulation

The following simulations attempt to reproduce the results in (22).

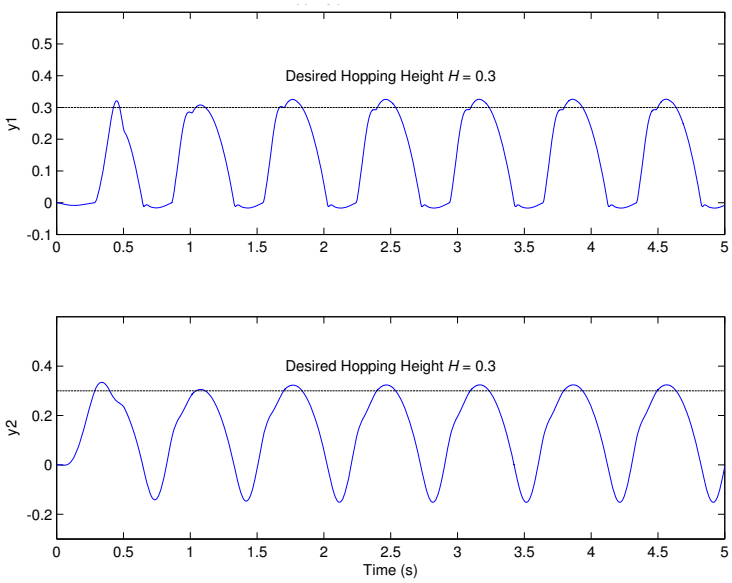

Fig. 4.7: NIV-vertical hopping with desired height 0.3

Figure 4.7 shows the response of the system in the time domain when the desired

height $H = 0.3$. The height of the top mass, y_2 and the height of the foot, y_0 are shown in the top plot.

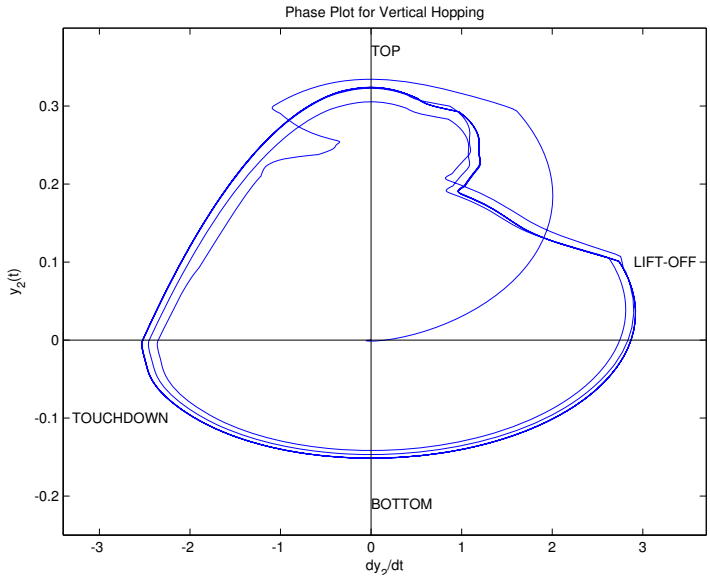

Fig. 4.8: NIV-phase plot for vertical hopping

Figure 4.8 is a phase plot of the $\dot{y}_2$ and y_2. The system starts off at $(\dot{y}_2, y_2) = (0, 0)$, in the middle of the phase plot. The system then approaches a periodic orbit. In the above figure plot, we use the height of the body relative to its rest position on the ground. (i.e. suppose y_2 is at 1.9 m and the length from the body to the foot is 1.5, then $1.9 - 1.5 = 0.4$ is used instead of 1.9).

We chose to reproduce their results in order to avoid the extensive simulations required to determine the coefficients $a_1, \cdots, a_6$ for a new system.

Chapter 5. **LINEAR QUADRATIC REGULATOR**

The optimal Linear Quadratic Regulator (LQR) method is a powerful technique for designing controllers for complex systems that have stringent performance requirements. For most realistic applications, the LQR problem must be solved by using a Computer-Aided-Design package. A simulation is conducted by using MATLAB.

Consider the system

$$\dot{x} = A(t)x + B(t)u, \quad x(0) = x_0 \tag{5.0.1}$$

where $x \in R^n, u \in R^m$ with associated quadratic performance index

$$J = \frac{1}{2}x^T(T)S(T)x(T) + \frac{1}{2}\int_{t_0}^{T}(x^TQ(t)x + u^TR(t)u)dt \tag{5.0.2}$$

$[t_0, T]$ is the time interval over which we are interested in the behavior of the plant.

We want to determine the control $u^*(t)$ on $[t_0, T]$ that minimizes J for the case where the final state is fixed. In this case u^* will turn out to be an open-loop control.

We assume that the final time T is fixed and known, and that no function of the final state $\psi(x(T))$ is specified. The initial plant state $x(t_0)$ is given. Weighting matrices S(T) and Q(T) are symmetric and positive semidefinite, and $R(t)$ is symmetric and positive definite, for all $t \in [t_0, T]$.

5.1 The State and Costate Equations

The Hamiltonian is

$$H(t) = \frac{1}{2}(x^TQx + u^TRu) + \lambda^T(Ax + Bu) \tag{5.1.3}$$

where $\lambda(t) \in R^n$ is an undetermined multiplier, commonly referred to as the costate variables. The state and costate equations are

$$\dot{x} \;=\; \frac{\partial H}{\partial \lambda} = Ax + Bu \tag{5.1.4}$$

$$-\dot{\lambda} = \frac{\partial H}{\partial x} = Qx + A^T\lambda \tag{5.1.5}$$

and the stationary condition is

$$0 = \frac{\partial H}{\partial u} = Ru + B^T\lambda \tag{5.1.6}$$

Solving (5.1.6) yields the optimal control in terms of the costate

$$u(t) = -R^{-1}B^T\lambda(t) \tag{5.1.7}$$

Using (5.1.7) in the state equation yields the homogeneous *Hamiltonian system.*

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} \tag{5.1.8}$$

5.2 Fixed Final State and Open Loop Control

Suppose that the initial state is known to be $x(t_0)$ and that the control objective is to drive the state exactly to the given fixed reference value $r(T)$ at the final time T. Since $x(T)$ is fixed at $r(T)$, it is redundant to include a final state weighting in the cost index. Let $S(T) = 0$. Also, for our LQR design, we are only interested in minimizing the control effort, so let $Q = 0$ so that the cost function reduces to

$$J = \frac{1}{2}\int_{t_0}^{T} u^T R u dt \tag{5.2.9}$$

The state and costate equation are now

$$\begin{aligned} \dot{x} &= Ax - BR^{-1}B^T\lambda \\ \dot{\lambda} &= -A^T\lambda \end{aligned} \tag{5.2.10}$$

Setting $Q = 0$ decoupled the costate equation from the state equation, so its solution is just

$$\lambda(t) = e^{A^T(T-t)}\lambda(t) \tag{5.2.11}$$

where $\lambda(t)$ is still unknown. Using the expression in the state equation yields

$$\dot{x} = Ax - BR^{-1}B^T e^{A^T(T-t)}\lambda(t) \tag{5.2.12}$$

whose solution is

$$x(t) = e^{A(t-t_0)}x(t_0) - \int_{t_0}^{t} e^{A(t-\tau)}BR^{-1}Be^{A^T(T-\tau)}\lambda(T)d\tau \tag{5.2.13}$$

To find $\lambda(T)$, evaluate this at $t = T$ to get

$$x(T) = e^{A(T-t_0)}x(t_0) - G(t_0, T)\lambda(T) \tag{5.2.14}$$

where the weighted continuous reachability gramian is

$$G(t_0, T) = \int_{t_0}^{T} e^{A(T-\tau)}BR^{-1}B^T e^{A^T(T-\tau)}d\tau \tag{5.2.15}$$

According to the final condition

$$\lambda(T) = -G^{-1}(t_0, T)[r(T) - e^{A(T-t_0)}x(t_0)] \tag{5.2.16}$$

Finally the optimal control can be written as

$$u^*(t) = R^{-1}B^T e^{A^T(T-t)}G^{-1}(t_0, T)[r(T) - e^{A(T-t_0)}x(t_0)] \tag{5.2.17}$$

This is our result: it is the minimum-energy control that drives the given initial state $x(t_0)$ to the desired final reference value of $x(T) = r(T)$.

5.2.1 Dynamics of the Hopper

For the vertical case, the hopper can be modeled as follows:

$$\begin{aligned} M_1\ddot{y}_1 &= F_y - F_T - M_1 g \\ M_2\ddot{y}_2 &= F_T - M_2 g \\ y_0 &= y_1 - r_1 \\ w &= y_2 - y_1 + r_1 - r_2 \end{aligned} \tag{5.2.18}$$

where

$$F_T = \begin{cases} K_L(k_0 - w + \chi) & \text{if} \quad (k_0 - w + \chi) > 0 \\ K_{L2}(k_0 - w + \chi)) - B_{L2}\dot{w} & \text{otherwise} \end{cases} \tag{5.2.19}$$

$$F_y = \begin{cases} -K_G y_0 - B_G \dot{y}_0 & \text{if} \quad y_0 < 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.2.20}$$

During stance period, $F_y = -K_G y_0 - B_G \dot{y}_0$ and $F_T = K_L(k_0 - w + \chi)$. As a result, (5.2.18) can be reduced to the following state-space model

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{F_y - F_T}{M_1} - g \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{F_T}{M_2} - g \end{aligned} \tag{5.2.21}$$

by defining the transformation

$$\begin{aligned} x_1 &= y_1 \\ x_2 &= \dot{y}_1 \\ x_3 &= y_2 \\ x_4 &= \dot{y}_2 \end{aligned} \tag{5.2.22}$$

The state space model can be represented as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_G+K_L}{M_1} & -\frac{B_G}{M_1} & \frac{K_L}{M_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{K_L}{M_2} & 0 & -\frac{K_L}{M_2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_G r_1 - K_L(k_0 - r)}{M_1} \\ 0 \\ \frac{K_L(k_0 - r)}{M_2} - g \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{K_L}{M_1} \\ 0 \\ \frac{K_L}{M_2} \end{bmatrix} \tag{5.2.23}$$

where $u = \chi$.

Since during flight phase, there is no control applied to the system and the control can only be applied during the stance phase. Thus the problem can be formulated as how to transfer the states from

$$\begin{bmatrix} r_1 \\ -\sqrt{2gH_{i-1}} \\ k_0 + r_2 + \chi_0 \\ -\sqrt{2gH_{i-1}} \end{bmatrix} \quad \text{to} \quad \begin{bmatrix} r_1 \\ \frac{M_1+M_2}{M_2}\sqrt{2gH_i} \\ k_0 + r_2 + \chi_0 \\ \frac{M_1+M_2}{M_2}\sqrt{2gH_i} \end{bmatrix}$$

within time $T = \pi\sqrt{\frac{M_2}{K_L}}$ (see (3.2.12)). H_{i-1} is the previous hopping height and H_i is the desired hopping height for this cycle.

After the coordinate transformation, the state space model (5.2.23) is transformed into

$$\begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \\ \dot{\bar{x}}_3 \\ \dot{\bar{x}}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_G+K_L}{M_1} & -\frac{B_G}{M_1} & \frac{K_L}{M_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{K_L}{M_2} & 0 & -\frac{K_L}{M_2} & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{K_L}{M_1} \\ 0 \\ \frac{K_L}{M_2} \end{bmatrix} \chi \tag{5.2.24}$$

where

$$\begin{aligned} \bar{x}_1 &= x_1 + \frac{(M_1+M_2)g}{K_G} - r_1 \\ \bar{x}_2 &= x_2 \\ \bar{x}_3 &= x_3 + \frac{(M_1+M_2)g}{K_G} - r_1 - k_0 + r + \frac{M_2}{K_L}g \\ \bar{x}_4 &= x_4 \end{aligned}$$

Next, we examine the controllability of the linear system.

$$\text{A:} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_G+K_L}{M_1} & -\frac{B_G}{M_1} & \frac{K_L}{M_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{K_L}{M_2} & 0 & -\frac{K_L}{M_2} & 0 \end{bmatrix} \qquad \text{B:} \begin{bmatrix} 0 \\ -\frac{K_L}{M_1} \\ 0 \\ \frac{K_L}{M_2} \end{bmatrix} \tag{5.2.25}$$

$$[\lambda I - A \vdots B] \sim \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ \lambda(\lambda + \frac{B_G}{M_1}) + \frac{K_G+K_L}{M_1} & 0 & -\frac{K_L}{M_1} & 0 & -\frac{K_L}{M_1} \\ 0 & 0 & 0 & -1 & 0 \\ \lambda^2 + \frac{B_G}{M_1}\lambda + \frac{K_G}{M_1} & 0 & \frac{M_2}{M_1}\lambda^2 & 0 & 0 \end{bmatrix} \tag{5.2.26}$$

for $\lambda = 0$ Full Rank

for $\lambda \neq 0$ Full Rank

So the system is controllable.

We choose $R = 0.5 * K_L = 500$, which is a scalar because the system has only one input (i.e. $m = 1$). Since the weighted controllability gramian is only related to t_0 and T so it can be computed offline. Let $t_0 = 0$ and $T = T_{STANCE} = 0.3142$

$$G(t_0, T) = \int_{t_0}^{T} e^{A(T-\tau)} B R^{-1} B^T e^{A^T(T-\tau)} d\tau \tag{5.2.27}$$

Using Matlab we get

$$G(t_0, T) = \begin{bmatrix} 0.0015 & 0.0000 & 0.0026 & -0.0189 \\ 0.0000 & 13.5853 & 0.0193 & 0.0784 \\ 0.0026 & 0.0193 & 0.0296 & 0.0018 \\ -0.0189 & 0.0784 & 0.0018 & 2.4661 \end{bmatrix} \tag{5.2.28}$$

The controller u is as follows:

$$u^*(t) = R^{-1} B^T e^{A^T(T-t)} G^{-1}(t_0, T)[r(T) - e^{A(T-t_0)} x(t_0)] \tag{5.2.29}$$

5.3 Simulation

The following results represent the case when the initial height is 0.2 m and the desired height, $H = 2m$. In this simulation, the control input χ was calculated using the LQR method and actuated right after touchdown. Compared with energy based control method, this will give us more time to do the control and the control difficulty is reduced in some sense. At touchdown, the position actuator starts to compress the spring based on the calculations done by the controller. After lift-off, χ was retracted back to an initial position. For these simulations, position actuator's initial position is 0.

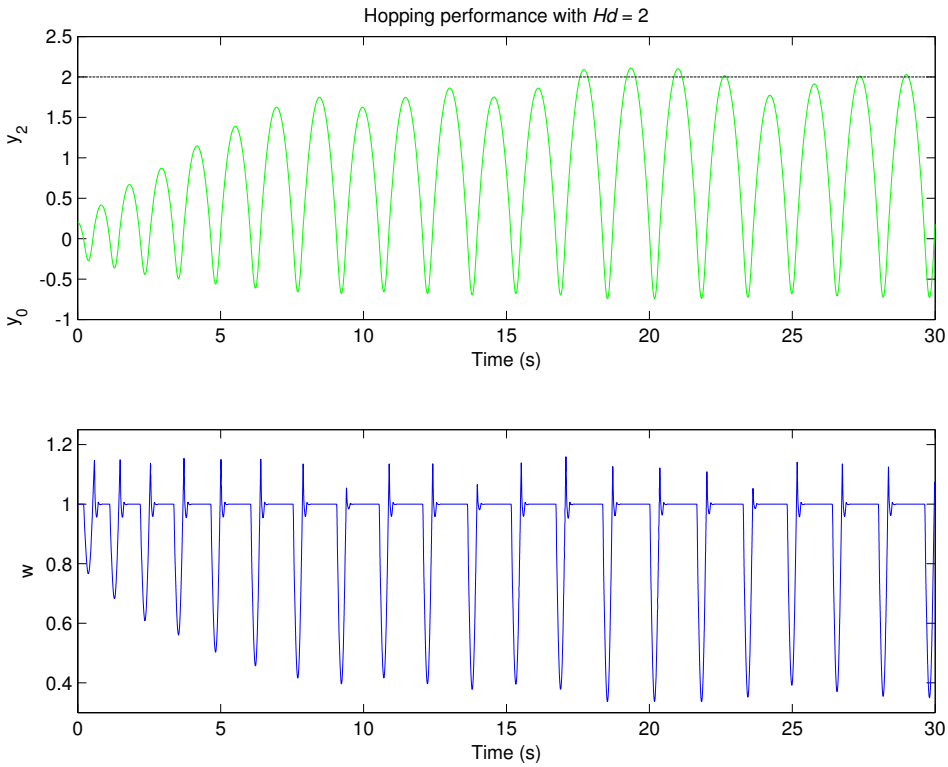

Fig. 5.1: LQR-vertical hopping start from rest

Figure 5.1 shows the response of the system in the time domain when the desired height $H = 2$. The height of the top mass, y_2 is shown in the top plot. The bottom plot shows w, the distance from the hip to the foot. Recall that the spring is compressed when $\chi - w < k_0$.

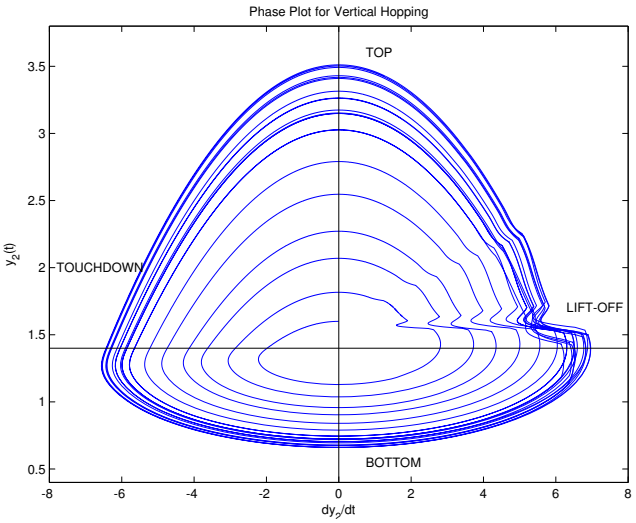

Fig. 5.2: LQR-phase plot for vertical hopping

Figure 5.2 is the phase plot of $\dot{y}_2$ and y_2. The system starts off at $(\dot{y}_2, y_2) = (0, 1.6)$, in the middle of the phase plot. The system then approaches a periodic orbit.

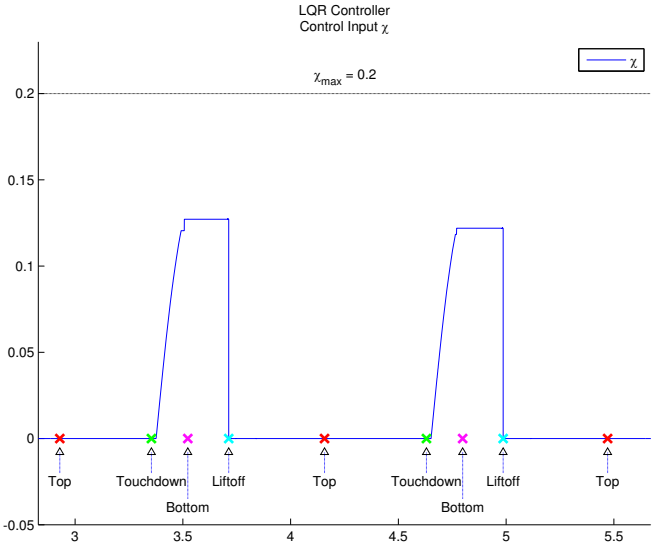

Fig. 5.3: LQR-χ plot for vertical hopping

Figure 5.3 is a plot of the actuator length for the two selected cycles. Note that the actuator starts to compress the spring at the touchdown event so that it has more freedom to choose control. The stance period is very short, leaving the controller little time to bring the system to the desired state. The most difficult problem for the vertical hopping control is that it has to be done at a short period. By choosing to start control at the touchdown event, this control difficulty can be somewhat reduced.

Chapter 6. **PD CONTROLLER**

In previous chapter, we try to use the Linear Quadratic Regulator to control the vertical hopping. By doing so, two things are expected. First, the controller will initiate the hopping, drive the robot to the desired height and then maintain the desired hopping height. Second, the controller use minimum energy to actuate the robot from the initial hopping height to the desired height. However, the controller involves complicated calculation and not easy to implement. In the simulation, it is not smooth at the steady state hopping (i.e. to hop at constant level) and there are fluctuations. However, it is the first time, we are trying to apply optimal control for one-legged hopping robot. In this chapter, a PD controller is proposed for vertical control. Compared with the LQR, it is simpler and the simulation results are better.

6.1 Control Strategy

The design of a PD controller is explained into details in this section. By using PD controller, the actuator starts to compress the spring right after the touchdown and it enable us to control the hopping robot during the whole stance period. Also, the incremental displacement of the actuator becomes largest once it reaches the bottom. For a spring compressed 4 cm, the energy injected into the spring at the second 2 cm is larger than first 2 cm. For the same reason, the actuator injects maximum energy at the bottom event and this is optimized in some sense.

We are going to take two values as our reference point.

First, let's consider the height of the body at the bottom. Once the robot reaches its desired height, it will maintain the hopping height. At the steady state hopping, there is a balance between the energy injected into the system and the lost energy and for every cycle the spring will be compressed the same amount.

At desired height H, let's assume the body's height at the bottom event is y_{bot} at the steady state hopping and it can be calculated by

$$MgH = \frac{1}{2}K_L y_{bot}^2 - Mgy_{bot} \tag{6.1.1}$$

$$y_{bot} = -(Mg + \sqrt{(Mg)^2 + 2K_L MgH})/K \tag{6.1.2}$$

Second, when the robot reaches its steady state, the maximum speed of the body from top to the bottom will be taken as the second reference point. Right after the top event, the robot starts to fall and it will reach its maximum speed soon after the touchdown event and then reduce to zero at the bottom. Then it will accelerate upward and leave the ground. The pattern and magnitude of the speed is similar for each hopping cycle at the steady state.

The maximum speed v_{ref} can be calculated by using the following formula.

$$MgH = \frac{1}{2}Mv_{ref}^2 + \frac{1}{2}K_L x_{bal}^2 - Mgx_{bal} \tag{6.1.3}$$

$$v_{ref} = \sqrt{2(MgH + Mgx_{bal} - \frac{1}{2}K_L x_{bal}^2)/M} \tag{6.1.4}$$

where x_{bal} is displacement that the spring is compressed so that the spring's elastic force equals to the gravitational force and after passing this point the body will be deaccelerated.

$$x_{bal} = M_2 g/K_L = 0.1 \tag{6.1.5}$$

By using the above two values as reference, the controller can be written in the following way.

$$\Delta\chi = k_p(y_1 - 1.4 - y_{bot}) + k_d||y_2| - v_{ref}| \tag{6.1.6}$$

where $\Delta\chi$ is the incremental displacement of the actuator and y_1 is the body's height and y_2 is the body's velocity. k_p and k_d are the parameters that we can tune.

6.2 Simulation Results

Simulation is done in Matlab and the desired height H is set to be 2. In this simulation, the two reference points are set to be $y_{bot} = -0.7753$, $v_{ref} = 6.1965$ and

the parameters are chosen to be $k_p = 4 * 1e - 5$ and $k_d = 1.2 * 1e - 5$. The initial hopping height is still 0.2m which is the same as the set up for energy based control simulation.

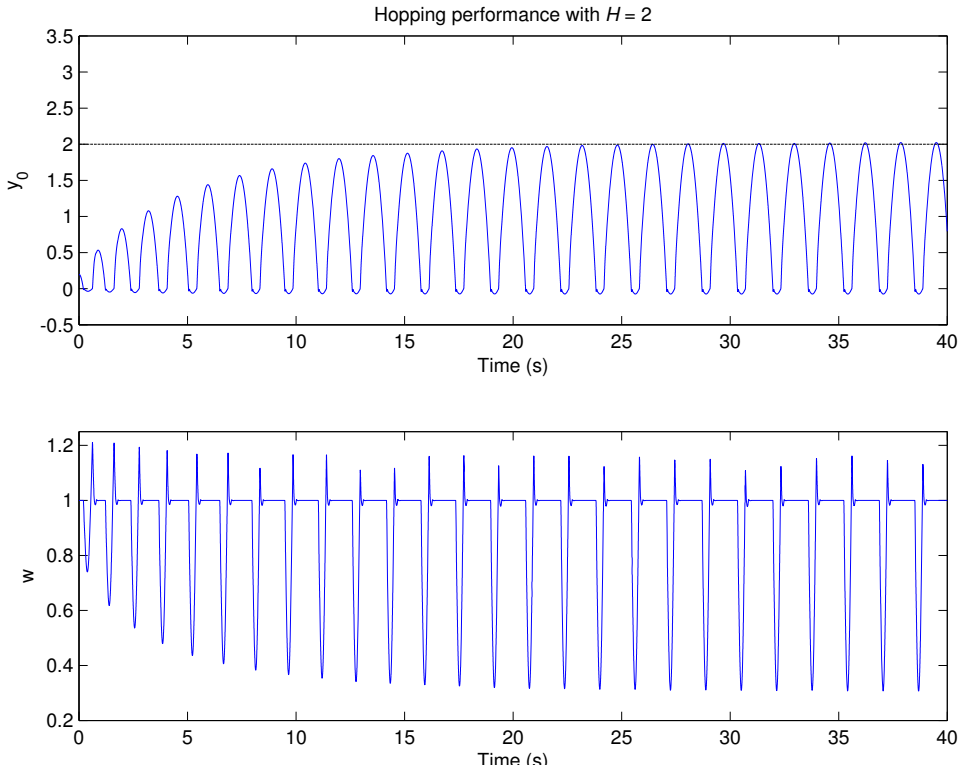

Fig. 6.1: PD controller-hopping start from rest

Figure 6.1 shows the response of the system in the time domain. The height of the body, y_0, is shown in the top plot. In this simulation we choose to plot the body. There are two reasons behind. First at the touchdown event, the foot hits the ground and the speed of the foot goes to zero immediately. Second, at liftoff event the height and the spring hits the mechanical stop and this causes the spring to fluctuate and it affects the foot seriously. However the body is much heavier than the foot and it is less impacted and its height and velocity changed more smoothly. After several hopping cycles, the PD controller drives the robot to the desired height and then maintain the steady state hopping there. It can converge to the desired hopping height exactly.

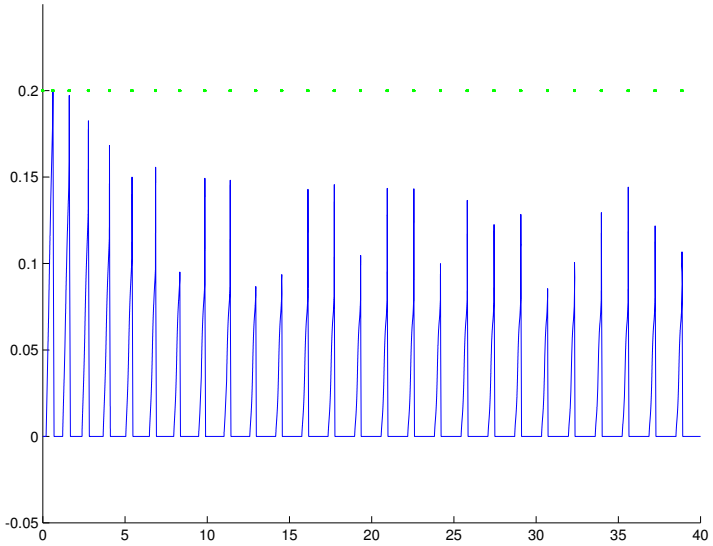

Fig. 6.2: PD controller-position actuator χ

Figure 6.2 is a plot of the actuator length. Note that the actuator does start to compress the spring right after the touchdown event. Also note that there are sudden changes of the χ at the end of the stance period which is the liftoff event. This behavior will be explained later.

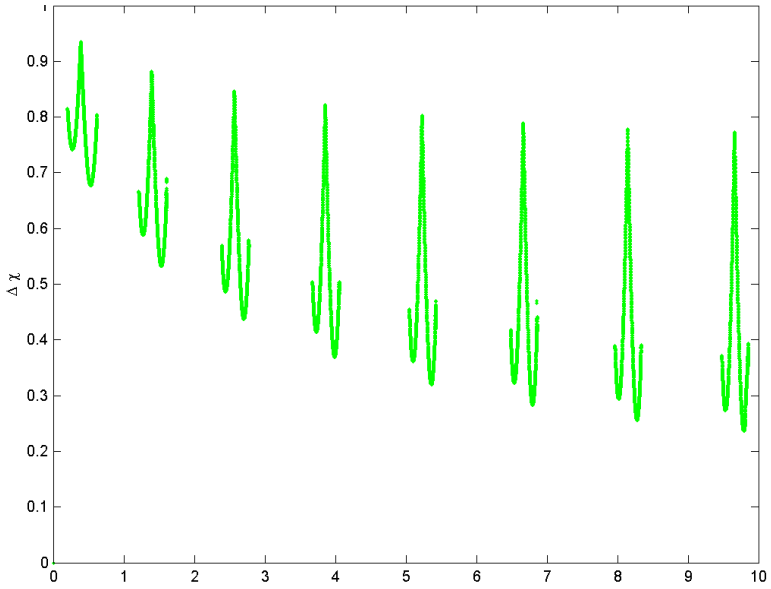

Fig. 6.3: PD controller-incremental displacement $\Delta\chi$

Figure 6.3 plots the incremental displacement of the actuator. Note that, the incremental displacement achieves its biggest amount at the bottom during stance period. The actuator's displacement is larger and also the spring is compressed most at the bottom, so more energy is injected around the bottom event and this lead to

the optimum performance.

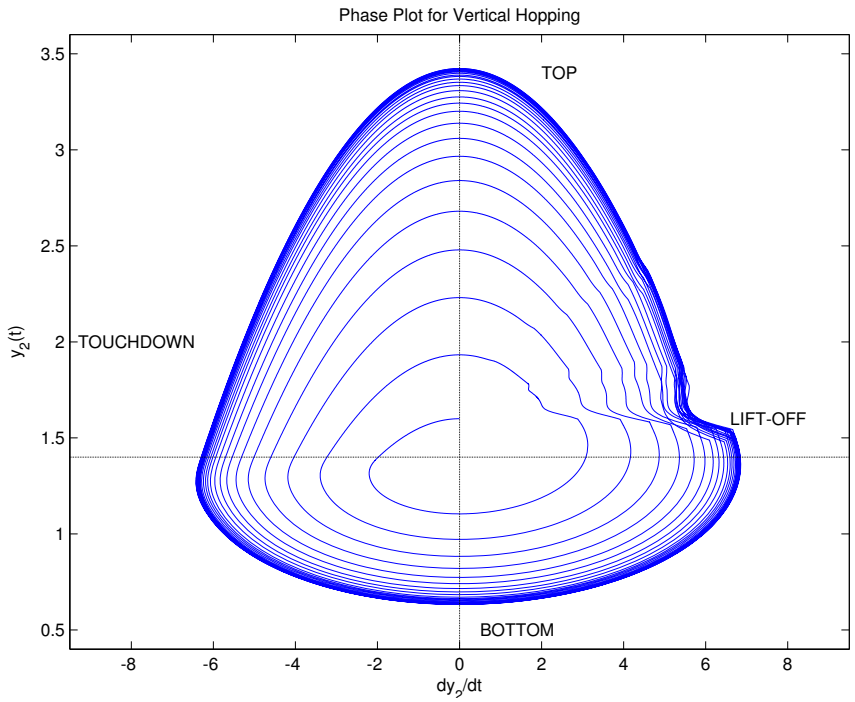

Fig. 6.4: PD controller-phase plot for vertical hopping

Figure 6.4 is a phase plot of the $\dot{y}_2$ and y_2. The system starts off at $(\dot{y}_2, y_2) = (0, 1.6)$, in the middle of the phase plot. After several cycles, the system then approaches a periodic orbit.

6.3 Summary

In reality, it is not easy to detect the bottom event. It is even more difficult if one choose to control the actuator and compress the leg spring starting from the bottom event. Here we are trying to demonstrate how the PD controller works and we start the control at the beginning of the stance period and the control can utilize the whole period. Also, it is easier to detect the touchdown event compared with the bottom event.

Figure 6.2 shows some fluctuations of χ, however, it happens near the liftoff event. During that time, the spring is not compressed that much and so this sudden change won't affect the system's performance that much. At the steady state hopping, the system maintain its hopping height well.

The actuator compress the spring a lot around the bottom event and this lead to the optimal performance in an energy point of view.

Chapter 7. **CONCLUSIONS AND FUTURE WORK**

7.1 Conclusions

In this thesis, a generic vertical hopping machine model is developed. by a springy leg with nonzero mass, a simple body, and an actuated hinge-type hip. The body is forms an elevated mass that must be balanced atop the leg and that it forms a structure from which torques can be applied to the leg. Legs typically do two things during locomotion: they change length and orientation with respect to the body. The dynamics of the one-legged robot is described in Chapter 3. The dynamics are used to develop the simulations of the vertical hopping and test the proposed controllers.

Two historical methods are shown in Chapter 4. One is energy based control and the other one is Near Inverse Method. The objective of the vertical controller is to initiate and terminate hopping, as well as switch between hopping heights quickly. The control problem is complicated by the fact that the hopping can only be controlled during the stance phase. Both control methods can take care of the vertical control problem with their special characters.

The vertical hopping control problem is studied in this thesis and we proposed two methods. The first one is the Linear Quadratic Regulator (LQR) method. The LQR method is a powerful technique for designing controllers for complex systems that have stringent performance requirements. It is the first time we are trying to solve the vertical hopping control using optimal control method. However, the controller is complicated in terms of calculations and implementation. But it inspired our second method, the PD controller. We take two reference points from the steady state hopping, one is the body's height at the bottom and the other is the maximum speed of the body during the period between the touchdown and the bottom. By using this two reference point, we designed the PD controller. The simulation results are very

good. Compared with the LQR method, it is easier and its performance is better.

7.2 Future Works

In the future, this type of work can be extended by figuring out the motors to drive the actuator. The PD controller design is proposed in last chapter. However, we did not focus on developing a real prototype in this thesis and there is not much effort putted in figuring out the motors which can actually drive the actuator. And PD controller is just a beginning of developing optimal vertical hopping control. I believe there will be better ways to do the control.

Also, we studied only the vertical hopping motion in this thesis. The study can be extended in to 2-D, 3-D. By adding horizontal control problem in, the control of a hopping robot is much more complicated and there are many many existing solutions out there. However by studying the horizontal control, the control of a one-legged robot will be completed. Since then, it can be even extended into two legs, four legs and so on. Those are the untouched areas.

BIBLIOGRAPHY

[1] B*ü*hler M. and Koditschek, D.E., From Stable to Chaotic Juggling: Theory Simulation, and Experiments. In *Proc. IEEE Int. Conf. Robotics and Automation*, Cincinnati, OH, May 1990.

[2] , M. and Koditschek, D. E., Analysis of a simplified hopping robot, *The International Journal of Robotics Research*, Vol. 10, pp. 587-605, 1991.

[3] Ehrlich, A. *Vehicle propelled by steppers*. U. S. Pattent 1,691,233, 1928. Adolf Ehrlich, Budapest, Hungary, 1928.

[4] Gurfinkel, V.S., Gurfinkel, E.V., Shneider, A.Y., Devjanin, E.A., Lensky, A.V., and Shitilman, L.G., Walking robot with supervisory control, *Mech. Mach. Theory* 16 (1981), pp. 31-36.

[5] Hirose, S., A study of design and control of a quadruped walking vehicle, *Int. J. Robotics Res*, 3(1984), pp. 113-133.

[6] Hirose, S. and Umetani, Y., The basic motion regulation system for a quadruped walking vehicle, In *ASME Conference on Mechanisms* (Los Angeles, California), 1980.

[7] Lapshin, V. V., Vertical and horizontal motion control of a one-legged hopping machine, *The International Journal of Robotics Research*, Vol. 11, pp. 491-498, 1992.

[8] Lebaudy, A., Prosser, J. and Kam, M., Control Algorithms for a vertically-Constrained One-Legged Hopping Robot, *Proceedings of the 32nd Conference on Decision and Control*, Vol.3, pp. 2688-2693, San Antonio, TX, December 1993.

[9] Liston, R.A. Increasing vehicle agility by legs: The quadruped transporter. In *38th National Meeting of the operations Research Society of America*, 1970.

[10] Lucas, E. Huitieme recreation - La machines a marcher. *Recreat. Math.* 4(1894), 198-204.

[11] Matsuoka, K., A mechanical model of repetitive hopping movements, *Biomechanisms 5*(1980), pp 251-258.

[12] McGeer, T., Powered flight, child's play, silly wheels and walking machines, *IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 1592-1597, 1989.

[13] McGhee, R. B. Vehicular legged locomotion. *Advances in Automation and Robotics*, 1985.

[14] M'Closkey, R. T. and Burdick, J. W., Periodic motions of a hopping robot with vertical and forward motion, *The International Journal of Robotics Research*, Vol. 12, pp. 197-218, 1993.

[15] Mehrandezh, M., Surgenor, B.W. and Dean, S.R.H., Jumping Height Control of an Electrically Actuated, One-Legged Hopping Robot: Modelling and Simulation, *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, U.S.A., Dec. 13-15, 1995.

[16] Miura, H. and Shimoyama, I., Dynamic walk of a biped, *Int. J. Robotics Res.* 3(1983), pp 60-74.

[17] Morrison, R.A. Iron mule train. In *Porceedings of Off-Road Mobility Research Symposium*. International Society for Terrain Vehicle Systems, Washington, D.C., 1968, pp. 381-400.

[18] Mosher, R.S. Test and evaluation of a versatile walking track, *Off Road Mobility Research*, pp. 359-379, Washington, 1968. International Society for Terrain Vehicle systems.

[19] Muybridge. E. *Animals in Motion*, New Dover Edition, Dover Publicatoin, Inc., New York, (first published in 1899), 1957.

[20] Muybridge. E. *The Human Figure in Motion*. Dover Publications, New York, 1955. First ed., Chapman and Hall, London, 1901.

[21] Prosser, J. and Kam, M.(1992a), Height Control of a One-Legged Hopping Machine Using a Near-Inverse Model, *Proceddings of the conference of Information Science and Systems*, Princetor, NJ, pp. 995-1002, March 1992.

[22] Prosser, J. and Kam, M.(1992b), Vertical Control for a Mechanical Model of the One-Legged Hopping Machine, *Frist IEEE Conf. on Cont. Application*, pp.136-141, Princeton, NJ, September 1992.

[23] Rad, H., Gregorio, P. and Bühler M. (1993), Design, Modeling and Control of a Hopping Robot, *Proc of the 1993 IEEE/RSJ Int Conf on Intelligent Robots and Systems*, Yokohama, Japan, July 26-30, pp. 1778-1785.

[24] Raibert, M. H., Hopping in legged systems–modeling and simulation for the 2D one-legged case, *IEEE Trans. Systems, Man and Cybernetics*, Vol. 14, pp. 451-463, 1984.

[25] Raibert, M. H., Brown, H. B. Jr. and Chepponis, M., Experiments in balance with a 3D one-legged hopping machine, *The International Journal of Robotics Research*, Vol. 3, pp. 75-92, 1984.

[26] Raibert, M.H. and Wimberly, F.C., Tabular Control of Balance in a Dynamic Legged System, *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-14, No. 2, pp.451-463, March/April 1984.

[27] Raibert, M. H., *Legged Robots That Balanced*, MIT Press, Cambridge, MA, 1986.

[28] Rao, S. S., *Mechanical Vibrations*, Addison-Wesley, 1984.

[29] Shigley, R. *The Mechanics of Walking Vehicles*. Rep. 7. Land Locomotion Laboratory, Detroit, Mich., 1957.

[30] Snell, E. *Reciprocating Load Carrier*. Patent 2,430,537, 1947.

[31] Sznier, M. and Damborg, M. J., "An Adaptive Controller for a One-Legged Mobile Robot." *IEEE Transitions on Robotics and Automation*, Vol. 5, No. 2, pp. 253-259, April 1989.

[32] Urschel, W.E. *Walking tractor*. Patent 2,491,064, Valparaiso Ind., 1949.

[33] Vakakis, A.F. and Burdick, J.W., Chaotic Motions of A Simplified Hoppong Robot. In *Proc. IEEE Int. Conf. Robotics and Automotion*, pp. 1464-1469, Cincinnati, OH 1990.

[34] Z. Li and J. He., An Energy Perturbation Approach to Limit Cycle Analysis in Legged Locomotion Systems. In *Proc. IEEE Int. Conf. Decision and Control*, pp. 1989-1994, 1990.