Robust Bipedal Locomotion on Unknown Terrain

by

Hongkai Dai

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

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2012

Author .....
Department of Electrical Engineering and Computer Science
Aug,31,2012

Certified by .....
Russ Tedrake
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Theses

Robust Bipedal Locomotion on Unknown Terrain

by

Hongkai Dai

Submitted to the Department of Electrical Engineering and Computer Science
on Aug,31,2012, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

A wide variety of bipedal robots have been constructed with the goal of achieving natural and efficient walking in outdoor environments. Unfortunately, there is still a lack of general schemes enabling the robots to reject terrain disturbances. In this thesis, two approaches are presented to enhance the performance of bipedal robots walking on modest terrain. The first approach searches for a walking gait that is intrinsically easily stabilized. The second approach constructs a robust controller to steer the robot towards the designated walking gait.

Mathematically, the problem is modeled as rejecting the uncertainty in the guard function of a hybrid nonlinear system. Two metrics are proposed to quantify the robustness of such systems. The first metric concerns the 'average performance' of a robot walking over a stochastic terrain. The expected LQR cost-to-go for the post-impact states is chosen to measure the difficulty of steering those perturbed states back to the desired trajectory. A nonlinear programming problem is formulated to search for a trajectory which takes the least effort to stabilize. The second metric deals with the 'worst case performance', and defines the L_2 gain for the linearization of the hybrid nonlinear system around a nominal periodic trajectory. In order to reduce the L_2 gain, an iterative optimization scheme is presented. In each iteration, the algorithm solves a semidefinite programming problem to find the quadratic storage function and integrates a periodic differential Riccati equation to compute the linear controller. The simulation results demonstrate that both metrics are correlated to the actual number of steps the robot can traverse on the rough terrain without falling down. By optimizing these two metrics, the robot can walk a much longer distance over the unknown landscape.

Thesis Supervisor: Russ Tedrake
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

First and foremost, I would like to thank my advisor, Russ Tedrake, for his invaluable mentorship and steady encouragement. Russ's sharp insight in transforming a complicated problem into a tractable formulation greatly helps me sprouting research ideas. When I tried to define the robustness measure for the periodic system, his suggestion on sticking to the continuous system significantly simplifies the analysis of a infinite horizon signal. Moreover, Russ is the role model in combining intelligence and devotion, whose frequent midnight svn commits always remind me that thinking smart and working hard are the qualities essential to producing great works.

My fellow labmates in the Robot Locomotion Group contributed intriguing perspectives through the years. I would like to thank John Roberts and Zack Jackowski for the discussion we made on designing the trajectory, as well as Mark Tobenkin for the help of robust control, Anirudha Majumdar, Michael Posa and Amir Ali Ahmadi, Joseph Moore for the help of semidefinite programming, and Andy Barry, Mark Tobenkin and Joseph Moore for their great help on the thesis writing.

I want to thank specially to my girlfriend, Jiahui Shi, who has been with me in every step, even when we were separated by the whole continent. Finally, I would like to thank my parents as well as my grandparents, their love are always the sweetest comfort for me.

Contents

List of Figures

Chapter 1

Introduction

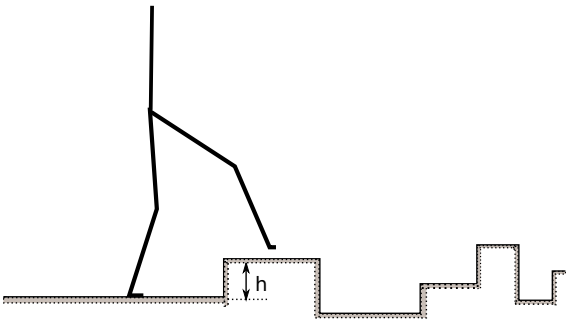
1.1 Motivation

Legged animals exhibit the extraordinary ability to adapt to variable terrain elevation during walking, yet spend little effort tweaking their natural walking gait. Such an ability is essential for robots to operate in an outdoor environment. For decades, engineers have been analyzing the animal walking patterns with the aim of bringing about legged robots that are comparably robust on unknown terrain. A wide variety of legged robots capable of overcoming rough terrain have been built to examine diverse locomotion approaches. Usually, each control strategy for these robots has been tailored to the specific robot model, and a general framework is still missing to improve the robustness for robots walking on uneven terrain. This work presents a new control strategy applicable to general robot model, with simulation validation on some robots.

One reason why most current robots are incapable of walking in an unknown environment is that they are not robust enough to tolerate the diversified sources of uncertainty. Such uncertainty sources include gusts of wind, uncontrolled slip, unmodeled friction forces, etc. Among all those uncertainties, the terrain uncertainty is of particular interest, as it pertains only to hybrid dynamical systems. For a hybrid system like the bipedal robot, its dynamical behavior is changed by the contact force every time a ground impact occurs. Uncertainty in the ground impact can result in severe drift of the post-impact state away from the desired walking pattern. Underactuated systems like legged robots do not possess enough control authority to cancel out the state error arising from unscheduled ground impacts; thus the robot will fall down.

We examine two possible approaches to reduce the failure risk of a robot walking over unknown terrain. The first one is to walk in a gait that is intrinsically easily stabilized on the rough terrain. For humans, one such possible gait is to bend the knees and to take a shorter step. The other approach is to control the

(a) A dog running on uneven road, by Jochen Bongaerts "http://www.flickr.com/photos/jochenbongaerts/2938148467/sizes/z/in/photostream/"



(b) An illustrative figure depicting a bipedal robot walking on an unknown terrain, with terrain height at the next foot collision point to be unknown variable h

Figure 1-1: Legged locomotion on uneven terrain

body motion so that the robot will follow the designated gait. Again, for humans, one candidate motion is to swing the torso so as to balance the body by regulating the center of mass. We will show that both approaches can be mathematically formulated as solving optimization problems.

The goal of this work is to bring about a general framework to reduce the failure risk of a robot walking over uneven terrain. Two metrics are defined to quantify the robustness of bipedal robots to the terrain disturbances, and control strategies are developed by optimizing such robustness metrics.

1.2 Contributions

The contributions of this thesis include two metrics to quantify the robustness of the legged robots walking on the unknown terrain. These metrics are derived for separate models of ground profile. When the terrain elevation is modeled as a realization of a stochastic process, one metric measures the average cost for stabilizing the post-impact state towards the desired trajectory. When the terrain elevation is bounded, the other metric measures the worst-case sensitivity of the error signal to the terrain disturbance. Moreover, for each metric, an optimization scheme is presented to reduce the sensitivity. A trajectory optimization is formulated to improve the robustness on the stochastic terrain, and an iterative semidefinite programming is presented to alleviate the worst-case sensitivity. Simulation of simple legged robot models shows that these two schemes increase the number of steps the robot can traverse over an uneven terrain.

1.3 Organization

The thesis is organized as follows, Chapter 2 reviews the previous work on robots walking over uneven terrain. Chapter 3 formulates the problem as a hybrid system with uncertainty at the guard. Chapter 4 describes the trajectory optimization scheme for finding a robust walking gait, Chapter 5 presents the robust control synthesis scheme, Chapter 6 concludes with a discussion of our approach and future work.

Chapter 2

Previous Work

2.1 Passive Robot walker

Honda's ASIMO are widely considered as the state-of-the-art walking robot. However, as a humanoid, its gait is apparently distinct from the natural human walking pattern. Such robot uses high-gain feedback controllers to cancel out the natural dynamics, so as to strictly follow the desired trajectory. This requires large torque, and the robot needs to be fully actuated, i.e, one actuator for one degree of freedom. This mechanical design and control strategy causes it energy-inefficient in its walking.

On the other hand, a passive walker, introduced by the seminal work of McGeer [23, 24, 22, 25], can walk down a shallow slope solely due to the gravity force. Thus they are extremely energy efficient. This walker is also inherently stable with respect to small disturbances [9]. Unfortunately passive walkers have notoriously fragile basins of attraction, and minor terrain elevation will cause them to fall. In this thesis, we will add actuators to the passive walker to improve its robustness through actively steering the robot.

2.2 Robots robust walking

Extensive efforts have been put to study the locomotion patterns of legged robots by robotics communities [33, 45, 8], and several successful control strategies have been devised for certain robot models. For the spring-mass model, swing leg retraction proves to improve its stability of running [36]. Furthermore, through extensive simulation, Ernst discovered the control strategy for the *Spring-Loaded Inverted Pendulum* (SLIP) model to running at a constant speed on uneven terrain [12]. This result demonstrates that by carefully designing the control strategy and the walking gait, a certain type of robot can perfectly reject the terrain disturbance. We are motivated by such result, and desire to study if a general control strategy exists for all

Figure 2-1: A passive walker by Steven Collins [9]

robot models. Hopefully, our more general approach can reproduce their model-specific result.

There has been several motion planning and control strategies developed for robots navigating known terrain. For example, the quadruped robot, the *Little Dog*, can adapt to a large set of terrain profiles [20, 37, 19, 49]. Although greatly successful on the indoor challenging terrain, all those strategies require perfect terrain modeling, which is not realistic in the outdoor tasks due to perception error. In our work, we model the terrain height at the incoming step as an unknown variable, so as to explicitly address such ground perception error.

By adopting the notion of transversal surface, Manchester demonstrates that a general linear controller can stabilize the system in the vicinity of the desired trajectory [21]. On the *Compass Gait* example, the robot can safely step down a stair with stair height equals to 6% of its leg length. This result is a clear evidence that a linear controller is powerful enough to stabilize a bipedal robot, which is a nonlinear hybrid system. It also indicates that selecting the appropriate phase being tracked on the nominal trajectory can make a big difference on the overall stability.

Morimoto considers the modeling error as the uncertainty in the continuous mode of the walking robot [28]. By mini-maxing the H_∞ related cost function, he designs a nominal trajectory together with the control strategy to track that trajectory. Experiment shows that a bipedal robot can take much larger steps on a flat terrain, compared to a PD controller. This approach implies that H_∞ cost, and thus L_2 gain, is a good indicator of the robustness of a walking robot on a flat terrain. We want to explicitly focus on the disturbance on the hybrid guard function and also explore such a good indicator of robustness to the terrain disturbances.

A common approach used in analyzing periodic walking is to compute the Poincaré map, and to model the step-to-step behavior as a discrete system [7, 38, 35, 29]. To enable MABEL robot walking over unknown terrain, Park linearizes the one-step Poincaré map of MABEL, and represents the actuator input within each step as a Bézier curve. He then constructs an H_∞ controller to adjust the values of the two Bézier coefficients in every step [31]. The simulation results shows that such a discrete H_∞ controller can reduce the failure rate. In our scheme, rather than focusing on the discretized system and controlling only finite number of coefficients, we aim to work on the continuous system directly, and designs the continuous H_∞ controller for the whole period.

2.3 Trajectory Optimization

Trajectory optimization is a powerful tool for motion planning to design the nominal trajectory [43, 42, 11, 44, 17, 34, 10, 13, 32]. Well established nonlinear optimization techniques, such as the sequential quadratic programming, and software packages like *SNOPT* and *IPOPT* are extensively used [14, 46]. By minimizing a meaningful cost function (duration time, control cost, cost of transport), the performance of the robot with respect to some benchmarks gets improved if the robot follows the designed trajectory. There are basically two types of formulations in the trajectory optimization, the indirect (shooting) method and direct method [1]. The indirect method, known as the adjoint method in the optimal control community, treats the control inputs as decision variables, and explicitly integrates the dynamics function to incorporate it into the optimization scheme. This approach suffers from the 'dog wagging the tail' problem, as a small change in the initial control input will cause huge variation of the final state. Another approach, the direct one, regard both states and control inputs as decision variables, and the dynamics are treated as constraints in the optimization program. This approach is much more numerically better conditioned. Besides, although large in size, the Jacobian matrix of the dynamics constraints is generally sparse, and thus can be easily handled by modern linear solvers. Moreover, it is easier to impose state constraints with the direct approach. The state constraint is important in the hybrid system case, as we need to constrain each state in their corresponding mode. We will stick to the direct approach in our work.

2.4 Periodic Motion

Walking on the flat terrain has a periodic pattern. In nonlinear control theory, a periodic motion that is asymptotically stable or unstable is called a limit cycle. There exist many criteria to characterize the stability of a limit cycle, like the *Largest Floquet Multiplier* [40] and the *Gait Sensitivity Norm* [18]. Mombaur presents a non-differentiable optimization algorithm to design a robust limit cycle for a passive walking machine [27]. Her work demonstrates that constructing a walking gait that is robust by itself is a viable approach to improve the overall robustness of the passive system, and motivates us to design a differentiable optimization scheme, so as to improve the robustness of a controlled limit cycle.

An important tool in the linear control design is the Riccati equations. Extensive study has shown that on a periodic orbit, the Riccati equation also has a periodic solution [3, 2]. Both LQR and H_∞ control synthesis on a periodic orbit rely on the periodic Riccati solution [47]. In our work, we will design the linear controller on the periodic orbit, and the periodic Riccati equation will play an important role in determining the controller formulation.

Chapter 3

Problem Formulation

As is commonplace in optimization-based trajectory design for legged robots [27, 39, 31], we describe the dynamics of a legged robot as a hybrid dynamical system, which has both continuous and discrete variables [5]. In our case, we consider the system with continuous states and control inputs, and discrete modes. A guard function determines when to trigger the mode transition, and the mode transition function maps the pre-impact state to a new state in the destination mode. In the case of robot walking, the distance from the foot to the ground is the guard function. We model the ground as a rigid body, and the collision between the robot and the ground as an impulsive transition between different modes. For simplicity we focus on the case where the hybrid system has only one mode and a single transition function.

$$\dot{x} = f(x, u) \quad \text{if } \phi(x, u) > h \tag{3.1}$$

$$x_h^+ = \Delta(x_h^-, u_h^-) \quad \text{if } \phi(x_h^-, u_h^-) = h \tag{3.2}$$

Where x is the continuous state, u is the control action, $\phi(x, u)$ is the current height of the foot, h is the terrain height at the next impact position. When the foot is above the ground ($\phi(x, u) > h$), the system dynamics is governed by the continuous differential equation $\dot{x} = f(x, u)$; when the foot touches the ground with the terrain height h, the mode transition function Δ maps the pre-impact state x_h^- to the post-impact state x_h^+. Here we suppose we have full access to the state information. Figure (3-1) illustrates the dynamical system behavior.

Note that the unknown variable h only exists in the guard function, the 'hybrid' part of the system. Controlling a hybrid system with uncertainty in the guard function is challenging, since unlike tradition disturbance which exists in the continuous dynamics, in our system, the disturbance changes the occurrence time of the mode switch. And the drift in the impact time will cause large perturbation of the post-impact

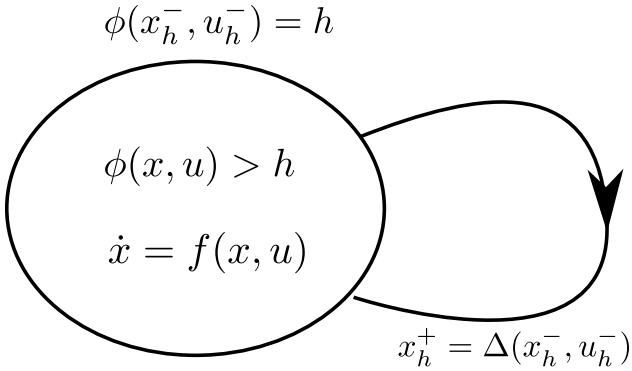


Figure 3-1: A hybrid system with single continuous mode and single transition

states, making those states hard to stabilize.

Chapter 4

Optimizing the robust limit cycle

4.1 Intuition

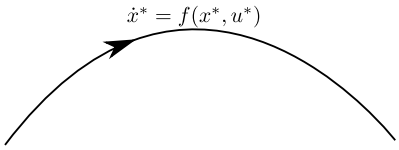
Suppose the terrain height h at the incoming step is a random variable, and its distribution is known beforehand. Rather than designing one control trajectory for each terrain scenario, our goal is to design a single nominal one-step limit cycle that can minimize the effect of all possible terrain disturbances. Thus all possible ground impacts should be examined for that single trajectory. Denote the nominal state trajectory as $x^*(\tau)$, and the nominal control trajectory as $u^*(\tau)$, we depict how the state propagates along the nominal trajectory in Fig 4-1 for three different ground scenarios. The figures are explained as follows:

- Suppose h_0 is the nominal terrain, and the nominal ground impact happens at the phase τ_0^-, the transition functions Δ lands the post-impact state x_0^+ exactly on the nominal state trajectory x^*, due to the existence of a periodic orbit on the nominal terrain, as in Fig 4-1a-4-1c

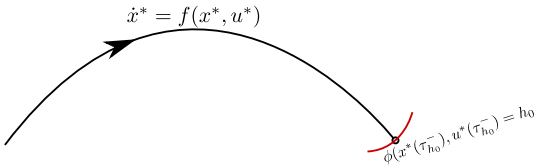
- In Fig 4-1d, if the terrain is higher than the nominal one, with $h_1 > h_0$, then the nominal trajectory will hit the impact surface $\{(x^*(\tau_1^-), u^*(\tau_1^-))|\ \phi(x^*(\tau_1^-), u^*(\tau_1^-)) = h_1\}$ at the pre-impact phase $\tau_1^- < \tau_0^-$. And the state transition function maps the pre-impact state $x^*(\tau_1^-)$ to the post-impact state x_1^+.

- Likewise, in Fig 4-1e if the terrain is lower than the nominal one, with $h_{-1} < h_0$, the nominal trajectory should be extended to hit the impact surface $\{(x^*(\tau_{-1}^-), u^*(\tau_{-1}^-))|\ \phi(x^*(\tau_{-1}^-), u^*(\tau_{-1}^-)) = h_{-1}\}$, with the pre-impact phase $\tau_{-1}^- > \tau_0^-$. The post-impact state x_{-1}^+ is mapped by the transition function Δ from the pre-impact state $x^*(\tau_{-1}^-)$.

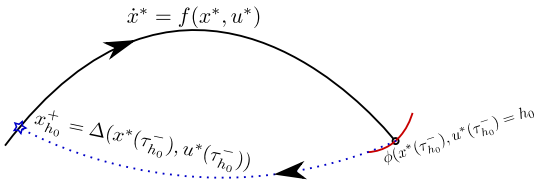
- We wish to use time-varying controller to track the nominal trajectory x^*. In Fig 4-1f, the post impact states x_{-1}^+, x_1^+ will be projected to the nominal trajectory to determine the reset phase τ_{-1}^+, τ_1^+ respectively, and the controller will track the reset phase on the nominal trajectory.



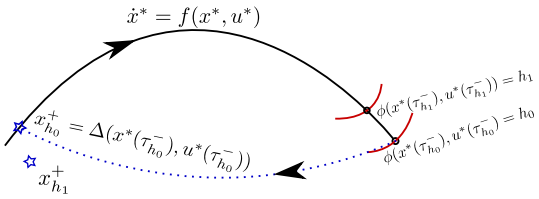
(a) continuous dynamics



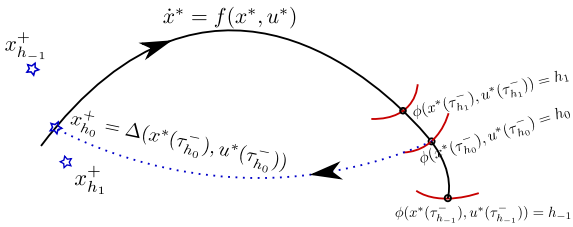
(b) Ground impact for nominal terrain h_0



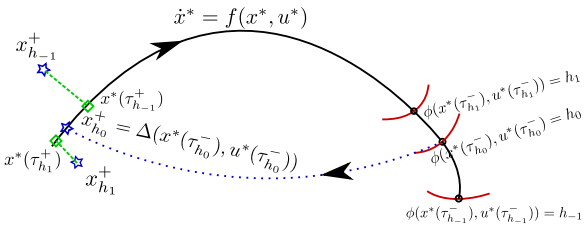
(c) Mode transition for nominal terrain h_0



(d) Ground impact for higher terrain h_1

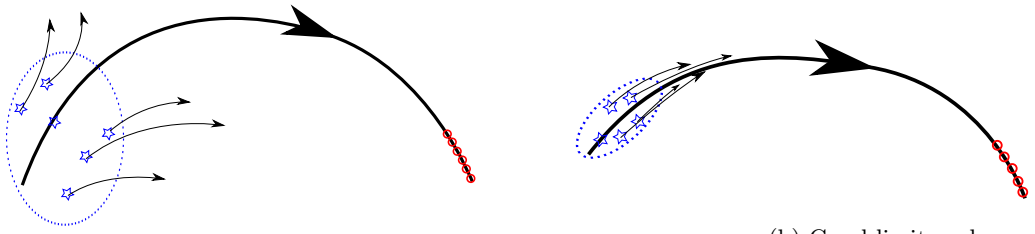


(e) Ground impact for lower terrain h_{-1}



(f) Determine phases to be tracked for the post-impact states

Figure 4-1: State propagation along the nominal trajectory



(a) Bad limit cycle

(b) Good limit cycle

Figure 4-2: Squeeze the region of post-impact states

One of reasons why a robot easily falls down after hitting unexpected terrain is that the post impact states are far away from the nominal trajectory, and leaves little hope for the controller to capture those post-impact states within the narrow regions of attraction, thus the post-impact states will diverge from the nominal trajectory. Intuitively, we want to squeeze the regions of the possible post-impact states, and constrain them to be in the vicinity of the nominal trajectory. Fig 4-2 illustrates this idea. Each post-impact states (blue stars) corresponds to a ground impact at a specific terrain height. When the regions of all possible post impacts (the blue circle) spans a large region away from the nominal cycle (blue ellipsoid), those post-impacts will not track the nominal cycle, and the robot will fall down. On the contrary, when the possible post-impact states are all close to the nominal cycle, they will be stabilized by the tracking controller. The location of the post-impact states are determined by the pre-impact states, and thus by the whole nominal trajectory. The improvement from Fig 4-2a to 4-2b can also be accomplished by elegant controller design, but we decide that the nominal trajectory has a major effect, and we focus on designing such nominal trajectory in this chapter. The controller design will be discussed in the next chapter.

To summarize the intuition, the objective is to design a nominal limit cycle, such that the expected distance from the post-impact states to the nominal trajectory is small. Such nominal limit cycle is intrinsically robust to the terrain disturbances. In the following section, discussion will be drawn on selection of the distance metric.

4.2 Distance metric

To explicitly constrain the post-impact states close to the nominal trajectory, a specific distance metric must be chosen to measure the 'closeness'. Two candidates, the (Weighted) Euclidean distance and the cost-to-go will be discussed.

4.2.1 (Weighted) Euclidean Distance

The (weighted) Euclidean distance between two points x, y is $d(x, y) = \sqrt{(x-y)'Q(x-y)}$. This metric is simple in computation, but perform poorly in the state space for two reasons

1. The (weighted) Euclidean distance presumes that there always exists a straight path between two points x and y, while in the state space, all feasible paths are governed by the system dynamics. Such straight path does not exists between most points in the state space, especially for an underactuated system, which lack the control authority to regulate the path.

2. The shortest path in the state space may not the the desired path when other factors like energy consumption or time become of interest.

A good example of why Euclidean distance is bad in the state space is that when swinging up the inverted pendulum as in Figure 4-3, in which a mass m is attached to the end of a massless rod with length l, and the system is frictionless. Define the state as $x = [\theta\ \omega]^T$. The goal is to swing the inverted pendulum to the upright configuration with state $x_D = [\pi\ 0]^T$ (the light-colored configuration in the middle), by applying a torque u at the base. Both the left configuration **A** and the right one **B** have same angle θ to the vertical position, and they are both swinging at the same angular speed ω counterclockwise. Denote the state of configuration **A** as $x_A = [\theta\ \omega]^T$ and the state of configuration **B** as $x_B = [-\theta\ \omega]^T$. Furthermore, suppose both x_A and x_B have the same kinematic energy as the upright desired configuration state x_D, namely,

$$\frac{1}{2}ml\omega^2 + mgl\cos(\theta) = mgl$$

Although both states x_A and x_B are equidistant in the Euclidean metric to the desired state x_D, these two states are not equivalent. The right configuration **B** is swinging **to** the desired position while the configuration **A** rotates **away** from the upright configuration, indicating that in an ideal metric, x_B should be closer to x_D than x_A is. Moreover, the shortest path in the state space from configuration **B** to upright position requires large torques to drag the pendulum back, while there exists another path, which applies no force and the pendulum swings a full cycle effortlessly to the upright position. Apparently when energy becomes a concern, the latter passive path, instead of the shortest path, becomes more preferable.

Based upon the discussion above, the desired metric should be a good predictor of the long-term performance of the system.

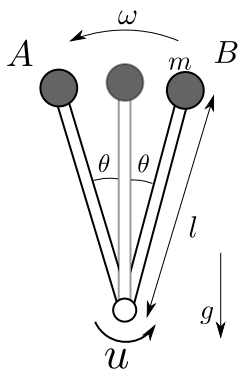


Figure 4-3: Two configurations in swinging up the pendulum

4.2.2 optimal cost-to-go

In the optimal control literature, the cost-to-go is a scalar objective function which scores the long-term performance of running each candidate control policy from each initial condition. When the control goal is to steer the initial condition to the desired state, such objective function measures how much cost a certain controller would incur to regulate such initial condition to the goal state. The optimal cost-to-go is the minimum cost incurred among all possible controllers. This function is ideal to measure the distance in the state-space [6, 16].

Denoting the optimal cost-to-go from a post-impact state x_h^+ to the nominal trajectory as $J(x_h^+)$, our goal is to minimize the expected cost-to-go for all possible x^+. The problem is formulated as

$$\begin{aligned} \min_{x(.),u(.),J(.)} \quad & E(J(x_h^+)) \\ \text{s.t} \quad & \dot{x} = f(x,u) \quad \text{if } \phi(x,u) > h \\ & x_h^+ = \Delta(x^-,u^-) \quad \text{if } \phi(x^-,u^-) = h \\ & x_{h_0}^+ \in \Gamma \\ & J \text{ subject to cost-to-go dynamics} \\ & J \text{ subject to periodicity constraints} \\ & h \text{ drawn from a random distribution} \\ & x \in \mathcal{X} \quad u \in \mathcal{U} \end{aligned} \tag{4.1}$$

Where $\Gamma = \{x^* | x^* \text{ is on the periodic state trajectory on the flat terrain}\}$. $\mathcal{X}$, $\mathcal{U}$ are the admissible set of state and control respectively.

Computing the true optimal cost-to-go $V(x^+)$ analytically is prohibitively hard for a general nonlinear sys-

tem. But an approximation of the true optimal cost still preserves the metric property to a large extent. When the terrain elevation is minor, the post-impact states land not far from the nominal trajectory, thus the linearized dynamics is a good approximation of the nonlinear system in the neighborhood of the nominal trajectory. For the linearized system, the time integral of the quadratic state error and extra control effort, namely, the Linear Quadratic Regulator (LQR) cost is widely used when designing the optimal controller.

LQR cost

For a linear time-varying system with dynamical equation

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{4.2}$$

Where $x \in R^n$ to be state and $u \in R^m$ to be control input. $A \in R^{n\times n}$ $B \in R^{n\times m}$

To design an optimal controller steering the system from the initial state $x(t) = x_0$ to the desired state $x_d = 0$, the cost-to-go is defined as the integral of quadratic cost

$$J^*(x_0, t) = \min_{u(.)} x(t_f)'Q_f x(t_f) + \int_t^{t_f} x(\tau)'Qx(\tau) + u(\tau)'Ru(\tau)d\tau$$

Where $Q \in R^{n\times n}, , Q = Q', Q > 0$ is a given state cost matrix. $R \in R^{m\times m}, R = R', R > 0$ is a given control cost matrix, $Q_f \in R^{n\times n}, Q_f = Q_f', Q_f > 0$ is the final cost matrix.
Suppose the optimal cost is a quadratic function

$$J^*(x_0, t) = x_0' S(t) x_0$$

Where $S \in R^{n\times n}, S = S', S > 0$
Using **Hamilton-Jacobi-Bellman Equation**, the sufficient condition of S is

$$0 = \min_u [x'Qx + u'Ru + 2x'S(Ax + Bu)] \tag{4.3}$$

Due to the convexity of the Equation 4.3 in u, the optimal control law is

$$u^* = -R^{-1}B'S(t)x$$

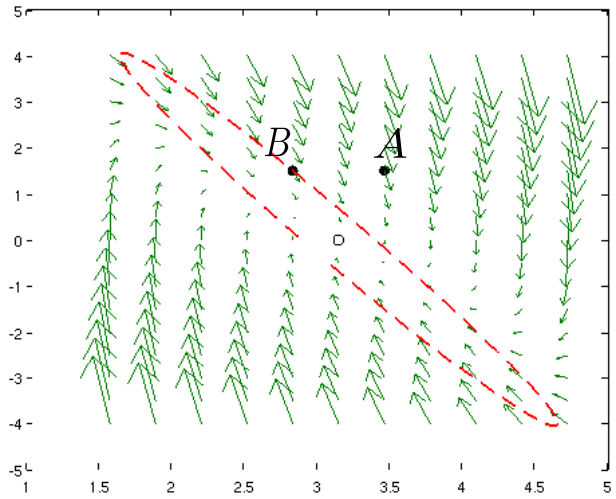


Figure 4-4: The vector flow and cost-to-go level set of pendulum around [π 0]

Where S satisfies the following differential Riccati Equation

$$\dot{S}(t) = -S(t)A(t) - A(t)'S(t) + S(t)B(t)R^{-1}B'(t)S(t) - Q$$

And the terminal condition

$$S(t_f) = Q_f$$

This LQR cost-to-go incorporates the system dynamics, and can be used as a distance metric in the state space. Back to the pendulum swing up example, an LQR controller is constructed at the top position for stabilization, and the vector flows around the state [π 0] is shown in Fig 4-4 as green arrows. The two filled black circles represent the two configurations **A,B** in Fig 4-3. The red ellipse is the contour with the same cost-to-go as configuration **B**. The black circle representing configuration **B**, is outside of the ellipsoid, indicating it has a larger cost-to-go, and thus configuration **B** is more distant away from the desired state than configuration **A** is. This cost-to-go result agrees with the previous discussion on the distance from the two configurations to the desired state.

Periodic Riccati Equation

Theorem 4.2.1. *The Riccati equation satisfying*

$$-\dot{S}(t) = A(t)'S(t) + S(t)A(t) - S(t)B(t)R^{-1}B(t)S(t) + Q$$

is the periodic Riccati Equation (PRE), where $A(t+T) = A(t)\ \forall t,\ B(t+T) = B(t)\ \forall t$ *are matrices with period* T*. There exists a unique symmetric, periodic and positive semidefinite (SPPS) solution* S *if and only if* $(A(.), B(.))$ *is stabilizable. [3]*

The usual way to compute such periodic solution is to integrate the Riccati equation backward with a given final condition until the solution converges [30]. Under the controllability assumptions, all solutions to the PRE with positive semidefinite final condition converge to the unique SPPS solution. For the Riccati solution in the LQR controller design, select Q_f to be any positive semidefinite symmetric matrix, such solution S indicates that $x(t_0)'S(t_0)x(t_0)$ is cost-to-go from initial condition $x(t_0)$ to the infinite horizon.

4.3 LQR cost-to-go for rough terrain walking

For the rough terrain walking, we focus on the linearized dynamics around the nominal trajectory and consider the infinite horizon LQR cost-to-go as the distance from the post-impact state to the nominal trajectory.

To obtain the linearized dynamics, the phase at which to be linearized needs to be defined first

4.3.1 Phase propagation

Denote the phase being tracked on the nominal trajectory as τ. For simplicity, we assume that in the continuous mode, the clock of phase τ is running at the same speed of the world clock, namely

$$\dot{\tau} = 1 \tag{4.4}$$

Every time an impact occurs, the phase right after the impact is reset, by the projection function π, which uniquely determines the reset phase after the impact, based upon the terrain height h, pre-impact phase τ_h^-, the post-impact state $x^+(h)$ and the nominal trajectory $x^*(.)$. The projection is of the form

$$\Pi(x^+(h), x^*(\tau_h^+), h, \tau_h^-) = 0 \tag{4.5}$$

One example of such phase projection function for a bipedal walking robot is to select the post-impact phase such that the nominal post-impact state $x^*(\tau_h^+)$ and the perturbed post-impact state x_h^+ have the same stance leg angle. Note that the phase projection function π might be an implicit function of τ_h^+, and no analytic formulation of the reset post-impact phase is available.

4.3.2 Linearized state dynamics

Define the state error $\bar{x}$ and the control error $\bar{u}$ as

$$\bar{x}(\tau) = x(t) - x^*(\tau)$$
$$\bar{u}(\tau) = u(t) - u^*(\tau)$$

Where τ is the nominal phase being tracked by the perturbed trajectory.
The state error dynamics for the continuous mode is trivial, computed as follows

$$\dot{\bar{x}}(\tau) = A(\tau)\bar{x}(\tau) + B(\tau)\bar{u}(\tau) \quad \tau \neq \tau_c \tag{4.6}$$

Where τ_c is the collision time on the nominal trajectory. A, B are defined as

$$A(\tau) = \left.\frac{\partial f}{\partial x}\right|_{x^*(\tau),u^*(\tau)}, \quad B(\tau) = \left.\frac{\partial f}{\partial u}\right|_{x^*(\tau),u^*(\tau)} \tag{4.7}$$

Suppose that for the simulated trajectory, the ground impact also happens at the nominal impact time τ_c, with proper ground height. So the linearized jump dynamics is

$$\bar{x}^+ = T_x\bar{x}^- + T_u\bar{u} \tag{4.8}$$

Where

$$T_x = \left.\frac{\partial \Delta}{\partial x}\right|_{x^*(\tau_c),u^*(\tau_c)}, \quad T_u = \left.\frac{\partial \Delta}{\partial u}\right|_{x^*(\tau_c),u^*(\tau_c)} \tag{4.9}$$

The dynamics of the cost matrix S is also a hybrid system

$$-\dot{S}(\tau) = A(\tau)'S(\tau) + S(\tau)A(\tau) - S(\tau)B(\tau)R^{-1}B(\tau)'S(\tau) + Q \quad \text{if } \tau \neq \tau_c \tag{4.10}$$
$$S(\tau_c^-) = T_x'S(\tau_c^+)T_x \tag{4.11}$$

Equation 4.11 is the jump Riccati equation at the nominal impact. It is easily verified that the dynamics of S is periodic, and thus there exists a periodic solution to the Riccati equation 4.10 and 4.11. The LQR cost $x'Sx$ represents the cost that the LQR controller would incur to steer the state x back to the nominal trajectory in the infinite horizon, which is the analytical representation of the distance metric used in the rough terrain case.

4.3.3 Optimizing expected cost-to-go

With the discussion above, we linearize the dynamics around the nominal trajectory and compute the LQR cost. The problem 4.1 is reformulated as

$$\min_{\substack{x^*(.),u^*(.),S^*(.),x_h^+ \\ \tau_h^-,\tau_{h_0}^-, \\ \tau_h^+,\tau_{h_0}^+}} \quad E[(x_h^+ - x^*(\tau_h^+))' S(\tau_h^+)(x_h^+ - x^*(\tau_h^+))] \tag{4.12}$$

$$\begin{aligned}
\text{s.t} \quad & \dot{x}^* = f(x^*, u^*) \quad \forall \tau < \tau_h^- \\
& \phi(x^*(\tau), u^*(\tau)) > h \quad \forall \tau < \tau_h^- \\
& x_h^+ = \Delta(x^*(\tau_h^-), u^*(\tau_h^-)) \\
& \phi(x^*(\tau_h^-), u^*(\tau_h^-)) = h \\
& -\dot{S}(\tau) = A(\tau)'S(\tau) + S(\tau)A(\tau) - S(\tau)B(\tau)R^{-1}B(\tau)'S(\tau) + Q \quad \forall \tau < \tau_{h_0}^- \\
& S(\tau_{h_0}^+) = T_x' S(\tau_{h_0}^-) T_x \\
& \pi(x_h^+, x^*(\tau_h^+), h, \tau_h^-) = 0 \\
& \pi(x_{h_0}^+, x^*(\tau_{h_0}^+), h_0, \tau_{h_0}^-) = 0 \\
& x_{h_0}^+ = x^*(\tau_{h_0}^+) \\
& h \quad \sim \quad \text{a given distribution} \\
& x \in \mathcal{X} \quad u \in \mathcal{U}
\end{aligned}$$

Where h_0 is the nominal terrain, for which a periodic state and cost matrix trajectory exist.

Notice that both state x^*, control u^* are decision variables in the optimization problem 4.12, so we are parameterizing the problem using direct method. Also the impact time are decision variables, so instead of performing zero-crossing to detect the impact time, the guard function ϕ is treated as a constraint also. To find the periodic solution to the Riccati Equation, the traditional approach, which is to integrate the Riccati equation backward until it converges, is not viable in this case. This is because the periodic state and control trajectories are not obtained prior to the optimization ends, thus during the optimization process, the backward integration approach is not applicable. To this end, the cost-to-go matrix is parameterized as a decision variable also, and the Riccati equation is the dynamical constraint on this decision variable. The periodic constraint is enforced by equalizing the values at the two sides of the jump Riccati equation. Thus the state/control and cost-to-go matrix are solved simultaneously in the optimization program.

4.3.4 Sampling and discretization

The problem 4.12 is not viable for numerically computation. First, the expected value of the cost-to-go cannot be computed analytically for a continuous distribution. Second, the continuous differential dynamical equation cannot be dealt with the optimization solver, which only allows finite decision variables. The remedy is to sample the terrain for computing expected value, and to discretize the state/control and cost-to-go matrices and numerically approximate the time derivative.

For the terrain sampling, $K+1$ terrain values of $[h_0 \ h_1 \ \dots \ h_K]$ are taken with impulsive probability $Pr(h = h_i) = p_i$. Notice that the nominal terrain height h_0 is always sampled, for its specialty related to the nominal trajectory on the flat terrain. The nominal trajectory includes the earliest possible taking off time at the highest sampled terrain, and the latest possible ground impact time with the lowest sampled terrain. Namely if the terrain samples are sorted as $h_i < h_{i+1}$ $i = 1, \dots, N$, then the nominal trajectory interval $[\tau_s \ \tau_e]$ satisfies $\tau_s \leq \tau_{h_K}^+$ and $\tau_e \geq \tau_{h_1}^-$, as illustrated in Fig 4-1f.

To discretize the state/control and cost-to-go matrices, the nominal trajectory is split into N small intervals, with the interval duration between knot i to knot $i+1$ to be dt_i. Among the $N+1$ knot points, there are $2(K+1)$ of them that are special, where each special knot corresponds to a ground impact at a sampled terrain height. The knot points $p_n, n = 0 \dots K$ are assigned to be the projected knot of $x_{h_n}^+$, the post-impact state at terrain height h_n. The knot point $c_n, n = 0 \dots K$ are assigned to be the pre-impact knot at terrain height h_n. The state, control and cost-to-go matrix trajectories are discretized at those knot points as x_i, u_i and S_i respectively. To rebuild the continuous trajectory from discretized knot point, the state and cost-to-go trajectory are approximated by cubic Hermite interpolation of x_i and S_i respectively, and control profile is linearly interpolated with u_i. The same interpolation approach is used in the aerospace applications [17].

Cubic interpolation of state

Suppose the state x on time interval $[0, 1]$ is a cubic polynomial of time $s \in [0, 1]$

$$x = c_0 + c_1 s + c_2 s^2 + c_3 s^3 \tag{4.13}$$

Where c_i are constants. The values of c_i can be determined by the values of the state and its derivatives at the tailing points. Suppose $x(0) = x_0, x(1) = x_1, \frac{dx}{ds}(0) = \dot{x}_0, \frac{dx}{ds}(1) = \dot{x}_1$, then

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ x_1 \\ \dot{x}_1 \end{bmatrix}$$

The matrix on the left is nonsingular, thus c_i can be computed from $x_0, \dot{x}_0, x_1, \dot{x}_1$. Substitute c_i with $x_0, \dot{x}_0, x_1, \dot{x}_1$, and map the time interval to $[0, T]$, the value of x at the middle of the interval is

$$x^c = x(T/2) = (x_0 + x_1)/2 + T(\dot{x}_0 - \dot{x}_1)/8$$

Based upon the cubic polynomial assumption, the slope at the middle point is

$$x'^c = -3(x_0 - x_1)/(2T) - (\dot{x}_0 + \dot{x}_1)/4$$

The slope at the middle point evaluated from the dynamics equation is $f^c = f(x^c, u^c)$, where u^c is interpolated by the control at the two tailing points, $u^c = (u(0) + u(1))/2$. The dynamics constraint $\dot{x} = f(x, u)$ is then transformed to the constraint that the slope at the middle point f^c computed from dynamics function should match x'^c, the slope computed from the cubic interpolation.

$$f^c - x'^c = 0$$

Apply this interpolation to every small segment with interval from knot i to knot $i + 1$, the continuous differential dynamics constraint $\dot{x} = f(x, u)$ is formulated as an algebraic constraint on $x_i, x_{i+1}, u_i, u_{i+1}, dt_i$. The same approach can be applied to the differential Riccati equation.

After sampling and discretization, the optimization formulation to be handled by the optimizer is

$$\min_{\substack{x_i,u_i,S_i,\\ x^+_{h_n},dt_i}} \quad \sum_n pr(h_n)[(x^+_{h_n} - x_{p_n})' S_{p_n} (x^+_{h_n} - x_{p_n})] \tag{4.14}$$

$$\text{s.t} \quad x^c_i = (x_i + x_{i+1})/2 + dt_i(f(x_i, u_i) - f(x_{i+1}, u_{i+1}))/8 \tag{4.15}$$

$$u^c_i = (u_i + u_{i+1})/2 \tag{4.16}$$

$$f(x^c_i, u^c_i) = -3(x_i - xi + 1)/(2dt_i) - (f(x_i, u_i) + f(x_{i+1}, ui + 1)/4 \tag{4.17}$$

$$\dot{S}_i = -(A'_i S_i + S_i A_i - S_i B_i R^{-1} B'_i S_i + Q) \tag{4.18}$$

$$S^c_i = (S_i + S_{i+1})/2 + dt_i(\dot{S}_i - \dot{S}_{i+1})/8 \tag{4.19}$$

$$-(A'^c_i S^c_i + S^c_i A^c_i - S^c_i B^c_i R^{-1} B'^c_i S^c_i + Q) = -3(S_i - S_{i+1})/(2dt_i) - (\dot{S}_i + \dot{S}_{i+1})/4 \tag{4.20}$$

$$h_n = \phi(x_{c_n}, u_{c_n}) \tag{4.21}$$

$$x^+_{h_n} = \Delta(x_{c_n}, u_{c_n}) \tag{4.22}$$

$$\tau^-_{h_n} = \sum_{j=0}^{c_n} dt_j \tag{4.23}$$

$$0 = \Pi(x^+_{h_n}, x_{p_n}, h_n, \tau^-_{h_n}) \tag{4.24}$$

$$x_{p_0} = x^+_{h_0} \tag{4.25}$$

$$S_{p_0} = T'_{c_0} S_{c_0} T_{c_0} \tag{4.26}$$

$$x_i \in \mathcal{X}, \quad u_i \in \mathcal{U} \tag{4.27}$$

$$i = 1 \dots N-1, \quad n = 0 \dots K \tag{4.28}$$

Equation 4.15 and 4.16 interpolate the state and control at the middle of each small segment from knot i to knot $i+1$. Equation 4.17 imposes the constraints that the state dynamics at the middle of each interval matches that computed from cubic interpolation. Equation 4.18 is the dynamics of the cost-to-go matrix at each knot point, where A_i, B_i are the linearized dynamics matrices at those knots, $A_i = \frac{\partial f}{\partial x}\Big|_{x_i,u_i}, B_i = \frac{\partial f}{\partial u}\Big|_{x_i,u_i}$. Equation 4.19 interpolates the cost-to-go matrix at the middle of each small segment, and Equation 4.20 constrains that the cost-to-go dynamics at the middle of each segment matches that computed from cubic interpolation, where $A^c_i = \frac{\partial f}{\partial x}\Big|_{x^c_i,u^c_i}, B^c_i = \frac{\partial f}{\partial u}\Big|_{x^c_i,u^c_i}$. Equation 4.21 constrains that at knot c_n, the robot foot touches ground with height h_n. Equation 4.22 is the state transition at the ground height h_n. Equation 4.23 and 4.24 project the post-impact state $x^+_{h_n}$ to x_{p_n} on the nominal trajectory, where $\tau^-_{h_n}$ is the pre-impact phase at terrain h_n. Equation 4.25 and 4.26 are the periodic constraints on state x and cost-to-go matrix S respectively, both periodic orbit exist for the nominal terrain h_0.

To improve the computation speed, the gradient of each constraint is computed analytically. Note that to compute the gradient of equation 4.19, computing the second order differentiation of dynamics f is necessary.

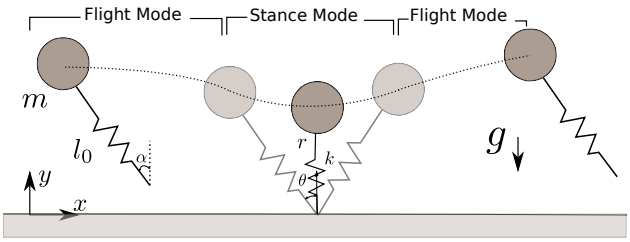


Figure 4-5: Spring Loaded Inverted Pendulum running from apex to apex

The Jacobian matrix of the constraint optimization problem is sparse, since the dynamics at each interpolated middle point only depends on states, controls or cost-to-go matrices in the neighboring two knots. SNOPT is a powerful solver for such constraint nonlinear optimization with sparse Jacobian matrix [15].

4.4 Results

4.4.1 Spring Loaded Inverted Pendulum

The Spring Loaded Inverted Pendulum (SLIP) model [4], as shown in Fig 4-5, consists of a point mass attached to a massless spring. There are two modes for a SLIP model running on the ground, the flight mode in which the system is solely governed by the gravity, when the foot touches the ground, the system enters the stance mode, in which the gravity force, the normal force from the ground and the spring elastic force act together on the system. The state in the flight mode is $[x\ y\ \dot{x}\ \dot{y}]$, and the state in the stance mode is $[r\ \theta\ \dot{r}\ \dot{\theta}]$, It is widely known that appropriate leg angle of attack and/or spring stiffness can achieve a perfect apex-to-apex return map on the flat terrain. On unknown terrain, by exhaustive simulation, Ernst discovers the controller that encodes a time law of swing leg angle and the spring stiffness in the flight mode, so as to achieve perfect return map [12]. Such control law can perfectly reject the terrain disturbances. We show that using our general optimization scheme, we can recover such model-specific controller.

Suppose the swing leg angle is freely actuated during the flight mode, and no actuation during the stance mode. The phase projection function is chosen to select the nominal state which has the same vertical position as the post-impact state, namely $\Pi(x_h^+, x^*(\tau_h^+), h, \tau_h^-) = y^+ - y^*(\tau_h^+) = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} (x_h^+ - x^*(\tau_h^+))$. It is worth noting that the mode transition function Δ that maps the pre-impact (pre-landing) state to the post-impact (post-taking-off) state is not impulsive, but rather governed by the dynamics in the stance mode. To numerically compute such mode transition function and its derivative, the Real Time Recurrent Learning (RTRL) method is adopted [42]. Denote the mapping from pre-landing to post-landing as Δ_1, the mapping from pre-taking-off to post-taking-off as Δ_2, continuous dynamics in the stance mode as f_2, the

pre-landing state as x_l^-, pre-landing angle of attack as u^-, and define $\alpha = \begin{bmatrix} x_l^- \\ u^- \end{bmatrix}$. Also define the matrix P as the Jacobian matrix of the state

$$P(t) = \frac{\partial x(t)}{\partial \alpha}$$

During the stance mode, the dynamics of such Jacobian matrix is

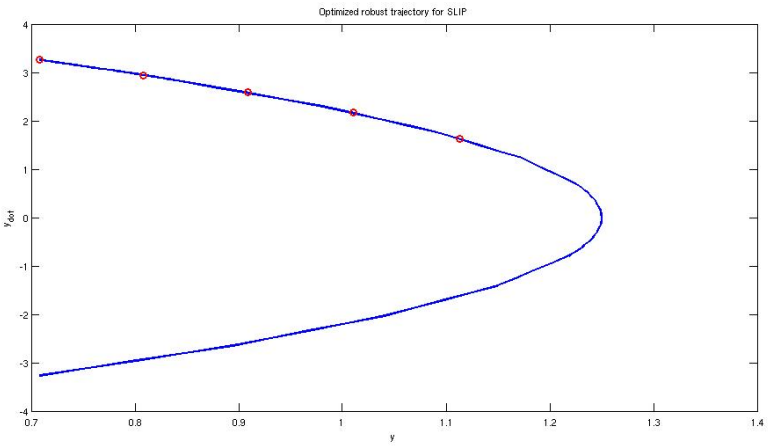
$$\dot{P}(t) = \frac{\partial f_2}{\partial x} P \tag{4.29}$$

By numerically integrating this P matrix, and multiplying the Jacobians of the functions Δ_1 and Δ_2, the Jacobian of Δ is computed. Using the same idea, the second order derivative of Δ can be computed similarly. Special attention should be paid that the taking off time is an implicit function of α, since a change in the pre-impact state will change the time the take-off guard being triggered. So the implicit function theorem is used when computing the gradients at the time of taking off the ground.

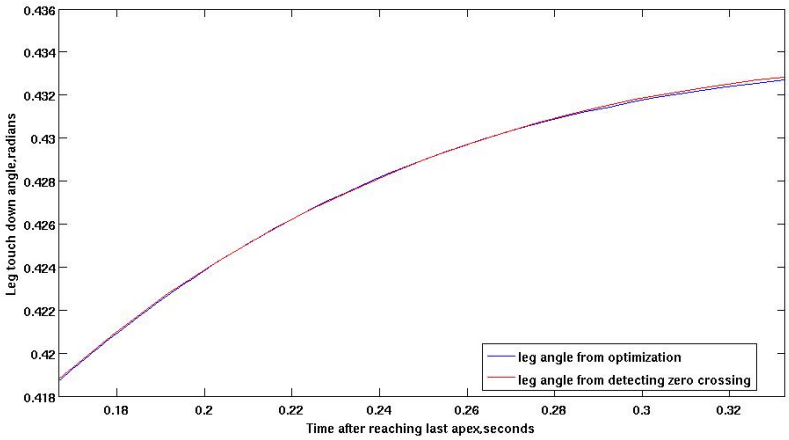
Fig 4-6a shows the optimized state trajectory in the flight mode. The red circles are the post-taking-off states. As can be seen on the plot, the post-taking-off states land exactly on the nominal state trajectory, indicating the terrain disturbances are perfectly rejected such that the post-taking-off state can follow the nominal trajectory exactly whilst the existence of the unknown terrain elevation. Fig 4-6b compares the optimized control input (the swing leg angle) and the control profile computed from exhaustive simulation. The good match between those two suggest Ernst model-specific approach can be integrated into our general optimization scheme.

4.4.2 Compass Gait Walker

The configuration of a compass gait walker is shown in Fig 4-6. It has one actuator at the hip, and two degrees of freedom. Such an underactuated robot can maintain a passive limit cycle when walking down a shallow slope due to the gravitational force only. Although self-stable, such passive limit cycle is extremely fragile under slight terrain disturbances, even with a stabilizing controller Fig 4-8a. Instead of designing a more robust controller, our first solution is to search for a more robust limit cycle. The phase projection function Π is chosen such that the projected state on the nominal trajectory has the same stance leg angle as the post-impact state, namely $\Pi(x_h^+, x^*(\tau_h^+), h, \tau_h^-) = \theta_{st}^+ - \theta_{st}^*(\tau^+) = 0$. Further constraint can be imposed to make this projection to be injective, like the constraint that the stance leg angle around the nominal post-impact time is monotonic. The terrain disturbance is the angle of the virtual slope from step to step. Equivalently, the incoming terrain slope changes after every ground impact. As is widely known, the compass gait walker needs to penetrate into the virtual slope so as to maintain a limit cycle, so ground clearance is



(a) Optimized state trajectory for SLIP in the flight mode



(b) Leg angles computed from optimization and exhaustive simulation

ignored in the simulation of compass gait.

In the optimization problem settings, the terrain slope is drawn from a uniform distribution within [2° 8°], with the nominal terrain angle equals to 5°. Five sampled terrain $\gamma = [2° \ 3.5° \ 5° \ 6.5° \ 8°]$ are considered. The whole trajectory is discretized into 18 intervals. The LQR cost matrix is taken as $Q = \text{diag}[1 \ 1 \ 10 \ 10]$ and $R = 1$. The initialization of the state trajectory is the passive limit cycle, the initialization of the cost-to-go matrix is the LQR periodic solution on the passive limit cycle. After optimization, the expected cost-to-go drops from 45.1462 to 0.7859. Due to the size of the problem, it takes around 10 minutes for SNOPT to find the local minimum.

The phase plots of the optimized robust limit cycle and the passive limit cycle for the compass gait are depicted in Fig 4-7. For each cycle, both phase plots for the stance leg (the lower curve) and the swing leg

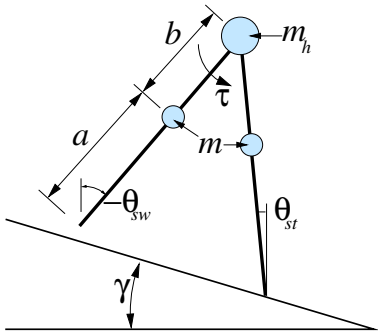


Figure 4-6: A compass gait walker on a slope

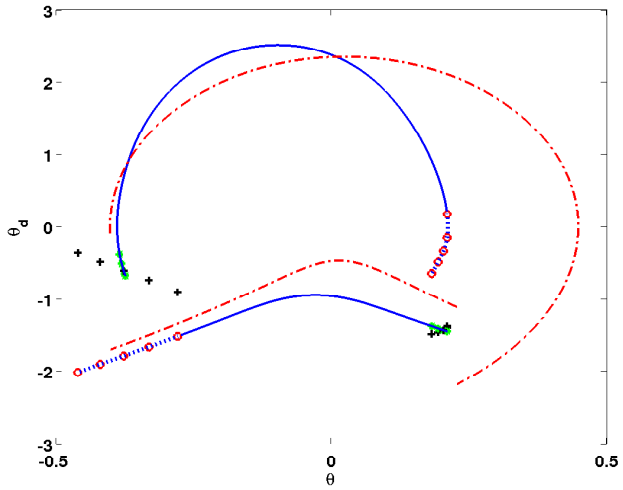
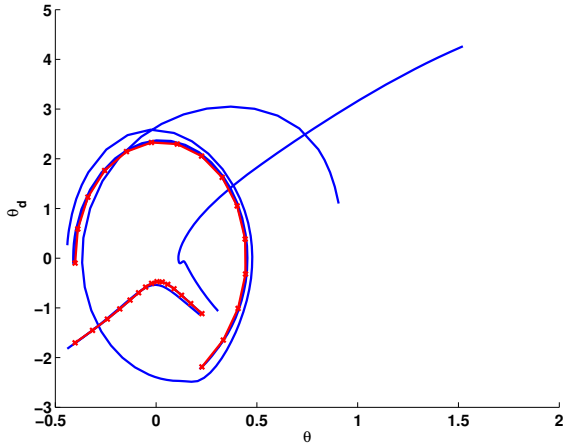


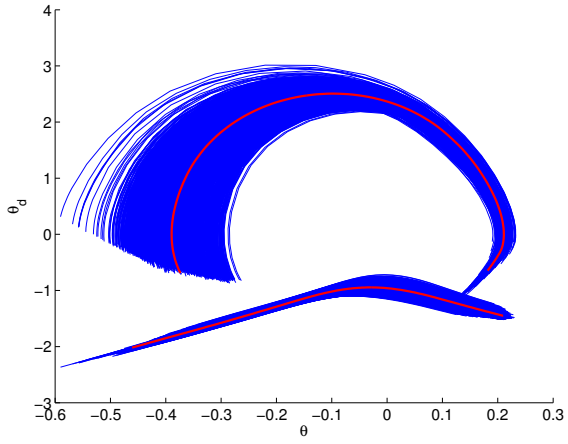
Figure 4-7: The robust and passive limit cycle

(the upper curve) are drawn. The red curve is for the passive cycle, while the blue one is for the robust cycle. The red circles on the robust cycle are the pre-impact states, the black plus signs are the post-impact states, the green asterisks are the projected states on the nominal trajectory from the post-impact states. Note that one of the black plus sign and a green asterisk exactly overlay, indicating the periodic solution on the nominal terrain.

The compass gait walker is simulated on the rugged terrain as in the problem settings. An LQR controller using the same Q and R as in the optimization is constructed to stabilize the trajectory. In most simulation cases, by following the passive cycle, the robot falls down within 10 steps. On the contrary, by following the optimized robust cycle, the robot walks for 10,000 steps and still does not fall down. The phase plots tracking both passive limit cycle and the robust cycle are drawn in Fig 4-8a and 4-8b. In both plots, the red curves are the nominal cycle with which the compass gait struggles to track. In the left plot, the



(a) Phase plot of simulating the passive limit cycle on a rugged terrain



(b) Phase plot of simulating the robust limit cycle on the same terrain

passive limit cycle case, the cure diverges after its third collisions, indicating the robot falls down in three steps. The right plot is the simulation of 10,000 steps by following the robust cycle. It is also interesting to notice that in the robust cycle plot, the trajectories converge to the nominal cycle, as the volume of the trajectory bundle shrinks, indicating the asymptotically stability.

The big decrease of the expected cost-to-go (from 45.1462 to 0.7859) and the huge improvement of the number of steps on the rough terrain indicates the strong correlation between the expected cost-to-go and the ability to traverse the rough terrain, thus when the terrain distribution is known prior, the expected cost-to-go is a good measure of the ability for a bipedal robot to reject unknown terrain disturbances, and our trajectory optimization scheme can drastically improve the robustness of the bipedal robot.

Chapter 5

Robust Control Synthesis

As the robust walking gait can be designed using the trajectory optimization scheme in Chapter 4, the next goal is to construct a controller so as to regulate the robot tracking the designated walking gait on the fly. The walking robot is again modeled as a hybrid system with only one continuous mode and one guard

$$\dot{x} = f(x, u) \quad \text{if} \quad \phi(x) > h \tag{5.1}$$

$$x^+ = \Delta(x^-) \quad \text{if} \quad \phi(x^-) = h \tag{5.2}$$

Notice that the guard function ϕ depends only on the state x now. This assumption is valid for most robots since the state x contains all the robot configuration information, so the height of the foot can be solely determined by the state x. The transition function also depends on pre-impact state x^- only, meaning the uncontrolled mode switch is studied in this case.

5.1 Background of robust control

5.1.1 L_2 gain

As defined in [26, 48], for a continuous time state space system $\mathcal{S}$

$$\dot{x} = Ax + Bw$$

$$e = Cx + Dw$$

Where $x \in \mathbb{R}^m, w \in \mathbb{R}^l, e \in \mathbb{R}^s$ are the state, the noise, and the error output respectively. The L_2 gain for such a system $\mathcal{S}$ is defined as the maximal lower bound for $\gamma \geq 0$ such that

$$\inf_{T \geq 0} \int_0^T \{\gamma^2 |w(t)|^2 - |e(t)|^2\} dt > -\infty \quad \forall\, v = [w; e] \in \mathcal{L}^2 \tag{5.3}$$

Where $\mathcal{L}^2$ is the locally square integrable signal space.

5.1.2 Dissipation inequality

Define the supply rate function σ as

$$\sigma(e, w) = \gamma^2 |e|^2 - |w|^2$$

The dissipation inequality associated with a continuous time state-space model $\mathcal{S}$ and a supply rate σ is the condition

$$\frac{\partial V}{\partial x}(Ax + Bw) \leq \sigma(x, w) \quad \forall\, x, u$$

Where $V : \mathbb{R}^n \to \mathbb{R}$ (the associated storage function) is continuously differentiable, and satisfies $V(x) \geq 0 \; \forall\, x$.

Then the dissipation inequality establishes the following condition

$$\int_0^T (\gamma^2 |e|^2 - |w|^2) dt \geq \int_0^T \dot{V}(x) dt = V(x(T)) - V(x(0)) \geq -V(x(0)) > -\infty \tag{5.4}$$

Thus the dissipation inequality becomes a sufficient condition that the L_2 gain of the linear system $\mathcal{S}$ is no larger than γ .

5.1.3 KYP lemma for linear system

Equation 5.4 is satisfied for some continuous V and linear function $f(x, w) = Ax + Bw$ if and only if they are satisfied for some quadratic storage function $V(x) = x'Sx, S = S', S \succeq 0$ [26]. i.e, to bound the L_2 gain of a linear system $\mathcal{S}$, rather than searching over the storage function V of all forms, we can restrict the storage function to a quadratic form, and only search over a storage matrix S satisfying the dissipation inequality.

5.2 L_2 gain for periodic linearized system

For a simulated trajectory with infinite time horizon, the input disturbance is the terrain perturbation. Define the n^{th} terrain elevation as $\bar{h}[n] = h[n] - h_0$, also define the output error function as

$$e(t) = \begin{bmatrix} \sqrt{Q}\bar{x}(t) \\ \sqrt{R}\bar{u}(t) \end{bmatrix} \tag{5.5}$$

Where $Q \in \mathbb{R}^{n\times n}, Q = Q', Q \succeq 0, R \in \mathbb{R}^{m\times m}, R = R', R \succeq 0$ are given matrix, $\bar{x}(t) = x(t) - x^*(\tau), \bar{u}(t) = u(t) - u^*(\tau)$ are the state and control error signals respectively, τ is the phase on the nominal trajectory to be tracked by the perturbed trajectory. The L_2 gain of the input/output system $(\bar{h}, e)$ is no larger than a constant γ if and only if the following condition holds

$$\gamma^2 \sum_{n=1}^{\infty} (\bar{h}[n])^2 - \int_0^{\infty} (|\bar{x}(t)|_Q^2 + |\bar{u}(t)|_R^2) dt > -\infty \tag{5.6}$$

Where $|y|_M$ is short for $y'My$.
Equation 5.6 is the condition to check if the system has L_2 gain no larger than γ. But it is hard to handle, for two reasons:

1. The time horizon stretches to infinity, and the notion of convergence in the infinite horizon is vague in our case. The effect of the previous terrain disturbance get accumulated and are prohibitively difficult to compute explicitly, as the disturbance exists in the guard function instead of in the continuous mode as the traditional manner.

2. The error output signal e is a continuous time signal, while the terrain disturbance $\bar{h}$ is discrete.

In the following sections, these two problems will be addressed by focusing on one period and linearizing the dynamics. The analysis of the periodic system starts with defining the nominal phase within one cycle.

5.2.1 Periodic phase tracking system

To define the error signal e as in 5.5, the phase being tracked system needs to be defined. By leveraging the periodic property of a walking pattern on the flat terrain, our analysis focuses on a single period. We define one period as starting from the phase of one certain event (for example, the robot on the nominal trajectory reaches its apex), and the phase propagating according to phase dynamics

$$\dot{\tau} = 1 \quad \text{if } \phi(x^*(\tau) + \bar{x}(t)) > h \tag{5.7}$$

$$\Pi(x_h^+, x^*(\tau_h^+), h, \tau_h^-) = 0 \quad \text{if } \phi(x^*(\tau_h^-) + \bar{x}(t)) = h \tag{5.8}$$

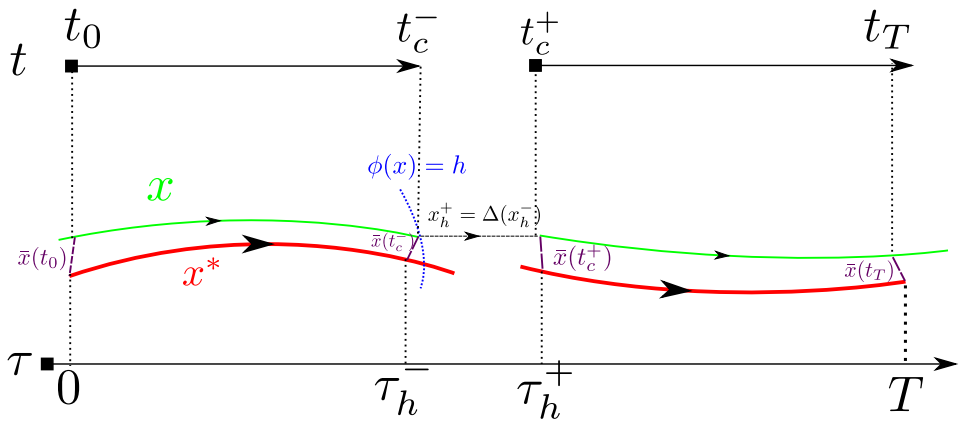


Figure 5-1: The nominal and simulated trajectories in one period

until again τ reaches the same event T, where T is the period of the nominal trajectory on the flat terrain. 5.7 means when the state $x = x^*(\tau) + \bar{x}(t)$ does not trigger the ground impact guard, the tracked phase runs at the same clock speed as the world clock; while the ground impact happens, the phase projection function Π in 5.8 resets the tracked phase after the impact, based upon the post-impact state, the nominal trajectory, the terrain height and the pre-impact phase. After the impact, the phase continues to propagate at $\dot{\tau} = 1$ until the phase reaches $\tau = T$. After one period, the phase is reset back to 0. The whole period is illustrated in Fig 5-1. The simulated trajectory (the green curve x) and the nominal trajectory (the red curve x^*) both propagate according to the hybrid system equations 5.1 and 5.2. When the green curve hits the blue surface $\phi(x) = h$, the impact transition function Δ maps the pre-impact state x_h^- to the post-impact state x_h^+. The purple dashed lines, representing the state error between the simulated trajectory x and the nominal trajectory x^*, are defined as

$$\bar{x}(t) = x(t) - x^*(\tau) \tag{5.9}$$

Similarly, the extra control effort $\bar{u}$ is defined as

$$\bar{u}(t) = u(t) - u^*(\tau) \tag{5.10}$$

Note that since $\dot{\tau} = 1$ in the continuous phase, the following relations hold

$$\tau_h^- = t_c - t_0$$
$$T - \tau_h^+ = t_T - t_c$$

When the simulated trajectory $x(t)$ hits the impact surface, the nominal trajectory $x^*(\tau)$ does not necessarily trigger the same guard function at the same moment, as can be seen from the plot; when the tracked phase $\tau = T$, the simulated trajectory may not trigger the same event (for example, reach the apex). And the cycle is defined as the **tracked phase** runs over one cycle, not the simulated trajectory. Thus, the dynamics of the error state $\bar{x}$ is governed by

$$\dot{\bar{x}}(t) = f(x^*(\tau) + \bar{x}(t), u^*(\tau) + \bar{u}(t)) - f(x^*(\tau), u^*(\tau)) \quad \text{if} \quad \phi(x^*(\tau) + \bar{x}(t)) > h \tag{5.11}$$

$$\bar{x}(t_c^+) = \Delta(x^*(\tau_h^-) + \bar{x}(t_c^-)) - x^*(\tau_h^+) \quad \text{if} \quad \phi(x^*(\tau_h^-) + \bar{x}(t_c^-)) = h \tag{5.12}$$

Equations 5.7, 5.8, 5.11 and 5.12 together governed the dynamics of $\bar{x}$ within one cycle. Each cycle starts with the nominal phase $\tau = 0$, and ends when the nominal phase $\tau = T$.

5.2.2 Periodic storage function

With each period defined in the previous section, the infinite time horizon can be split into individual periods, and each period is analyzed separately. Suppose period n starts at time $t_a[n]$ and ends at time $t_a[n+1]$, and within period n, the robot leg hits the ground at terrain height $h[n]$, then the L_2 gain defined in the infinite horizon (equation 5.6) can be reformulated as

$$\sum_{n=1}^{\infty} \left[\gamma^2 (\bar{h}[n])^2 - \int_{t_a[n]}^{t_a[n+1]} |\bar{x}(t)|_Q^2 + |\bar{u}(t)|_R^2 dt \right] > -\infty \tag{5.13}$$

Lemma 5.2.1. *A sufficient condition for 5.13 and 5.6 is that there exists a scalar storage function $V(\tau, \bar{x})$, such that*

$$\gamma^2 (\bar{h}[n])^2 - \int_{t_a[n]}^{t_a[n+1]} |\bar{x}(t)|_Q^2 + |\bar{u}(t)|_R^2 dt \geq V(T, \bar{x}(t_a[n+1])) - V(0, \bar{x}(t_a[n])) \quad \forall \bar{x}(t_a[n]) \tag{5.14}$$

and

$$V(T, x) \geq V(0, x) \quad \forall\, x \tag{5.15}$$

and the nonnegative constraint

$$V(T, x) \geq 0 \quad \forall\, x \tag{5.16}$$

Proof.

$$
\begin{aligned}
&\sum_{n=1}^{\infty}\left[\gamma^2(\bar{h}[n])^2-\int_{t_a[n]}^{t_a[n+1]}|\bar{x}(t)|_Q^2+|\bar{u}(t)|_R^2dt\right]\\
\geq&\sum_{n=1}^{\infty}\left[V(T,\bar{x}(t_a[n+1]))-V(0,\bar{x}(t_a[n]))\right]\\
=&\sum_{n=2}^{\infty}\left[V(T,\bar{x}(t_a[n]))-V(0,\bar{x}(t_a[n]))\right]+\lim_{m\to\infty}V(T,\bar{x}(t_a[m]))-V(0,\bar{x}(t_a[1]))\\
\geq&-V(0,\bar{x}(t_a[1]))\\
>&-\infty
\end{aligned}
$$

□

Within one period $\tau \in [0\ T]$, the system dynamics involves both continuous mode and the impact transition. As in the left hand-side of the equation 5.14, the time integral of the error state and discrete terrain disturbance are added together. To separate the discrete and continuous part, a sufficient condition for 5.14 is imposed as

$$\dot{V}(\tau,\bar{x}) \leq -|\bar{x}|_Q^2-|\bar{u}|_R^2 \quad \text{if} \quad \phi(x^*(\tau)+\bar{x})>h \tag{5.17}$$

$$V(\tau_h^+,\bar{x}(t_c^+))-V(\tau_h^-,\bar{x}(t_c^-)) \leq \gamma^2h^2 \quad \text{if} \quad \phi(x^*(\tau_h^-)+\bar{x}(t_c^-))=h \tag{5.18}$$

Where 5.17 is for the continuous mode, $\dot{V}$ is the total differentiation of V over t; and 5.18 is for the jump transition.

The state error $\bar{x}$ is parameterized by the world time t, since the nominal phase τ is a function of the world time t, and we are analyzing the behavior of state error within one cycle, it is more convenient to parameterize the state error by τ, in the same manner as equation 5.9

$$\bar{x}(\tau)=x(t)-x^*(\tau) \tag{5.19}$$

$$\bar{u}(\tau)=u(t)-u^*(\tau) \tag{5.20}$$

And condition 5.17 and 5.18 are reformulated as

$$\frac{dV(\tau,\bar{x})}{d\tau} \leq -|\bar{x}|_Q^2-|\bar{u}|_R^2 \quad \text{if} \quad \phi(x^*(\tau)+\bar{x})>h \tag{5.21}$$

$$V(\tau_h^+,\bar{x}(\tau_h^+))-V(\tau_h^-,\bar{x}(\tau_h^-)) \leq \gamma^2h^2 \quad \text{if} \quad \phi(x^*(\tau_h^-)+\bar{x}(\tau_h^-))=h \tag{5.22}$$

Conditions 5.21 and 5.22 are a conservative formulation for the L_2 gain definition 5.6 since first the analysis

focuses within one period, as in Lemma 5.2.1, and the long term behavior is not analyzed; second, within one cycle, by separating the continuous and discrete dynamics, the condition becomes more conservative, but such condition can be easily dealt by the existing tools in linear control theory.

5.2.3 Linearized system

To leverage the rich control tools for linear system, the dynamics of $\bar{x}$ is linearized around the nominal trajectory x^*, u^* and the terrain height h_0. For the continuous mode, the linearization is simple

$$\dot{\bar{x}}(\tau) = A(\tau)\bar{x}(\tau) + B(\tau)\bar{u}(\tau) \quad \text{if } \phi(x^*(\tau) + \bar{x}(\tau)) > h \tag{5.23}$$

Where $A = \frac{\partial f}{\partial x}, B = \frac{\partial f}{\partial u}$. (A, B) is controllable.
It is tricky for the ground impact transition. One caveat is that the pre-impact phase τ_h^- is determined by the guard function with the pre-impact state and terrain height. Since we separate the continuous dynamics and the jump transition, we write $\bar{x}(\tau_h^-) = \bar{x}_h^-$ and treat it as a free variable, namely, the value of $\bar{x}_h^-$ can be freely chosen and independent of the continuous dynamics. Similarly we denote $\bar{x}(\tau_h^+) = \bar{x}_h^+$. $\bar{x}_h^-$ and $\bar{x}_h^+$ are solely used in the jump transition analysis. Suppose the nominal trajectory ($\bar{x} = 0$) hits the mode transition surface with nominal terrain h_0 at τ_0^-, and linearize the guard function around τ_0^-

$$\phi_{x^*} f(x^*(\tau_0^-), u^*(\tau_0^-))d\tau_h^- + \phi_{x^*}\bar{x}_h^- = \bar{h}$$

Where $\phi_{x^*} = \frac{d\phi}{dx}\Big|_{x^*(\tau_0^-)}$. Suppose the nominal impact is not grazing on the ground surface, namely $\phi_{x^*} f(x^*(\tau_0^-), u^*(\tau_0^-)) \neq 0$, the perturbation of the pre-impact phase can be computed as

$$d\tau_h^- = \frac{\bar{h} - \phi_{x^*}\bar{x}_h^-}{\phi_{x^*} f(x^*(\tau_0^-), u^*(\tau_0^-))} \tag{5.24}$$

By differentiating equation 5.12, the following condition should hold

$$f(x^*(\tau_0^+), u^*(\tau_0^+))d\tau_h^+ + \bar{x}_h^+ = \Delta_{x^*}(f(x^*(\tau_0^-), u^*(\tau_0^-))d\tau_h^- + \bar{x}_h^-) \tag{5.25}$$

Since the post-impact phase is determined by the phase projection function Π, differentiate Π to compute the perturbation of the post-impact phase

$$\Pi_{x(t_c^+)}(f(x^*(\tau_0^+), u^*(\tau_0^+))d\tau^+ + \bar{x}_h^+) + \Pi_{x^*} f(x^*(\tau_0^+), u^*(\tau_0^+))d\tau_h^+ + \Pi_h\bar{h} + \Pi_{\tau^-} d\tau_h^- = 0 \tag{5.26}$$

Combining equation 5.24, 5.25 and 5.26, the following linear equations should be solved

$$\begin{bmatrix} \phi_{x^*} f(x^*(\tau_0^-), u^*(\tau_0^-)) & 0 & 0 \\ -\Delta_{x^*} f(x^*(\tau_0^-), u^*(\tau_0^-)) & f(x^*(\tau_0^+), u^*(\tau_0^+)) & I \\ \Pi_{\tau^-} & (\Pi_{x^+} + \Pi_{x^*}) f(x^*(\tau_0^+), u^*(\tau_0^+)) & \Pi_{x^+} \end{bmatrix} \begin{bmatrix} d\tau_h^- \\ d\tau_h^+ \\ \bar{x}_h^+ \end{bmatrix} = \begin{bmatrix} 1 & -\phi_{x^*} \\ 0 & \Delta_{x^*} \\ -\Pi_h & 0 \end{bmatrix} \begin{bmatrix} \bar{h} \\ \bar{x}_h^- \end{bmatrix} \quad (5.27)$$

Suppose that the phase projection function Π explicitly depends on the nominal trajectory and $\Pi_{x^*} f(x^*(\tau_0^+), u^*(\tau_0^+)) \neq 0$, the left matrix in equation 5.27 is nonsingular, thus by computing the inverse of such matrix, $d\tau_h^-, d\tau_h^+$ and $\bar{x}_h^+$ can be explicitly represented as a linear combination of $\bar{h}$ and $\bar{x}_h^-$. Suppose one of such representations is

$$\bar{x}_h^+ = T_1 \bar{h} + T_2 \bar{x}_h^- \quad (5.28)$$

which will be used in the following section.

5.2.4 Quadratic storage function within one cycle

For the linearized system, suppose the storage function is quadratic

$$V(\tau, \bar{x}) = \bar{x}' S(\tau) \bar{x} \quad (5.29)$$

Where $S \in \mathbb{R}^{n \times n}, S = S'$. For the continuous mode, substitute equation 5.29 into 5.21, and the following condition should hold

$$2\bar{x}' S(A\bar{x} + B\bar{u}) + \bar{x}' \dot{S} \bar{x} \leq -\bar{x}' Q \bar{x} - \bar{u}' R \bar{u} \quad (5.30)$$

If a linear controller is constructed as

$$\bar{u} = K\bar{x} \quad (5.31)$$

where $K \in \mathbb{R}^{m \times n}$ is the control gain, then by substituting all the $\bar{u}$ in 5.30, the following condition should be valid

$$\bar{x}'((A + BK)'S + S(A + BK) + Q + K'RK + \dot{S})\bar{x} \leq 0 \quad \forall\, \bar{x} \quad (5.32)$$

This quadratic inequality holds for all $\bar{x}$, thus it is equivalent to the negative semi-definiteness of the matrix in the middle, thus

$$-\dot{S} \succeq (A + BK)'S + S(A + BK) + Q + K'RK \quad (5.33)$$

For the ground impact transition, substituting 5.29 into 5.22 gives rise to

$$\bar{x}_h^{+\prime} S(\tau_h^+) \bar{x}_h^+ - \bar{x}_h^- S(\tau_h^-) \bar{x}_h^- \leq \gamma^2 \bar{h}^2 \quad (5.34)$$

Equation 5.34 contains terms $S(\tau_h^+)$ and $S(\tau_h^-)$, which do not have a closed form representation. To make problem viable, the second order expansion around $\bar{x}_h^- = 0$ and $\bar{h} = 0$ is taken to approximate 5.34. Since when for the nominal trajectory ($\bar{x} = 0$) and the nominal terrain height $\bar{h} = 0$, the periodic cycle exists, namely $\bar{x}_h^+ = 0$ for the case $\bar{x}_h^- = 0, \bar{h} = 0$, ($\tau_0^-, \tau_0^+$ are the pre- and post-impact phase on the nominal trajectory for the nominal terrain height), the second order expansion is simple

$$(T_1\bar{h} + T_2\bar{x}_h^-)'S(\tau_0^+)(T_1\bar{h} + T_2\bar{x}_h^-) - \bar{x}_h^- S(\tau_0^-)\bar{x}_h^- \leq \gamma^2\bar{h}^2 \quad \forall\, \bar{x}_h^-, \bar{h} \tag{5.35}$$

Where T_1, T_2 are defined in the previous section in equation 5.28.
This constraint can be reformulated as a quadratic constraint on the variable $[\bar{x}^-\ \bar{h}]$

$$\begin{bmatrix} \bar{x}_h^- \\ \bar{h} \end{bmatrix}' \left(\begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S(\tau_0^+) \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S(\tau_0^-) & 0 \\ 0 & \gamma^2 \end{bmatrix} \right) \begin{bmatrix} \bar{x}_h^- \\ \bar{h} \end{bmatrix} \preceq 0 \quad \forall\, \bar{x}^-, \bar{h} \tag{5.36}$$

Since the negativity holds for all $\bar{x}^-$ and $\bar{h}$, this is equivalent to the negative definiteness of the matrix in the middle

$$\begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S(\tau_0^+) \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S(\tau_0^-) & 0 \\ 0 & \gamma^2 \end{bmatrix} \preceq 0 \tag{5.37}$$

The following theorem concludes this section

Theorem 5.2.2. *For the linearized periodic system as defined in section 5.2.3 with a linear controller $\bar{u} = K\bar{x}$, a sufficient condition for the L_2 gain is less than a constant γ, is that there exists a matrix function $S(\tau) : [0,\ T] \to \mathbb{R}^{n\times n}$ satisfying the condition in the continuous mode*

$$-\dot{S} \succeq (A + BK)'S + S(A + BK) + Q + K'RK \quad \forall \tau \neq \tau_0^+, \tau_0^- \tag{5.38}$$

the jump transition

$$\begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S(\tau_0^+) \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S(\tau_0^-) & 0 \\ 0 & \gamma^2 \end{bmatrix} \preceq 0 \tag{5.39}$$

And

$$S(T) \succeq S(0) \tag{5.40}$$

$$S(T) \succeq 0 \tag{5.41}$$

Where 5.40 and 5.41 are obtained by substituting the quadratic storage function $V(\tau, x) = x'S(\tau)x$ into the conditions 5.15 and 5.16 respectively.

To minimize the L_2 gain of the hybrid system, we need to search for a controller K and a storage matrix S such that the infimum of γ is achieved. Unfortunately, this is not a semidefinite programming problem, as constraint 5.38 is quadratic in K. Thus we fix the controller first and search for the storage function, and then search for the controller.

5.2.5 Computing L_2 gain through LMI

According to Theorem 5.2.2, for the linearized periodic system defined in section 5.2.3, given a fixed periodic linear controller $\bar{u} = K\bar{x}$ satisfying $K(\tau + T) = K(\tau)\ \forall\tau$, denote the closed loop system as $\dot{x} = Fx$, where $F = A + BK$. To compute its L_2 gain is equivalent to find the smallest γ and a feasible quadratic storage function parameterized by $S(\tau) : [0,\ T] \to \mathbb{R}^{n\times n}$ as the optimal solution to the following program

$$\min_{\gamma,S} \quad \gamma \tag{5.42a}$$

$$\text{s.t} \quad \begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S(\tau_0^+) \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S(\tau_0^-) & 0 \\ 0 & \gamma^2 \end{bmatrix} \preceq 0 \tag{5.42b}$$

$$-\dot{S} \succeq (A+BK)'S + S(A+BK) + Q + K'RK \quad \forall\tau \neq \tau_0^+, \tau_0^- \tag{5.42c}$$

$$S(T) \succeq S(0) \tag{5.42d}$$

$$S(T) \succeq 0 \tag{5.42e}$$

Unfortunately this optimization problem is hard to handle for two reasons:

1. The decision variable S is a phase-parameterized matrix function, so there are infinitely many S matrices, instead of a single matrix.
2. The constraint 5.42c is a differential inequality for infinitely many time instances. It is hard to compute $\dot{S}$ in the closed form representation while regarding S as a decision variable.

However, as can be seen in the following analysis, the optimal solutions to the optimization problem 5.42a possess some special properties, and a semidefinite programming can be formulated to find such optimal solutions. The analysis starts with reformulating the differential inequality 5.42c first.

Lemma 5.2.3. *For a general differential Lyapunov function*

$$-\dot{P} = F'P + PF + M \tag{5.43}$$

The solution is given by

$$P(t) = \Phi_F(t_f, t)'P(t_f)\Phi_F(t_f, t) + \int_t^{t_f} \Phi_F(\sigma, t)'M(\sigma)\Phi_F(\sigma, t)d\sigma \tag{5.44}$$

Where $\Phi_F(\sigma, t)$ is the state transition matrix of the system $\dot{U} = FU$ from time t to time σ.

In our case 5.42c, we have a general differential Lyapunov inequality. For this case, the following lemma will help

Corollary 5.2.4. *For the differential Lyapunov Inequality*

$$-\dot{S} \succeq F'S + SF + M$$

The solution satisfies

$$S(t) \succeq \Phi_F(t_f, t)'S(t_f)\Phi_F(t_f, t) + \int_t^{t_f} \Phi_F(\sigma, t)'M(\sigma)\Phi_F(\sigma, t)d\sigma \quad \forall\, t < t_f$$

Proof. Note that the condition is equivalent to $\exists\ N \succeq 0$

$$-\dot{S} = F'S + SF + M + N$$

With Lemma (5.2.3), we know the solution to the equations above are

$$\begin{aligned} S_(t) &= \Phi_F(t_f, t)'S(t_f)\Phi_F(t_f, t) + \int_t^{t_f} \Phi_F(\sigma, t)'(M(\sigma) + N(\sigma))\Phi_F(\sigma, t)d\sigma \\ &\succeq \Phi_F(t_f, t)'S(t_f)\Phi_F(t_f, t) + \int_t^{t_f} \Phi_F(\sigma, t)'M(\sigma)\Phi_F(\sigma, t)d\sigma \end{aligned}$$

□

Applying Corollary (5.2.4 to the differential inequality constraint 5.42c, and substitute the square term $\gamma^2 = \zeta$, the optimization program is simplified as

$$\min_{\zeta, S(0), S(\tau_0^-), S(\tau_0^+), S(T)} \quad \zeta \tag{5.45a}$$

$$\text{s.t} \begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S(\tau_0^+) \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S(\tau_0^-) & 0 \\ 0 & \zeta \end{bmatrix} \preceq 0 \tag{5.45b}$$

$$S(\tau_0^+) \succeq \Phi_F(T, \tau_0^+)'S(T)\Phi_F(T, \tau_0^+) + \int_{\tau_0^+}^{T} \Phi_F(\sigma, \tau_0^+)'(Q + K'RK)\Phi_F(\sigma, \tau_0^+)d\sigma \tag{5.45c}$$

$$S(0) \succeq \Phi_F(\tau_0^-, 0)'S(\tau_0^-)\Phi_F(\tau_0^-, 0) + \int_0^{\tau_0^-} \Phi_F(\sigma, 0)'(Q + K'RK)\Phi_F(\sigma, 0)d\sigma \tag{5.45d}$$

$$S(T) \succeq S(0) \tag{5.45e}$$

$$S(T) \succeq 0 \tag{5.45f}$$

Note that for the given matrix A, B, K, the state transition matrix Φ_F and the integral term $\int_{t_1}^{t_2} \Phi_F(\sigma, t_1)'(Q + K'RK)\Phi_F(\sigma, t_1)d\sigma$ in equations 5.45c and 5.45d can all be numerically computed. Thus the optimization problem 5.45a is a semidefinite programming problem for the decision variables $\zeta, S(0), S(\tau_0^-), S(\tau_0^+), S(T)$.

Moreover, this semidefinite formulation can be further simplified by investigating the optimal condition.

Lemma 5.2.5. *For the semidefinite programming problem 5.45a-5.42e, the optimal solution satisfy* $S^*(0) = S^*(T)$

Proof. Notice that if there exists a feasible solution $(\zeta_1, S_1(0), S_1(\tau_0^-), S_1(\tau_0^+), S_1(T))$ to the constraint 5.45b-5.42e, and a set of matrices $S_2(0), S_2(\tau_0^-), S_2(\tau_0^+), S_2(T)$ satisfying $S_2(0) = S_2(T), S_2(\tau_0^+) = S_1(\tau_0^+), S_2(T) = S_1(T)$, then there exists an $S_2(\tau_0^-)$ such that $S_2(0), S_2(\tau_0^-)$ satisfy constraint 5.45d and $S_2(\tau_0^-) \succeq S_1(\tau_0^-)$. Actually since $S_2(0) = S_2(T) = S_1(T) \succeq S_1(0)$, by setting

$$S_2(\tau_0^-) = S_1(\tau_0^-) + \frac{\lambda_{\min}(S_2(0) - S_1(0))}{\lambda_{\max}(\Phi_F(\tau_0^-)'\Phi_F(\tau_0^-, 0))} I \tag{5.46}$$

we have $S_2(\tau_0^-) \succeq S_1(\tau_0^-)$ and $S_2(0), S_2(\tau_0^-)$ satisfy constraint 5.45d. So $(S_2(0), S_2(\tau_0^-), S_2(\tau_0^+), S_2(T))$ satisfy constraint 5.45c-5.42e. For constraint 5.45b, we have

$$\begin{aligned}
&\begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S_2(\tau_0^+) \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S_2(\tau_0^-) & 0 \\ 0 & \zeta_1 \end{bmatrix} \\
&= \begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S_1(\tau_0^+) \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S_1(\tau_0^-) & 0 \\ 0 & \zeta_1 \end{bmatrix} + \begin{bmatrix} S_1(\tau_0^-) - S_2(\tau_0^-) & 0 \\ 0 & 0 \end{bmatrix} \\
&\preceq 0
\end{aligned}$$

then (ζ_1, S_2) also satisfies the jump constraint 5.45b, thus $(\zeta, S_2(0), S_2(\tau_0^-), S_2(\tau_0^+), S_2(T))$ is also a feasible solution to the same optimization problem, indicating the infimum of the set

$$\{\zeta | S = S_2; (\zeta, S) \text{ satisfying constraint } 5.45b\}$$

is no larger than the infimum of the set

$$\{\zeta | S = S_1; (\zeta, S) \text{ satisfying constraint } 5.45b\}$$

For the given matrices S_1, S_2 constructed as before. Thus the optimal solution always satisfy $S^*(0) = S^*(T)$. □

Thus all $S(0)$ in the previous optimization program can be replaced by $S(T)$, and only three matrix decision variables are left. In the next lemma, we will see that only two matrix decision variables are necessary.

Lemma 5.2.6. *Suppose* $(\zeta, S_2(0), S_2(\tau_0^-), S_2(\tau_0^+), S_2(T))$ *satisfy constraint 5.45b-5.42d, then* $S(T) \succeq 0$ *if and only if* $S(\tau_0^+) \succeq 0$.

Proof. If $S(\tau_0^+) \succeq 0$, then by Schur Complement, for constraint 5.45b, in the upper left corner of the matrix in the left hand-side, $S(\tau_0^-) \succeq T_2' S(\tau_0^+) T_2 \succeq 0$. By constraint 5.45d, $S(0) \succeq 0$. By 5.42d, $S(T) \succeq S(0) \succeq 0$. If $S(T) \succeq 0$, then by constraint 5.45c, $S(\tau_0^+) \succeq 0$. □

Based on lemma 5.2.5 and 5.2.6, the decision variables $S(0), S(T)$ can be dropped. And the optimization program to find the L_2 gain can be reformulated as follows, for simplicity, denote $S(\tau_0^+) = S^+, S(\tau_0^-) = S^-$,$\Phi_F(\tau_0^-, 0)\Phi_F(T, \tau_0^+) = \Psi_F, \Phi_F(T, \tau_0^+)' \left(\int_0^{\tau_0^-} \Phi_F(\sigma, 0)'(Q + K'RK)\Phi_F(\sigma, 0) d\sigma \right) \Phi_F(T, \tau_0^+) + \int_{\tau_0^+}^{T} \Phi_F(\sigma, \tau_0^+)'(Q + K'RK)\Phi_F(\sigma, \tau_0^+) d\sigma = \bar{M}$.

$$
\begin{aligned}
\min_{\zeta, S^+, S^-} \quad & \zeta \\
\text{s.t} \quad & \begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S^+ \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S^- & 0 \\ 0 & \zeta \end{bmatrix} \preceq 0 \\
& S^+ \succeq \Psi_F' S^- \Psi_F + \bar{M} \\
& S^+ \succeq 0
\end{aligned}
\tag{5.47}
$$

And an upper bound L_2 equals to $\sqrt{\zeta}$. By using semidefinite programming software like *SeDuMi* [41], the optimization problem 5.47 can be efficiently solved.

5.3 Robust Controller Design for a given L_2 gain

In the last section, for a given fixed periodic linear controller $\bar{u} = K\bar{x}$, the scheme for computing an upper bound of the L_2 gain is derived. Since the ultimate goal of this chapter is to minimize the L_2 gain by designing a robust controller, such that the robot can walk more steadily on the rough terrain, naturally, the next question would be, given a constant γ as an upper bound of the L_2 gain, is there a scheme to construct such a periodic linear controller K? We will answer this question in this section. Based on theorem 5.2.2, and lemma 5.2.6, a sufficient condition for such a robust controller is that there exists a quadratic storage

function $V(\tau, \bar{x}) = \bar{x}' S(\tau) \bar{x}$ where $S(\tau) : [0,\ T] \to \mathbb{R}^{n \times n}$, such that

$$\begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S(\tau_0^+) \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S(\tau_0^-) & 0 \\ 0 & \gamma^2 \end{bmatrix} \preceq 0 \tag{5.48a}$$

$$-\dot{S} \succeq (A + BK)'S + S(A + BK) + Q + K'RK \ \forall \tau \neq \tau_0^+, \tau_0^- \tag{5.48b}$$

$$S(T) \succeq S(0) \tag{5.48c}$$

$$S(\tau^+) \succeq 0 \tag{5.48d}$$

This controller cannot be determined through semidefinite programming problem as in the last section, since equation 5.48b is quadratic in the control gain K instead of a matrix linear inequality, causing it intractable for the semidefinite programming. However, we will see that in there exist a solution K_1, S_1, γ satisfying all constraints 5.48a-5.48d, then there will be a constructive way to compute another solution K_2, S_2, γ satisfying the same constraints.

As is discussed in Lemma 5.2.5, if S_1, K_1, γ satisfy constraints 5.48a-5.48d, S_2, K_2 satisfy constraint 5.48b-5.48d and extra constraint $S_2(\tau_0^+) = S_1(\tau_0^+), S_2(\tau_0^-) \succeq S_1(\tau_0^-)$, then S_2, γ satisfy constraint 5.48a, thus S_2, K_2, γ satisfy all constraints 5.48a-5.48d, and K_2 would also be a γ sub-optimal controller.

The constraint 5.48b involves controller K, and also it is a differential inequality constraint. We would show that there exists a sub-optimal controller K as a function of matrix S, and we can restrain 5.48b to be an equality equation. We start by proving the following lemma

Lemma 5.3.1. *For two matrix functions $X, Y : \mathbb{R} \to \mathbb{R}^{n \times n}$ satisfying*

$$\begin{aligned} -\dot{X} &\preceq A'X + XA + Q - XBR^{-1}B'X \\ -\dot{Y} &= A'Y + YA + Q - YBR^{-1}B'Y \\ X(0) &= Y(0) \end{aligned}$$

Where (A, B) is controllable,$Q \in \mathbb{R}^{n \times n}, Q = Q' \succeq 0, R \in \mathbb{R}^{m \times m}, R = R' \succeq 0$. The following condition holds

$$X(t) \succeq Y(t) \quad \forall\, t > 0$$

Proof. Define

$$Z = Y - X$$

The dynamics of Z satisfies

$$\begin{aligned} -\dot{Z} &\succeq (A - BR^{-1}B'Y)'Z + Z(A - BR^{-1}B'Y) + ZBR^{-1}B'Z \\ Z(0) &= 0 \end{aligned}$$

Denote $A - BR^{-1}B'Y = \bar{A}, ZBR^{-1}B'Z = N, N \succeq 0$, by lemma 5.2.4, the solution of $Z(t), t > 0$ satisfies

$$Z(0) \succeq \Phi_{\bar{A}}(t,0)'Z(t)\Phi_{\bar{A}}(t,0) + \int_0^t \Phi_{\bar{A}}(\sigma,0)'N(\sigma)\Phi_{\bar{A}}(\sigma,0)d\sigma$$

Since (A, B) is controllable, Y is the solution to the differential Riccati equation, we know that Φ_A is nonsingular. Since $Z(0) = 0$, we then know $Z(t) \preceq 0 \; \forall t > 0$. Thus $X(t) \succeq Y(t) \quad \forall\, t > 0$. □

Corollary 5.3.2. *For two matrix functions $X, Y : \mathbb{R} \to \mathbb{R}^{n\times n}$ satisfying*

$$\begin{aligned} -\dot{X} &\succeq A'X + XA + Q - XBR^{-1}B'X \\ -\dot{Y} &= A'Y + YA + Q - YBR^{-1}B'Y \\ X(0) &= Y(0) \end{aligned}$$

Where (A, B) is controllable,$Q \in \mathbb{R}^{n\times n}, Q = Q' \succeq 0, R \in \mathbb{R}^{m\times m}, R = R' \succeq 0$. The following condition holds

$$X(t) \preceq Y(t) \quad \forall\, t > 0$$

Proof. $\exists N \in \mathbb{R}^{n\times n}, N = N' \succeq 0$, s.t

$$\begin{aligned} -\dot{X} &= A'X + XA + (Q + N) - XBR^{-1}B'X \\ -\dot{Y} &\preceq A'Y + YA + (Q + N) - YBR^{-1}B'Y \\ X(0) &= Y(0) \end{aligned}$$

By the previous lemma,

$$X(t) \preceq Y(t) \quad \forall\, t > 0$$

□

So for any storage function S and K satisfying the constraints 5.48a-5.48d, by using completion of squares,

it can be shown that

$$\begin{aligned}\dot{S} &\preceq -((K+R^{-1}B'S)'R(K+R^{-1}B'S)+A'S+SA+Q-SBR^{-1}B'S) \\ &\preceq -(A'S+SA+Q-SBR^{-1}B'S)\end{aligned} \tag{5.49}$$

Namely, the derivative of this storage function is upper bounded by the Riccati function $-(A'S+SA+Q-SBR^{-1}B'S)$. So by constructing a new storage function $\bar{S}$ and a controller $\bar{K}$ satisfying

$$\begin{aligned}\bar{K} &= -R^{-1}B'\bar{S} \\ -\dot{\bar{S}} &= (A'\bar{S}+\bar{S}A+Q-\bar{S}BR^{-1}B'\bar{S}) \\ \bar{S}(\tau_0^+) &= S(\tau_0^+), \quad \bar{S}(0)=S(0)\end{aligned}$$

By the corollary 5.3.2, we know the storage function satisfies $\bar{S}(T) \succeq S(T) \succeq S(0) = \bar{S}(0)$, and $\bar{S}(\tau_0^-) \succeq S(\tau_0^-)$, so the newly constructed storage function $\bar{S}$ and controller $\bar{K}$ satisfy the constraints 5.48a-5.48d. Namely, if there exists a γ sub-optimal controller for the system, then the controller K satisfying the following constraints is such a controller.

$$\exists\, S:[0,\ T]\to\mathbb{R}^{n\times n} \quad \text{s.t} \quad \begin{bmatrix}T_2'\\ T_1'\end{bmatrix} S(\tau^+)\begin{bmatrix}T_2 & T_1\end{bmatrix} - \begin{bmatrix}S(\tau^-) & 0\\ 0 & \gamma^2\end{bmatrix} \preceq 0 \tag{5.50a}$$

$$-\dot{S} = A'S+SA+Q-SBR^{-1}B'S \tag{5.50b}$$

$$S(T) \succeq S(0) \tag{5.50c}$$

$$S(\tau_0^+) \succeq 0 \tag{5.50d}$$

$$K = -R^{-1}B'S \tag{5.50e}$$

The next step is to show that the constraint 5.50c can be strict. To show that, consider the following lemma

Lemma 5.3.3. *For the Riccati equation*

$$-\dot{S} = A'S+SA+Q-SBR^{-1}B'S$$

And two solutions to this Riccati equation, with $X(0) \preceq Y(0)$. Then the following condition holds

$$X(t) \preceq Y(t) \quad \forall\, t>0$$

Proof. Define $Z = X - Y$, the dynamics of Z is

$$-\dot{Z} = (A - BR^{-1}B'X)'Z + Z(A - BR^{-1}B'X) + ZBR^{-1}B'Z$$

Denote $\bar{A} = A - BR^{-1}B'X, M = ZBR^{-1}B'Z \succeq 0$, then the solution of Z satisfies

$$Z(0) = \Phi_{\bar{A}}(t,0)'Z(t)\Phi_{\bar{A}}(t,0) + \int_0^t \Phi_{\bar{A}}(\sigma,0)'M(\sigma)\Phi_{\bar{A}}(\sigma,0)d\sigma$$

Since $Z(0) = X(0) - Y(0) \preceq 0$, then $Z(t) \preceq 0$, namely $X(t) \preceq Y(t) \quad \forall\, t > 0$ □

So for any storage function S satisfying 5.50a-5.50d, we can construct a matrix function $\bar{S}$, such that $\bar{S}(0) = S(T) > S(0), \bar{S}(\tau_0^+) = S(\tau_0^+), -\dot{\bar{S}} = A'\bar{S} + \bar{S}A + Q - \bar{S}BR^{-1}B'\bar{S}$. Thus the newly constructed $\bar{S}$ satisfies $\bar{S}(\tau_0^-) \succeq S(\tau_0^-), \bar{S}(T) = S(T) = \bar{S}(0)$, so $\bar{S}$ also satisfies constraint 5.50a-5.50d. So construct a new controller $\bar{K} = -R^{-1}B'\bar{S}$, and it will also be a γ sub-optimal controller. In all, if there exists a γ sub-optimal controller, then the controller K satisfying the following constraint would be such a controller

$$\exists\, S : [0,\ T] \to \mathbb{R}^{n\times n} \quad \text{s.t} \quad \begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S(\tau_0^+) \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S(\tau_0^-) & 0 \\ 0 & \gamma^2 \end{bmatrix} \preceq 0 \tag{5.51a}$$

$$-\dot{S} = A'S + SA + Q - SBR^{-1}B'S \quad \forall \tau \neq \tau_0^+, \tau_0^- \tag{5.51b}$$

$$S(T) = S(0) \tag{5.51c}$$

$$S(\tau_0^+) \succeq 0 \tag{5.51d}$$

$$K = -R^{-1}B'S \tag{5.51e}$$

By Schur's Complement, when $\gamma^2 > T_1'S(\tau_0^+)T_1$, the jump transition 5.51a is equivalent to

$$S(\tau_0^-) \succeq T_2'S(\tau_0^+)[I - \gamma^{-2}T_1T_1'S(\tau_0^+)]^{-1}T_2$$

An important observation is that the current conditions for a γ sub-optimal controller to exist for the hybrid system, is equivalent to the case that the following linearized periodic system $\bar{\mathcal{S}}$ has an L_2 gain no larger than γ

$$\begin{aligned} \dot{x}(t) =& A(t)x(t) \quad \forall t \neq \tau_0^- + NT, \tau_0^+ + NT, N \in \mathbb{Z} \\ x(t^+) =& T_2x(t^-) + T_1\bar{h} \quad t = \tau_0 + NT, N \in \mathbb{Z} \end{aligned}$$

It is known that the following statements are equivalent [47]

1. $\bar{\mathcal{S}}$ has an L_2 gain no larger than γ

2. If $\gamma^2 > T_1' S(\tau_0^+) T_1$, then there exists a periodic solution to the following periodic Riccati equation

$$S(\tau_0^-) = T_2' S(\tau_0^+)[I - \gamma^{-2} T_1 T_1' S(\tau_0^+)]^{-1} T_2 \tag{5.52a}$$

$$-\dot{S} = A'S + SA + Q - SBR^{-1}B'S \quad \forall \tau \neq \tau_0^+, \tau_0^- \tag{5.52b}$$

Thus the controller K can be computed by first solving the periodic Riccati equations 5.52a, 5.52b to find the storage matrix S, and $K = -R^{-1}B'S$. From the previous theorem 4.2.1 on the periodic Riccati equation, such solution is unique, and can be computed by backward integration until the solution converges.

5.4 Iteratively minimize the L_2 gain

The goal is to minimize the upper bound of the L_2 gain of the hybrid system by designing a robust controller. Based on theorem 5.2.2, this is not a semidefinite programming problem, thus cannot be efficiently solved. Instead, an iterative approach is used here. In each iteration, given a linear controller, a storage function is searched through a semidefinite programming, so as to compute the upper bound γ of L_2 associated with such controller. Then for such upper bound, solve the periodic Riccati equation to determine a controller, such that the feedback system has an L_2 gain no larger than γ. And this controller is used as the given controller in the next iteration. The algorithm is stated as in Algorithm 1. The initial guess of the controller can be any periodic linear controller (the LQR controller, zero control, etc.)

Apparently the upper bound γ_i is nondecreasing in each iteration. Since in each iteration, the semidefinite programming will find a lower γ. This alternative search stops when the γ gap between two consecutive iterations gets below some threshold.

5.5 Result

5.5.1 Compute L_2 gain

The algorithm above are applied to the compass gait walker. We first compare the upper bound of the L_2 gain for two types of limit cycles, the robust limit cycle computed in Chapter 4, and the passive limit cycle. For both cycles with the periodic LQR controller $Q = diag([1\ 1\ 10\ 10])$ and $R = 1$, the robust cycle has $\gamma = 21.3043$, while the passive cycle has $\gamma = 292.6646$, more than 13 times larger than that of the robust cycle. Such large discrepancy in the L_2 gain and the huge distinction in the number of steps on the rough terrain ($\geq 10,000$ vs ≈ 10) indicates that L_2 gain is a good measurement of the actual ability to traverse the rough terrain.

Algorithm 1 Iterative optimization

1: At iteration i, given γ_i as an upper bound of the L_2 gain, find a periodic solution to the following Riccati equation

$$P_i(\tau_0^-) = T_2' P_i(\tau_0^+)(I - \gamma^{-2} T_1 T_1' P_i(\tau_0^+))^{-1} T_2$$
$$-\dot{P}_i = A'P_i + P_i A + Q - P_i B R^{-1} B' P_i \quad \forall\, \tau \neq \tau_0^+, \tau_0^-$$

Find the γ_i-suboptimal controller K_i

$$K_i = -R^{-1} B' P_i$$

2: With the controller K_i, solve the following semidefinite programming problem

$$\begin{aligned} \min_{S_i^+, S_i^-, \zeta_i} \quad & \zeta_i \\ \text{s.t} \quad & \begin{bmatrix} T_2' \\ T_1' \end{bmatrix} S_i^+ \begin{bmatrix} T_2 & T_1 \end{bmatrix} - \begin{bmatrix} S_i^- & 0 \\ 0 & \zeta_i \end{bmatrix} \preceq 0 \\ & S_i^+ \succeq \Psi_{F_i}' S_i^- \Psi_{F_i} + \bar{M}_i \\ & S_i^+ \succeq 0 \\ & \zeta_i > T_1' S_i^+ T_1 \end{aligned}$$

Where

$$\begin{aligned} F_i =& A + B K_i \\ \bar{M}_i =& \int_t^{t_f} \Phi_{F_i}(\sigma, t)'(Q + K_i' R K_i) \Phi_{F_i}(\sigma, t) d\sigma \end{aligned}$$

3: $i \leftarrow i + 1$, with $\gamma_{i+1} = \sqrt{\zeta_i}$

5.5.2 Minimize the L_2 gain

We minimize the L_2 gain of the passive limit cycle by designing the robust controller associated with it. In 6 iterations, the L_2 gain drops from 292.6646 to 269.9908, the decreasing sequence of γ_i is plotted in Fig 5-2. The γ sequence drops quickly in the first two iterations, and then converges. We simulate the compass gait tracking the passive cycle on the rough terrain with three controllers, the LQR controller with $\gamma = 292.6646$, the robust controller 1 with $\gamma = 269.9908$, and another robust controller 2 with $\gamma = 274.2038$. The robust controller 1 is computed after the sixth iteration in the optimization algorithm, while robust controller 2 is computed after the second iteration. The terrain slope is a uniform distribution within $[4^\circ,\ 6^\circ]$. Note that the terrain disturbance is very small in magnitude. The reason is that the linearized method can only be effective in the small neighborhood of the nominal terrain (slope equals to 5°). The simulation results are depicted in Fig 5-3. 100 simulations are run, and the number of steps traversed before falling down are drawn in each plots. In Fig 5-3a, the number of steps traversed in the same unknown terrain is compared between the LQR controller ($\gamma = 292.6646$) and the robust controller 1 ($\gamma = 269.9908$), in a large amount of

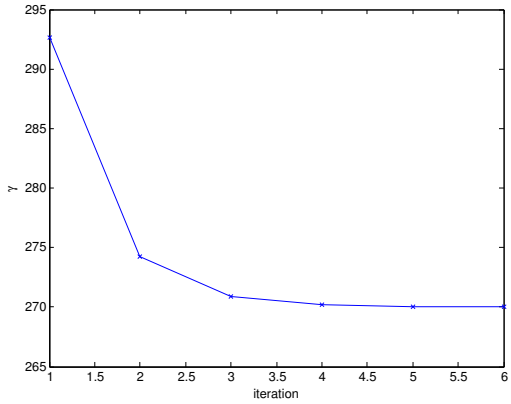
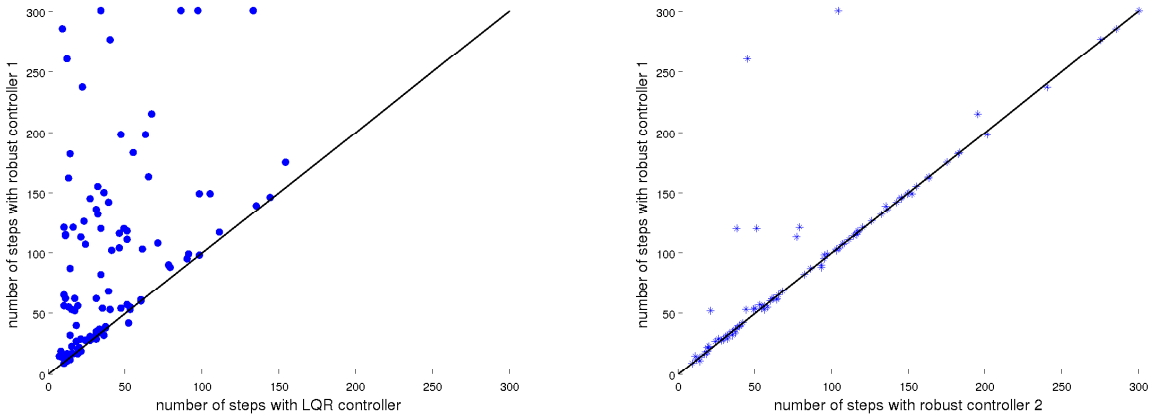


Figure 5-2: γ for the passive cycle in each iteration



(a) Compare LQR and robust controller 1

(b) Compare robust controller 2 and robust controller 1

Figure 5-3: number of steps traversed on the rough terrain in each simulation

trials, the robust controller 1 makes the robot takes much more steps than the LQR does. In Fig 5-3b, the robust controller 1 ($\gamma = 269.9908$) and the robust controller 2 ($\gamma = 274.2038$) perform very similarly on the unknown terrain. But the control gain of those two robust controllers are noticeably different. The control gain at τ_0^- of the robust controller 1 is around 2.3 times larger than that of the robust controller 2. This big difference suggest that although the controller are quite different in the gain value, as their L_2 gain are close, their performance on the rough terrain are similar, implies that L_2 gain is a good measurement of the terrain sensitivity.

The average number of steps for the three controllers are shown in the following table. By decreasing the

controller	LQR	Robust2	Robust1
γ	292.6646	274.2038	269.9908
average number of steps before falling down	40.61	89.27	96.21

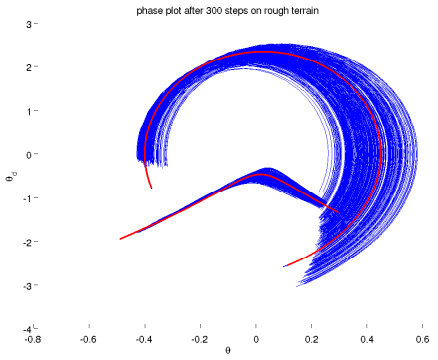


Figure 5-4: Phase plot of compass gait following passive cycle with a robust controller, walking 300 steps on the rough terrain

L_2 gain of the linearized system from 292.6646 to 269.9908, the average number of steps traversed on the rough terrain gets doubled. Given the compass gait is extremely sensitive to the terrain disturbances, it is not strange that a 10% drop of the L_2 gain can double the number of steps.

A phase plot of the compass gait on the rough terrain with robust controller $\gamma = 269.9908$ is shown in Fig 5-4. The red curve is the passive limit cycle, and the blue curves are the simulated trajectory for 300 steps. Although the robot does not fall down, it is clear that the volume of the trajectory bundle does not shrink along the nominal trajectory, which happens in the robust cycle case. This comparison indicates that constructing a linear robust controller is not as powerful as designing a robust limit cycle, in terms of improving the robustness of the hybrid system.

Chapter 6

Conclusion

6.1 Contributions

In this thesis, to improve the ability of a bipedal robot traversing the rough terrain, we model the problem as a hybrid system with uncertainty in the guard function, and propose two metrics quantifying the robustness of such systems. For each metric, an optimization scheme is presented to bring down the sensitivity measurement.

We design a walking gait that is intrinsically more robust on a stochastic terrain. The metric considered in this case is the expected LQR cost-to-go of the post-impact states. And we use nonlinear optimization techniques to solve this trajectory optimization problem. Simulations shows that the number of steps on the rough terrain can be drastically improved.

We then design a linear controller to steer the robot following the designated walking gait. The metric used is the worst-case performance for the linearized system, and an iterative optimization approach is used to bring down the sensitivity of the linearized system. In each iteration, a semidefinite programming problem is optimized to compute the upper bound of the L_2 gain, and a periodic Riccati equation is solved to construct the linear controller with the desired L_2 gain. Simulation demonstrate that this approach can marginally improve the number of steps on a terrain with minor elevation.

6.2 Discussion

From our simulation results, designing an intrinsically robust walking gait is much more effective than designing a robust controller, as by following the robust limit cycle, the robot never falls down, while on a less rugged terrain, even with the robust controller, the passive cycle still falls down within around 100 steps. This matches our common intuition, as human adapt to different terrains by noticeably changing

his walking gait (lowering the center of mass, taking smaller steps, etc). Although powerful, solving this trajectory optimization problem is not very efficient, for at least two reasons

1. The trajectory optimization is a nonlinear optimization problem, instead of a convex optimization problem. Solving NLP is generally more time-consuming, and subject to the local minimum.

2. This NLP has a large number of decision variables. To compute the cost-to-go, a matrix with size $n \times n$ is chosen as a decision variable for every discretized knot. So the size of the optimization problem grows as n^2. For system with large *degrees of freedom*, solving such a large-scale nonlinear optimization problem is extremely challenging.

The second approach, to linearize the system and minimize the L_2 gain, is not as powerful as the first one. In the compass gait example, the L_2 gain drops by less than 10% after the optimization, and can only marginally increase the number of steps traversed in the landscape with slight terrain elevation. One major reason for its shortage is that this approach only handles the linearized system, while for the hybrid system with uncertainty in the guard function, the dynamics is extremely nonlinear, thus the linearization is only valid within a small neighborhood of the nominal trajectory. For this reason, the **nonlinear** trajectory optimization can drastically improve the robust performance, while optimizing the L_2 gain for the linearized system does not produce equally satisfactory results. On the other hand, the second approach has its own advantage, as it is a small-scale semidefinite programming problem, and can be solved very efficiently, and it scales well to more complicated robot with high degrees of freedom.

6.3 Future work

To apply the optimization scheme to real humanoids, the algorithm should scale well when the degrees of freedom increases. Unfortunately, optimizing the robust limit cycle suffers from the 'curse of dimensionality', mostly because the cost-to-go matrix at every knot point is chosen as decision variables. Other than LQR cost-to-go, we will seek for a better metric in the state space with less computation burden.
As discussed above, when designing the robust controller for the bipedal walker, a linear controller for the linearized system is not very satisfactory. The future work will be to design the controller for the nonlinear hybrid system directly. The optimization objective can be the largest terrain perturbation that can be tolerated. To design such a controller, we can use the *Sum of Squares* technique to construct the funnel around the nominal trajectory, and such funnel should be invariant under the terrain disturbances.

Bibliography

[1] Betts and J.T. *Practical methods for optimal control and estimation using nonlinear programming.* Society for Industrial & Applied, 2010.

[2] Robert R. Bitmead, Michel R. Gevers, Ian R. Petersen, and R.John Kaye. Monotonicity and stabilizability- properties of solutions of the riccati difference equation: Propositions, lemmas, theorems, fallacious conjectures and counterexamples. *Systems and Control Letters*, 5(5):309 – 315, 1985.

[3] S. Bittanti, P. Colaneri, and G. De Nicolao. The periodic riccati equation. In S. Bittanti, A. J. Laub, and J. C. Willems, editors, *The Riccati Equation*, chapter 6. Springer-Verlag, 1991.

[4] R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(1112):1217 – 1227, 1989.

[5] M.S. Branicky. *Studies in hybrid systems: Modeling, analysis, and control.* PhD thesis, 1995.

[6] Peng Cheng, LaValle, and S.M. Reducing metric sensitivity in randomized trajectory design. *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 1:43–48 vol.1, 2001.

[7] Christine Chevallereau, J. W. Grizzle, and Ching-Long Shih. Asymptotically stable walking of a five-link underactuated 3-d bipedal robot. *IEEE Transactions on Robotics*, 25(1):37–50, February 2009.

[8] Steven H. Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, February 18 2005.

[9] Steven H. Collins, Martijn Wisse, and Andy Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *International Journal of Robotics Research*, 20(7):607–615, July 2001.

[10] Elnagar, G., Kazemi, M.A., Razzaghi, and M. The pseudospectral legendre method for discretizing optimal control problems. *Automatic Control, IEEE Transactions on*, 40(10):1793 –1796, oct 1995.

[11] Paul J. Enright and Bruce A. Conway. Optimal finite-thrust spacecraft trajectories using collocation and nonlinear programming. *J. Guidance*, 14(5):981–985, Sept-Oct 1991.

[12] Michael Ernst, Hartmut Geyer, and Reinhard Blickhan. Spring-legged locomotion on uneven ground: a control approach to keep the running speed constant. In *International Conference on Climbing and Walking Robots*, pages 639–644, 2009.

[13] Garg, Divya, Patterson, Michael, Francolin, Camila, Darby, Christopher, Huntington, Geoffrey, Hager, William, Rao, and Anil. Direct trajectory optimization and costate estimation offinite-horizon and infinite-horizon optimal control problems using a radau pseudospectral method. *Computational Optimization and Applications*, 49:335–358, 2011. 10.1007/s10589-009-9291-0.

[14] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[15] Philip E. Gill, Walter Murray, and Michael A. Saunders. *User's Guide for SNOPT Version 7: Software for Large -Scale Nonlinear Programming*, February 12 2006.

[16] Elena Leah Glassman. A quadratic regulator-based heuristic for rapidly exploring state space. Master's thesis, Massachusetts Institute of Technology, February 2010.

[17] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *J Guidance*, 10(4):338–342, July-August 1987.

[18] Hobbelen, D.G.E., Wisse, and M. A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm. *Robotics, IEEE Transactions on*, 23(6):1213 –1224, dec. 2007.

[19] Mrinal Kalakrishnan, Jonas Buchli, Peter Pastor, Michael Mistry, and Stefan Schaal. Learning, planning, and control for quadruped locomotion over challenging terrain. *I. J. Robotic Res.*, 30(2):236–258, 2011.

[20] J. Zico Kolter and Andrew Y. Ng. The stanford littledog: A learning and rapid replanning approach to quadruped locomotion. *I. J. Robotic Res.*, 30(2):150–174, 2011.

[21] Ian R. Manchester, Uwe Mettin, Fumiya Iida, and Russ Tedrake. Stable dynamic walking over rough terrain: Theory and experiment. In *Proceedings of the International Symposium on Robotics Research (ISRR)*, 2009.

[22] T. Mcgeer. Passive bipedal running. *Proceedings of the Royal Society of London*, 240(1297):107–134, May 1990.

[23] Tad McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, April 1990.

[24] Tad McGeer. Passive walking with knees. pages 1640 – 1645. IEEE International Conference on Robotics and Automation (ICRA), 1990.

[25] Tad McGeer. Dynamics and control of bipedal locomotion. *Journal of Theoretical Biology*, 163:277–314, 1993.

[26] Alexandre Megretski. Personal communication, 2008.

[27] Katja D. Mombaur. *Stability Optimization of Open-Loop Controlled Walking Robots*. PhD thesis, Heidelberg, August 2001.

[28] Jun Morimoto and Christopher Atkeson. Minimax differential dynamic programming: An application to robust biped walking. *Advances in Neural Information Processing Systems*, 2002.

[29] Jun Morimoto, Jun Nakanishi, Gen Endo, Gordon Cheng, Christopher G Atkeson, and Garth Zeglin. Poincare-map-based reinforcement learning for biped walking. *ICRA*, 2005.

[30] Toshimitsu Nishimura and Hiroyuki Kano. Periodic strong solutions of periodically time-varying matrix riccati equations. *International Journal of Control*, 49(1):193–205, 1989.

[31] Hae-Won Park. *Control of a Bipedal Robot Walker on Rough Terrain*. PhD thesis, University of Michigan, 2012.

[32] Michael Posa and Russ Tedrake. Direct trajectory optimization of rigid body dynamical systems through contact. In *To appear in the Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, 2012.

[33] Marc H. Raibert. *Legged Robots That Balance*. The MIT Press, 1986.

[34] C. David Remy. *Optimal Exploitation of Natural Dynamics in Legged Locomotion*. PhD thesis, ETH ZURICH, 2011.

[35] Shai Revzen. *Neuromechanical control architectures in arthropod locomotion*. PhD thesis, University of California, Berkeley, 2009.

[36] Andre Seyfarth, Hartmut Geyer, and Hugh Herr. Swing-leg retraction: a simple control model for stable running. *The Journal of Experimental Biology*, 206:2547–2555, Feb 2003.

[37] Alexander Shkolnik, Michael Levashov, Ian R. Manchester, and Russ Tedrake. Bounding on rough terrain with the littledog robot. *The International Journal of Robotics Research (IJRR)*, 30(2):192–215, Feb 2011.

[38] Sreenath, K., Park, H.W., Poulakakis, I., Grizzle, and JW. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *International Journal of Robotics Research*, 2010.

[39] Manoj Srinivasan and Andy Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439:72–75, January 5 2006.

[40] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Perseus Books, 1994.

[41] Jos F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625 – 653, 1999.

[42] Russ Tedrake. *Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines: Course Notes for MIT 6.832*. Working draft edition, 2012.

[43] Russ Tedrake, Ian R. Manchester, Mark M. Tobenkin, and John W. Roberts. LQR-Trees: Feedback motion planning via sums of squares verification. *International Journal of Robotics Research*, 29:1038–1052, July 2010.

[44] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37:357–373, 1992.

[45] Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, Boca Raton, FL, 2007.

[46] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1):25–57, May 2006.

[47] Lihua Xie, de Souza, and C.E. H infin; state estimation for linear periodic systems. *Automatic Control, IEEE Transactions on*, 38(11):1704 –1707, nov 1993.

[48] Zhou, Kemin, Doyle, John C., Glover, and Keith. *Robust and optimal control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[49] Matthew Zucker, Nathan D. Ratliff, Martin Stolle, Joel E. Chestnutt, J. Andrew Bagnell, Christopher G. Atkeson, and James Kuffner. Optimization and learning for rough terrain legged locomotion. *I. J. Robotic Res.*, 30(2):175–191, 2011.