Feedback Control of Highly Dynamic 3D Bipedal Locomotion

by

Yukai Gong

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Robotics)
in The University of Michigan
2022

Doctoral Committee:

Professor Jessy W. Grizzle, Chair
Assistant Professor Maani Ghaffari
Professor Odest Chadwicke Jenkins
Associate Professor Ramanarayan Vasudevan

Yukai Gong

ykgong@umich.edu

ORCID iD: 0000-0003-2448-8891

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Jessy Grizzle, for giving me the opportunity to do research in bipedal robotics. I am grateful for his instruction and patient help, and admire his energy, broad-mindedness, wisdom, caring, and tolerance. I want to express my appreciation to the members of my dissertation committee, Prof. Chad Jenkins, Prof. Ram Vasudevan, and Prof. Maani Ghaffari, for their support and insightful feedback.

I am thankful to the members of our lab. Especially to Xingye Da for spending so much time on teaching me bipedal control one on one. To Ross Hartley for the initial work on Cassie. To Ayonga Hereid and Omar Harib for building and perfecting the powerful tool FROST on which my controller was built. To Zhenyu Gan for inspiring discussions and sharing the burden of maintaining hardware. To Lu Gan, Rui Zhang and Bruce Huang for the fun and pain we experienced together in the autonomy experiments. To Eva Mungai, Grant Gibson, and Wami Ogunbi for your help in experiments and emotional support. To Dianhao Chen for extraordinary work in the autonomy experiments.

Lastly, I want to thank my parents, who provided great support and help. Being aware that there was always an unconditional haven comforted me and helped me get through many difficult times.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Bipedal robots have the potential to free humans from tedious or dangerous tasks. Compared to robots in other forms, a bipedal robot has similar morphology to humans and thus can work in almost all spaces where humans work and requires little to no facility modifications. However, while many other robots are deployed in real life and are beginning to have an impact, bipedal robots are hardly seen outside of a lab due to stability issues. Bipeds are inherently unstable due to their morphology. A bipedal system is nonlinear, high dimensional, hybrid, and underactuated, which poses significant challenges to controller design. This thesis will therefore focus on developing control methods for biped locomotion.

First, we discuss a controller for a Cassie-series 3D bipedal robot designed with gait-library methods. The full 20 degrees of freedom dynamic model of Cassie and optimization are used to design seven periodic gaits for walking in place, forward, and backward while meeting key physical constraints. Importantly, we show how to practically implement these gaits on the robot.

Next, we conduct a more general study of the dynamics of bipedal robots. We establish connections between various approximate pendulum models that are commonly used for heuristic controller design and those that are more common in the feedback control literature where formal stability guarantees are the norm. We clarify commonalities and differences in the two perspectives for using low-dimensional models. In the process of doing so, we argue that models based on angular momentum about the contact point provide more faithful representations of robot state than models based on linear velocity. Specifically, we show that an approximate (pendu-

lum or zero dynamics) model parameterized by angular momentum provides better predictions for a physical robot (e.g., legs with mass) than does a related approximate model parameterized in terms of linear velocity. We call the pendulum model parameterized by angular momentum ALIP.

Finally, we discuss general mechanisms in bipedal balance, explain why foot placement is the most effective method, and select it as our primary method to stabilize a bipedal gait. We focus on regulating angular momentum about the contact point with the ALIP model. We implement a one-step-ahead angular-momentum-based walking controller on Cassie and demonstrate high agility and robustness in experiments. We also design a running controller with the same methodology and demonstrate the results in simulation and experiments.

CHAPTER I

Introduction

1.1 Motivation

Robots are created to help humans work more efficiently or to free humans from tedious or dangerous tasks (Fig. 1.1). Legged robots have the advantage of being able to work under certain conditions that other robots with different morphology find difficult. Compared to a wheeled robot, a legged robot can move agilely in environments designed for humans, where stairs or on-ground-obstacles are present, and can travel outdoors in areas lacking infrastructure, such as a paved road. Compared to a flying robot, a legged robot consumes less energy per unit weight and thus can work or standby for a longer interval of time. Within the class of legged robots, bipeds, quadrupeds, and hexapods have so far attracted the most interest. In environments designed for humans, a bipedal robot may be more suitable than robots with more legs, especially when space is narrow since it can carry more load with the same occupied area, see Fig. 1.2. Furthermore, methods implemented on a bipedal robot can be transferred to exoskeletons that strengthen human power or rehabilitate people with disabilities.

However, despite numerous potential advantages, bipedal robots are rarely seen in real life. One of the significant challenges preventing them from being utilized is difficulties in controlling them. A bipedal robot, seen from a control perspective, is

Figure 1.1: Cassie walking through burnt ground. Legged Robots have the potential to replace humans in dangerous situations. [1]

nonlinear, high dimensional, hybrid, underactuated, and inherently unstable. This makes bipedal locomotion a very complex control problem, which cannot be modeled and stabilized with classical control techniques. This proposal will therefore focus on developing control methods for biped locomotion.

1.2 Review

1.2.1 Legged Robot Prototypes

Legged locomotion is defined by alternating contact of leg ends with the ground. Legged locomotion is adopted by most animals living on land because of its versatility: it is fast, energy-efficient, robust to wild terrains, and biologically feasible. However, unlike wheeled locomotion or flying, legged locomotion has never been utilized as a practical tool or machine in human history because of multiple technical difficulties. Despite all limitations, the interest in such machines appeared very early

[1] Photographed by Bruce JK Huang

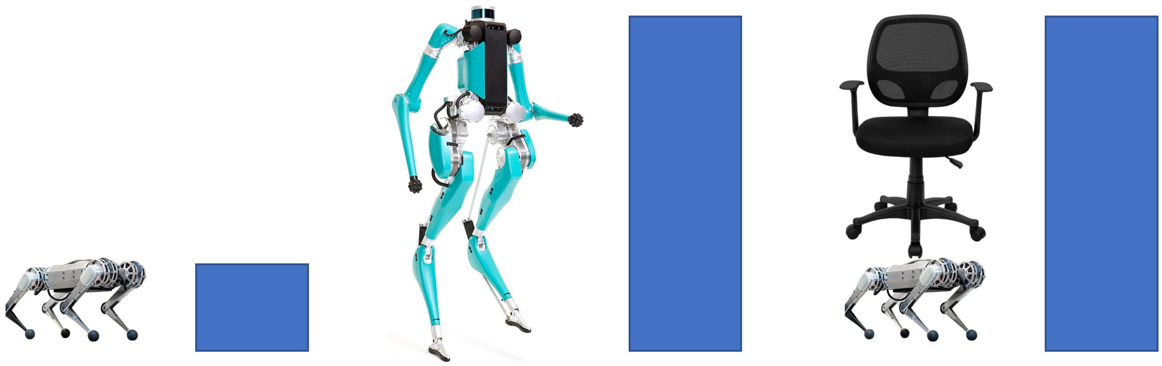


Figure 1.2: Quadrupedal robots are often believed to be inherently more stable than bipedal robots because of their low Height/Support Area Ratio. However, when a quadrupedal robot is required to carry a load, for example, a chair, the quadruped's ratio will become similar to that of a biped while the biped's ratio is hardly changed when it carries the same chair. A large quadruped would avoid the dramatic ratio change when carrying a load, but it would require a large space in which to maneuver. Therefore, solving balancing problems for robots with high Height/Support Area Ratio is critical to making legged robots practically useful.

in the modern history of robotics. The first evidence of such machine is no later than the nineteenth century when Rygg created his mechanical horse[5], which moves with a fixed gait generated by gears and levers. Since the latter half of the 20th century, more and more legged robots have been created. The General Electric Walking Truck was constructed by Mosher [6] in 1965. With four legs, it weighed 1400kg and could walk at 8km/h. A human operator was required to control the stepping of the robot. The first legged robot coordinated by computer was Phony Pony created in 1968[7]. With two degrees of freedom on each leg, it can walk in a straight line only. The first bipedal robot was created by Kato in 1972[8]. In 1980 Raibert created a monopod hopper that demonstrated dynamic balancing ability[9]. In 2000, Asimo was created by Honda Corporation[10]. It is able to walk, run, climb stairs, and dance. BigDog, created by Boston Dynamics in 2005[11], might be the first legged robot that had practical uses. It could carry 150kg and walk over difficult terrains. In the past decade, numerous legged robots have been created; famous ones include Atlas, Spot, ANYMAL, Cheetah, Cassie, Laikago... and the list goes on. Biped or Quadruped, all of them demonstrate the ability to move limbs with agility.

1.2.2 Bipedal Robot Control

Compared to quadruped or hexapod robots, bipedal robots pose a more significant challenge to control because of a smaller or no support polygon. Some bipeds are created with large feet and, as a result, are inherently more stable. However, this kind of configuration sacrifices many advantages of bipeds such as walking on stepping stones or on uneven terrain. Instead of balancing the robot with a pair of large feet, we focused on bipeds with realistic configurations, i.e., normal size feet. To balance these kinds of robots, instead of quasi-static stability which balances the robot by moving slowly and always keeping the Center of Mass (CoM) within the support polygon, we choose to achieve stability dynamically, which balances the robot through proper movement. We believe achieving dynamic stability is necessary to realize the full potential of a biped robot.

The first actively controlled dynamic balancing was achieved by Raibert with his monopod [9], using a model-free method. The monopod had three links and two joints: one rotational joint between the torso and upper leg and one prismatic joint connecting the upper and lower legs. Three intuitive control targets were identified: torso pitch, leg length, and leg angle. During the stance phase, the torso pitch is regulated to upright; leg length first retracts then extends so that the monopod can jump. During the flight phase, the leg first retracts for foot clearance and then extends to initiate contact with the ground. Leg angle is adjusted based on the difference between desired velocity and actual velocity: when the speed is too slow, the leg will pitch backward so that the body can be pushed forward during the stance phase and vice versa. This method enabled the 2D monopod to jump agilely without falling; similar methods were implemented later on a 3D monopod[12] and a biped[13]. Although this method is model-free and stability was achieved by parameter tuning, its result is successful and it highlights foot placement as one of the basic mechanisms of dynamic balancing.

The Linear Inverted Pendulum Model (LIP) was proposed by Kajita, which initially assumes a point mass with massless leg travel on a horizontal plane[14], and later allows a weaker assumption where the CoM travels on an inclined plane, and angular momentum about the CoM is constant[15]. A linear system can be obtained with this setup and a closed-form solution can be found for this system, which provides a straightforward way to plan foothold position and step time. Although this simplified model ignores some important aspects of biped walking, such as varying angular momentum and CoM height, it captures the basic dynamics of walking and provides a closed-form solution, and thus is widely used in the biped community[16, 17, 18].

The Spring Loaded Inverted Pendulum(SLIP) was proposed by Blickhan to simulate human and animal running and hopping[19]. Despite its simplicity, the model generates motions that are close to data collected from humans and animals.

By designing certain virtual constraints, the stability of a high dimension nonlinear model could be analyzed in a low dimension space with no simplification necessary. The dynamics in the low dimensional space is called the Zero Dynamic[20]. Grizzle et al. proposed a Hybrid Zero Dynamic framework [21] in which 2D robots' stability could be analyzed in a two-dimension space. The original holonomic constraints were extended to non-holonomic constraints by Griffin in [22] and the phase used to define constraints was extended from mechanical phase to time phase by Da in Generalized Hybrid Zero Dynamic(GHZD) [23].

The idea of Zero Moment Point was first introduced in 1968 by Miomir Vukobratović[24]. It specifies a point in the support polygon at which the horizontal moment is zero. ZMP is an important variable indicating the stability of a robot, especially for bipeds whose support polygons are small. When ZMP is inside the support polygon, the contact between foot and ground is considered solid; when ZMP is at the edge of the support polygon, the contact foot will begin to roll. Using this method, a controller designed for walking generally tries to avoid generating a ZMP outside of the support polygon unless foot rolling is desired. LIP models are sometimes used in conjunction with ZMP, where a ZMP path is planned and the corresponding CoM trajectory of a LIP model will be used as a reference for the real robot[25].

Whole Body Control focuses on how to coordinate multiple tasks in a robot with redundancy[26]. In [27], Kajita calculates the reference joint velocities for the humanoid HRP-2 to simultaneously obtain desired centroidal linear and angular momentum. The resulting controller enables HRP-2 to kick a ball and walk with arms swinging naturally.

A middle ground found between full order dynamics and a simplified model is Centroidal Dynamics. Centroidal Dynamics describes the linear and angular momentum at the CoM of a robot. The rates of those momenta are decided by the gravitational force and force generated at contact. Dai et al. [28] proposed an optimization method that has Centroidal Dynamics and full kinematics constraints. The resulting optimization problem can be solved much faster than one using a full order dynamic model while still respecting the real robot's dynamics. However, the motor torque could not be constrained in this setup.

Reinforcement learning for bipedal locomotion has been studied in simulations[29, 30] and with robots with large feet[29] for about two decades. More recently RL algorithms have been applied to robots with small feet in experiments. Xie et al trained a neural network for bipedal robot Cassie in simulation and then successfully transferred it to a physical robot[31]. Li et al. designed a reinforcement algorithm that tracks pre-optimized periodic gaits[32]. Siekmann et al. designed a two-layer network that takes phase command as input and produces reference joint trajectory, which enables Cassie to perform multiple two-beat gaits, including walking, hopping, skipping, and switching between those gaits[33].

1.3 Overall Objective of this Dissertation

The goal of this research is to design a robust controller for a three-dimensional bipedal robot that allows it to move quickly on mildly varying ground such as a sidewalk, pass over challenging terrain without falling, and reject large external disturbances.

The specific objectives are to:

- Incorporate real robot model's dynamics and kinematics into optimization, which generates feasible gait library for a bipedal robot, enabling it to walk at different speeds. Develop techniques for standing control and implement them in experiments. Enable Cassie to ride a Segway.

- Study the relation between pendulum models, real models, and Hybrid Zero Dynamics. Demonstrate the commonality in the HZD across different robots when specific zero dynamic states and virtual constraints are chosen.

- Utilize the commonality in HZD, plan motions for a real robot with Pendulum models, and accurately obtain desired evolution for chosen zero dynamic states. Demonstrate agile and robust motion in experiments with Cassie.

The structure of this dissertation is as follows: Chapter IIintroduces the robot models and hardware we use, and reviews several related control methods. Chapter III describes the work on generating a gait library with optimization on a full order model of a real robot and corresponding controller design and implementation. Chapter IV discusses the relation between pendulum models, real biped models, and zero dynamics. Chapter V describes the implementation of a controller which utilizes the properties of Angular momentum about contact point, plans foot placement for a real robot with a pendulum model, and obtains desired states with high accuracy. Chapter VI introduces a similar but reformed controller that enables the robot to run. Chapter VII concludes the dissertation and discusses future work.

CHAPTER II

Background

In this chapter, we introduce the robot testbeds we work on, the mathematical models we use, and several related control methods.

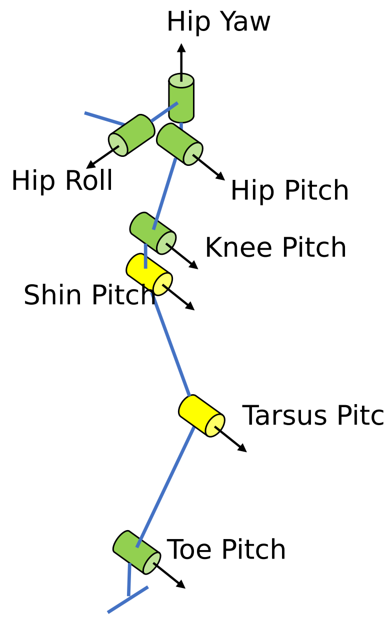
2.1 Robots

In this section, we introduce the configurations of two robots, on which we will test our control algorithms. Cassie, a 3D robot with mass concentrated near the robot's CoM, imitating birds; and Rabbit, a 2D robot with mass distributed throughout its legs, more similar to humans.

2.1.1 Cassie

Cassie is a three-dimensional bipedal robot designed and manufactured by Agility Robotics. It went through several prototypes before the design was finalized in 2017. It is all electric and can walk for approximately four hours on a single battery charge. The robot's morphology and name are inspired by the *Cassowary*, a flightless bird similar to an ostrich. It weighs 32 kg and it's height is about 1 m. There are 7 joints

(a) Cassie Robot



(b) Left Leg Can Model

Figure 2.1: Cassie Robot

on each leg. The robot's generalized coordinates are taken as

$$\begin{aligned} q := [&q_x, q_y, q_z, q_{\text{yaw}}, q_{\text{pitch}}, q_{\text{roll}}, \\ &q^{\text{L}}_{\text{hip roll}}, q^{\text{L}}_{\text{hip yaw}}, q^{\text{L}}_{\text{hip pitch}}, q^{\text{L}}_{\text{knee}}, q^{\text{L}}_{\text{knee spring}}, q^{\text{L}}_{\text{ankle}}, q^{\text{L}}_{\text{toe}}, \\ &q^{\text{R}}_{\text{hip roll}}, q^{\text{R}}_{\text{hip yaw}}, q^{\text{R}}_{\text{hip pitch}}, q^{\text{R}}_{\text{knee}}, q^{\text{R}}_{\text{knee spring}}, q^{\text{R}}_{\text{ankle}}, q^{\text{R}}_{\text{toe}}]^{\mathsf{T}}. \end{aligned} \tag{2.1}$$

A can model of Cassie is shown in Fig. 2.1. The yellow cans in the figures denote joints that are not actuated but constrained by two four-bar-linkages, each with one link being springs(not shown in the can model but can be seen in Cassie's photo). The springs have high stiffness and if they are assumed rigid, the coordinates would have the following relation:

$$\begin{cases} q_{\text{ knee spring}} = 0 \\ \qquad q_{\text{ ankle}} = -q_{\text{ knee}} + 13^\circ \end{cases}$$

Removing the joints constrained by springs, Cassie still has 5 free joints on each leg. These free joints enable the foot to move freely in 3D Cartesian space, change the

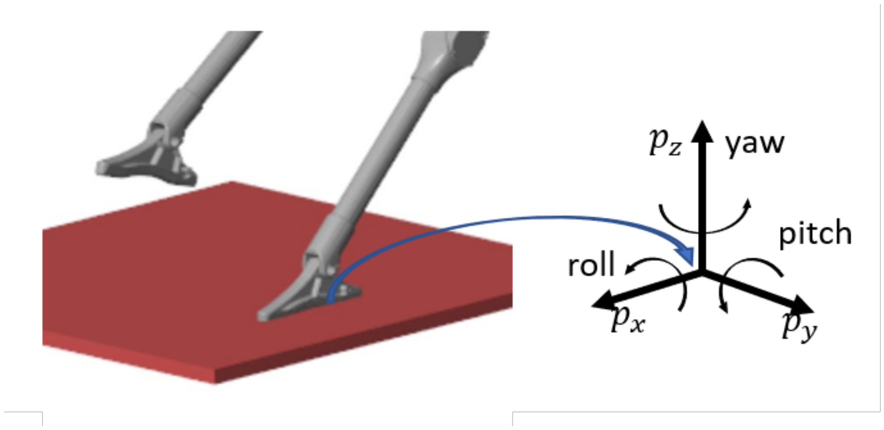


Figure 2.2: Cassie's feet and its constraint. When it is in contact ground, all motions are constrained except yaw.

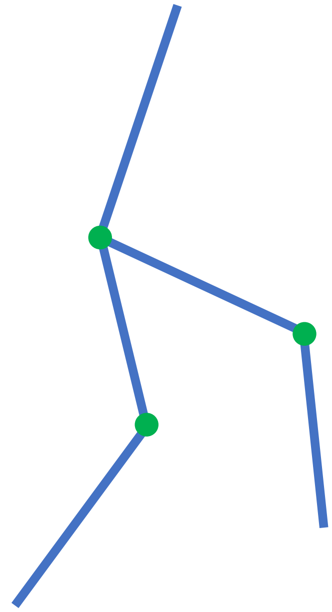
direction of the toe with respect to the horizontal plane, and tilt upward or downward. These degrees of freedom are enough for Cassie to move agilely in a 3D space.

Cassie has blade-shaped feet and is able to stand on two feet but could not stand on only one foot. The length of its foot is about 17 cm. The foot-body ratio is similar to human, making it have limited ability to balance itself with ankle torque. This makes Cassie an underactuated robot, distinct from robots with large feet.

2.1.2 Rabbit

Rabbit, shown in Fig. 2.3, is a planar bipedal walking robot made in 2002 with joint efforts from several labs and universities in France and United States[34]. The robot is 1.4m tall and weighs 32kg. It has five links, one for the torso and two for each leg. On each leg there are two actuated joints located at the hip and knee. This morphology enables the robot to walk in the longitudinal direction. A radial bar attached at the hip limits the movement in the lateral direction. The robot is designed to have no feet, literally, it has point feet, with the goal of demonstrating actuated ankles are not necessary for many styles of locomotion and inspiring feedback stabilization methods that can address underactuation.

(a) Cassie Robot



(b) Rabbit Stick Model

Figure 2.3: Rabbit Robot

2.2 Dynamic Models

One characteristic of biped dynamics is that the dynamics model changes as foot contact changes. The dynamics can be roughly classified into a continuous model when the foot contact is not changing, and an impact model when a swinging foot touches the ground, during which there is a sudden change in the states because of the force impulse on the foot.

2.2.1 Continuous model

Given generalized coordinates q and their time derivative $\dot{q}$, Lagrangian Equations of Motion are often used to describe the dynamics of a robot. For a pinned model, where the stance foot is considered as "pinned" on the ground, we write the EoM as:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu, \tag{2.2}$$

where D is the Inertia Matrix, C is the Centrifugal and Coriolis Matrix, G arises from gravity, B is the input distribution matrix and u is the input.

For a floating-base model, where the robot is assumed floating in the air and the ground contact is added as explicit constraints, the EoM can be written as:

$$\begin{cases} D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu + J(q)^{\top}F \\ J(q)\ddot{q} + \dot{J}(q,\dot{q})\dot{q} = 0, \end{cases} \tag{2.3}$$

where J is the Jacobian of the constraints, and the second line of the model indicates that the second time derivative of the constraints is zero. Equation (2.3) can also be written in matrix form

$$\begin{bmatrix} D & -J^{\top} \\ J & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F \end{bmatrix} + \begin{bmatrix} C\dot{q} + G \\ \dot{J}\dot{q} \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} u \tag{2.4}$$

where the first matrix is invertible as long as D is invertible and J is full rank.

2.2.2 Impact Model

The transition from single support to double support is captured by the height of the swing foot from the ground decreasing to zero.

Instantaneous impacts are modeled through a discrete map that results in a discontinuity in the velocity of the system $\dot{q}^-$ just before impact and the velocity of the system $\dot{q}^+$ just after impact, while the positions do not change [35]. Moreover, just after impact, the former swing foot is assumed to satisfy the same constraints as those imposed on the stance foot. Letting $c_{\mathrm{L}}(q)$ denote the corresponding holonomic constraint, the pre- and post-impact velocities then satisfy

$$\begin{bmatrix} D(q) & -J_{\mathrm{L}}^{T}(q) \\ J_{\mathrm{L}}(q) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^{+} \\ \delta F_{\mathrm{L}} \end{bmatrix} = \begin{bmatrix} D(q)\dot{q}^{-} \\ 0 \end{bmatrix}, \tag{2.5}$$

where δF_{L} is the vector of contact impulses. Because the Jacobian $J_{\mathrm{L}}(q)$ has full row rank and $D(q)$ is positive definite, the left hand side of (2.5) is invertible. Projecting the solution of (2.5) to the velocity components defines the impact map,

$$\dot{q}^{+} =: \Delta_{\mathrm{R}\to\mathrm{L}}(\dot{q}^{-}). \tag{2.6}$$

2.3 Control Approaches Related to this Work

2.3.1 Hybrid Zero Dynamics

For a system with multiple states, after several outputs are driven to zero by inputs, the remaining dynamics of the system are called zero dynamics[36, 37]. One theory that has been widely adopted as a framework for designing gaits for bipedal robots and analyzing their stability is Grizzle et al's theory of Hybrid Zero Dynamics (HZD)[35, 21] . We can think of driving those outputs to zero as adding virtual constraints on a robot. Virtual constraints are imposed by inputs, unlike physical constraints which are imposed mechanically. In the early stage of HZD theory, virtual constraints were always holonomic constraints. The constraints were often written in the following form:

$$h(q) = 0 \tag{2.7}$$

Usually, the dimension of $h(q)$ is the same as the dimension of inputs. For a 2D point contact robot model like Rabbit, such constraints would leave two states free. Here we denote the two free states $[\xi_1, \xi_2]$, and the zero dynamics for these two states is:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = f_{\mathrm{zero}}(\xi_1, \xi_2) \tag{2.8}$$

The choice of $[\xi_1, \xi_2]$ is decided by the controller designer. Usually, they are chosen to be states that are not strongly actuated. This can be CoM position and velocity, or the angle of stance leg in the world frame and its angular velocity. ξ_2 does not have to be the time derivative of ξ_1. For example, ξ_1 can be stance leg angle, while ξ_2 is angular momentum about the contact point. The proof of why (2.8) still holds in the last case can be found in [22].

The design of virtual constraints $h(q)$ decides the zero dynamics for $[\xi_1, \xi_2]$, and once the virtual constraints are decided and imposed, $[\xi_1, \xi_2]$ evolves independently on the low dimensional zero dynamics manifold defined by virtual constraints $h(q)$. This greatly facilitates the analysis of robot stability. Hybrid here refers to the invariant of the zero dynamics manifold at impact, i.e. $h(q)$ and $\dot{h}(q, \dot{q})$ equals to zero before and after impact. At impact, the zero dynamics is described by:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \Delta(\xi_1, \xi_2) \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \tag{2.9}$$

Equations (2.8) and (2.9) define the classic HZD for bipedal robots. By choosing proper virtual constraints, we could obtain the desired Zero Dynamics, which in return could make the robot attain desired behavior. It is proved that if the periodic orbit of $[\xi_1, \xi_2]$ is stable, the periodic orbit of the full states is stable, as long as $h(q)$ equals zero, or it converges to zero sufficiently quickly[35]. Proper virtual constraints are usually found by an optimization program, with the goal of making the robot walk periodically and minimizing the consumed energy.

Hybrid Zero Dynamics successfully enable robots to walk with robust gaits with provable stability. However, holonomic constraints limit its potential because the joints of a robot must follow a fixed trajectory. When rejecting disturbance, the robot joints can only move on the fixed trajectories faster or slower and cannot modify the trajectory to tackle the disturbance. This limits the controller's ability to reject disturbance. A non-holonomic constraints method was proposed by Griffin et al. [22], in which angular momentum about the contact point (L) is included in the virtual constraints:

$$h(q, L) = 0 \tag{2.10}$$

while the form of the zero dynamics still remains the same as when using holonomic constraints (2.8) (2.9).

This form of virtual constraints gives the controller the choice to modify the trajectory when needed. Robust optimization is run to obtain a set of virtual constraints which not only minimize the cost but also maximize the converging speed to a periodic orbit when disturbances occur. The nonholonomic constraints-based controller successfully enables the bipedal robot MARLO to walk through uneven terrains, specifically, terrain which otherwise would destabilize a controller based on holonomic constraints.

With only mechanical constraints, walking in place has been a challenge: How can we make the swing foot move up and down when there is no monotonically increasing mechanical phase? In an attempt to answer this, Da et al [23] included time in the virtual constraints:

$$h(q, \dot{q}, \tau) = 0 \tag{2.11}$$

where τ is the time elapsed since the beginning of the current step. And the corresponding zero dynamics becomes

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = f_{\text{zero}}(\xi_1, \xi_2, \tau). \tag{2.12}$$

Including time in the virtual constraints enabled MARLO to walk in place. Furthermore, transition gaits between different periodic orbits are also generated by optimization, and a large zero dynamic manifold is formed with machine learning techniques. Multiple zero dynamic manifolds (virtual constraints) are integrated into a gait library and are chosen based on the command given by the human opera-

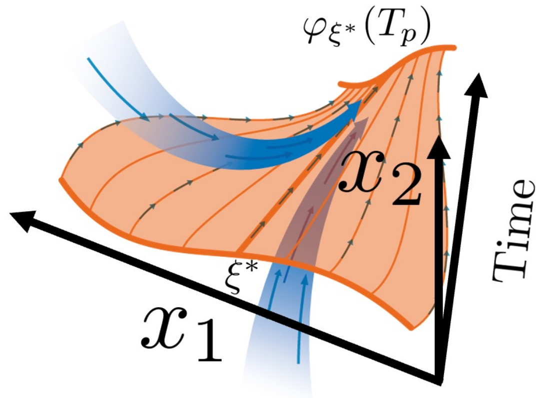


Figure 2.4: Submanifold defined in G-HZD. $x_1 = [\xi_1, \xi_2]$ is the vector of weakly actuated states, x_2 is the vector of strongly actuated states. With a low-level controller driving the states to this submanifold, the robot is guaranteed to be stable.

tor, which enables Marlo, with one controller, to walk at different speeds and reject varying disturbances, including uneven ground and strong pushes.

2.3.2 Linear Inverted Pendulum Model

Biped researchers have been studying walking through inverted pendulum models for a long time because of its simplicity and the widely held belief that walking is a sequence of repeated actions falling and catching oneself. Kajita et al proposed a Linear Inverted Pendulum Model, for which a simple linear dynamic can be found[15, 38]. The 2D case of the model is shown in Figure 2.5. It assumes the time derivative of angular momentum about the Center of Mass ($\dot{L}_c$) is zero, and the Center of Mass (CoM) travels on a straight line.

We will briefly discuss how to derive LIP dynamics by analyzing ground reaction force. Assumption $\dot{L}_c = 0$ indicates that Ground Reaction Force F points toward the CoM. So we have

$$\frac{x}{z} = \frac{F_x}{F_z} = \frac{\ddot{x}}{\ddot{z}} \tag{2.13}$$

Assuming the CoM travels on a straight line means the following relation holds,

$$z = kx + H. \tag{2.14}$$

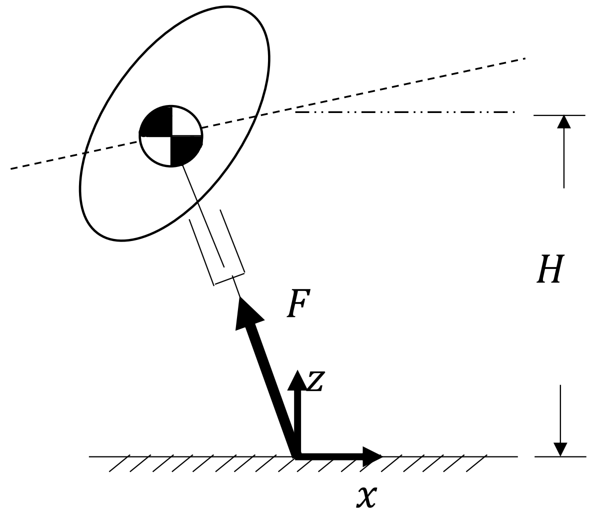


Figure 2.5: 2D Linear Inverted Pendulum(LIP) model. It assumes angular momentum about CoM is 0 and CoM travels on a straight line..

Combining (2.13) and (2.14), we obtain the LIP model dynamics,

$$\ddot{x} = \frac{g}{H}x. \tag{2.15}$$

This LIP model is popular for having a closed-form solution and thus facilitating fast step planning. However, the assumption $\dot{L}_c = 0$ is too strong for a real robot. In some controllers[39], L_c has been planned to be close to zero to reduce LIP model error. Others have used a moving ZMP, or ankle torque to offset the effect of $\dot{L}_c$.[40]

CHAPTER III

Gait Library with Full order Model

Although the simplified models for bipedal robots found in the literature are easy to analyze, they do not accurately capture a real robot's dynamics and thus fail to plan behavior that is feasible, robust, and energy-efficient on real hardware. Since real models are high dimensional, nonlinear, and hybrid, they are too complicated to analyze with traditional control methods. Here we utilize optimization with the full order model to generate optimal gaits for Cassie. The generated gaits are dynamically and kinematically feasible, respect multiple real-world constraints such as ground friction cone and torque limits, and are energy-efficient.

Because the optimization with a full order model is time-consuming, we choose to run the optimization offline and store the result in a gait library. The gait library includes gaits at different walking speeds. A controller is designed to implement these gaits in simulation and in experiments. We also design a standing controller which enables Cassie to maintain balance on swaying ground and ride a Segway.

3.1 Walking Controller

This section presents the initial walking gait library controller implemented on Cassie. The control design is based on the method of *virtual constraints*, a *gait library* limited to the sagittal plane, and *leg-angle adjustment* in the sagittal and frontal planes [41].

3.1.1 Virtual Constraints

Virtual constraints are functional relations among the generalized coordinates of the robot that are asymptotically imposed on the system through feedback control. In particular, the virtual constraints are expressed as outputs of the model (5.23) in the form

$$y = h(q, \tau, \alpha) = h_0(q) - h_d(\tau, \alpha), \tag{3.1}$$

where h_0 specifies the quantities being regulated, h_d encodes their desired evolution, and α is a matrix of real coefficients that parameterizes the spline h_d. The phasing variable, τ, satisfies

$$\begin{aligned} \dot{\tau} &= \frac{1}{T}, & (q; \dot{q}^-) &\notin \mathcal{S}_{\mathrm{R}\to\mathrm{L}} \cup \mathcal{S}_{\mathrm{L}\to\mathrm{R}} \\ \tau^+ &= 0, & &\text{otherwise,} \end{aligned} \tag{3.2}$$

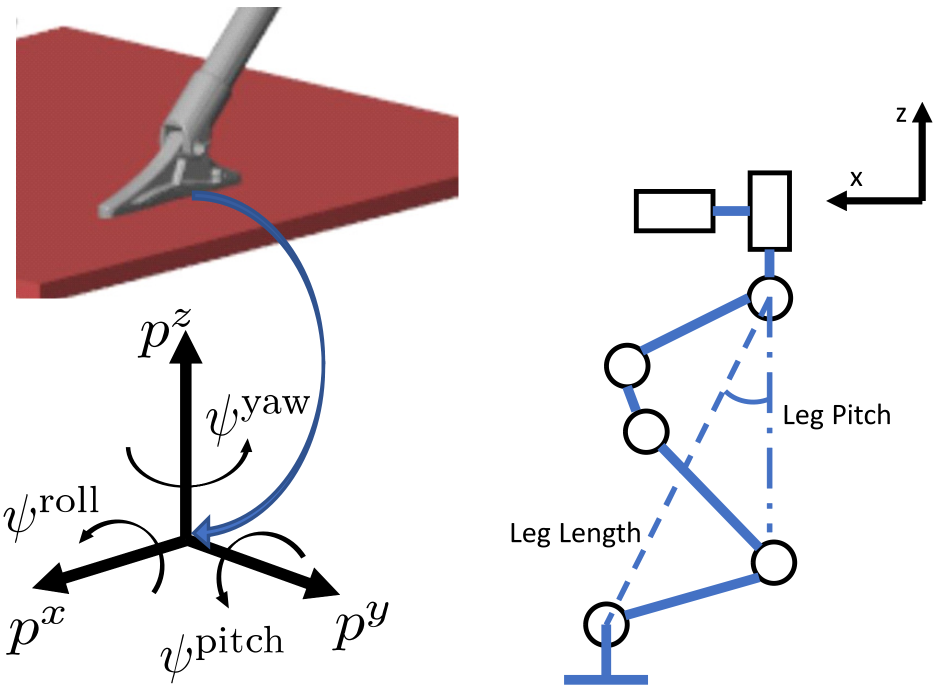
where T is the nominal step duration.

A controller is then designed with the objective of zeroing the outputs, i.e., $y \equiv 0$, thereby achieving the virtual constraints. The zeroing of the output value will be at best accomplished asymptotically, and in practice, on a physical robot, only approximately.

3.1.2 Choice of What to Control

The most direct choice for the regulated quantities, h_0, would be the actuated joints of Cassie, which are a subset of the body coordinates. On previous planar robots, such as Rabbit and MABEL [42, 43], this made sense because these robots had a simple (human-inspired) morphology and the control objectives could be associated in an intuitive manner with hip and knee angles. On our lab's first 3D biped, MARLO, the legs, and hence the actuated coordinates in the sagittal plane, were associated with four-bar linkages, which gave rise to synthesizing coordinates that were associated with a virtual leg connecting the hip to the end of the leg [22]. MARLO also had a rather tall torso that provided adequate inertia about the roll axis so that adjustments by the stance leg hip motor to maintain the torso roll angle approximately zero would not cause oscillations on the swing leg roll angle. For the land-bird-inspired Cassie, if one is not a biologist, the actuated joints have limited physical meaning with respect to the walking behavior of the robot. Hence, we choose instead to regulate torso orientation, stance and swing leg lengths, swing leg orientation, and swing foot pitch angle. We find these quantities to be rather universal across bipedal platforms (whether human or bird inspired) and directly relatable to gait outcomes. For example, leg lengths are directly related to the height of the torso and foot clearance; swing leg pitch and roll angles at impact are commonly used in bipeds to regulate walking speed, and the yaw angle of the swing leg sets the direction of the robot for the next step. Moreover, because Cassie is roughly a basketball-sized sphere with one meter-long legs attached to it, its (spherical) torso provides little "mechanical filtering" between the motion of the stance and swing legs.

With this in mind, we define the following nine outputs, with the stance foot pitch



(a) (b)

Figure 3.1: (a) The coordinates on the feet are shown. When a foot is in contact with the ground, there are five independent constraints leaving only the roll angle of the foot free. (b) The virtual leg is the dotted line from the hip to the top of the foot; its length is called Leg Length. The relative leg pitch is the angle of the virtual leg relative to the hip while absolute leg pitch is the relative angle plus the pitch angle of the torso.

angle left passive[1],

$$h_0(q) = \begin{bmatrix} q_{\text{roll}} \\ q_{2\ \text{st}} \\ q_{\text{pitch}} \\ q_{\text{LL st}} \\ q_{\text{LR sw}} \\ q_{2\ \text{sw}} \\ q_{\text{LP sw}} \\ q_{\text{LL sw}} \\ q_{\text{FP sw}} \end{bmatrix} \begin{pmatrix} \text{torso roll} \\ \text{stance hip yaw} \\ \text{torso pitch} \\ \text{stance leg length} \\ \text{swing leg roll} \\ \text{swing hip yaw} \\ \text{swing leg pitch} \\ \text{swing leg length} \\ \text{swing foot pitch} \end{pmatrix}, \tag{3.3}$$

[1]For now, it is simply observed that leaving it passive avoids "foot roll"; when walking on soft sand, we will come back to this feature.

where, when the right leg is stance and the left leg is swing,

$$\begin{aligned}
q_{\text{LL st}} =& \sqrt{0.5292\cos(q_{4\text{R}} + 0.035) + 0.5301} \\
q_{\text{LR sw}} =& q_{\text{roll}} + q_{1\text{L}} \\
q_{\text{LP sw}} =& -q_{\text{pitch}} + q_{3\text{L}} \\
-\arccos(& \frac{0.5(\cos(q_{4\text{L}} + 0.035) + 0.5292)}{\sqrt{0.5292\cos(q_{4\text{L}} + 0.035) + 0.5301}}) + 0.1 \\
q_{\text{LL sw}} =& \sqrt{0.5292\cos(q_{4\text{L}} + 0.035) + 0.5301} \\
q_{\text{FP sw}} =& -q_{\text{pitch}} + q_{7\text{L}} + 1.1.
\end{aligned} \tag{3.4}$$

The forward kinematics of leg length and leg pitch are calculated based on the configuration of the robot as shown in Fig. 3.1b. For clarity, leg pitch refers to the absolute pitch angle of the virtual leg when torso roll and yaw are zero; leg roll is defined in a similar manner.

For later use in control implementation, we note that the to-be-regulated quantities in (3.4) can be expressed in terms of the *actuated joints* via

$$\begin{aligned}
q_1 =& q_{\text{LR}} - q_{\text{roll}} \\
q_3 =& q_{\text{LP}} + q_{\text{pitch}} \\
&+ \arccos\left(\frac{0.9448\ q_{\text{LL}}^2 - 0.0284}{q_{\text{LL}}}\right) - 0.1 \\
q_4 =& \arccos\left(1.8896\ q_{\text{LL}}^2 - 1.0017\right) - 0.035 \\
q_7 =& q_{\text{FP}} + q_{\text{pitch}} - 1.1,
\end{aligned} \tag{3.5}$$

where the distinction between stance and swing has been dropped.

3.1.3 Gait Library Generation

The desired evolution of the virtual constraints is defined by h_d in the output equation (3.1). This function is constructed using linear interpolation of a discrete library of gaits, each encoding a particular forward walking speed. Here, seven gaits were generated where the average velocity in the sagittal plane, $\bar{v}_x$, ranged from -0.5 m/s to +1.0 m/s in 0.25 m/s increments. We assume that the virtual constraints of each of these "open-loop" gaits have a desired trajectory, $h_d^i(\tau, \alpha_i)$, that is parameterized by a set of 5$^{\text{th}}$-order Bézier polynomials with the corresponding matrix of coefficients denoted as α_i. The step time for all gaits was chosen to be a constant. Trajectory optimization is then used to independently solve for each α_i.

The nonlinear optimization problems were constructed and solved using FROST [44], which internally uses the direct-collocation trajectory optimization framework developed by Hereid et. al. [45]. Each hybrid optimization was performed over two domains (right stance then left stance), where the following cost function was minimized:

$$\begin{aligned}\text{Domain Cost} = \int_{\tau=0}^{\tau=1} \Big(& ||u||^2 + c\,|q_{pitch}|^2 + c\,|q_{roll}|^2 \\ & + c\,|q_{1\text{L}}|^2 + c\,|q_{2\text{L}}|^2 + c\,|q_{1\text{R}}|^2 + c\,|q_{2\text{R}}|^2 \Big)\, d\tau.\end{aligned}$$

The addition of the torso pitch/roll and the hip roll/yaw angles into the cost function (multiplied by a large weight, $c = 10,000$) guides the optimizer to find gaits with minimal roll and yaw movement. Constraints are placed on the optimization problem to ensure that the optimized gait is periodic over two steps and that the left and right stances are symmetric[2]. Torque, joint angle, and joint velocity limits were imposed to ensure that the gait can be physically-realized on the actual robot. Additional

[2]Due to the enforced symmetry, it is possible to write this optimization problem using a single domain. However, the general two-step formulation allows for the future design of non-symmetric gaits through the removal of this particular constraint.

constraints are outlined in Table 3.1.

Average sagittal velocity, $\bar{v}_x$	$= v_i$ m/s
Average lateral velocity, $\bar{v}_y$	= 0 m/s
Step time	= 0.4 s
Torque for stance foot pitch	= 0 Nm
Friction cone, μ	¡ 0.6
Mid-step swing foot clearance	¿ 0.15 m
Absolute swing foot pitch	= 0 rad
Distance between feet	¿ 0.2 m
Distance between pelvis and stance foot	$\in$ (0.5, 1) m
Swing foot velocity on impact (x and y)	= 0 m/s
Swing foot velocity on impact (z)	$\in$ (-1, 0) m/s

Table 3.1: Constraints used in gait optimizations. For each of the seven optimizations, the average sagittal velocity was constrained to a different value between -0.5 and 1 m/s.

Each of the 7 optimization problems yields a single parameter matrix, α_i, and takes approximately 3 min to solve using IPOPT in MATLAB.

Remark: *A C++ implementation of the optimization problem formed by FROST has been posted on GitHub [46]. It allows parallel computation of the gaits and cloud-based gait optimization.*

3.1.4 Approximately Implementing the Virtual Constraints

If the overall dynamic model and joint angular velocity estimates were sufficiently accurate, we could implement the virtual constraints via input-output linearization. Indeed, the outputs (3.1) have relative degree two [47] and the row rank of the decoupling matrix is full rank on the control-design model.

On the actual robot, however, the power amplifiers, motor dynamics, network delays, and walking surface are not adequately characterized to allow model-based torque control (the mechanical model itself is not the main source of uncertainty). Consequently, the virtual constraints are approximately imposed through decoupled

PD controllers, as in [42, 43, 22]. To do this, (3.5) is used to rewrite (3.1) as

$$\widetilde{y} = \widetilde{h}_0(q) - \widetilde{h}_d(\tau, q_{\text{pitch}}, q_{\text{roll}}, \alpha), \tag{3.6}$$

with (right leg in stance)

$$\begin{aligned} \widetilde{h}_0(q)^\top = [&q_{\text{roll}}, q_{2\text{R}}, q_{\text{pitch}}, q_{4\text{R}}, \\ &q_{1\text{L}}, q_{2\text{L}}, q_{3\text{L}}, q_{4\text{L}}, q_{7\text{L}}]^\top \end{aligned} \tag{3.7}$$

and

$$\widetilde{h}_d(\cdot) = \begin{bmatrix} h_{d\ 1}(\cdot) \\ h_{d\ 2}(\cdot) \\ h_{d\ 3}(\cdot) \\ \arccos\left(1.8896\ \ [h_{d\ 4}(\cdot)]^2 - 1.0017\right) - 0.035 \\ h_{d\ 5}(\cdot) - q_{\text{roll}} \\ h_{d\ 6}(\cdot) \\ h_{d\ 7}(\cdot) + q_{\text{pitch}} + 0.1 \\ \quad + \arccos\left(\frac{0.9448\ \ (h_{d\ 8}(\cdot))^2 - 0.0284}{h_{d\ 8}(\cdot)}\right) \\ \arccos\left(1.8896\ \ [h_{d\ 8}(\cdot)]^2 - 1.0017\right) - 0.035 \\ h_{d\ 9}(\cdot) + q_{\text{pitch}} - 1.1 \end{bmatrix}. \tag{3.8}$$

Remark: *Though not proven here,* (3.6) *implements the same virtual constraints as* (3.1) *and* (3.4)*; one is zero if, and only if, the other is zero.*

In (3.7) and (3.8), the outputs are ordered so that they correspond to the first four actuators on the stance leg followed by the five actuators on the swing leg. Recall that the torque on the stance foot is set to zero. The virtual constraints are approximately

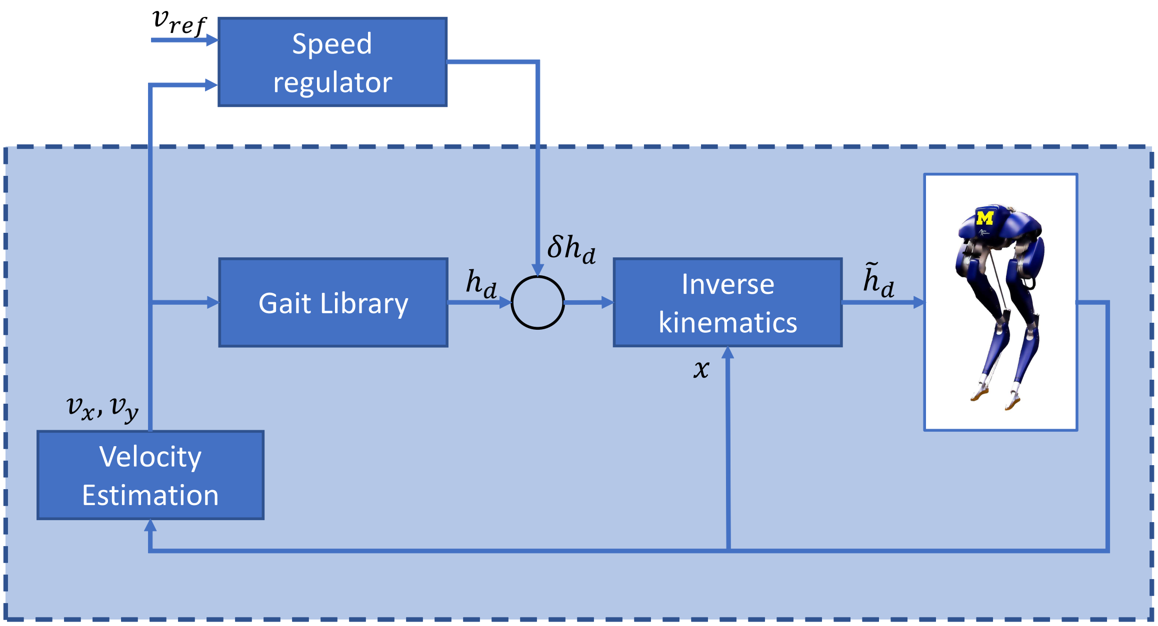


Figure 3.2: Control Diagram for Walking. The feedback loop implementing the virtual constraints and the gait library (blue box) maintains the robot's posture and synchronizes the legs for walking. Moreover, the resulting closed-loop system renders the dynamics of the center of mass velocity close enough to that of an inverted pendulum that it can be regulated by adjusting the pitch and roll angles of the swing leg.

zeroed with a classical PD controller,

$$u = -K_P \widetilde{y} - K_D \dot{\widetilde{y}}, \tag{3.9}$$

where the 9×9 matrices K_P and K_D are diagonal.

Remark: *The torso pitch and roll angles are world frame coordinates. When the stance foot is firmly on the ground (i.e., not slipping), they can be controlled through the hip motors. The remaining outputs are directly actuated. With springs ignored, the output has vector relative degree two.*

3.1.5 Gait Library and Stabilization by Leg Angle Adjustment

The *Gait Library* is an interpolation of the seven discrete gaits into a continuum of gaits valid for $-0.5 \leq v_x \leq 1.0$ m/s. The interpolation parameter is the robot's filtered sagittal velocity. The implementation of the gait library is done exactly as in [41, Eq. (8)-(10)] and does not introduce any new parameters into the controller.

With the gait library implemented, the closed-loop system is unstable in the sense that the $(\dot{x}, \dot{y})$ Cartesian coordinates are approximate integrators; see [41, Sect. III-C] for the explanation. Leg angle adjustment is added to stabilize the closed-loop system. The implementations for longitudinal and lateral velocity stabilization are based on [41, Eq. (13) and (17)]. These controllers add four more control parameters. The overall control strategy is shown in Fig. 3.2.

3.1.6 Parameter Tuning

The right and left legs of the robot are sufficiently symmetric that control parameters for the left and right legs are the same. The controller was implemented in Realtime Simulink and the parameters tuned by hand on a SimMech model provided to us by Agility Robotics. A process for tuning the 18 joint-level PD parameters and the 4 leg-angle PD parameters on the robot is posted with the code on GitHub.

3.1.7 Experiments

A first version of the walking controller was implemented on Cassie Blue six weeks after arrival on campus and was demonstrated to the Associated Press (AP) on October 23rd, 2017 [48]. On June 2, 2018, we damaged a leg on Cassie Blue and sent her back for repairs. While the robot was in the shop, Agility Robotics upgraded the hip roll and yaw joints to match those on their current production model, significantly reducing friction in them. At the same time, Agility also modified the MATLAB environment in which a user's controller is implemented, breaking our controller. Be-

(a) (b) (c) (d)

Figure 3.3: Cassie Blue walking on various (unmodeled) terrains.

cause we would be soon modifying the robot with the addition of a 15 kg torso, we did not spend much time re-tuning the controller.

The remainder of the section discusses some of the many terrains on which the robot has been challenged to operate over the past 11 months as documented in [49]. The dates of the experiments are noted below. In each experiment, the robot is being directed by an operator via an RC Radio with commands “stand quietly” or “walk”. When walking, the robot is sent desired $\bar{v}_x^{tg}$, $\bar{v}_y^{tg}$, and turn rate.

3.1.7.1 Initial Testing in the Laboratory

After the controller was successfully working in closed-loop with the SimMech model, it was transferred to the robot and the PD gains tuned over a period of a few days. During this process, an overhead gantry was used to catch the robot in case of a fall. The gains were initially tuned for walking in place. Once that milestone was achieved, walking at various speeds came quickly. A typical limit cycle is shown in Fig. 3.4.

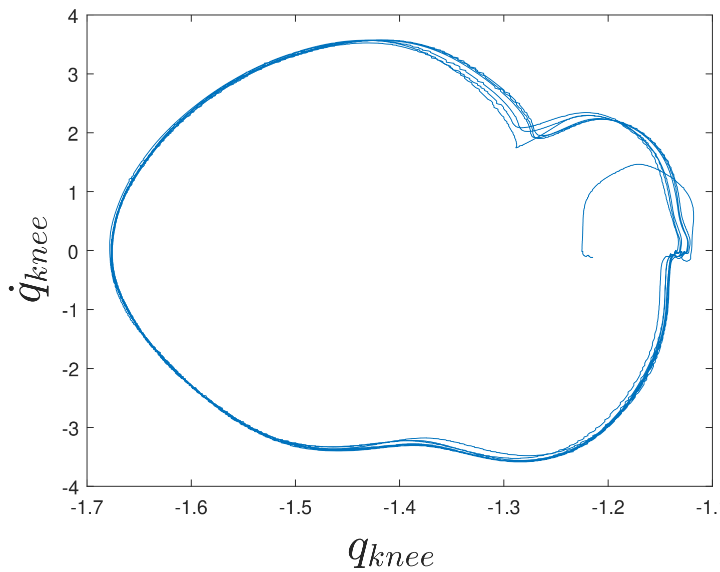


Figure 3.4: The phase portrait of the left knee when Cassie is walking in place, starting from a standing position. The units are rad and rad/s.

3.1.7.2 North Campus Grove

Cassie Blue was taken outdoors for the first time and demonstrated to the Associated Press [48] on 23 October 2017. Initially, a safety gantry was used. After walking on a sidewalk with no difficulty, Cassie was released from the gantry. After initial walking on a level concrete area, Cassie was steered onto the grass. Due to a nervous operator hitting the wrong button on the RC-controller, the robot sped up and had to be stabilized by a researcher. The softer nature of the grass caused no difficulties in detecting foot impacts via spring deflection. For the next 11 minutes, the robot was steered on and off grass and hard surfaces, onto sloped grass surfaces, turned, and walked in place. When the experiment happened to bring the robot and researchers near a grassy knoll, the decision was made to see if the robot could handle it. The robot unexpectedly sped up when heading up the slope, leaped over a bench, and fell at least 1.5 m onto the concrete. This put a slight dent in the battery, but caused no other observable damage as the robot was quickly rebooted and walked in place. This ended the experiment.

3.1.7.3 Waxed Floors and Snow

Cassie Blue was taken to the UM Dental School on 11 December 2017 at the invitation of Dean Laurie McCauley. The video can be found here [50]. The robot handled well on surfaces with reduced friction. An unplanned bump into a pillar and walking in the snow are shown here [49].

3.1.7.4 Controlled Burn for Native Grasses

On 22 April 2018, a controlled burn was conducted on the UM campus to promote the growth of native grasses. After clearing it with the personnel conducting the burn, Cassie Blue walked over sloped ground, in heavy smoke, and over short burning grass, branches, and leaves [51]. The robot never fell. While the exercise demonstrated our general confidence in the robot's controller, it was done to drive home the fact that a battery-powered robot does not suffer from smoke inhalation and can take some heat.

3.1.7.5 Sand Volley Ball Court

On 09 May 2018, the Discovery Channel filmed Cassie Blue. The Segway riding, reported later, was their main interest. Since we were near the sand volley ball court, we challenged the robot to walk on it [52]. The narrow feet sunk into the sand, with the "heel" digging in the most. Because the stance foot is passive, the robot's gait remained quite stable. The robot walked more slowly than on grass (possibly due to foot slip) and it traversed the entire course, passing under the net. For the second pass, a pair of tennis shoes were placed on the robot. Cassie then walked no differently than when on grass or concrete.

3.2 Quiet Standing and Riding a Segway

For standing, all ten actuators are used. Foot actuation is required to prevent rotation of the robot about the y-axis in the body frame. The standing controller also allows Cassie to ride a Segway; yes, a robot riding another robot.

3.2.1 Quiet Standing

The standing condition is here assumed to be reached either from a stepping-in-place gait or by a user booting up the robot with the torso suspended approximately a half meter off the ground. The feet are assumed to be flat on the ground and beneath the torso.

Let $p^{\text{CoM}} = (p_x^{\text{CoM}}, p_y^{\text{CoM}}, p_z^{\text{CoM}})$ be the center of mass of the robot. With the feet flat on the ground and $(p_x^{\text{CoM}}, p_y^{\text{CoM}})$ within the convex hull of the feet, the robot can maintain a static pose. To maintain the desired center of mass position, p_x^{CoM} is regulated with the actuators for the pitch angle of the feet, while p_y^{CoM} is regulated in an indirect way: the roll angle, instead of p_y^{CoM}, is controlled to be zero. The roll is controlled by adjusting the leg length difference in the two legs. A zero roll angle is equivalent to a centered p_y^{CoM} when all other joints are symmetric. One reason we are using roll angle for feedback is that its value is less noisy than p_y^{CoM}, which is calculated via a kinematic chain. Another reason is that if the ground is sloped in the frontal plane, causing the robot to lean to the right[3], the right leg will automatically

[3]This feature is useful even on flat ground. The PD control of leg length should be thought of

be extended and the left leg compressed, as shown in Fig. 3.5. The height of the standing pose p_z^{CoM} is set by adjusting the average of the two leg lengths. The hip roll and yaw joints are regulated to constant values. With this controller, and the two feet roughly 0.3 m apart, Cassie is able to squat almost flat on the ground and stand to a height of approximately one meter. Quiet standing, lowering to a squat, and standing back up are illustrated in the video associated with the paper [49].

3.2.2 Riding a Segway

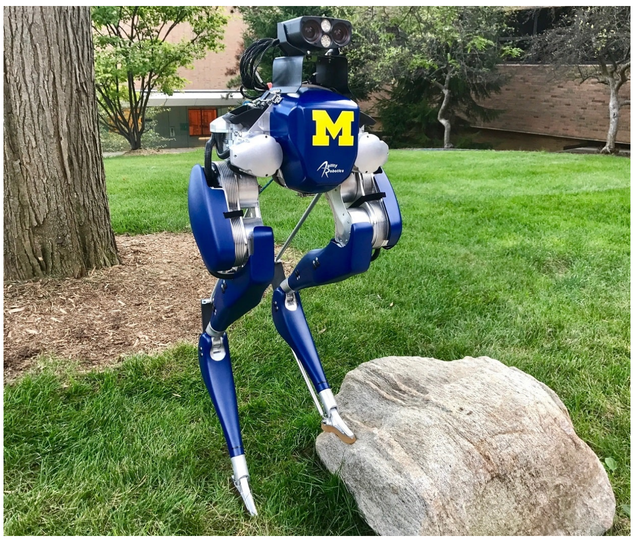
Figure 3.5 shows Cassie Blue riding a Segway miniPro (robot). The dynamics of the Segway are unknown and its states are not measured. The acceleration and direction of the Segway are determined by body lean, that is, by adjusting the target center of mass position. As elsewhere, the commands are sent by an operator via radio control.

To accelerate or decelerate the robot-Segway system, p_x^{CoM} is shifted forward or backward, respectively. To turn, Cassie needs to lean into the center bar with her legs, which is accomplished by shifting p_y^{CoM}. With the nominal standing controller, the Segway would oscillate when Cassie was placed on its platform. The feedback gains on the feet were reduced and the oscillations ceased.

With the crouched posture seen in Fig. 3.5, Cassie Blue was able to ride on sidewalks and grassed areas at roughly 4 m/s [53]. To be clear, the robot was placed on the Segway by an operator. Mounting and dismounting of the Segway were not addressed.

---

as soft springs. A bit of lean to the right places more weight on the right leg, which causes further compression of the spring in the right leg, which causes further leaning, etc., until p_y^{CoM} moves to the right of the foot and the robot falls. With high PD gains, this cascading effect can be avoided, but regulating p_y^{CoM} solves the problem with lower gains.



(a)

(b)

Figure 3.5: (a) Cassie standing on an uneven surface. (b) A stack of robots. Cassie is riding a Segway miniPro. The Segway will accelerate forward if the foot platform leans forward. Turning is controlled by pushing against the central vertical bar.

CHAPTER IV

Pendulum Models and Zero Dynamics

Models of realistic bipedal robots tend to be high-dimensional, hybrid, nonlinear systems, thus posing a great challenge to controller design and stability analysis. This chapter is concerned with two major themes in the literature for "getting around" the analytical and computational obstructions posed by realistic models of bipeds.

On the one hand are the broadly used, simplified pendulum models [54, 38, 19, 16, 55, 56, 57] that provide a computationally attractive model for the center of mass dynamics of a robot. When used for control design, the fact that they ignore the remaining dynamics of the robot generally makes it impossible to prove any stability properties of the closed-loop system. Despite the lack of analytical backing, the resulting controllers often work in practice when the center of mass is well regulated to match the assumptions underlying the model. Within this context, the dominant low-dimensional pendulum model by far is the so-called linear inverted pendulum model, or LIP model for short, which captures the center of mass dynamics of a real robot correctly when, throughout a step, the following conditions hold: (i) the center of mass (CoM) moves in a straight line; and (ii), the robot's angular momentum about the center of mass (L_c) is zero (or constant). This latter condition can be met by designing a robot to have light legs, such as the Cassie robot by Agility Robotics [58], or by deliberately regulating L_c to zero [39, 59]. When L_c cannot be regulated to a small value, an MPC feedback control law based on the LIP model has been proposed to minimize zero moment point (ZMP) tracking error and CoM jerk [60, 61, 62]. The effects of L_c can be compensated with ZMP, making a real robot's CoM dynamics the same as those of a LIP [40]. Alternatively, L_c can be approximately predicted and used for planning [63, 64].

On the other hand, the control-centric approach called the Hybrid Zero Dynamics provides a mathematically-rigorous gait design and stabilization method for realistic bipedal models [65, 66, 67, 68, 69, 70, 71, 72, 73, 74] without restrictions on robot or gait design. In this approach, the links/joints of the robot are synchronized via the imposition of "virtual constraints", meaning the constraints are achieved through the action of a feedback controller instead of contact forces. As opposed to physical constraints, virtual constraints can be re-programmed on the fly. Like physical constraints, imposing a set of virtual constraints results in a reduced-dimensional model. The term "zero dynamics" for this reduced dynamics comes from the original work of [36, 75]. The term "hybrid zero dynamics" or HZD comes from the extension of zero dynamics to (hybrid) robot models in [76]. A downside of this approach, however, has been that it lacked the "analytical tractability" provided by the pendulum models[1], and it requires non-trivial time to find optimal virtual constraints for a realistic model.

While CoM velocity is the most widely used variable "to summarize the state" of a bipedal robot, angular momentum about the contact point has also been valued by multiple researchers. In [80], angular momentum is chosen to represent a biped's state and it is regulated by stance ankle torque. In [21], the relative degree three property of angular momentum motivated its use as a state variable in the zero dynamics. In [81, 82], control laws for robots with an unactuated contact point were proposed to exponentially stabilize them about an equilibrium. In [71], angular momentum is

[1] The approaches in [77, 78, 79] to build reduced-order models via embeddings is a step toward attaching physical significance to the zero dynamic models.

explicitly used for designing nonholonomic virtual constraint. In [83], angular momentum is combined with the LIP model to yield a controller that stabilizes the transfer of angular momentum from one leg to the next through continuous-time (single-support-phase) control coupled with a hybrid model that captures impacts that occur at foot strike. In [84], the accuracy of the angular-momentum-based LIP model during the continuous phase is emphasized; as opposed to [83], angular momentum is allowed to passively evolve according to gravity during each single support phase, and foot placement is used to regulate the estimated angular momentum at the end of the ensuing step.

The objectives of this chapter are two-fold. Firstly, we seek to contribute insight on how pendulum models relate to one another and to the dynamics of a physical robot. We demonstrate that even when two pendulum models originate from the same (correct) dynamical principles, the approximations made in different coordinate representations lead to non-equivalent approximations of the dynamics of a (realistic) bipedal robot. Secondly, we seek a rapprochement of the most common pendulum models and the hybrid zero dynamics of a bipedal robot. Both of these objectives are addressed for planar robot models. The extension to 3D is not attempted here, primarily to keep the arguments as transparent as possible.

The first point, that approximations made in different coordinate representations lead to non-equivalent approximations of the dynamics of a real robot, is important in practice; hence we elaborate a bit more here, with details given in Sect. 4.4. Let's only consider trajectories of a robot where the center of mass height is constant, and therefore, the velocity and acceleration of the center of mass height are both zero. In a realistic robot, the angular momentum about the center of mass, denoted by L_c, contributes to the longitudinal evolution of the center of mass, though it is routinely dropped in the most commonly used pendulum models. Can dropping L_c have a larger effect in one simplified model than in another?

In the standard 2D LIP model, the coordinates are taken as the horizontal position and velocity of the center of mass *and the time derivative of L_c is dropped from the differential equation for the velocity.* It follows that the term being dropped is a high-pass filtered version of L_c, due to the derivative. Moreover, the derivative of L_c is directly affected by the motor torques, which are typically "noisy" (have high variance) in a realistic robot. On the other hand, in a less frequently used representation of a 2D inverted pendulum [71, 22, 83, 84], the coordinates are taken as angular momentum about the contact point and the horizontal position of the center of mass, *and L_c is dropped from the differential equation for the position.* In this model, L_c (and not $\dot{L}_c$) shows up in the second derivative of the angular momentum about the contact point. It follows that variations in L_c are low-pass filtered in the second representation as opposed to high-pass filtered in the first, and thus, speaking intuitively, neglecting L_c should induce less approximation error in the second model. More quantitative results are shown in the main body of the chapter.

In this chapter, we focus on the underactuated single support phase dynamics and assume an instantaneous double support phase. The reader is referred to existing literature on how pendulum models [25, 85] and Hybrid Zero Dyanmics [86, 87, 88] handle non-instantaneous double support phases; the topic is not discussed in this chapter. We provide models for a robot with non-trivial feet. While most of the results are demonstrated for robots with point feet, we briefly show that the conclusions we obtained for robots with point feet still apply to robots with non-trivial feet. Further studies of how pendulum models and Hybrid Zero Dynamics handle non-trivial feet can be found in [60, 89, 90, 91]

4.1 Swing Phase and Hybrid Model

This section introduces the full-dimensional swing-phase model that describes the mechanical model when the robot is supported on one leg and a hybrid representation used for walking that captures the transition of support legs. The section concludes with a summary of a few model properties that are ubiquitous when discussing low-dimensional pendulum models of walking.

4.1.1 Full-dimensional Single Support Model

We assume a planar bipedal robot satisfying the specific assumptions in [65, Chap. 3.2] and [21], which can be summarized as a revolute point contact with the ground, no slipping, all other joints are independently actuated, and all links are rigid and have mass. The gait is assumed to consist of alternating phases of single support (one "foot" on the ground), separated by instantaneous double support phases (both feet in contact with the ground), with the impact between the swing leg and the ground obeying the non-compliant, algebraic contact model in [92, 93] (see also [65, Chap. 3.2]).

The contact point with the ground, which we refer to as the stance ankle, can be passive or actuated. Even when actuated, the stance ankle is "weak" in the sense that only limited torque can be applied before the foot rolls about one of its extremities. The swing ankle is not weak, however, because it only needs to regulate the orientation of the swing foot. To accommodate both actuation scenarios, we will routinely separate the stance ankle actuation from other actuators on the robot so that it can be either set to zero or appropriately exploited.

We assume a world frame (x, z) with the right-hand rule. We assume the swing-phase (pinned) Lagrangian model is derived in coordinates $q := (q_0, q_1, \ldots, q_n) \in Q$, where q_0 is an absolute angle (referenced to the z-axis of the world frame) and $q_b := (q_1, \ldots, q_n)$ are body coordinates. Furthermore, we reference the contact point (i.e., stance ankle) to the origin of the world frame.

With the above sets of assumptions, the robot in single-support is either fully actuated or has one degree of underactuation. Moreover, q_0 is a cyclic variable (of the kinetic energy). It follows that the dynamic model can be expressed in the form

$$D(q_b)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)u, \tag{4.1}$$

where the vector of motor torques $u \in \mathbb{R}^n$ and the torque distribution matrix has full column rank. The model is written in state space form by defining

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \dot{q} \\ D^{-1}(q_b)\left[-C(q,\dot{q})\dot{q} - G(q) + B(q)u\right] \end{bmatrix} \\ &=: f(x) + g(x)u \end{aligned} \tag{4.2}$$

where $x := (q; \dot{q})$. The state space of the model is $\mathcal{X} = T\mathcal{Q}$. For each $x \in \mathcal{X}$, $g(x)$ is a $2(n+1) \times (n+1)$ matrix. In natural coordinates $(q; \dot{q})$ for $T\mathcal{Q}$, g is independent of $\dot{q}$.

4.1.2 Full Dimensional Hybrid Model

In the above, we implicitly assumed left-right symmetry in the robot so that we could avoid the use of two single-support models—one for each leg playing the role of the stance leg—by relabeling the robot's coordinates at impact, thereby swapping their roles. Immediately after swapping, the former swing leg is in contact with the ground and is poised to take on the role of the stance leg. The result of the impact and the relabeling of the states provides an expression

$$x^+ = \Delta(x^-) \tag{4.3}$$

where $x^+ := (q^+; \dot{q}^+)$ (resp. $x^- := (q^-; \dot{q}^-)$) is the state value just after (resp. just before) impact and

$$\Delta(x^-) := \begin{bmatrix} \Delta_q(q^-) \\ \Delta_{\dot{q}}(q^-)\,\dot{q}^- \end{bmatrix}. \tag{4.4}$$

A detailed derivation of the impact map is given in [65], showing that it is linear in the generalized velocities.

A hybrid model of walking is obtained by combining the single support model and the impact model to form a system with impulse effects [94]. A non-instantaneous double support phase can be added [86, 88], but we choose not to do so here. Even though the mechanical model of the robot is time-invariant, we will allow feedback controllers for (4.2) that are time-varying. So that the hybrid model in closed loop can be analyzed with tools developed for time-invariant hybrid systems, we do the standard "trick" of adding time as a state variable via $\dot{\tau} = 1$. The guard condition (aka switching set) for terminating a step is

$$\mathcal{S} := \{(q, \dot{q}) \in T\mathcal{Q} \mid p^{\mathrm{z}}_{\mathbf{sw}}(q) = 0,\ \dot{p}^{\mathrm{z}}_{\mathbf{sw}}(q, \dot{q}) < 0\}, \tag{4.5}$$

where $p^{\mathrm{z}}_{\mathbf{sw}}(q)$ is the vertical height of the swing foot. It is noted that $\mathcal{S}$ is independent of time. Combining (4.2), (4.3) with the guard set and time gives the hybrid model

$$\Sigma : \begin{cases} \dot{x} = f(x) + g(x)u & x^- \notin \mathcal{S} \\ \dot{\tau} = 1 & \\ x^+ = \Delta(x^-) & x^- \in \mathcal{S} \\ \tau^+ = 0. & \end{cases} \tag{4.6}$$

It is emphasized that the guard condition for re-setting the "hybrid time variable", τ, is determined by foot contact.

4.1.3 Center of Mass Dynamics in Single Support

While (4.2) is typically high dimensional and nonlinear, standard mechanics yields simpler equations for the evolution of the center of mass. For succinctness, we only consider the planar case and define the following variables:

- $(x_{\mathrm{c}}, z_{\mathrm{c}})$: CoM position in the frame of the contact point.
- v_{c} : CoM velocity in x-direction. The velocity in z-direction is denoted by $\dot{z}_{\mathrm{c}}$
- L_{c} : y-component of Angular momentum about CoM.
- L : y-component of Angular momentum about contact point.
- u_a : ankle torque at the contact point.

In addition, we note the following (standard) result

$$L = L_c + m \begin{bmatrix} x_c \\ z_c \end{bmatrix} \wedge \begin{bmatrix} \dot{x}_c \\ \dot{z}_c \end{bmatrix} \tag{4.7}$$

where $\wedge$ is the 2D version of cross product

$$\begin{bmatrix} x_c \\ z_c \end{bmatrix} \wedge \begin{bmatrix} \dot{x}_c \\ \dot{z}_c \end{bmatrix} := \left(\begin{bmatrix} x_c \\ 0 \\ z_c \end{bmatrix} \times \begin{bmatrix} \dot{x}_c \\ 0 \\ \dot{z}_c \end{bmatrix} \right) \bullet \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

We refer to (4.7) as the *angular momentum transfer formula* because it relates angular momentum determined about two different points.

In the following, we provide the CoM dynamics for two sets of coordinates

- $(x_{\mathrm{c}}, v_{\mathrm{c}})$
- (x_{c}, L), and

- (θ_c, L),

where

$$\theta_c := \mathrm{atan}(x_c/z_c) \tag{4.8}$$

and we assume that $z_c > 0$. We will subsequently dedicate Sect. 4.3 to establishing connections between pendulum models and zero dynamics, which will allow the zero dynamics to be intuitively grounded in physics.

Case 1: (x_c, v_c) **Horizontal Position and Velocity** Differentiating (4.7) and using $v_c = \dot{x}_c$ results in

$$\begin{aligned} \dot{x}_c &= v_c \\ \dot{v}_c &= \frac{g}{z_c}x_c + \frac{\ddot{z}_c}{z_c}x_c - \frac{\dot{L}_c}{mz_c} + \frac{u_a}{mz_c}. \end{aligned} \tag{4.9}$$

In general, z_c depends on q, $\dot{z}_c$ and L_c depend on both q and $\dot{q}$. While $\dot{L}_c$ and $\ddot{z}_c$ depend on q, $\dot{q}$, and the motor torques u, it is more typical to replace the motor torques by the ground reaction forces. In particular, one uses $\ddot{z} = g - \frac{1}{m}F_z$ and $\dot{L}_c := \frac{d}{dt}L_c = x_cF_z - z_cF_x + u_a$, where F_x and F_z are the horizontal and vertical components of the ground reaction forces. In turn, the ground reaction forces can be expressed as functions of q, $\dot{q}$, and the motor torques, u.

Case 2: (x_c, L) **Angular Momentum and Horizontal Position:** Manipulating (4.7) and using $\dot{L} = mgx_c + u_a$ results in

$$\begin{aligned} \dot{x}_c &= \frac{L}{mz_c} + \frac{\dot{z}_c}{z_c}x_c - \frac{L_c}{mz_c} \\ \dot{L} &= mgx_c + u_a. \end{aligned} \tag{4.10}$$

The remarks made above on z_c, $\dot{z}_c$, and L_c apply here as well.

Case 3: (θ_c, L) **Alternative absolute angle (cyclic variable):** Differentiating (4.8) yields

$$\dot{\theta}_c = \frac{L - L_c}{mr_c^2(q_b)}. \tag{4.11}$$

Combining (4.7) and (4.11) yields

$$\begin{aligned}\dot{\theta}_c &= \frac{L - L_c}{mr_c^2}\\ \dot{L} &= mgr_c\sin(\theta_c) + u_a.\end{aligned} \tag{4.12}$$

It is remarked that the derivatives of the generalized coordinates only appear through L_c. In the following, we will keep the discussion primarily focused on (4.10), but most of the results apply to (4.12) as well; see Appendix A.2.

4.2 Angular Momentum about the Contact Point

In this chapter, we are focusing on the angular momentum about the contact point, L, as a replacement for the center of mass velocity, $v_{\rm c}$, which is used as an indicator of walking status in many other papers [95, 74, 57, 96]. Specific to this paper, L is also a state of the zero dynamics. Before we proceed to that, it is beneficial to explain why L can replace $v_{\rm c}$, summarize some general properties of L, and highlight some of its advantages versus $v_{\rm c}$. More specific advantages of using L in the zero dynamics and the LIP model will be discussed in later sections.

We first need to answer why L can replace $v_{\rm c}$ as an indicator of walking. The relationship between angular momentum and **linear momentum** for a 3D bipedal robot is

$$L = L_{\rm c} + p_{\rm c} \wedge mv_{\rm c}, \tag{4.13}$$

where $L_{\rm c}$ is the angular momentum about the center of mass, $v_{\rm c}$ is the linear velocity of the center of mass, m is the total mass of the robot, and $p_{\rm c}$ is the vector emanating from the contact point to the center of mass.

For a bipedal robot that is walking instead of doing somersaults, it is reasonable to focus on gaits where the angular momentum about the center of mass oscillate about zero (e.g., arms are not rotating as in a flywheel). The oscillating property of $L_{\rm c}$ is discussed in [1, 2]. When L oscillates about zero, (4.13) implies that the difference between L and $p_{\rm c} \wedge m v_{\rm c}$ also oscillates about zero, which we will write as

$$L - p_{\rm c} \wedge m v_{\rm c} = L_{\rm c} \text{ oscillates about } 0. \tag{4.14}$$

From (4.14), we see that we approximately obtain a desired linear velocity by regulating L. Hence, in walking robots without a flywheel, one can replace the control of linear velocity with control of angular momentum about the contact point.

What are there advantages to using L?

(a) The first advantage of controlling L is that it provides a more comprehensive representation of current walking status because it is the sum of angular momentum about the center of mass, $L_{\rm c}$, and linear momentum, $p_{\rm c} \wedge m v_{\rm c}$. From (4.9), we see that there exists momentum transfer between these two quantities. If $L_{\rm c}$ increases, it must "take" some momentum away from $v_{\rm c}$, and vice versa. For normal bipedal walking, $L_{\rm c}$ oscillates about zero. $L_{\rm c}$ functions to store momentum[97], but importantly it can an only store it for a short amount of time. When designing a foot placement strategy, it is important to take the "stored" momentum into account.

    When balancing on one foot for example, some researchers plan $L_{\rm c}$ and $v_{\rm c}$ separately [98], or use $L_{\rm c}$ as an input to regulate balance by waving the torso, arms, or swing leg[99, 100] or even a flywheel [18]. Here, instead of moving limbs to generate a certain value of $L_{\rm c}$, we view $L_{\rm c}$ as a result of the legs and torso moving to fulfill other tasks. In this paper, we observe $L_{\rm c}$ and take it into consideration through L and do not seek to regulate it directly as an independent quantity.

(b) Secondly, because $\dot{L} = mgx_{\rm c} + u_a$ depends only on the CoM position, it follows that L has relative degree three with respect to all inputs except the stance

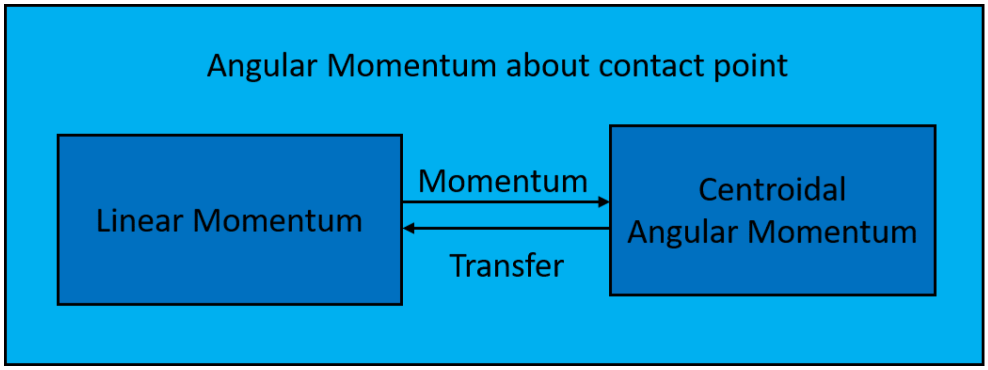


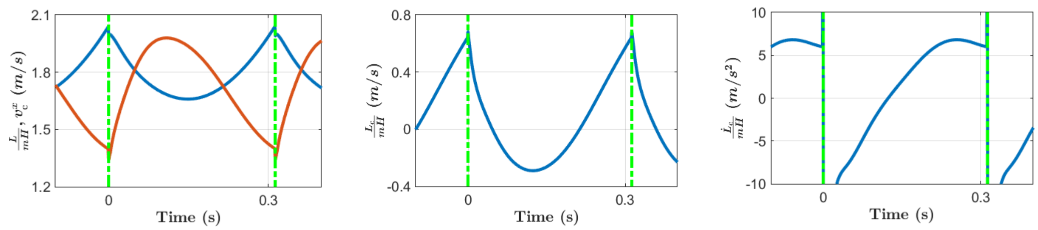
Figure 4.1: The relation between L, L_c, and v_c^x. Equation (4.13) shows L is the sum of L_c and a term that is linear in v_c, while the second line of (4.9) shows the transfer of momentum between L_c and v_c. The relation is an analogue of mechanical, kinetic and potential energy.

ankle torque, where it has relative degree one. Consequently, the evolution L is only weakly affected by motor torques of the body, that is u_b, during a step. In Fig. 4.2 (a) and (d) and Fig. 4.3 (a) we see that the trajectory of L consistently has a convex shape when stance ankle torque is zero, irrespective of model or speed. We'll see later the same property in experimental data.

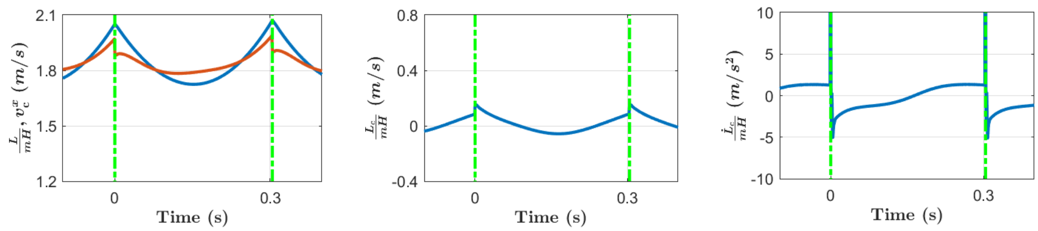
(c) The discussion so far has focused on the single support phase of a walking gait. Bipedal walking is characterized by the transition between left and right legs as they alternately take on the role of stance leg (aka support leg) and swing leg (aka non-stance leg). In double support, the transfer of angular momentum between the two contact points satisfies

$$L_2 = L_1 + p_{2\to 1} \wedge m v_c, \tag{4.15}$$

where $p_{2\to 1}$ is the vector from point 2 (the new stance leg position) to point 1 (the previous stance leg position). Hence, the change of angular momentum between two contact points depends only on the vector defined by the two contact points and the center of mass velocity. In particular, angular momentum about a given contact point is invariant under the impulsive force generated at that contact



(a) Rabbit $\frac{L}{mH}$ and v_c^x (b) Rabbit $\frac{L_c}{mH}$ (c) Rabbit $\frac{\dot{L}_c}{mH}$

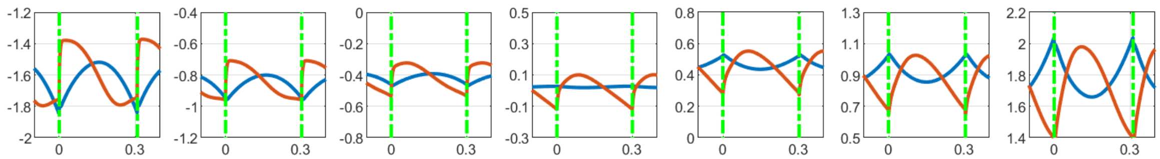


(d) Cassie $\frac{L}{mH}$ and v_c (e) Cassie $\frac{L_c}{mH}$ (f) Cassie $\frac{\dot{L}_c}{mH}$

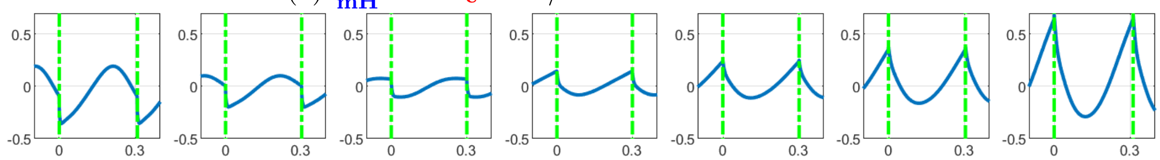
Figure 4.2: Plots of L, v_c, and $\dot{L}_c$ for the bipedal robots Rabbit and Cassie walking at about 2m/s, while z_c is carefully regulated to 0.6m. The vertical green lines indicate the moment of impact. For both robots, the angular momentum about the contact point, L, has a convex shape (due to $\dot{L} = mgx_c + u_a$, $u_a = 0$ and CoM passes the contact point only once), similar to the trajectory of a LIP model, while the trajectory of the longitudinal velocity of the center of mass, v_c, has no consistent shape. The variation of L_c throughout a step, which is caused by the legs of the robot having mass, is what leads to a difference in the CoM velocity between a real robot and a LIP model. The patterns of L_c shown above are not specific to certain robot or controller but match the walking mechanism described in [1, 2].
In this figure, L is continuous at impact, which is based on two conditions: $v_c^z = 0$ at impact and the ground is level. Even when these two conditions are not met, the jump in L at impact can be easily calculated with (4.15).

point. Consequently, we can easily determine the angular momentum about the new contact point by (4.15) when impact happens without resorting to approximating assumptions about the impact model. Moreover, if $\dot{z}c$ is zero and the ground is level, then $p_{2\to 1} \wedge m v_c = 0$, and hence $L_2 = L_1$. We note that pendulum models parameterized with CoM velocity often assume continuity at impact, which is not generally true for real robots.

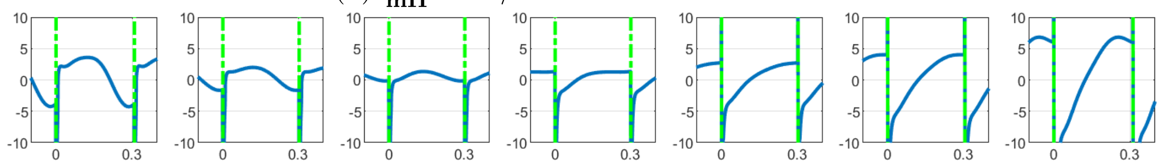
Figure 4.2 shows the evolution of L, v_c, L_c and $\dot{L}_c$ during a step for both Cassie and Rabbit, when walking speed is about 2 m/s, $\dot{z}_c = 0$, and no stance ankle torque is applied. Figure 4.3 shows the evolution of L, v_c, L_c for Rabbit walking at a range of speeds from -1.8 m/s to 2.0 m/s. Figures 4.2 and 4.3 also show the continuity property of L at impact.



(a) $\frac{\mathbf{L}}{\mathbf{mH}}$ and $\mathbf{v_c}$ in m/s versus time in seconds.



(b) $\frac{\mathbf{L_c}}{\mathbf{mH}}$ in m/s versus time in seconds.



(c) $\frac{\dot{\mathbf{L}}_\mathbf{c}}{\mathbf{mH}}$ in m/s^2 versus time in seconds.

Figure 4.3: Plots of L, v_c, and $\dot{L}c$ for Rabbit walking at different speeds. The green vertical lines indicate the moment of impact. (a) shows that L always has a convex or concave shape like the LIP model, while v_c has no determinant shape. The shape of L is a direct consequence of $\dot{L} = mgxc$. The quantities $\frac{\mathbf{L}}{\mathbf{mH}}$ and $\mathbf{v_c}$ are close in scale and oscillate about one another. This shows that directly regulating L does indeed indirectly regulate v_c. (b) and (c) show the scales of L_c and $\dot{L}c$. It is seen that $\dot{L}c$ is much larger in scale and thus omitting it in (4.16) can create a larger error than neglecting L_c in (4.17).

4.3 Comparison of Approximate Models for Center of Mass Dynamics

Each of the dynamical models(4.9), (4.10), and (4.12) is valid along all trajectories of the full-dimensional model. This section systematically goes through the models in Sec. 4.1.3 and looks for connections with low-dimensional pendulum models. Subsequently, Sect. 4.4 makes connections between pendulum models and the zero dynamics.

4.3.1 Constant Pendulum Height

If CoM height is constant, i.e., $z_\mathrm{c} = H$, $\dot{z}_\mathrm{c} = 0$, and $\ddot{z}_\mathrm{c} = 0$, then (4.9) and (4.10) become

$$\begin{aligned} \dot{x}_\mathrm{c} &= v_\mathrm{c} \\ \dot{v}_\mathrm{c}^x &= \frac{g}{H} x_\mathrm{c} - \frac{\dot{L}_\mathrm{c}}{mH} + \frac{u_a}{mH}, \end{aligned} \tag{4.16}$$

and

$$\begin{aligned} \dot{x}_\mathrm{c} &= \frac{L}{mH} - \frac{L_\mathrm{c}}{mH} \\ \dot{L} &= mgx_\mathrm{c} + u_a, \end{aligned} \tag{4.17}$$

respectively. Equation (4.17) can be rewritten as

$$\begin{aligned} \dot{x}_\mathrm{c} &= v_p - \frac{L_\mathrm{c}}{mH} \\ \dot{v}_p &= \frac{g}{H} x_\mathrm{c} + \frac{u_a}{mH}, \end{aligned} \tag{4.18}$$

where $v_p = \frac{L}{mH}$, which is more directly comparable to (4.16). In this paper we frequently plot L scaled by the coefficient $\frac{1}{mH}$, so that it can be more directly compared to v_c (same units and similar magnitudes).

At this point, no approximations have been made and both models are valid everywhere that $z_\mathrm{c}(q) \equiv H$. Hence, the two models are still *equivalent* representations of the center of mass dynamics for all trajectories satisfying $z_\mathrm{c}(q) \equiv H$. *We'll next argue that the models are not equivalent when it comes to approximations.*

Dropping the $\dot{L}_\mathrm{c}$ term in (4.16) results in:

$$\begin{aligned} \dot{x}_\mathrm{c} &= v_\mathrm{c} \\ \dot{v}_\mathrm{c} &= \frac{g}{H} x_\mathrm{c} + \frac{u_a}{mH}. \end{aligned} \tag{4.19}$$

This is the well-known LIP model proposed by [15].

Dropping L_c in (4.17) results in

$$\begin{aligned}\dot{x}_\mathrm{c} &= \frac{L}{mH} \\ \dot{L} &= mgx_\mathrm{c} + u_a,\end{aligned} \tag{4.20}$$

which is used in [83, 84]. In [83], (4.20) is used instead of (4.19) so that the easy "update" property of L at impact can be used. In this paper, we demonstrate that during the continuous phase, the states of (4.20) much more accurately capture the evolution of (x_c, L) in a real robot than the states of (4.19) capture the evolution of $(x_\mathrm{c}, v_\mathrm{c})$. Moreover, we will make use of this improved accuracy in the design of a feedback controller. To distinguish the model (4.20) from (4.19), we will denote it by ALIP, where A stands for Angular Momentum.

For a robot with a point mass, the two models (4.19) and (4.20) are equivalent, because L_c is then identically zero. For a real robot with L_c and $\dot{L}_\mathrm{c}$ that are nonnegligible, however, we argue that (4.20) is more accurate than (4.19) primarily because of three properties,

(a) Relative Amplitude. Based on our observations, the ratio of $\dot{L}_\mathrm{c}/mgx_\mathrm{c}$ is much larger than L_c/L over a wide range of walking velocities; thus the simplification (4.20) introduces relatively less error than (4.19).

(b) Relative degree. L has relative degree two with respect to L_c and three with respect to $\dot{L}_\mathrm{c}$, whereas v_c has relative degree one with respect to $\dot{L}_\mathrm{c}$. Because integration is a form of low-pass filtering, the lower relative degree makes v_c more sensitive to the omission of the L_c term.

(c) L_c oscillates about zero. What makes (4.20) even more accurate is that, based on our own observation and references [1, 2], the sagittal plane component of L_c oscillates about zero for periodic and non-periodic gaits. The oscillation of L_c results in the effect of L_c on x_c roughly averaging out to zero over a step.

In Fig. 4.4, we have used the models (4.19) and (4.20) to predict the values of v_{c} and L at the end of a step. We plot $\frac{L}{mH}$ instead of L to make the scale and units comparable. The blue line is the true trajectory of L (resp. v_{c}) during a step. The red line shows the prediction of L (resp. v_{c}) at the end of each of step, at each moment throughout a step, based on the instantaneous values of x_{c} and L (v_{c}) at that moment. The red line would be perfectly flat if (4.20) and (4.19) perfectly captured the evolution of L (v_{c}), respectively, in the full simulation model, and the flatter the estimate, the more faithful is the representation.

The prediction errors of (4.19) and (4.20) caused by neglecting L_{c} and $\dot{L}_{\mathrm{c}}$, respectively, satisfy

$$\begin{aligned} \dot{x}_e &= v_e \\ \dot{v}_e &= \frac{g}{H} x_e - \frac{\dot{L}_{\mathrm{c}}}{mH}, \end{aligned} \tag{4.21}$$

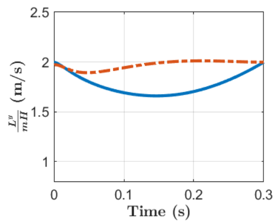
and

$$\begin{aligned} \dot{x}_e &= \frac{L_e}{mH} - \frac{L_{\mathrm{c}}}{mH} \\ \dot{L}_e &= mgx_e, \end{aligned} \tag{4.22}$$

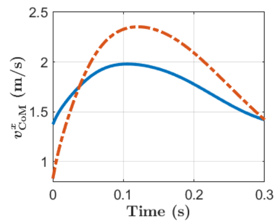
where (x_e, L_e) are the differences in the trajectories of (4.19) and (4.18); similarly, (x_e, v_e) are the differences in the trajectories of (4.20) and (4.16). Direct solution of these two sets of differential equations for zero initial conditions leads to

$$\begin{aligned} v_e(t_2, t_1) &= e_1(t_2, t_1) \\ &= e_2(t_2, t_1) + e_3(t_2, t_1) \end{aligned} \tag{4.23}$$

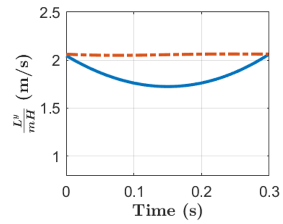
$$\frac{L_e(t_2, t_1)}{mH} = e_2(t_2, t_1), \tag{4.24}$$



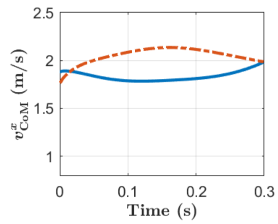
(a) Rabbit L^y prediction



(b) Rabbit v_{c} prediction



(c) Cassie L^y prediction



(d) Cassie v_{c} prediction

Figure 4.4: Comparison of the ability to predict velocity vs angular momentum at the end of a step. The instantaneous values are shown in **blue** and the predicted value at the end of the step is shown in **red**, where a perfect prediction would be a flat line that intercepts the terminal point of the blue line. The most crucial decision in the control of a bipedal robot is where to place the next footfall. In the standard LIP controller, the decision is based on predicting the longitudinal velocity of the center of mass. In Sect. 5.2 we use angular momentum about the contact point. We do this because on realistic bipeds, a LIP-style model provides a more accurate and reliable prediction of L than v_{c}. The comparison is more significant on Rabbit, whose leg center of mass is further away from the overall center of mass.

where

$$
\begin{aligned}
e_1(t_2,t_1) &= -\frac{1}{mH}\int_{t_1}^{t_2} \cosh(\ell(t_2-\tau))\dot{L}_{\mathrm{c}}(\tau)\,d\tau \\
e_2(t_2,t_1) &= -\frac{1}{mH}\int_{t_1}^{t_2} \ell\sinh(\ell(t_2-\tau))L_c(\tau)\,d\tau \\
e_3(t_2,t_1) &= -\frac{1}{mH}\big(L_{\mathrm{c}}(t_2) - \cosh(\ell(t_2-t_1))L_c(t_1)\big).
\end{aligned}
$$

Figure 4.5 shows the (relative) sizes of these error terms. If we view L_{c} as a disturbance and prediction error as an output in (4.21) and (4.22), we obtain the corresponding Laplace transforms and Bode plots shown in Fig. 4.6.

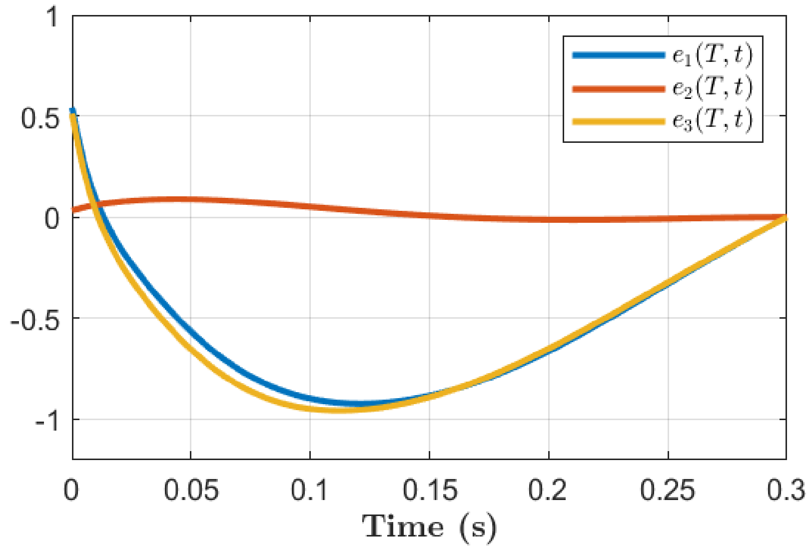


Figure 4.5: A plot of the error terms in (4.23) and (4.24) resulting from dropping $\dot{L}_c$ and L_c, respectively, for the Rabbit model walking at 2 m/s. The take-home message is that of the terms $e_2(t_1, t_2) + e_3(t_1, t_2)$ in (4.23) comprising the velocity error of the LIP model, the term $e_3(t_1, t_2)$ shown in the **yellow** line contributes by far the largest portion of the total error shown by the **blue** line. The error of the ALIP model, however, is given only by $e_2(t_1, t_2)$, which results in the significantly reduced prediction error shown by the **red** line.

4.3.2 Simulation Comparison

We compare controllers designed on the basis of the ALIP and LIP models in simulation. The results shown in Fig. 4.7 demonstrate the advantage of using ALIP over LIP for controller design. The initial hip velocity is set to 0.5 m/s and hip position is centered over the contact point. The goal of each controller is to regulate v_c (resp., L) to zero, with foot placement as the decision variable and step duration constant. In the plots, we observe that the ALIP-based controller regulates L closely to zero and thus has an average v_c close to zero, while the LIP-based controller is unable to regulate v_c effectively. The reason is that, at the end of a step, the linear momentum was transferred to centroidal angular momentum L_c due to the movement of Rabbit's heavy legs (see Eqn (4.18)), resulting in a small v_c, which misleads the LIP controller into choosing a small foot displacement. In the ALIP model, L is less affected by momentum transfer between v_c and L_c because L captures their sum, and thus the ALIP model suggests better foot placement. Though with a LIP controller it is possible to regulate velocity through ZMP (ankle torque) during continuous phase, we argue that with an ALIP controller, the capability of ZMP can be reserved for better purposes than compensating for model error.

4.3.3 Non-zero Ankle Torque

In previous subsections we have demonstrated the accuracy of pendulum model parameterized with L when ankle torque is zero. According to Eqn (4.10), the effect of L_c and u_a on the system are independent due to the superposition property. So if dropping L_c term has little effect on the model accuracy when u_a is zero, it should still has little effect on the model accuracy when u_a is non-zero. Though L_c trajectory itself will be changed when u_a is non-zero, its pattern is still similar. Here for completeness we run a simulation on Rabbit. The results are shown in Fig. 4.8

4.3.4 Accounting for L_c

The trajectory of L_c is determined by the robot's dynamic model and the movement of its links. In Fig. 4.3-(b), we observe that L_c has a simple shape[2]. This motivates us to estimate the trajectory of L_c and plug it into (4.17) to improve our prediction.

For each of a range of walking speeds, we fit the time-based trajectory of L_c during a step with a third-order polynomial in τ. For online use, the four polynomial coefficients in $\widehat{L}_c(\tau)$ were regressed as second-order polynomials in $L(0)$, the value of L at the beginning of each step. Given the estimate $\widehat{L}_c(\tau, L(0))$, we can plug it into (4.17) and view $\widehat{L}_c(\tau, L(0))$ as a time-varying forcing term. Its contribution to

[2]The trajectories are the result of a controller described in Sect. 5.3.

predicted angular momentum is then

$$\int_{t}^{T} e^{A(T-\tau)} B \widehat{L}_c(\tau, L(0)) d\tau, \tag{4.25}$$

which can be added to an estimate of L. Figure 4.9 shows the resulting improvements in estimates of scaled angular momentum when Rabbit is walking at various constant speeds. Figure 4.10 shows that the improvements in estimation persist under transient operation of the robot.

4.4 Pendulum Models, Zero Dynamics, and Overall System Stability

This section establishes connections between the pendulum models of Sec. 4.1.3 and the swing phase zero dynamics as developed in [65], or more precisely, approximations of the zero dynamics. This is accomplished by analyzing how the zero dynamics are driven by the states of a bipedal robot's full-order model and its feedback controller when the closed-loop system is evolving off the zero dynamics manifold. As a main contribution, the analysis will yield conditions under which the driving terms are small and hence do not adversely affect the stability predictions associated with the exact zero dynamics. A secondary contribution of the section will be a presentation of the swing phase zero dynamics for a more general set of "virtual constraints" than those developed in [65, 101, 71].

4.4.1 Intuitive Background

An initial sense of the meaning and mathematical foundation of the swing phase zero dynamics can be gained by considering a floating-base model of a bipedal robot, and then its pinned model, that is, the model with a point or link of the robot, such as a leg end or foot, constrained to maintain a constant position respect to the ground. The given contact constraint is holonomic and constant rank, and thus using Lagrange multipliers (from the principle of virtual work), a reduced-order model compatible with the (holonomic) contact constraint is easily computed. When computing the reduced-order model, no approximations are involved, and solutions of the reduced-order model are solutions of the original floating-base model, with inputs (ground reaction forces and moments) determined by the Lagrange multiplier.

Virtual constraints are relations (i.e., constraints) on the state variables of a robot's model that are achieved through the action of actuators and feedback control instead of physical contact forces. They are called *virtual* because they can be re-programmed on the fly without modifying any physical connections among the links of the robot or its environment. We use virtual constraints to synchronize the evolution of a robot's links, so as to create exponentially stable motions. Like physical constraints, under certain regularity conditions, they induce an exact low-dimensional invariant model, called the *zero dynamics*, due to the highly influential paper [75].

Each virtual constraint imposes a relation between joint variables, and by differentiation with respect to time, a relation between joint velocities. As a consequence, for the virtual constraints studied in this paper, the dimension of the zero dynamics is the number of states in the robot's (pinned) model minus twice the number of virtual constraints (which can be at most the number of independent actuators). As explained in [102], the computation of the motor torques to impose virtual constraints parallels the Jacobian computations for the ground reaction forces in a pinned model.

4.4.2 Allowing Non-holonomic, Time-varying Virtual Constraints

In this section, we choose L as one of the states of the zero dynamics. So that fully actuated and underactuated biped models can be addressed simultaneously, we suppose that the torque distribution matrix $B(q)$ in (4.1) can be split so that

$$B(q)u =: B_a(q)u_a + B_b(q)u_b, \tag{4.26}$$

where u_a is the torque affecting the stance ankle as in Sect. 4.1.3 and $u_b \in \mathbb{R}^n$ are actuators affecting the body coordinates, q_b. When the robot is underactuated, $B_a(q)$ is an empty column vector.

We define n virtual constraints as an output zeroing problem of the form

$$y = h(q, L, \tau) = h_0(q) - h_d(x_c, L, \tau), \tag{4.27}$$

where τ captures time dependence. As in Sect. 4.1.3, we use L instead of other functions of $\dot{q}$ because L has relative degree three with respect to all actuators except stance ankle torque, while $\dot{q}$ has relative degree one. Hence, the relative degree of y is determined by q once u_a is fixed. Indeed, while u_b is used for imposing the virtual constraints, u_a can be used for shaping the evolution of x_c and L directly. We assume a feedback law, u_a, of the form

$$u_a = \alpha(x_c, L, \tau), \tag{4.28}$$

and note that u_a should respect relevant ankle torque limits and ZMP constraints when $y \equiv 0$.

Following [75, 36, 65], we make the following specific *regularity assumptions* for the virtual constraints:

A1: h is at least twice continuously differentiable and u_a is at least once differentiable.

A2: The virtual constraints (4.27) are designed to identically vanish on a desired nominal solution (gait) $(\bar{q}(t), \dot{\bar{q}}(t), \bar{u}(t))$ of the dynamical model (4.6) with $\tau(t) = t$, where the solution meets relevant constraints on motor torque, motor power, ground reaction forces, and workspace. To be clear, y vanishing means

$$h_0(\bar{q}) - h_d(\bar{x}_c, \bar{L}, \tau) \equiv 0 \tag{4.29}$$

for $0 \leq t \leq T$, where $\bar{L}(t)$ is the angular momentum about the contact point, evaluated along the trajectory.

A3: The *decoupling matrix*

$$A(q) := \frac{\partial h(q, L, t)}{\partial q} D^{-1}(q) B_b(q) \tag{4.30}$$

is square and invertible along the nominal trajectory, so that, from [36] and [103], by treating u_a as a known signal, there exists a feedback controller of the form

$$u_b =: \gamma(q, \dot{q}, \tau) + \gamma_a(q, L, \tau) u_a \tag{4.31}$$

resulting in the closed-loop dynamics

$$\ddot{y} + K_d \dot{y} + K_p y = 0, \tag{4.32}$$

with $K_d > 0$ and $K_p > 0$ positive definite.

A4: The function

$$\begin{bmatrix} y \\ \dot{y} \\ x_c \\ L \end{bmatrix} \tag{4.33}$$

is full rank and injective in an open neighborhood of the nominal solution $(\bar{q}(t), \dot{\bar{q}}(t))$ $\forall t$.

From [75, 36, 65], the above assumptions imply that $(y, \dot{y}, x_c, L)$ is a valid set of coordinates for the full-order swing phase model (4.2). In particular,

1. there exists an invertible differentiable function Φ such that

$$\begin{bmatrix} q \\ \dot{q} \\ \tau \end{bmatrix} = \Phi(y, \dot{y}, x_c, L, \tau), \text{ and} \tag{4.34}$$

2. the swing phase zero dynamics, that is, the dynamics of the robot compatible with $y \equiv 0$, exists and can be parameterized by (x_c, L).

4.4.3 Zero Dynamics and Approximate Zero Dynamics

From Assumptions A1-A4, it follows that the swing phase zero dynamics exists and for $\xi = (x_c, L, \tau)$ can be expressed as

$$\dot{\xi} = f_{\text{zero}}(\xi), \tag{4.35}$$

when $y \equiv 0$. As with the popular pendulum models, the dimension of (4.35) is low, it has two states plus time. Different that the pendulum models, (4.35) is exact. Moreover, tools are known for relating periodic orbits of the hybrid version of (4.35) to corresponding orbits in the full-order model (4.6), including their stability properties; see Sect. 4.4.4.

On the basis of (4.10) evaluated at (4.34), the zero dynamics (4.35) can be written more explicitly as

$$\begin{aligned} \dot{x}_c &= \frac{L}{m z_c(x_c, L, \tau)} + \frac{\dot{z}_c(x_c, L, \tau)}{z_c(x_c, L, \tau)} x_c - \frac{L_c(x_c, L, \tau)}{m z_c(x_c, L, \tau)} \\ \dot{L} &= m g x_c + u_a(x_c, L, \tau) \\ \dot{\tau} &= 1. \end{aligned} \tag{4.36}$$

The state $\dot{\tau} = 1$ is included in (4.36) because, at hybrid transitions, τ is reset to zero, that is, $\tau^+ := 0$. As discussed above, this reduced-order model is exact along all trajectories of the full-order model for which $y \equiv 0$.

If one of the virtual constraints in (4.27) is $z_c - H$, that is, the center of mass height is regulated to a constant, then the zero dynamics (exactly) simplifies to

$$\begin{aligned} \dot{x}_c &= \frac{L}{mH} - \frac{L_c(x_c, L, \tau)}{mH} \\ \dot{L} &= mgx_c + u_a(x_c, L, \tau) \\ \dot{\tau} &= 1. \end{aligned} \tag{4.37}$$

This model is nonlinear and time-varying through L_c and possibly, the feedback control policy chosen for the stance ankle torque, u_a. We've argued in Sec. 4.3 that L_c can be dropped from the model. Doing so results in the ALIP model, (4.20). Hence, the ALIP model is an *approximate swing phase zero dynamics* when the center of mass height is controlled to a constant.

4.4.4 Consequences for Closed-loop Stability of the Full-order Model

When the foot placement policy (5.9) is applied to (4.37) with the rest map (5.2), the resulting closed-loop system is a (small) perturbation of a hybrid system that possesses a family of exponentially stable periodic orbits parameterized by $L^{\rm des}$. If the virtual constraints in (4.27) are hybrid[3] invariant for constant $L^{\rm des}$ [65, 106, 107], then

- (4.37) with impact map (5.2) is the hybrid zero dynamics, and
- an exponentially stable periodic solution of the hybrid zero dynamics is also an exponentially stable solution of the full order closed-loop system for appropriate choices of the feedback gains K_p and K_d in (4.32).

Consequently, the closed-loop system would possess a family of exponentially stable

[3]Hybrid invariance means that if y and $\dot{y}$ are zero before the impact, they will also be zero after the impact. References [104, 105] show how to systematically modify a given set of virtual constraints to achieve hybrid invariance.

periodic orbits parameterized by L^{des}. If the virtual constraints are not hybrid invariant, then (4.37) with (5.2) does not form a hybrid zero dynamics in the sense of [65], but rather a *limit restriction dynamics* [108, pp. 102]. Moreover, via the Brouwer Fixed Point Theorem, reference [108, Theorem 6, pp. 105] shows each exponentially stable periodic solution of the limit restriction dynamics corresponds to an exponentially stable periodic solution of the full model for appropriate design of the feedback gains in (4.32).

To illustrate the correspondence between exponentially stable motions of the ALIP and the full-order model, we turn to the Rabbit model controlled via virtual constraints that implement the foot placement control law (5.9), the center of mass at a constant height, the torso upright, and adequate foot clearance. We then numerically estimate the Jacobian of the Poincare map for the closed-loop full-order model and compare its dominant eigenvalues to the dominant eigenvalue of the closed-loop ALIP model; see (5.12) .

In Table 4.1, for various values of α in the step placement feedback controller, we show the dominant eigenvalue from the ALIP model and the dominant eigenvalue from the numerically estimated Poincaré map. We see that the dominant eigenvalue of the full-order closed-loop system corresponds to the dominant eigenvalue of the ALIP model for $0 \leq \alpha \leq 0.9$. The remaining eigenvalues of the full model are (very) small due to the gains chosen in (4.32). In fact, the zero dynamics captures the "weakly actuated", slow part of the full-order model that is evolving under the influence of gravity.

Remark: We numerically obtained the Jacobian of the Poincaré map for Rabbit with the foot placement controller by the method symmetric differences; δ deviations were applied on ten states (Rabbit has 5 degrees of freedom) and we measured the corresponding responses after two steps. The value of δ was chosen from the set $\{\pm 0.05, \pm 0.1, \pm 0.2, \pm 0.3\}$; see Table 4.2.

α	ALIP	Rabbit
0.9	0.81	0.781
0.8	0.64	0.601
0.7	0.49	0.442
0.6	0.36	0.299
0.5	0.25	0.168
0.4	0.16	0.052
0.3	0.09	0.013
0.2	0.04	2e-4
0.1	0.01	2e-4
0.0	0.00	1e-4

Table 4.1: Largest eigenvalues of ALIP and Rabbit under different α, for a two-step Poincaré map. Because the Poincaré map is computed over two steps, the ALIP's largest eigenvalue is α^2.

α	ALIP	$\delta = \pm 0.05$	$\delta = \pm 0.1$	$\delta = \pm 0.2$	$\delta = \pm 0.3$
0.9	0.81	0.780	0.783	0.781	0.779
0.8	0.64	0.602	0.602	0.601	0.599
0.7	0.49	0.442	0.443	0.441	0.439
0.6	0.36	0.299	0.301	0.300	0.299
0.5	0.25	0.170	0.170	0.168	0.164
0.4	0.16	0.054	0.053	0.052	0.051
0.3	0.09	0.014	0.014	0.012	0.011

Table 4.2: Numerical support for estimating the Jacobian of the two-step Poincaré map. The dominant eigenvalue of Rabbit model is insensitive to the perturbation used in estimating the Jacobian.

4.4.5 Non-periodic Walking

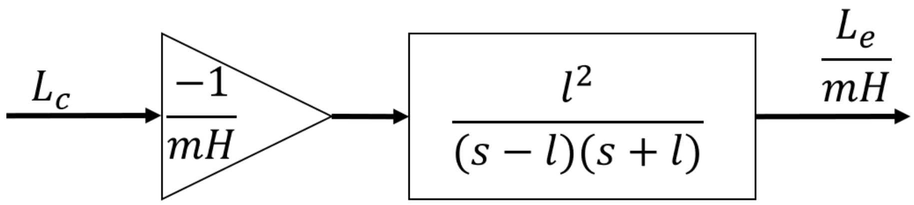
The desired angular momentum, L^{des}, determines the fixed point of the Poincaré map and hence the walking speed of the robot. While varying L^{des} causes the walking speed to change, the analysis of the controller has only been presented for a constant value of L^{des}. Reference [109] analyzes gait transitions in the formalism of the hybrid zero dynamics when L^{des} is switched "infrequently", meaning the closed-loop system is moving from a neighborhood of one periodic orbit to another. References [110, 111] generalize tools from Input-to-State Stability (ISS) of ODEs to the case of hybrid models. These results apply to time-varying L^{des}. The experimental work reported in

Sec. 5.5 includes examples of rapidly varying L^{des}, turn direction, and ground height.

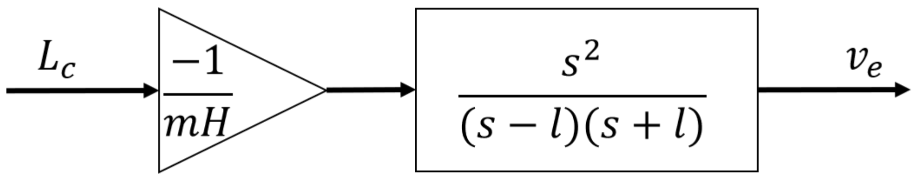
4.4.6 Varying Center of Mass Height

We have seen that the difference between the zero dynamics of a real robot and a pendulum model is the term related to L_c. In previous sections, we have shown that the L_c term has very little effect on the L dynamics when z_c is constant. This observation can be extended to the case when z_c is not constant but virtually constrained by (x_c, L, τ).

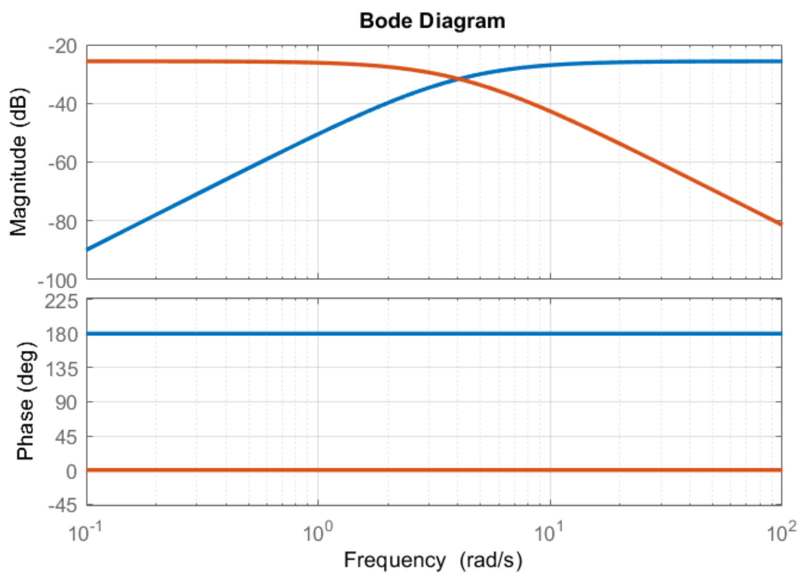
In Fig. 4.11, we illustrate that when the z_c is a function of time, the pendulum dynamics can still be used to predict accurately the zero dynamics of Rabbit.



(a) ALIPM

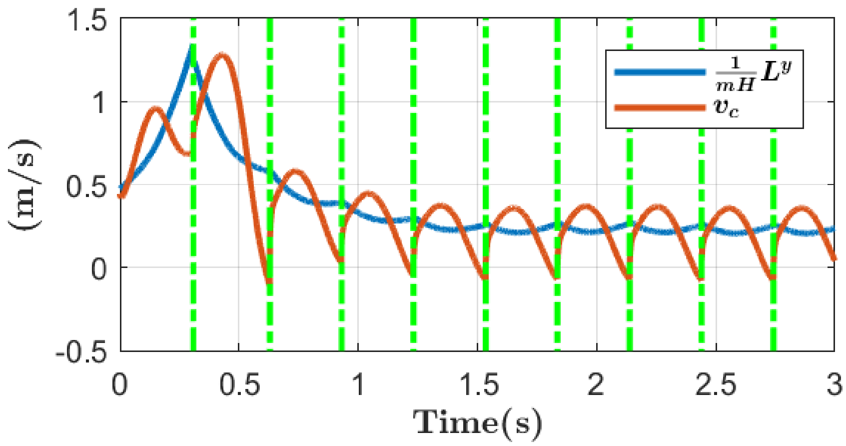


(b) LIPM

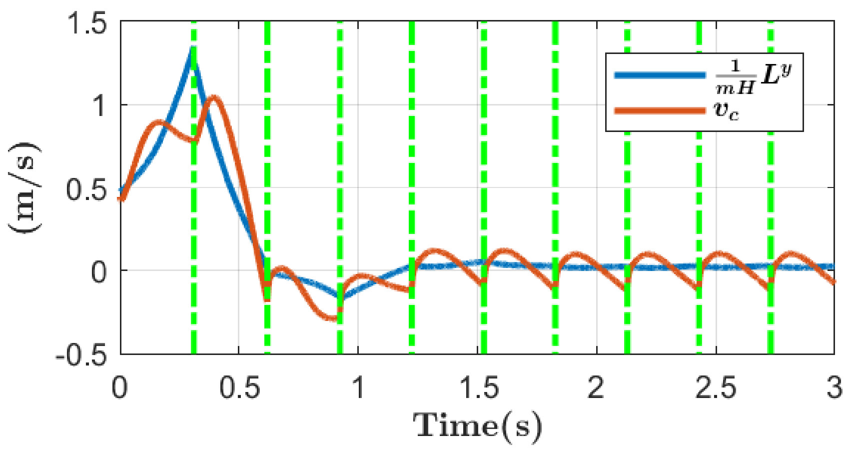


(c) Bode Plot

Figure 4.6: How neglecting L_c and $\dot{L}_c$ generates errors in ALIPM and LIPM. Note the low-pass (ALIPM in **red**) vs high-pass (LIPM in **blue**) nature of the respective transfer functions.

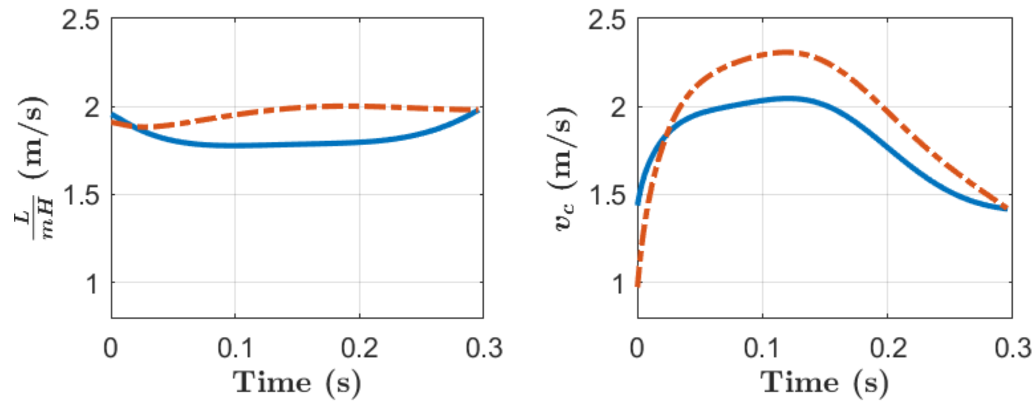


(a) LIP controller



(b) ALIP controller

Figure 4.7: Simulation results of Rabbit with controllers based on LIP and ALIP, following an identical design philosophy, based on foot placement. The details of the controller are described in Sec 5.2 and Sec 5.3. The controller based on the ALIP model is much more effective in regulating velocity to zero.



(a) Rabbit L prediction (b) Rabbit v_c prediction

Figure 4.8: Comparison of the ability to predict velocity vs angular momentum at the end of a step in a model with ankle torque $u_a = 30\sin(2\pi\tau/T)$, where τ varies from 0 to T during a step. The instantaneous values are shown in **blue** and the predicted value at the end of the step is shown in **red**. Because ankle torque is an input, we assume its trajectory is known when making predictions. For comparison purposes with Fig. 4.4, the ankle torque is chosen to be sufficiently large so that gravity is no longer dominant in $\dot{L} = mgx_{\mathrm{c}} + u_a$ and the trajectory of L is no longer convex.

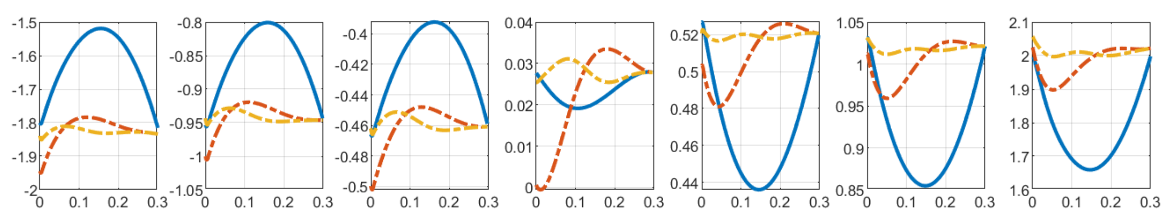


Figure 4.9: Prediction of angular momentum about the contact point at the end of a step, when Rabbit is walking at different speed. The **blue** line is the actual evolution of L in the simulation. The **red** line is the predicted value of L at step end when assuming $L_c = 0$. The **yellow** line utilizes a predicted trajectory for L_c. The x-axis is time in seconds. The **yellow** prediction is not perfectly flat because of fitting error in $L_c(t)$ and slight variation of CoM height in the simulation.

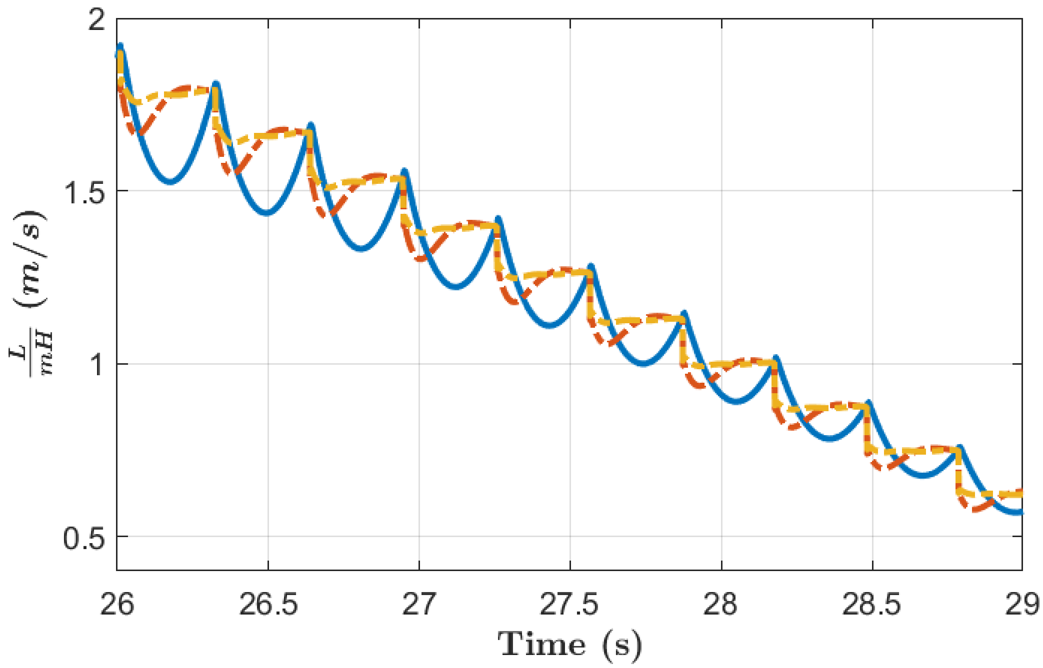


Figure 4.10: Though we make the bold assumption that the evolution of $L_c(t)$ over a step depends only on $L(0)$, the value of L at the beginning of a step, the improvement in the one-step-ahead prediction of L persists as walking speed decreases over three seconds from approximately 1.6 m/s to 0.6 m/s. The **blue** line is $L/(mH)$, the **red** line is the estimated value of $L/(mH)$ a the end of the current step when L_c is ignored, and the **yellow** line is the estimated value of $L/(mH)$ at the end of the current step when L_c is estimated each step from $L(0)$. The x-axis is time in seconds.

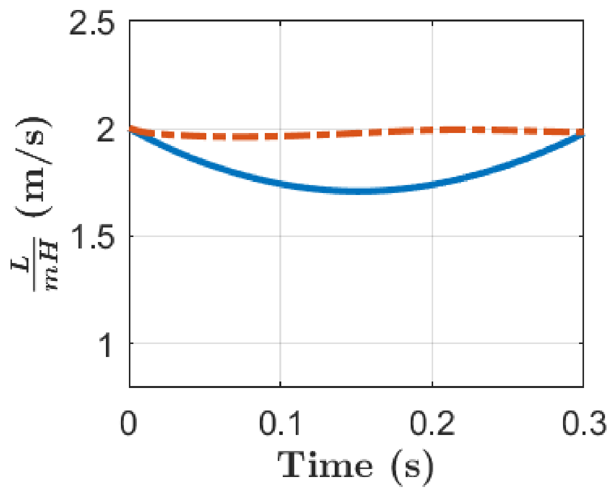


Figure 4.11: Trajectory of L and its prediction in a simulation of Rabbit. The instantaneous values are shown in **blue** and the predicted value at the end of the step is shown in **red**. In the prediction of L, the virtual constraint on center of mass height for the model model and for Rabbit are set to $z_c = 0.6 + 0.05\sin(\frac{T}{2\pi}\tau - \frac{\pi}{2}) + 0.05$, where T is the step time. Large z_c oscillations often occur in running. Here, we modify the ground model to pin the stance foot to the ground, so that we can impose a non-trivial z_c oscillation in periodic walking.

CHAPTER V

Angular Momentum Based Walking Controller

In Chapter IV we explained that the angular momentum about the contact point(L) can represent the walking status of a bipedal robot. We demonstrated that the evolution of L resembled that of a pendulum model and thus we could predict this state quickly and accurately with the closed form solution of ALIP. In this chapter, we design a controller that utilizes this prediction and calculates the foot placement to achieve a desired L at the end of the next step. Detailed experimental implementation for a 3D Cassie Robot is also discussed, including reference trajectory generation, state estimation for angular momentum, inverse kinematics, and a passivity-based controller. We demonstrate the robustness and agility of Cassie in both simulation and experiments. The controller code can be found at [112].

5.1 General Biped Balancing Mechanism

A bipedal robot is a hybrid under-actuated system. It is hybrid because the dynamics change when the model switches between left and right single support phases. The single support phase is under-actuated because of the limited foot size and unilateral ground constraint. Thus, the ground reaction force a bipedal robot can exert is limited, making it different from a fully-actuated robot arm whose based in pinned on the ground. As a result, the ability to keep balance during the single support phase is weak. However, humans have good balancing abilities when walking. This is because in the hybrid system we can set initial conditions when we switch between two phases. Though there isn't much we can do during the single support phase, we can decide how the phase starts. The initial condition for the single support phase has a dominant effect in balancing and we are allowed to set it only once per step. So it is important to set a proper initial condition. Setting an initial condition can be an easy task if we have a map associating the initial condition and its consequence.

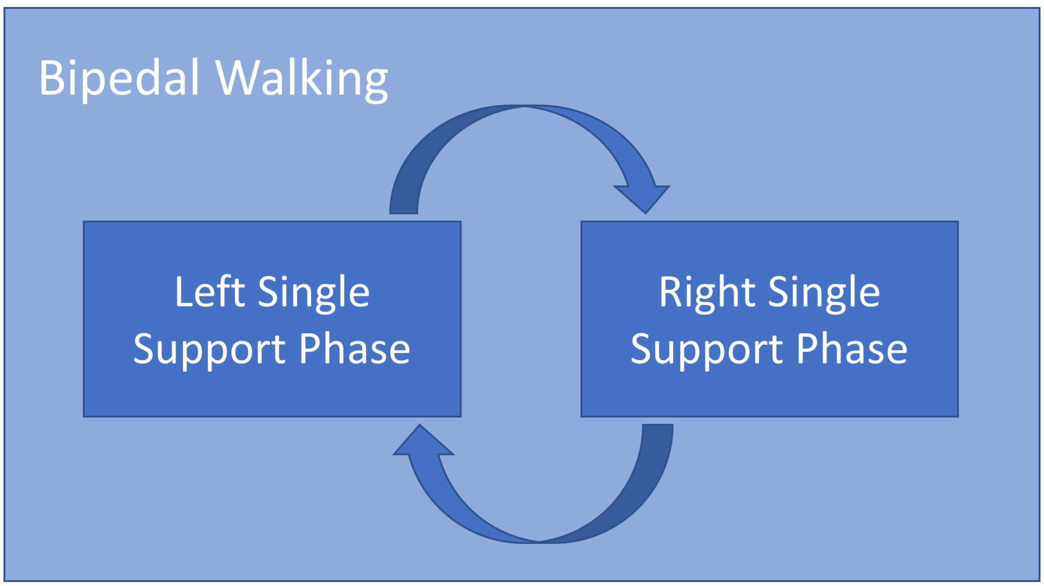


Figure 5.1: Bipedal Walking is a hybrid system composed of two weakly actuated continuous single support phases. Though during each single support phase the balancing ability is weak, the switching between these two phases provides a good balance ability in the hybrid system.

In the previous chapter, we argued that Angular Momentum about Contact Point (L) is a good indicator of robot status. L has relative degree three and is hardly affected by motor torque during a step, faithfully reflecting the underactuated nature of the single support phase. What's more, we proposed the ALIP model, a simplified model with good accuracy which provides the map associating the initial condition and the consequential L evolution.

In the following section, we will discuss how to use foot placement to obtain desired L. Here we briefly discuss general methods to regulate L, or "maintain balance". First, notice that during the single support phase, the dynamics of the robot (sagittal plane) can be written as:

$$\begin{cases} \dot{x} = \dfrac{L}{mz} + \dfrac{\dot{z}}{z}x - \dfrac{L_c}{mz} \\ \dot{L} = mgx + u_A. \end{cases} \tag{5.1}$$

- **Single Support Phase.** L can be regulated during the single support phase through L_c and z. L has a relative degree of three to related input. When on a tightrope, this is the only method we could use to keep balance. The effect of L_c is so small for a biped with normal morphology that even expert acrobats need to hold a long pole when they are walking on the tightrope. L can be also regulated by ankle torque u_A with a relative degree of one. The effect of u_A is stronger but still weak because of the unilateral constraint on the limited-size feet.

- **Hybrid System**. We can regulate L more effectively by exploiting the hybrid property of bipedal walking. We can set the $x(0)$ with foot placement, which we will discuss in detail in the following section. We can also set $L(0)$ with foot placement and step end vertical CoM velocity, using the angular momentum transfer formula: $L_2 = L_1 + p_{2\to1} \times mv$. We can also plan step time to regulate L, which is critical when foothold positions are fixed, such as stairs or stepping stones.

5.2 Stabilizing the ALIP Model

In this section, we provide a means to regulate angular momentum about the contact point to approximately achieve a desired walking speed. Specifically, the ALIP model (4.20) is used to form a one-step ahead prediction of angular momentum,

L. In combination with the angular momentum transfer formula (4.15), a feedback law results for where to place the swing foot at the end of the current step so as to achieve a desired angular momentum at the end of the ensuing step. In legged robot locomotion, this is typically called "foot placement control" [113, 114, 41, 57].

5.2.1 Gait assumptions

When designing the foot placement controller, we assume the gait of the robot is controlled such that:

(a) the height of the center of mass is constant, that is $z_c \equiv H > 0$;

(b) each step has constant duration $T > 0$; and

(c) a desired swing leg horizontal position, $p^{x\,\mathrm{des}}_{\mathrm{sw}\to\mathrm{CoM}}$, can be achieved at the end of the step.

We'll explain how to accomplish these objectives via the method of virtual constraints in Sections 4.4 and 5.3.

5.2.2 Notation

We distinguish among the following time instances when specifying the control variables.

- T is the step time.
- T_k is the time of the kth impact and thus equals kT.
- T_k^- is the end time of step k, so that
- T_k^+ is the beginning time of step $k+1$ and T_{k+1}^- is the end time of step $k+1$.
- $(T_k^- - t) = (T - \tau(t))$ is the time until the end of step k.

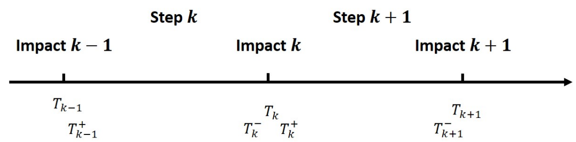


Figure 5.2: For a given time, T_k, the notation T_k^- means that we are evaluating a function as a limit from the left of T_k, while T_k^+ means we are taking a limit from the right. This is compatible with how trajectories are defined for the hybrid model (4.6).

The superscripts + and − on T_k are necessary because of the (potential) jump in a trajectory's values from the impact map; see [35]. As shown in Fig. 5.2, $x(T_K^-)$ is the limit from the left of the model's solution at the time of impact, in other words it's value "just before" impact, while $x(T_K^+)$ is the limit from the right of the model's solution at the time of impact, in other words it's value "just after" impact.

With this notation, the reset map for the ALIP becomes

$$\begin{aligned} x_c(T_k^+) &= p^x_{\text{sw}\to\text{CoM}}(T_k^-) \\ L(T_k^+) &= L(T_k^-), \end{aligned} \tag{5.2}$$

after noting that (4.15) simplifies to L being constant across impacts when the ground is level and the vertical velocity of the center of mass is zero.

We remind the reader that

- $p_{\text{st}\to\text{CoM}}$, $p_{\text{sw}\to\text{CoM}}$ are the vectors emanating from stance/swing foot to the robot's center of mass. The stance foot defines the current contact point, while the swing foot is defining the point of contact for the next impact and is therefore a control variable.

- Also, for the implementation of the control law on the 3D biped Cassie in Sect. 5.3, we need to distinguish between L^y and L^x, the y and x components of the angular momentum (sagittal and frontal planes), respectively.

5.2.3 Foot placement in longitudinal direction

The control objective will be to place the swing foot at the end of the current step so as to achieve a desired value of angular momentum at the end of the ensuing step. The need to regulate the angular momentum one-step ahead of the current step, instead of during the current step, is because in (4.20) L is passive without ankle torque, in other words, it is not affected by the control actions of the current step. The only way to act on its states is through the transition events.

The closed-form solution of (4.20) at time T and initial time t_0 is

$$\begin{bmatrix} x_c(T) \\ L^y(T) \end{bmatrix} = A(T - t_0) \begin{bmatrix} x_c(t_0) \\ L^y(t_0) \end{bmatrix}, \tag{5.3}$$

where

$$A(t) = \begin{bmatrix} \cosh(\ell t) & \sinh(\ell t)/(mH\ell) \\ mH\ell \sinh(\ell t) & \cosh(\ell t) \end{bmatrix}$$

and $\ell = \sqrt{\frac{g}{H}}$.

In the following, we breakdown the evolution of L^y from t to T_{k+1}^-, for three key time intervals or instances with the aim of forming a one-step-ahead estimate of angular momentum about the contact point.

5.2.3.1 From t to T_k^-

From the second row of (5.3), an *estimate* for the angular momentum about the contact point at the end of current step, $\widehat{L}^y(T_k^-, t)$, can be continuously updated by

$$\begin{aligned} \widehat{L}^y(T_k^-, t) =& mH\ell \sinh(\ell(T_k^- - t))x_c(t) \\ &+ \cosh(\ell(T_k^- - t))L^y(t). \end{aligned} \tag{5.4}$$

Forming the running estimate in (5.4), versus a fixed estimate based on the values of x_c and L^y at the beginning of the step, allows disturbances to be taken into account.

5.2.3.2 From T_k^- to T_k^+

This involves applying the reset map (5.2), yielding

$$x_c(T_k^+) = p^x_{\text{sw}\to\text{CoM}}(T_k^-) \tag{5.5}$$

$$\widehat{L}^y(T_k^+, t) = \widehat{L}^y(T_k^-, t). \tag{5.6}$$

5.2.3.3 From T_k^+ to T_{k+1}^-

Similar to (5.4), the angular momentum at the end of the next step is estimated by

$$\widehat{L}^y(T_{k+1}^-, t) = mH\ell \sinh(\ell T) x_c(T_k^+, t) + \cosh(\ell T)\widehat{L}^y(T_k^+, t). \tag{5.7}$$

Solving (5.4)-(5.7) so that

$$\widehat{L}^y(T_{k+1}^-, t) = L^{y\ \text{des}},$$

a desired value of angular momentum at the end of a step, (which can be obtained by $L^{y\ \text{des}} = mHv^{x\ \text{des}}$), yields a formula for the desired swing foot position at the end of the *current* step, given the value of desired angular momentum at the end of the *next* step,

$$p^{x\ \text{des}}_{\mathbf{sw}\to\mathbf{CoM}}(T_k^-, t) := \frac{L^{y\ \text{des}} - \cosh(\ell T)\widehat{L}^y(T_k^-, t)}{mH\ell \sinh(\ell T)}. \tag{5.8}$$

Remark: Instead of the deadbeat control (5.8), it is possible to asymptotically approach a desired value of L^{des} with the control law

$$\begin{aligned} p^{x\ \text{des}}_{\mathbf{sw}\to\mathbf{CoM}}(T_k^-, t) := {} & \frac{1-\alpha}{mH\ell \sinh(\ell T)} L^{y\ \text{des}} \\ & + \frac{\alpha - \cosh(\ell T)}{mH\ell \sinh(\ell T)} \widehat{L}^y(T_k^-, t), \end{aligned} \tag{5.9}$$

which achieves

$$(L^{y\ \text{des}} - \widehat{L}^y(T_{k+1}^-, t)) = \alpha(L^{y\ \text{des}} - \widehat{L}^y(T_k^-, t)) \tag{5.10}$$

for $\alpha \in [0, 1)$. Hence, for $\alpha = 0$, (5.9) reduces to (5.8).

5.2.4 Stability Analysis of the ALIP for L^{des}

Consider the ALIP model (4.20) with zero ankle torque and rest map (5.2). To compute the Poincaré map, we take the Poincaré section as $S := \{(x_c, L, \tau) \mid \tau = 0^+\}$, which is the set of states just after impact. Computing (4.20) over one step and using swing foot position with respect to the center of mass, u_{fp}, as an input, yields,

$$\begin{bmatrix} x_c(T^+) \\ L(T^+) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ mH\ell \sinh(\ell T) & \cosh(\ell T) \end{bmatrix} \begin{bmatrix} x_c(0^+) \\ L^(0^+) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_{fp}(T^-). \tag{5.11}$$

Next, applying the feedback law (5.9) with L^{des} a constant results in the Poincaré map being

$$\begin{aligned} \begin{bmatrix} x_c(T^+) \\ L(T^+) \end{bmatrix} = &\begin{bmatrix} \alpha - \cosh(\ell T) & \frac{(\alpha - \cosh(\ell T))\cosh(\ell T)}{mH\ell \sinh(\ell T)} \\ mH\ell \sinh(\ell T) & \cosh(\ell T) \end{bmatrix} \begin{bmatrix} x_c(0^+) \\ L(0^+) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{1-\alpha}{mH\ell \sinh(\ell T)} \\ 0 \end{bmatrix} L^{des}. \end{aligned} \tag{5.12}$$

The Poincaré map has fixed point

$$\begin{bmatrix} x_c^* \\ L^* \end{bmatrix} = \begin{bmatrix} \frac{1-\cosh(\ell T)}{mH\ell \sinh(\ell T)} L^{des} \\ L^{des} \end{bmatrix} \tag{5.13}$$

independent of α and the eigenvalues of the Poincaré map are $(\alpha, 0)$. Hence, for all $0 \leq \alpha < 1$, the fixed point is exponentially stable and moreover, (5.12) is bounded-input bounded-state stable with respect to the command, L^{des}.

5.2.5 Lateral Control and Turning

From (4.17), the time evolution of the angular momentum about the contact point is decoupled about the x- and y-axes. Therefore, once a desired angular momentum at the end of next step is given, Lateral Control is essentially identical to Longitudinal Control and (5.8) can be applied equally well in the lateral direction.[1] **The question becomes how to decide on $\mathbf{L^{x\ des}(T_{k+1}^-)}$,** since it cannot be simply set to zero for walking with a non-zero stance width.

For walking in place or walking with zero average lateral velocity, it is sufficient to obtain $L^{x\ \mathrm{des}}$ from a periodically oscillating LIP model,

$$L^{x\ \mathrm{des}}(T_{k+1}^-) = \pm\frac{1}{2}mHW\frac{\ell\sinh(\ell T)}{1+\cosh(\ell T)}, \tag{5.14}$$

where W is the desired step width. The sign is positive if next stance is left stance and negative if next stance is right stance. Lateral walking can be achieved by adding an offset to $L^{x\ \mathrm{des}}$.

To enable turning, we assume a target direction is commanded and associate a frame to it by aligning the x-axis with the target direction while keeping the z-axis vertical. To achieve turning, we then define the desired angular momentum $L^{y\ \mathrm{des}}$ and $L^{x\ \mathrm{des}}$ in the new frame and use the hip yaw-motors to align the robot in that direction.

5.3 Integrating Virtual Constraints and Angular-Momentum-based Foot Placement

In this section we generate virtual constraints for a 3D robot such as Cassie. As in [115], we leave the stance toe passive. Consequently, there are nine (9) control

[1] With a slight difference in the sign due to $L^x = -mgy_c$.

variables, listed below from the top of the robot to the end of the swing leg,

$$h_0 = \begin{bmatrix} \text{torso pitch} \\ \text{torso roll} \\ \text{stance hip yaw} \\ \text{swing hip yaw} \\ p^z_{\text{st}\to\text{CoM}} \\ p^x_{\text{sw}\to\text{CoM}} \\ p^y_{\text{sw}\to\text{CoM}} \\ p^z_{\text{sw}\to\text{CoM}} \\ \text{swing toe absolute pitch} \end{bmatrix}. \tag{5.15}$$

For later use, we denote the value of h_0 at the beginning of the current step by $h_0(T^+_{k-1})$. When referring to individual components, we'll use $h_{03}(T^+_{k-1})$, for example.

We first discuss variables that are constant. The reference values for torso pitch, torso roll, and swing toe absolute pitch are constant and zero, while the reference for $p^z_{\text{st}\to\text{CoM}}$, which sets the height of the CoM with respect to the ground, is constant and equal to H.

We next introduce a phase variable

$$s := \frac{t - T^+_{k-1}}{T} \tag{5.16}$$

that will be used to define quantities that vary throughout the step to create "leg pumping" and "leg swinging". The reference trajectories of $p^x_{\text{sw}\to\text{CoM}}$ and $p^y_{\text{sw}\to\text{CoM}}$ are defined such that:

- at the beginning of a step, their reference value is their actual position;
- the reference value at the end of the step implements the foot placement strategy in (5.8); and
- in between a half-period cosine curve is used to connect them, which is similar

to the trajectory of an ordinary (non-inverted) pendulum.

The reference trajectory of $p^z_{\text{sw}\to\text{CoM}}$ assumes the ground is flat and the control is perfect:

- at mid stance, the height of the foot above the ground is given by z_{CL}, for the desired vertical clearance.

The reference trajectories for the stance hip and swing hip yaw angles are simple straight lines connecting their initial actual position and their desired final positions. For walking in a straight line, the desired final position is zero. To include turning, the final value has to be adjusted. Suppose that a turn angle of ΔD^{des}_k radians is desired. One half of this value is given to each yaw joint:

- $+\frac{1}{2}\Delta D^{\text{des}}_k \to$ swing hip yaw; and
- $-\frac{1}{2}\Delta D^{\text{des}}_k \to$ stance hip yaw

The signs may vary with the convention used on other robots.

The final result for Cassie Blue is

$$h_{\text{d}}(s) := \begin{bmatrix} 0 \\ 0 \\ (1-s)h_{03}(T^+_{k-1}) + s(-\frac{1}{2}(\Delta D_k)) \\ (1-s)h_{04}(T^+_{k-1}) + s(\frac{1}{2}(\Delta D_k)) \\ H \\ \frac{1}{2}\left[(1+\cos(\pi s))h_{06}(T^+_{k-1}) + (1-\cos(\pi s))p^{x\ \text{des}}_{\text{sw}\to\text{CoM}}(T^-_k)\right] \\ \frac{1}{2}\left[(1+\cos(\pi s))h_{07}(T^+_{k-1}) + (1-\cos(\pi s))p^{y\ \text{des}}_{\text{sw}\to\text{CoM}}(T^-_k)\right] \\ 4z_{cl}(s-0.5)^2 + (H - z_{CL}); \\ 0 \end{bmatrix}. \tag{5.17}$$

When implemented with an Input-Output Linearizing Controller[2] so that h_0 tracks h_d, the above control policy allows Cassie to move in 3D in simulation.

[2]The required kinematic and dynamics functions are generated with FROST [116].

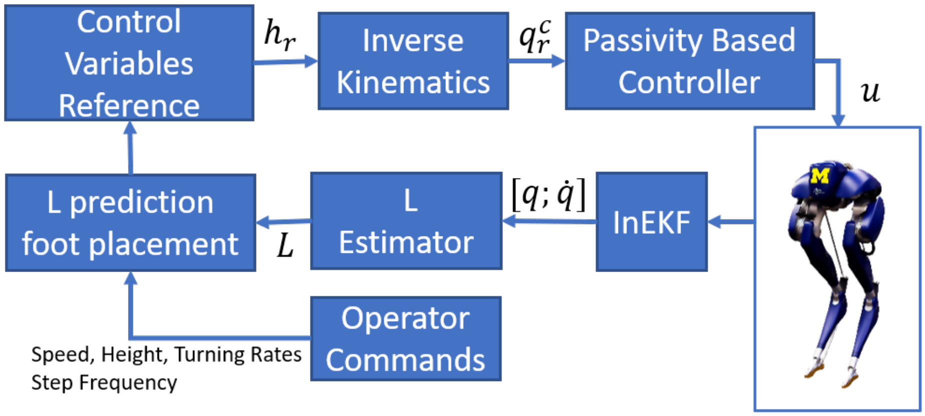


Figure 5.3: Block diagram of the implemented controller.

5.4 Practical Implementation on Cassie

This section resolves several issues that prevent the basic controller from being implemented on Cassie Blue.

5.4.1 IMU and EKF

In a real robot, an IMU and an EKF are needed to estimate the linear position and rotation matrix at a fixed point on the robot, along with their derivatives. Cassie uses a VectorNav IMU. We used the Contact-aided Invariant EKF developed in [117, 118] to estimate the torso velocity. With these signals in hand, we could estimate angular momentum about the contact point.

5.4.2 Filter for Angular Momentum

Angular Momentum about the contact toe could be computed directly from estimated $[q, \dot{q}]$, but it is noisy. We used a Kalman Filter to improve the estimation. The models we used are

$$\begin{aligned} \text{Prediction: } & L^y(k) = AL^y(k-1) + Bu(k) + \delta \\ \text{Correction: } & L^y{}_{\text{obs}}(k) = CL^y(k) + \epsilon \end{aligned} \tag{5.18}$$

where $A = B = C = 1$, $u(k) = (mgx_{\mathrm{c}}(k) + u_a(k))\Delta T$.

The update formula for angular momentum is

$$L^y(k) = (I - K(k)C)(AL^y(k-1) + Bu(k)) + K(k)L^y_{\text{obs}}(k) \tag{5.19}$$

The Kalman Gain $K(k)$ is obtained following the algorithm described in [119, Sec 3.2].

5.4.3 Inverse Kinematics

Input-Output Linearization does not work well in experiments[115, 120, 42]. To use a passivity-based controller for tracking that is inspired by [103], we need to convert the reference trajectories for the variables in (5.15) to reference trajectories for Cassie's actuated joints,

$$q^{\mathrm{a}} = \begin{bmatrix} \text{torso pitch} \\ \text{torso roll} \\ \text{stance hip yaw} \\ \text{swing hip yaw} \\ \text{stance knee pitch} \\ \text{swing hip roll} \\ \text{swing hip pitch} \\ \text{swing knee pitch} \\ \text{swing toe pitch} \end{bmatrix}. \tag{5.20}$$

Iterative inverse kinematics is used to convert the controlled variables in (5.15) to the actuated joints. Denote the un-actuated joint as q^{u}, then there exist a relation between the control variable h_0, q_0^{a} and q_0^{u}

$$
\begin{aligned}
h_0 &= f(q_0^{\mathrm{a}}, q_0^{\mathrm{u}}) \\
\dot{h}_0 &= J^{\mathrm{a}}(q_0^{\mathrm{a}}, q_0^{\mathrm{u}})\dot{q}_0^{\mathrm{a}} + J^{\mathrm{u}}(q_0^{\mathrm{a}}, q_0^{\mathrm{u}})\dot{q}_0^{\mathrm{u}}
\end{aligned}
\tag{5.21}
$$

We have discussed how to generate h_d and $\dot{h}_d$ in Sect. 5.3; here we want to find the corresponding q_d^{a} and $\dot{q}_d^{\mathrm{a}}$, such that

$$
\begin{aligned}
h_d &= f(q_d^{\mathrm{a}}, q_0^{\mathrm{u}}) \\
\dot{h}_d &= J^{\mathrm{a}}(q_d^{\mathrm{a}}, q_0^{\mathrm{u}})\dot{q}_d^{\mathrm{a}} + J^{\mathrm{u}}(q_d^{\mathrm{a}}, q_0^{\mathrm{u}})\dot{q}_0^{\mathrm{u}}
\end{aligned}
\tag{5.22}
$$

q_d^{a} and q_d^{a} are obtained iteratively with the following algorithm, the initial guess can be q_0^{a} or q_d^{a} from the previous iteration. The update at each loop is capped by max_update to avoid large linearization error.

Algorithm 1 Iterative Inverse Kinematics

$q_d^{\mathrm{a}} \leftarrow$ Initial Guess
while $||h_d - f(q_d^{\mathrm{a}}, q_0^{\mathrm{u}})|| >$threshold **do**
  update $\leftarrow J^{\mathrm{a}}(q_d^{\mathrm{a}}, q_0^{\mathrm{u}})^{-1}(h_d - f(q_d^{\mathrm{a}}, q_0^{\mathrm{u}}))$
  $q_d^{\mathrm{a}} \leftarrow$ median(update, max_update, - max_update)
end while
$\dot{q}_d^{\mathrm{a}} \leftarrow J^{\mathrm{a}}(q_d^{\mathrm{a}}, q_0^{\mathrm{u}})^{-1}(\dot{h}_d - J^{\mathrm{u}}(q_d^{\mathrm{a}}, q_0^{\mathrm{u}})\dot{q}_0^{\mathrm{u}})$

5.4.4 Passivity-based Controller

In this subsection, we discuss how to implement a Passivity-based Controller for joint-level tracking on Cassie.

Passivity-based control seeks feedback that renders the closed-loop system passive and thus stabilizes the equilibrium point or periodic orbit. Sadeghian et al[103] applied this technique to a planar robot in simulation and demonstrated that it can make the trajectories asymptotically converge to reference trajectories defined by holonomic virtual constraint. We adopt this method in our controller, and adapt it so that it can work with a floating-base model and a time-varying reference trajectory.

We first formulate the dynamics of Cassie as:

$$D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_{\mathrm{s}}^{\top}\tau_{\mathrm{s}} + J_{\mathrm{g}}(q)^{\top}\tau_{\mathrm{g}}, \tag{5.23}$$

with u the vector of motor torques, τ_{s} the spring torques, and τ_{g} the contact wrench. During the single-support phase, the blade-shaped foot on Cassie provides five holonomic constraints, leaving only the foot roll free. To simplify the problem, we also assume the springs are rigid, adding two constraints on each leg. These constraints leave the original 20-degree-of-freedom floating base model with 11 degrees of freedom.

The constraints mentioned above can be written, as

$$\begin{cases} J_{\mathrm{s}}\ddot{q} = 0 \\ J_{\mathrm{g}}(q)\ddot{q} + \dot{J}_{\mathrm{g}}(q)\dot{q} = 0. \end{cases} \tag{5.24}$$

Combining (5.23) and (5.24) yields the full model for Cassie in single support,

$$\underbrace{\begin{bmatrix} D & -J_{\mathrm{s}}^{\top} & -J_{\mathrm{g}}^{\top} \\ J_{\mathrm{s}} & 0 & 0 \\ J_{\mathrm{g}} & 0 & 0 \end{bmatrix}}_{\widetilde{D}} \underbrace{\begin{bmatrix} \ddot{q} \\ \tau_{\mathrm{s}} \\ \tau_{\mathrm{g}} \end{bmatrix}}_{f} + \underbrace{\begin{bmatrix} H \\ 0 \\ \dot{J}_{\mathrm{g}}\dot{q} \end{bmatrix}}_{\widetilde{H}} = \underbrace{\begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}}_{\widetilde{B}} u. \tag{5.25}$$

For simplicity, we assume that the components of q have already been ordered such that $q = [q_c, q_u]^T$, where q_c are the coordinates chosen to be controlled and q_u are the free coordinates. Define $\lambda = [q_u, \tau_s, \tau_g]^T$ and partition (5.25) as

$$\begin{cases} \widetilde{D}_{11}\ddot{q}_c + \widetilde{D}_{12}\lambda + \widetilde{H}_1 = \widetilde{B}_1 u \\ \widetilde{D}_{21}\ddot{q}_c + \widetilde{D}_{22}\lambda + \widetilde{H}_2 = \widetilde{B}_2 u, \end{cases} \tag{5.26}$$

where $\widetilde{D}_{11}$ is square. The vector λ can be eliminated from these equations, resulting in

$$\bar{D}\ddot{q}_c + \bar{H} = \bar{B}u, \tag{5.27}$$

where

$$\begin{aligned}
\bar{D} &= \widetilde{D}_{11} - \widetilde{D}_{12}\widetilde{D}_{22}^{-1}\widetilde{D}_{21} \\
\bar{H} &= \widetilde{H}_1 - \widetilde{D}_{12}\widetilde{D}_{22}^{-1}\widetilde{H}_2 \\
\bar{B} &= \widetilde{B}_1 - \widetilde{D}_{12}\widetilde{D}_{22}^{-1}\widetilde{B}_2.
\end{aligned}$$

Equation (5.27) is what we will focus on from here forward.

For the Passivity-based Controller, the error dynamics for $y := q_c - q_r$ is designed to be [121]

$$\bar{D}\ddot{y} + (\bar{C} + k_d)\dot{y} + k_p y = 0, \tag{5.28}$$

where $\bar{C}$ is the Coriolis/centrifugal matrix in $\bar{H}$ and it is chosen such that $\dot{\bar{D}} = \bar{C} + \bar{C}^\top$. From (5.27) and (5.28), we have

$$u = \bar{B}^{-1}(\bar{D}\ddot{q}_r + \bar{H}) - \bar{B}^{-1}(k_p y + (\bar{C} + k_d)\dot{y}). \tag{5.29}$$

Compared with a standard Input-Output Linearization controller, whose error dynamics and command torque are

$$\ddot{y} + k_d\dot{y} + k_p y = 0, \text{ and} \tag{5.30}$$

$$u = \bar{B}^{-1}(\bar{D}\ddot{q}_r + \bar{H}) - \bar{B}^{-1}\bar{D}(k_p y + k_d\dot{y}), \tag{5.31}$$

the passivity-based controller induces less cancellation of the robot's dynamics, and if k_p and k_d are chosen to be diagonal matrices, the tracking errors are approximately decoupled because, for Cassie, $\bar{B}^{-1}$ is close to diagonal. This controller provides improved tracking performance over the straight-up PD implementation in [115].

5.4.5 Springs

On the swing leg, the spring deflection is small and thus we are able to assume the leg to be rigid. On the stance leg, the spring deflection is non-negligible and hence requires compensation. While there are encoders on both sides of the spring to measure its deflection, direct use of this leads to oscillations. The deflection of the spring is instead estimated through a simplified model.

We assume the robot's links are massless and all the mass is concentrated at the CoM. During walking, the controller will try to keep the CoM height constant. With these assumptions, the Ground Reaction Force will point toward the CoM with a vertical component equal to gravity:

$$\mathrm{F} = mg \begin{bmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \\ 1 \end{bmatrix}. \tag{5.32}$$

Assuming the robot is static, with the principle of virtual work:

$$\tau = -J^T F \tag{5.33}$$

where τ is the general force acting on the coordinates q and $J = \frac{\partial p_{\text{stance foot}}}{\partial q}$.

With this roughly estimated load on the coordinates, we obtain the compression of the springs on the stance leg. The effect of the springs on the Leg Length (See Fig.3.1b) is then offset by modifying the stance knee reference position.

5.4.6 COM Velocity in the Vertical Direction

When Cassie's walking speed exceeds one meter per second, the assumption that $v^z_{\rm CoM} \approx 0$ breaks down due to spring deflection and imperfect low-level control, and thus (5.6) is no longer valid. Hence, we use

$$L^y(T_k^+) = L^y(T_k^-) + mv^z_{\rm c}(T_k^-)(p^x_{\rm sw \to CoM}(T_k^-) - p^x_{\rm st \to CoM}(T_k^-)). \tag{5.34}$$

From this, the foot placement is updated to

$$p^{x\ \rm des}_{\rm sw \to CoM}(T_k^-) = \frac{L^{y\ \rm des}(T^-_{k+1})}{m(H\ell \sinh(\ell T) - v^z_{\rm CoM}) \cosh(\ell T)} - \frac{(L^y(T_k^-) + mv^z_{\rm CoM}(T_k^-)p^x_{\rm st \to CoM}(T_k^-)) \cosh(\ell T)}{m(H\ell \sinh(\ell T) - v^z_{\rm CoM}) \cosh(\ell T)}. \tag{5.35}$$

Remark: In our experiments, $v^z_{\rm CoM}$ becomes negative at the end of a step when the robot is walking fast. If we still use (5.8) to decide foot placement, which is based on the reset map (5.6), in the lateral direction $L^x(T_k^+)$ will be overestimated. This in turn leads to the lateral foot placement being commanded further from the body than it should be. At the end of the next step, the magnitude of L^x will be larger than expected, requiring even further lateral foot placement from the body. The final phenomenon is abnormally large step width.

5.5 Experimental Results

The controller was implemented on Cassie Blue. The closed-loop system consisting of robot and controller was evaluated in a number of situations that are itemized below.

- **Walking in a straight line on flat ground.** Cassie could walk in place and walk stably for speeds ranging from zero to 2.1 m/s.

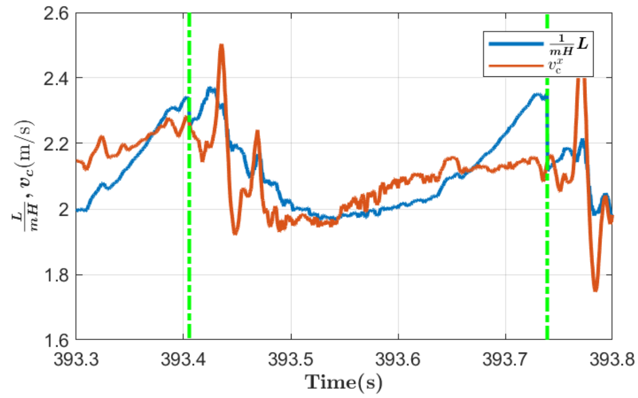
- **Diagonal Walking.** Cassie is able to walk simultaneously forward and sideways on grass, at roughly 1 m/s in each direction.

- **Sharp turn.** While walking at roughly 1 m/s, Cassie Blue effected a 90^o turn in six steps, without slowing down.

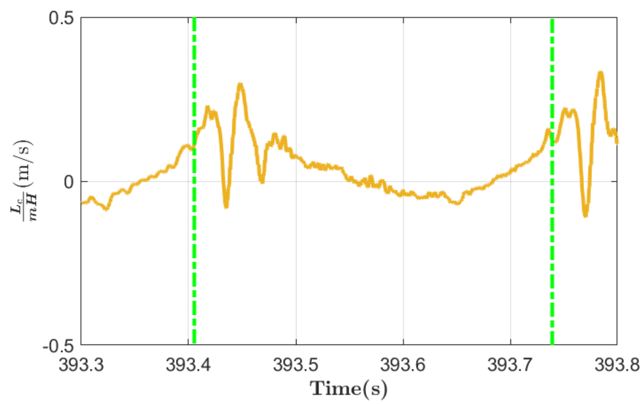
- **Rejecting the classical kick to the base of the hips.** Cassie was able to remain upright under "moderate" kicks in the longitudinal direction. The disturbance rejection in the lateral direction is not as robust as the longitudinal, which is mainly caused by Cassie's physical design: small hip roll motor position limits.

- **Finally we address walking on rough ground**. Cassie Blue was tested on the iconic Wave Field of the University of Michigan North Campus. The foot clearance was increased from 10 cm to 20 cm to handle the highly undulating terrain. Cassie is able to walk through the"valley" between the large humps with ease at a walking pace of roughly 0.75 m/s, without falling in all tests. The row of ridges running east to west in the Wave Field are roughly 60 cm high, with a sinusoidal structure. We estimate the maximum slope to be 40 degrees. Cassie is able to cross several of the large humps in a row, but also fell multiple times. On a more gentle, straight grassy slope of roughly 22 degrees near the laboratory, Cassie can walk up it with no difficulty with 20cm foot clearance.

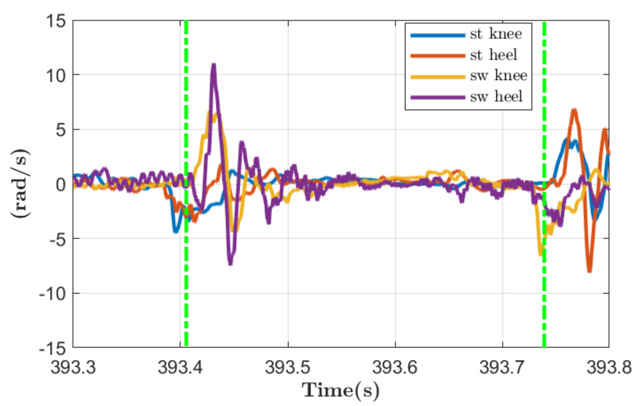
The experimental data is analyzed in Figs. 5.4 and 5.5. The figures support the advantages of using L to indicate robot status and the accuracy of ALIP model, as discussed in Sec 4.2 and 4.3.



(a) L and v_c



(b) L_c



(c) Spring deflection rates for stance and swing legs

Figure 5.4: Experimental data from Cassie walking forward at about 2m/s. To ensure a fair comparison, L is not smoothed by Kalman Filter described in Sect. 5.4.2. L, v_c and L_c are computed from the same states $[q, \dot{q}]$. v_c^x and L_c oscillates at the beginning of a step because of their relative degree one nature, in particular, they are heavily affected by the spring oscillation just after impact. L is mostly smooth because it has relative degree three, except near impact when the robot is in double support phase and L has relative degree one. The sudden jump in L at impact is caused by nonzero v_c^z, The smoothness difference shows another advantage of L: it can be used in feedback control without being heavily filtered.

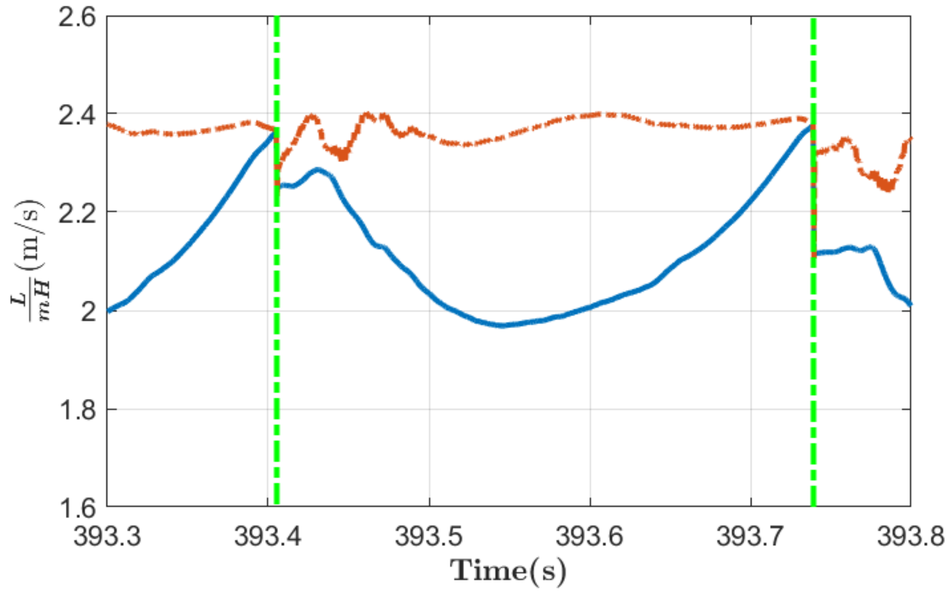
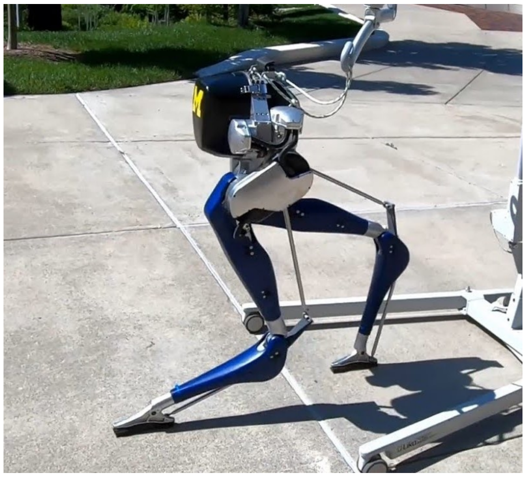


Figure 5.5: Prediction made in experiment from Cassie walking forward at about 2m/s. The instantaneous values are shown in **blue** and the predicted value at the end of the step is shown in **red**. The Kalman Filter described in Sect. 5.4.2 has been applied.



(a) Fast Walking

(b) Rough Terrain

(c) Disturbance Rejection

(d) A Fast 90 Degree Turn with a Long Stride

Figure 5.6: Images from several closed-loop experiments conducted with Cassie Blue and the controller developed in this paper. A short video compilation of these experiments is available in [3]. Longer versions can be found in [4].

CHAPTER VI

Angular Momentum Based Running Controller

Running is a more dynamic behavior than walking. The first running behavior on a legged robot was obtained with the heuristic Three-Part-Control method, dating back to 1984[9]. In [122, 123], running gaits are planned and stabilized with the simple inverted pendulum model. Running behaviors also emerged with the method of virtual constraints and Hybrid Zero Dynamics[124, 125]. In [33], a reinforcement learning method is applied to obtain a controller that generates many common bipedal gaits, including running, hopping, and skipping.

In this Chapter, we will design an angular momentum-based running controller, with the advantage of fast planning and good accuracy. The controller still assumes the L-dynamics of a pendulum model and a real robot are very similar. As shown in Sec. 4.4.6, the difference is that the CoM height will vary during the single support phase, and a flight phase will be included. We will discuss how to deal with these two new elements and demonstrate simulation and experimental results in the following sections.

6.1 Flight Phase Dynamics

Running is characterized by alternating single support phases and flight phases. The goal of the controller is still to regulate L, but during the flight phase there is no contact point and the dynamics of angular momentum about either foot are complicated: we no longer have the simple relation $\dot{L} = mgx_c + u_A$. Instead of predicting L during flight phase, we decompose L into linear CoM velocity $[\dot{x}_c, \dot{z}_c]$ and angular momentum about CoM L_c, and then predict those variables separately. At the end of the flight phase, we can compose these variables to obtain L. By doing so, we are able to find a relation between the angular momentum at the beginning and end of the flight phase.

According to the angular momentum transfer formula,

$$L = \dot{x}_c z_c - \dot{z}_c x_c + L_c. \tag{6.1}$$

During the flight phase,

$$\begin{aligned} \ddot{x}_c &= 0 \\ \ddot{z}_c &= g \\ \dot{L}_c &= 0. \end{aligned} \tag{6.2}$$

Note that gravity is the only external force during the flight phase.

We define the following notation:

- a, the contact point before the flight phase.
- b, the new contact after the flight phase.
- T_0, the time when the previous stance foot leaves the ground. T_0^- denotes the end of the previous stance phase and T_0^+ denotes the beginning of the flight phase.
- T_f, the time when the new stance foot touches the ground. T_f^- denotes the end of the flight phase and T_f^+ denotes the beginning of the new stance phase.

Our goal is to find the relation between $L_a(T_0^-)$ and $L_b(T_f^+)$. Since there is no impact at T_0,

$$L_a(T_0^-) = L_a(T_0^+). \tag{6.3}$$

Angular momentum about the contact point is invariant to the impact at contact, and thus

$$L_b(T_f^+) = L_b(T_f^-). \tag{6.4}$$

Decompose $L_a(T_0^+)$ and $L_b(T_f^-)$ by

$$\begin{aligned} L_a(T_0^+) &= \dot{x}_c(T_0^+)z_c(T_0^+) - \dot{z}_c(T_0^+)x_c(T_0^+) + L_c(T_0^+) \\ L_b(T_f^-) &= \dot{x}_c(T_f^-)z_c(T_f^-) - \dot{z}_c(T_f^-)x_c(T_f^-) + L_c(T_f^-). \end{aligned} \tag{6.5}$$

We have slightly abused notation here, because $x_c(T_0^+)$ is the CoM position defined in the frame a and $x_c(T_f^-)$ is defined in frame b. The velocity $\dot{x}_c$ is always defined in the world frame. Combining (6.3), (6.4), and (6.5) yields

$$\begin{aligned} L_b(T_f^-) =& L_a(T_0^+) + m\dot{x}_c(T_0^+)(z_c(T_f^-) - z_c(T_0^+)) \\ &+ m x_c(T_0^+)\dot{z}_c(T_0^+) - m x_c(T_f^-)\dot{z}_c(T_f^-), \end{aligned} \tag{6.6}$$

where $x_c(T_f^-)$ is the foot placement we must select to stabilize the running gait of the robot.

In (6.6), while $L_a(T_0^+)$ and $x_c(T_0^+)$ can be predicted during the previous stance phase and $z_c(T_f^-)$ $z_c(T_0^+)$ are chosen control variables, the quantity $\dot{x}_c(T_0^+)$ is neither easy to predict nor one of the control variables. The term $\dot{x}_c(T_0^+)$ disappears if we design the gait such that the robot enters and leaves the flight phase with the same CoM height, i.e., $z_c(T_f^-) = z_c(T_0^+)$. If we further assume $\dot{z}_c(T_f^-) = -\dot{z}_c(T_0^+)$, then

$$L_b(T_f^-) = L_a(T_0^+) + m(x_c(T_0^+) - x_c(T_f^-))\dot{z}_c(T_0^+). \tag{6.7}$$

Plugging (6.3) and (6.4) into (6.7) yields

$$L_b(T_f^+) = L_a(T_0^-) + m(x_c(T_0^-) - x_c(T_f^+))\dot{z}_c(T_0^-). \tag{6.8}$$

Equation (6.8) is the equation we need to describe how L evolves during flight phase

6.2 Single Support Phase with Varying Height

To obtain a flight phase in periodic running, the CoM height (z_c) has to vary during the single support phase, which differs from the constant height assumption we used in Chapters IV and V. In this Section, we will discuss how to obtain a feasible CoM height reference trajectory that enables running and meets contact constraints.

We represent the CoM height trajectory using a Bezier Curve with degree m,

$$z_c^r(s) = \sum_{i=0}^{m} \alpha_i B_{m,i}(s), \tag{6.9}$$

where $B_{m,i}(s) = \frac{n!}{i!(n-i)!} s^i (1-s)^i$, and s is the temporal phase variable defined as $s = \frac{t}{T_{\text{stance}}}$, which evolves from zero to one during the single support phase.

At the end and the beginning of the curve, the reference trajectory needs to satisfy certain constraints to enable running. If we make the same assumptions as the previous section that the robot enters and leaves the flight phase with the same

CoM height and velocity of the opposite sign, then the constraints will be:

$$\begin{aligned}
z_c^r(0) &= z_0 \\
z_c^r(1) &= z_0 \\
\frac{d}{dt} z_c^r(0) &= \frac{g}{2\ T_{\text{flight}}} \\
\frac{d}{dt} z_c^r(1) &= -\frac{g}{2\ T_{\text{flight}}} \\
\frac{d^2}{dt^2} z_c^r(0) &= 0 \\
\frac{d^2}{dt} z_c^r(1) &= 0,
\end{aligned} \tag{6.10}$$

where z_0 is the height when the robot enters or leaves the flight/stance phase and T_{flight} is the duration of the flight phase. These two parameters can either be determined by a higher-level planner or manually assigned by the control designer.

The CoM height trajectory also needs to satisfy the ground unilateral contact constraints. The vertical acceleration should be greater than -9.8 m/s^2 to avoid generating a negative vertical ground reaction force. We define N evenly spaced samples on the CoM height trajectory and impose the following constraint,

$$\frac{d^2}{dt^2} z_c^r(s) > -g \text{ for } s = \frac{1}{N}, \frac{2}{N}, \ldots \frac{N-1}{N}. \tag{6.11}$$

Finally, we seek to reduce the variation in acceleration to smooth the trajectory. The Bezier curve for the second derivative of $z_c^r(s)$ can be written as

$$\frac{d^2}{ds^2} z_c^r(s) = \sum_{i=0}^{m-2} \beta_i B_{m,i}(s), \tag{6.12}$$

where

$$\beta_i = n(n-1)(\alpha_{i+2} - 2\alpha_{i+1} + \alpha_i) \tag{6.13}$$

We formulate a Quadratic Program (QP) to obtain the vector of Bezier coefficients

β, namely

$$
\begin{aligned}
&\alpha^* := \min_{\alpha} \; Var(\beta) \\
&\textbf{subject to} \\
&\quad \text{Boundary Condition (6.10)} \\
&\quad \text{Unilateral Contact Constraint (6.11)} \\
&\beta \text{ depends on } \alpha \text{ per (6.13)},
\end{aligned}
\tag{6.14}
$$

where $Var(\beta)$ is the variance of the values $\{\beta_0, \ldots, \beta_{m-2}\}$.

6.3 Determining Foot Placement

In Section 5.2, foot placement is obtained with a closed-form solution. However, when the CoM height is time-varying, there is no general closed-form solution. Instead, we use the same method as [126] which partitions a phase into multiple small intervals and assumes the system is time-invariant within each interval. Then we construct an MPC problem and cast it in the form of a QP. We solve the QP to find a foot placement that will achieve the desired L at the end of the next single support phase.

The dynamics of a pendulum model with a varying CoM height can be written as,

$$
\begin{cases}
\dot{x}_c = \dfrac{L}{m z_c(t)} + \dfrac{\dot{z}_c(t)}{z_c(t)} x_c \\
\dot{L} = m g x_c + u_A.
\end{cases}
\tag{6.15}
$$

We partition the single support phase into N intervals, with $t_i = \frac{i}{N} T_{stance}$ denoting the start time of each interval, for $i \in \{0, 1, 2, ..., N\}$. Then we can approximate the dynamics of each interval by

$$\begin{cases} \dot{x}_c = \dfrac{L}{mz_c(t_i)} + \dfrac{\dot{z}_c(t_i)}{z_c(t_i)} x_c \quad \text{when } t \in [t_i, t_{i+1}) \\ \dot{L} = mgx_c + u_A. \end{cases} \tag{6.16}$$

On the basis of (6.16), we can discretize the stance phase,

$$X_{i+1} = A_i X_i + B u_{Ai}, \tag{6.17}$$

where $X_i = [x_c(t_i),\ L(t_i)]$.

To obtain a desired L at the end of the next stance phase, the optimization problem is formulated as

$$
\begin{aligned}
&\min_{u_{fp},\ u_A^{s1},\ u_A^{s2}} \quad ||X_N^{s2}(2) - L_{des}||_2 \\
&\textbf{subject to} \\
&\qquad \text{First stance Dynamics} \\
&\qquad\qquad X_{i+1}^{s1} = A_i^{s1} X_i^{s1} + B u_{Ai}^{s1} \\
&\qquad \text{Second stance Dynamics} \\
&\qquad\qquad X_{i+1}^{s2} = A_i^{s2} X_i^{s2} + B u_{Ai}^{s2} \\
&\qquad \text{Flight Phase Transition (from Eqn (6.8))} \\
&\qquad\qquad X_0^{s2}(2) = X_N^{s1}(2) + m(X_N^{s1}(1) - X_0^{s2}(1))\dot{z}_{cN}^{s1} \\
&\qquad \text{Foot Placement} \\
&\qquad\qquad u_{fp} = X_0^{s2}(1) \\
&\qquad \text{Current Measurement} \\
&\qquad\qquad X_0^{s1} = [x_c(t),\ L(t)] \\
&\qquad \text{Kinematics Constraint} \\
&\qquad\qquad x_c^{\text{lb}} < X_i(1) < x_c^{\text{ub}},
\end{aligned}
\tag{6.18}
$$

where the superscripts $s1$ and $s2$ denote the first stance phase and the second stance phase, respectively, u_{fp} is the foot placement we seek to achieve at the end of the flight phase. While X_i^{s2} is discretized over $[0,\ T_{\text{stance}}]$, X_i^{s1} is discretized over $[t,\ T_{\text{stance}}]$.

In the optimization problem, ankle torque u_A is an optional decision variable. In our implementation, we set u_A to zero.

6.4 Simulation Result

After obtaining the reference CoM height trajectory and foot placement, the implementation of the controller is basically the same as Sec. 5.3, with an added flight phase. The controller is evaluated on simulations of Rabbit and Cassie. Figures 6.1 and 6.2 show plots of L and v_c^x when Rabbit and Cassie are running forward and hopping in place. A sequence of stick figures is shown in Fig. 6.3.

For running, the target L at the end of stance is set to 4 m/s (normalized by mH_{nominal}, where the constant H_{nominal} is a rough estimate of the average CoM height). Cassie is able to reach this target speed with little error while Rabbit reaches steady-state speed with a tracking error of 0.3 m/s. The difference in behavior is because Cassie's leg mass is concentrated near the robot's CoM and generates small L_c when swinging, while Rabbit's leg mass is more evenly distributed along the leg and thus generates a larger L_c, resulting in a larger ALIP prediction error. Because of the tracking error, the L trajectories of Rabbit and Cassie are slightly different: Cassie has already reached its target L and thus the controller places the foot in front the CoM at the end of the flight phase, which results in a first-decreasing-then-increasing pattern in the ensuing stance phase. Rabbit, on the other hand, is always below the target L and thus the controller places the foot behind the CoM, trying to accelerate from the very beginning of the stance phase. Rabbit still fails to achieve the target L at the end because of the large L_c "disturbance". Though the patterns in the angular momentum are not exactly the same, they are both governed by $\dot{L} = mgx_c$ and thus convex, in contrast with the amorphous pattern seen in the linear velocity, v_c^x. The 0.3 m/s tracking error is not significant compared to the massive L_c generated by Rabbit's heavy leg swinging all the way from back to front in 0.15 seconds. The tracking error caused by omitting L_c can be alleviated by the method discussed in Sect. 4.3.4, which takes a nominal trajectory of L_c into consideration in the foot placement planning.

For hopping in place, both robots are able to hop high and obtain a long 0.5s flight phase. The leg still moves forward or backward when the robots are hopping in place (See Fig 6.3 (a) (c)) because the reference of the torso pitch is always upright and the controller has to swing the leg to maintain it. (The controller uses the non-landing leg to regulate torso pitch, the landing leg still focuses on achieving the planned foot placement to maintain balance.) To get a more natural or energy-efficient gait, the references for the controlled variables during the stance phase and the swing phase should be obtained by full-order model optimization rather than from simple hand-designed trajectories (5.17). However, being “natural” or “energy-efficient” is out of the scope of balancing and thus we do not discuss them further here.

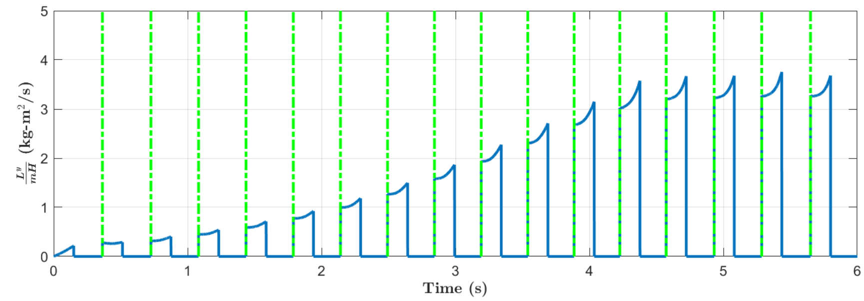
The simulations confirm that with a known rigid model and good joint level tracking, we can plan versatile and stable running gaits online in real-time using the ALIP model. However, the controller designed here ignores the motor torque and power constraints: the hopping with a 0.5s flight phase requires a very large motor torque that exceeds the torque limit we typically have on real robots.

6.5 Experimental Result

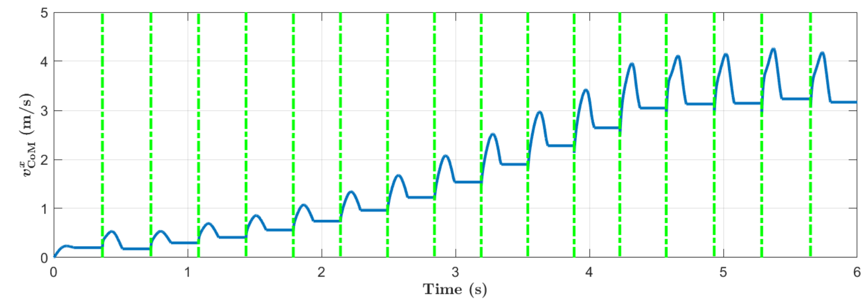
The controller is implemented on Cassie. The optimization problems (6.14) and (6.18) are formulated in Matlab using Casadi [127, 128]. An execution file compiled from autogenerated c-code is used to provide solutions. The QPs are solved at 1000 Hz in real-time.

In the experiment, we set the stance phase to 0.3s and the flight phase to 0.1s, which we found ensures that the motor torque will not exceed its limit. Cassie is able to hop in place, with an estimated flight phase of 0.05s. The actual flight time is shorter than desired because of the spring compliance on Cassie’s leg. Figure 6.4 shows a sequence of outtakes of Cassie while hopping.

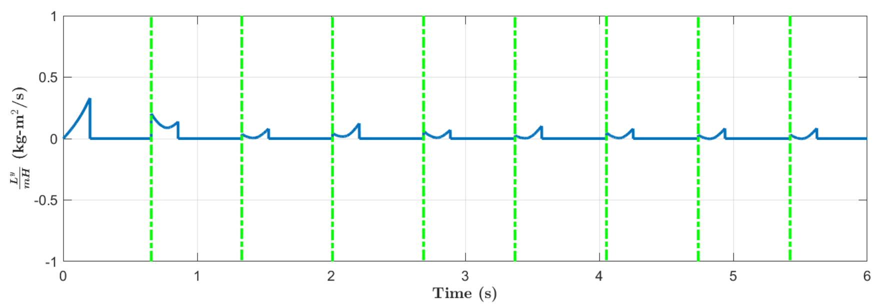
We gave a running forward command to Cassie in the experiment. An asymmetric gait emerged. Cassie ran with alternating large and small strides. Moreover, the gap in the stride length became larger as the speed increased and finally resulted in Cassie losing balance.



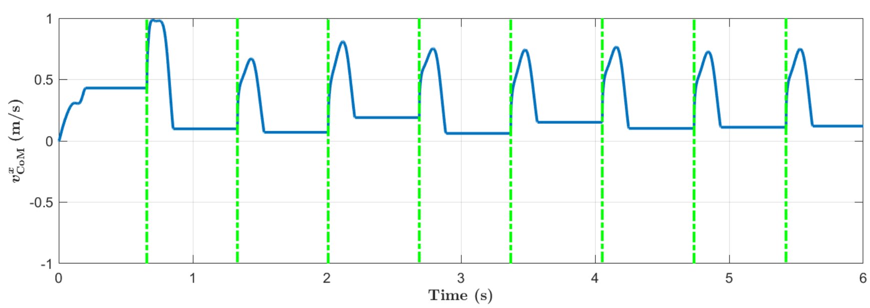
(a)



(b)

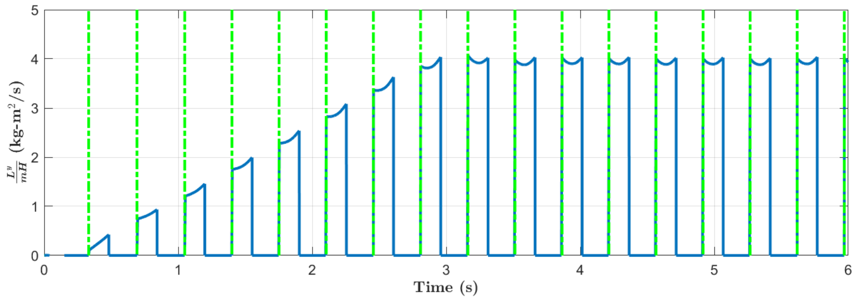


(c)

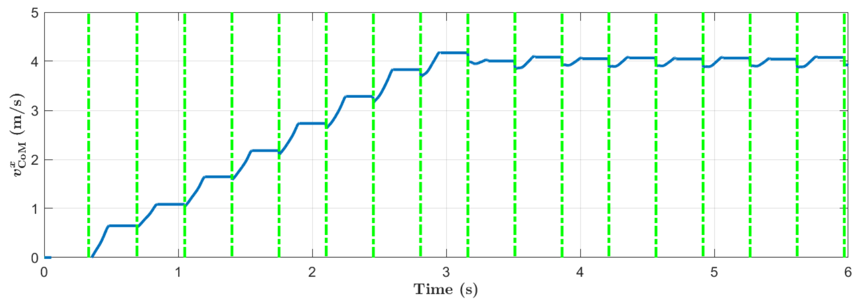


(d)

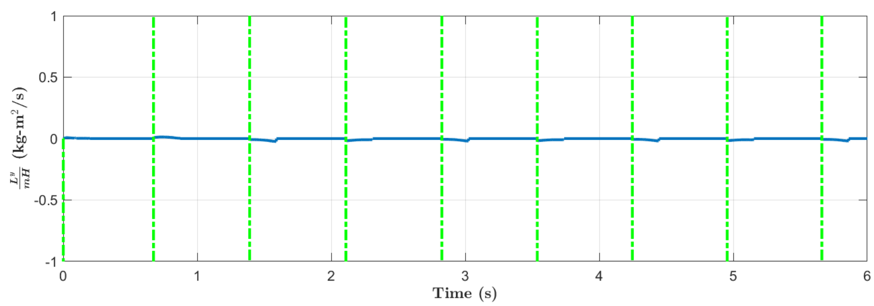
Figure 6.1: Trajectories of L and v_x when Rabbit is running. In (a) and (b), the single support and the flight duration are set to 0.15 s and 0.2s, respectively. The target $\frac{L_{des}}{mH_{nominal}}$ at the end of a step is set to 4 m/s. In the plot, L is reported as being zero during the flight phase because there is no contact point. In (c) and (d), the single support and flight duration are set to 0.2 s and 0.5s, respectively. The target $\frac{L_{des}}{mH_{nominal}}$ at the end of a step is set to zero.



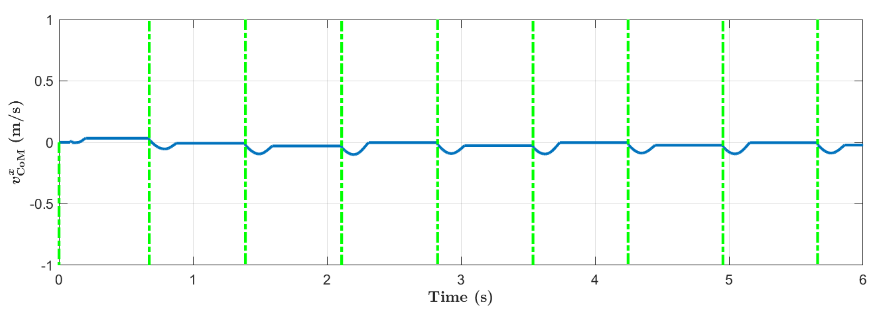
(a)



(b)

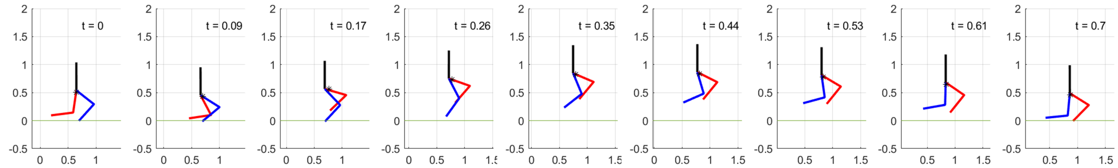


(c)

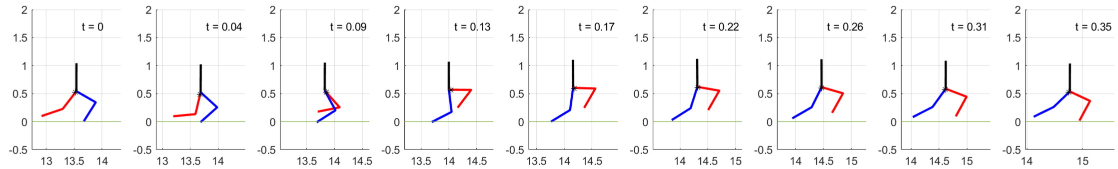


(d)

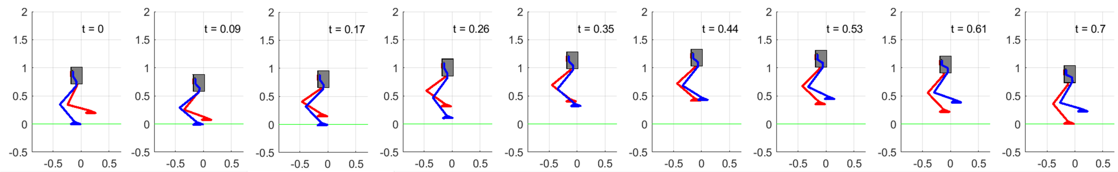
Figure 6.2: Trajectories of L and v_x when Cassie is running. In (a) and (b), the single support and flight duration are set to 0.15 s and 0.2s, respectively. The target $\frac{L_{des}}{mH_{nominal}}$ at the end of a step is set to 4 m/s. In the plot, L is reported as being zero during the flight phase because there is no contact point. In (c) and (d), the single support and flight duration are set to 0.2 s and 0.5s, respectively. The target $\frac{L_{des}}{mH_{nominal}}$ at the end of a step is set to zero.



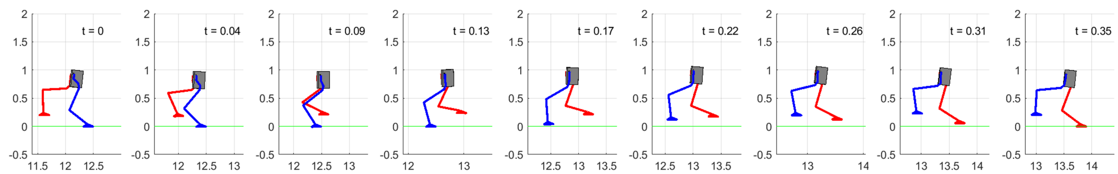
(a) Rabbit hopping in place



(b) Rabbit running forward



(c) Cassie hopping in place



(d) Cassie Running foward

Figure 6.3: Outtakes of Rabbit and Cassie hopping and running. When hopping, Rabbit's leg swings in the air to keep the torso upright. The robot is not moving forward with a large stride.

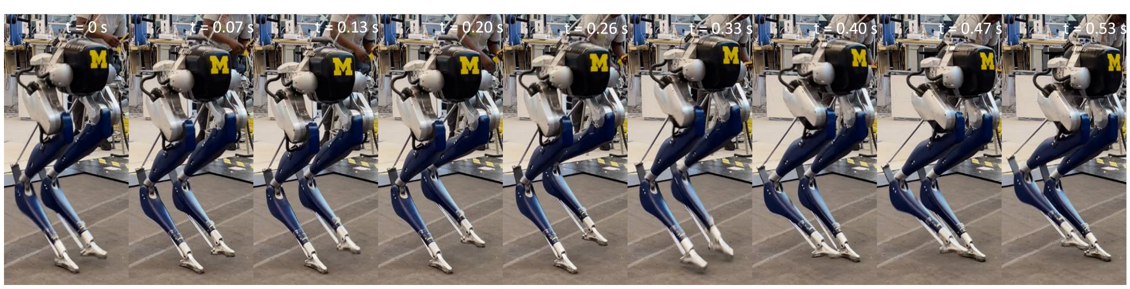


Figure 6.4: Cassie hopping in experiment.

A similar phenomenon used to occur in our walking experiments, caused by an unexpected vertical CoM velocity at the end of a step. As we have discussed in Sect. 5.1, besides the foot placement, the CoM velocity at the end of a step has an effect on the initial L of the next phase and thus affects balancing. In walking, though we set the reference for the CoM height to a constant and the reference vertical velocity to zero, the actual CoM height is always oscillating because of the springs on Cassie's leg. The springs are asymmetric, which leads to different CoM vertical velocities at the end of the left and the right stance phases (the differences get larger as the walking speed increases), thus causing different speeds in the ensuing stance phase, resulting in an asymmetric gait. The issue in walking was later avoided by measuring and estimating the CoM vertical velocity and adjusting foot placement correspondingly to offset this effect, as discussed in Sect. 5.4.6.

The same mitigation is not used for running at the present time because we don't have a good measurement of CoM vertical velocity during the flight phase. The foot placement is determined with the planned vertical velocity instead of the actual velocity. And thus we have an asymmetric gait caused by asymmetric CoM vertical velocities when Cassie was running forward.

We could not get a good measurement of CoM vertical velocity during the flight phase because the springs on the previous stance leg vibrantly oscillate when the stance foot leaves the ground, making the estimated CoM vertical velocity noisy. The noise in the CoM velocity estimation comes from the fusion of the slightly asynchronous measurement from IMU and encoders, which is amplified when the joints are moving fast. One mitigation is to use IMU vertical velocity instead of CoM vertical velocity. This approximation on Cassie generates little error (see Fig. 6.5), because Cassie's mass is concentrated near the torso, where the IMU is installed.

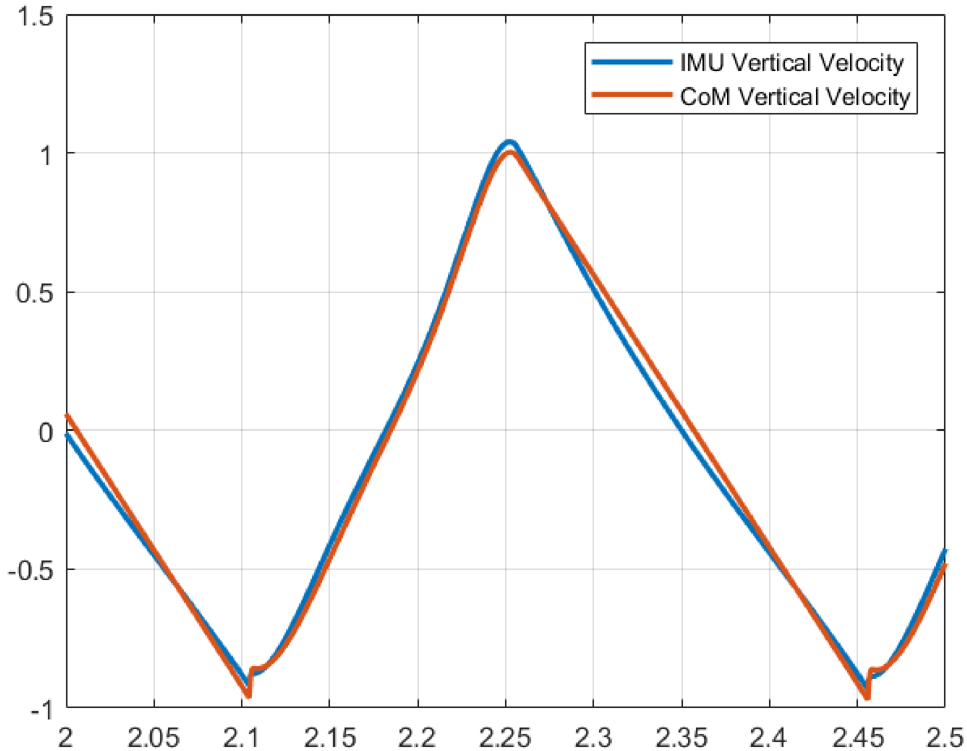


Figure 6.5: A simulation comparison of Cassie's IMU and CoM vertical velocity when Cassie is running forward. The measurement is noise-free.

CHAPTER VII

Conclusions and Future Work

7.1 Conclusions

In Chapter III, we discussed a controller designed with a gait library. The full 20 DoF dynamic model of Cassie and optimization was used to design seven gaits for walking in place, forward, and backward while meeting key physical constraints. The complicated morphology of the robot was translated into "universal," physically meaningful control objectives involving torso orientation, leg orientation, and leg length. Moreover, it was shown how to practically implement these control objectives via decoupled PD controllers on the robot. The final controller was demonstrated both in and out of the laboratory, including walking on sidewalks, grass, sand, waxed floors, snow, and a hill with short brush.

In Chapter IV, we established connections between various approximate pendulum models that are commonly used for heuristic controller design and those that are more common in the feedback control literature where formal stability guarantees are the norm. We clarified commonalities and differences in the two perspectives for using low-dimensional models. In the process of doing so, we argued that models based on angular momentum about the contact point provide more accurate representations of robot state than models based on linear velocity. Specifically, we showed that an approximate (pendulum or zero dynamics) model parameterized by angular momentum provides better predictions for foot placement on a physical robot (e.g., legs with mass) than does a related approximate model parameterized in terms of linear velocity. We use ALIP to denote the linear inverted pendulum model parameterized by angular momentum.

In Chapter V, we discussed the general mechanisms of bipedal balance and selected foot placement as our major method to stabilize the robot. We focused on regulating angular momentum about the contact point with the ALIP model. We implemented a one-step-ahead angular-momentum-based controller on Cassie, a 3D robot, and demonstrated high agility and robustness in experiments. Using our new controller, Cassie was able to accomplish a wide range of tasks with nothing more than common sense task-based tuning: a higher step frequency to walk at 2.1 m/s and extra foot clearance to walk over slopes exceeding 22 degrees.

In Chapter VI, we proposed a running controller for a bipedal robot. We discussed the dynamics of the flight phase and the design of CoM vertical trajectory for the stance phase. Finally, we formulated a QP problem based on the angular momentum pendulum model dynamics to determine foot placements that would stabilize a running gait. We demonstrated the result in simulation for both Rabbit and Cassie. The robots are able to hop and stay in the air for 0.5s and run at a speed above 3.5m/s. In experiments, the Cassie robot is able to hop in place and has a 0.05s flight phase.

7.2 Future Work

There are many interesting directions that still need to be pursued. In most of our discussions, the ankle torque is omitted and simply set to zero. We omitted it because though ankle torque can affect balancing during a step, the limitations of foot roll prevent ankle torque from achieving large changes. The effect of ankle torque is dwarfed compared to foot placement. However, for more agile maneuvering in narrow spaces where foot placement is limited, such as stepping stones and stairs, the ankle torque can be critical. Planning of the ankle torque with the ALIP model can be done by combining the MPC formulation described in [126] and (4.20) to obtain a QP problem similar to (6.18).

Another powerful balance mechanism we have not exploited is changing the step time. In a scenario where foot placements are limited, step time plays a more important role than ankle torque. While for gaits designed with mechanical phases the step time is exploited naturally for stabilization, it is less straightforward when we design gaits with a time phase. To see why, we can look at the solution of (4.20) (ignoring ankle torque),

$$\begin{bmatrix} x_c(t+\Delta T) \\ L^y(t+\Delta T) \end{bmatrix} = \begin{bmatrix} \cosh(\ell\Delta T) & \sinh(\ell\Delta T)/(mH\ell) \\ mH\ell\sinh(\ell\Delta T) & \cosh(\ell\Delta T) \end{bmatrix} \begin{bmatrix} x_c(t) \\ L^y(t) \end{bmatrix} \tag{7.1}$$

We observe that the state $[x_c(t+\Delta T), L^y(t+\Delta T)]$ has a linear relation with the previous state $[x_c(t), L^y(t)]$, making the states suitable decision variables for a QP due to the linear constraint between them. But the state always has a nonlinear relation with respect to the transition time ΔT, and as a result, ΔT is a fixed parameter instead of a decision variable in the QP problem described in [126] and (6.18).

To construct a linear relationship between the states and the transition time, we can first transform the states into a Convergent Component and a Divergent Component[129] (a.k.a Capture Point[130]) with the following transformation matrix,

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{\ell} \\ 1 & \frac{1}{\ell} \end{bmatrix} \begin{bmatrix} x_c \\ L^y \end{bmatrix}, \tag{7.2}$$

where a_1 is the Convergent Component and a_2 is the Divergent Component. The dynamics of $[a_1, a_2]$ is

$$\begin{bmatrix} a_1(t+\Delta T) \\ a_2(t+\Delta T) \end{bmatrix} = \begin{bmatrix} e^{-\ell \Delta T} & 0 \\ 0 & e^{\ell \Delta T} \end{bmatrix} \begin{bmatrix} a_1(t) \\ a_2(t) \end{bmatrix}. \tag{7.3}$$

When we apply the logarithm function to the vector a, defining

$$\begin{aligned} b_1 &= \log(a_1) \\ b_2 &= \log(a_2), \end{aligned} \tag{7.4}$$

the dynamics of b is then

$$\begin{bmatrix} b_1(t+\Delta T) \\ b_2(t+\Delta T) \end{bmatrix} = \begin{bmatrix} \ell \Delta T + b_1(t) \\ -\ell \Delta T + b_2(t) \end{bmatrix}. \tag{7.5}$$

Now the states b have a linear relation with respect to its previous states and time, both of which are suitable decision variables of a QP problem.

APPENDIX

APPENDIX A

A.1 A One-step Ahead Deadbeat Controller Based on Total Energy for the ALIP

We rewrite (5.3) as

$$\begin{bmatrix} x_c(T) \\ L(T) \end{bmatrix} = A(T) \begin{bmatrix} x_c(0) \\ L(0) \end{bmatrix}$$

and seek "fixed points", that is, conditions for periodicity when the impact map is included. With $\dot{z}_c = 0$, we have $L(T) = L(0)$, and a straightforward calculation with (5.3) shows that

$$L(T) = L(0) \implies x_c(T) = -x_c(0)$$

and hence periodic gaits are symmetric. Another straightforward calculation shows that if

$$\begin{bmatrix} x_c^* \\ L^* \end{bmatrix} = A(T) \begin{bmatrix} -x_c^* \\ L^* \end{bmatrix},$$

then

$$\frac{x_c^*}{L^*} = -\frac{1}{mH\ell} \frac{\sinh(\ell T)}{1 + \cosh(\ell T)} = -\frac{1}{mH\ell} \frac{\cosh(\ell T) - 1}{\sinh(\ell T)}$$

and one easily shows that

$$\frac{\sinh(t)}{1+\cosh(t)} = \frac{\cosh(t)-1}{\sinh(t)} = \frac{1-e^{-t}}{1+e^{-t}}.$$

For a symmetric gait with constant step duration T the average speed is $v_c^{x,\mathrm{avg}} = 2\frac{x_c(T)}{T}$. It follows that for a prescribed average walking speed, $v_c^{x,\mathrm{avg}}$, the fixed points satisfy

$$\begin{aligned} x_c^* &= \frac{T}{2} v_c^{x,\mathrm{avg}} \\ L^* &= -mH\ell \left(\frac{1+e^{-\ell T}}{1-e^{-\ell T}}\right) \frac{T}{2} v_c^{x,\mathrm{avg}}. \end{aligned}$$

After a bit of algebra, the (pseudo) energy associated with a fixed point can then be written as

$$\begin{aligned} E^*(v_c^{x,\mathrm{avg}}) &= \frac{1}{2} mg \left(\frac{T}{2} v_c^{x,\mathrm{avg}}\right)^2 \left[\left(\frac{1+e^{-\ell T}}{1-e^{-\ell T}}\right)^2 - 1\right] \\ &= 2mg \left(\frac{T}{2} v_c^{x,\mathrm{avg}}\right)^2 \frac{e^{-\ell T}}{(1-e^{-\ell T})^2}. \end{aligned}$$

We take the control objective to be $E(T_k^+) = E^*$, which of course, would need to be achieved subject to workspace limitations. If we assume that

$$L(T_k^+) = \widehat{L}(T_k^-),$$

our control law results from solving

$$-\frac{1}{2} mg \left(p_{\mathrm{sw}\to\mathrm{CoM}}^{x\,\mathrm{des}}(T_k^-)\right)^2 + \frac{1}{2}\frac{1}{mH}\left(\widehat{L}(T_k^-)\right)^2 = E^*(v_c^{x,\,\mathrm{des}})$$

for the desired swing foot position, to achieve a desired energy for step $k+1$. This is a one-step-ahead control law where we only need to run the angular momentum estimator for the end of the current step.

A.2 Constant Pendulum Length

Suppose that one component of the virtual constraints in (4.27) is $r_c(q) - R$, where R is a constant. Then $y \equiv 0$ yields $r_c = R$, simplifying (4.12) to

$$\begin{aligned} \dot{\theta}_c &= \frac{L - L_c}{mR^2} \\ \dot{L} &= mgR\sin(\theta_c) + u_a. \end{aligned} \tag{A.1}$$

At this point, no approximations have been made and the models is valid everywhere that $r_c(q) \equiv R$. An interesting aspect of this pendulum model is that it does not depend on $\dot{R}$, and thus imperfections in a achieving the virtual constraint $r_c = R$ have a smaller effect here than in (4.17), where $\dot{z}$ would appear when $z_c \neq H$, or in (4.16), where both $\dot{z}_c$ and $\ddot{z}_c$ would appear.

As with (4.17), the model (A.1) is driven by the strongly actuated states $q_b, \dot{q}_b$ through L_c and the same discussion applies. Dropping L_c in (4.10) results in

$$\begin{aligned} \dot{\theta}_c &= \frac{L}{mR^2} \\ \dot{L} &= mgR\sin(\theta_c) + u_a, \end{aligned} \tag{A.2}$$

which is nonlinear in θ_c. However, for $R = 1$ and a step length of 60 cm, $\max \theta_c \approx \pi/6$, and for 70 cm, $\max \theta_c \approx \pi/4$, giving simple bounds on the approximation error,

$$\begin{aligned} \frac{1}{\pi/6} \int_0^{\pi/6} (\theta - \sin(\theta)) d\theta &< 0.006 \\ \frac{1}{\pi/4} \int_0^{\pi/4} (\theta - \sin(\theta)) d\theta &< 0.02. \end{aligned}$$

Moreover, if desired, one can chose K to set

$$\frac{1}{\theta_{\max}}\left|\int_{0}^{\theta_{\max}}(K\theta-\sin(\theta))d\theta\right|=0.$$

For $\theta_{\max}=\pi/4$, the value is $K\approx 0.95$. While a linear approximation is useful for having a closed-form solution, numerically integrating the nonlinear model (A.2) in real time is certainly feasible.

The discussion on the approximate zero dynamics can be repeated here. The associated impact map is nonlinear and can be linearized about a nominal solution.

BIBLIOGRAPHY

[1] H-M Maus, S W Lipfert, M Gross, J Rummel, and A Seyfarth. Upright human gait did not provide a major mechanical challenge for our ancestors. *Nature Communications*, 1(6):70, January 2010.

[2] P-B Wieber. Holonomy and nonholonomy in the dynamics of articulated motion. In *Fast motions in biomechanics and robotics*, pages 411–425. Springer, 2006.

[3] Experiment Video Compilation. `https://youtu.be/o8msaembEu8`.

[4] Michigan Robotics: Dynamic Legged Locomotion Lab Youtube Channel. `https://www.youtube.com/channel/UCMfDV8rkQqWhUwnTAYAqOtQ`. Accessed: 2021-05-09.

[5] L.A.RYGG. Mechanical horse.

[6] RA Liston and RS Mosher. A versatile walking truck. In *Transportation Engineering Conference*, 1968.

[7] Andrew Alfonso Frank. *Automatic control systems for legged locomotion machines*. University of Southern California, 1968.

[8] I Kato. The hydraulically powered biped walking machine with a high carrying capacity. In *Proc. of the 4th Int. Symposium on External Control of Human Extremities, Dubrovnik*, pages 410–421, 1972.

[9] M.H. Raibert and Jr. Brown, H. Experiments in balance with a 2D one-legged hopping machine. *ASME Journal of Dynamic Systems Measurement, and Control*, 106:75–81, 1984.

[10] Wikipedia contributors. Asimo — Wikipedia, the free encyclopedia, 2021. [Online; accessed 16-November-2021].

[11] Marc Raibert. BigDog, the Rough-Terrain Quadruped Robot. In Myung J. Chung, editor, *Proceedings of the 17th IFAC World Congress, 2008*, volume 17, 2008.

[12] M. H. Raibert, H. B. Brown, and M. Chepponis. Experiments in Balance with a 3D One-Legged Hopping Machine. *The International Journal of Robotics Research*, 3(2):75–92, June 1984.

[13] J. K. HODGINS and M. H. RAIBERT. Adjusting step length for rough terrain locomotion. *IEEE Transactions on Robotics and Automation*, 7(3):289–298, June 1991.

[14] S. Kajita, T. Yamaura, and A. Kobayashi. Dynamic walking control of biped robot along a potential energy conserving orbit. *IEEE Transactions on Robotics and Automation*, 8(4):431–437, August 1992.

[15] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1405–1411 vol.2, 1991.

[16] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture point: A step toward humanoid push recovery. In *2006 6th IEEE-RAS international conference on humanoid robots*, pages 200–207. IEEE, 2006.

[17] Johannes Englsberger, Christian Ott, Máximo A Roa, Alin Albu-Schäffer, and Gerhard Hirzinger. Bipedal walking control based on capture point dynamics. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4420–4427. IEEE, 2011.

[18] Xiaobin Xiong and Aaron Ames. Sequential motion planning for bipedal somersault via flywheel slip and momentum transmission with task space control. *arXiv preprint arXiv:2008.02432*, 2020.

[19] R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(11-12):1217–1227, 1989.

[20] Alberto Isidori. The zero dynamics of a nonlinear system: From the origin to the latest progresses of a long successful story. In *Proceedings of the 30th Chinese Control Conference*, pages 18–25, 2011.

[21] E. Westervelt, J. W. Grizzle, and D.E. Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(1):42–56, January 2003.

[22] Brent Griffin and Jessy Grizzle. Nonholonomic virtual constraints and gait optimization for robust walking control. *The International Journal of Robotics Research*, page 0278364917708249, 2016.

[23] Xingye Da and Jessy W Grizzle. Combining Trajectory Optimization, Supervised Machine Learning, and Model Structure for Mitigating the Curse of Dimensionality in the Control of Bipedal Robots. *arXiv preprint arXiv:1711.02223*, 2017.

[24] Miomir Vukobratovic and Davor Juricic. Contribution to the Synthesis of Biped Gait. *IEEE Transactions on Biomedical Engineering*, BME-16(1):1–6, January 1969.

[25] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *ICRA*, volume 3, pages 1620–1626, 2003.

[26] Federico L Moro and Luis Sentis. Whole-body control of humanoid robots. *Humanoid Robotics: A Reference (A. Goswami, and P. Vadakkepat, eds.), Springer, to appear*, 2019.

[27] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Resolved momentum control: humanoid motion planning based on the linear and angular momentum. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, pages 1644–1650 vol.2, 2003.

[28] Hongkai Dai, Andrés Valenzuela, and Russ Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 295–302. IEEE, 2014.

[29] Hamid Benbrahim and Judy A Franklin. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, 22(3-4):283–302, 1997.

[30] J. Morimoto, G. Cheng, C.G. Atkeson, and G. Zeglin. A simple reinforcement learning algorithm for biped walking. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 3, pages 3030–3035 Vol.3, 2004.

[31] Zhaoming Xie, Glen Berseth, Patrick Clary, Jonathan Hurst, and Michiel van de Panne. Feedback control for cassie with deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1241–1246. IEEE, 2018.

[32] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. *arXiv preprint arXiv:2103.14295*, 2021.

[33] Jonah Siekmann, Yesh Godse, Alan Fern, and Jonathan Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7309–7315. IEEE, 2021.

[34] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, C. Canudas-de-Wit, and J. W. Grizzle. RABBIT: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, October 2003.

[35] Eric R Westervelt, Christine Chevallereau, Jun Ho Choi, Benjamin Morris, and Jessy W Grizzle. *Feedback control of dynamic bipedal robot locomotion.* CRC press, 2007.

[36] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, Berlin, third edition, 1995.

[37] A. Isidori and C. Moog. On the nonlinear equivalent of the notion of transmission zeros. In Byrnes C. and A. Kurzhanski, editors, *Proc. of the IIASA Conference: Modeling and Adaptive Control*, pages 146–157, Berlin, 1988. Springer-Verlag.

[38] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 1, pages 239–246. IEEE, 2001.

[39] A. Goswami and V. Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In *Proc. of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA*, volume 4, pages 3785–90, 2004.

[40] Johannes Englsberger and Christian Ott. Integration of vertical com motion and angular momentum in an extended capture point tracking controller for bipedal walking. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 183–189. IEEE, 2012.

[41] Xingye Da, Omar Harib, Ross Hartley, Brent Griffin, and Jessy W Grizzle. From 2D design of underactuated bipedal gaits to 3D implementation: Walking with speed tracking. *IEEE Access*, 4:3469–3478, 2016.

[42] E. R. Westervelt, G. Buche, and J. W. Grizzle. Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds. *International Journal of Robotics Research*, 24(6):559–582, June 2004.

[43] Hae-Won Park, Alireza Ramezani, and JW Grizzle. A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking. *IEEE Transactions on Robotics*, 29(2):331–345, 2013.

[44] Ayonga Hereid and Aaron D. Ames. Frost: Fast robot optimization and simulation toolkit. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, September 2017. IEEE/RSJ.

[45] Ayonga Hereid, Eric A Cousineau, Christian M Hubicki, and Aaron D Ames. 3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[46] Ayonga Hereid, Omar Harib, Ross Hartley, Yukai Gong, and Jessy W Grizzle. Rapid trajectory optimization using c-frost with illustration on a cassie-series

dynamic walking biped. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4722–4729. IEEE, 2019.

[47] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, Berlin, third edition, 1995.

[48] Cassie Blue and Associated Press. `https://youtu.be/h6hnfCo4a00`. Accessed: 2018-09-12.

[49] Cassie Blue Summary Video. `https://youtu.be/UhXly-5tEkc`. Accessed: 2018-09-14.

[50] Cassie Blue at the UM Dental School. `https://youtu.be/jhQMq6vpnAo`. Accessed: 2018-09-12.

[51] Cassie Blue Walks Through Fire. `https://youtu.be/cGb3bE6ZwrQ`. Accessed: 2018-09-12.

[52] Cassie Blue Plays in the Sand. `https://youtu.be/VoD7hbssu-M`. Accessed: 2018-09-12.

[53] Cassie Blue Hones Her Segway Riding Skills. `https://youtu.be/0gauVSUJzd0`. Accessed: 2018-09-12.

[54] H. Miura and I. Shimoyama. Dynamic walk of a biped. *International Journal of Robotics Research*, 3(2):60–74, 1984.

[55] Johannes Englsberger, Christian Ott, Máximo A Roa, Alin Albu-Schäffer, and Gerhard Hirzinger. Bipedal walking control based on capture point dynamics. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4420–4427. IEEE, 2011.

[56] Ting Wang and Christine Chevallereau. Stability analysis and time-varying walking control for an under-actuated planar biped robot. *Robotics and Autonomous Systems*, 59(6):444 – 456, 2011.

[57] Xiaobin Xiong and Aaron D Ames. Orbit characterization, stabilization and composition on 3d underactuated bipedal walking via hybrid passive linear inverted pendulum model. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4644–4651. IEEE, 2019.

[58] Agility Robotics. Robots. `https://www.agilityrobotics.com/robots#cassie`, 2021.

[59] M. Popovic, A. Hofmann, and H. Herr. Zero spin angular momentum control: definition and applicability. In *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, volume 1, pages 478–493 Vol. 1, 2004.

[60] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *ICRA'03*, 2003.

[61] Koichi Nishiwaki and Satoshi Kagami. High frequency walking pattern generation based on preview control of zmp. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2667–2672. IEEE, 2006.

[62] Pierre-Brice Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 137–142. IEEE, 2006.

[63] Tim Seyde, Apoorv Shrivastava, Johannes Englsberger, Sylvain Bertrand, Jerry Pratt, and Robert J Griffin. Inclusion of angular momentum during planning for capture point based walking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1791–1798. IEEE, 2018.

[64] Sung-Hee Lee and Ambarish Goswami. Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4667–4672. IEEE, 2007.

[65] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. Control and Automation. CRC Press, Boca Raton, FL, June 2007.

[66] T. Yang, E. R. Westervelt, A. Serrani, and J. P. Schmiedeler. A framework for the control of stable aperiodic walking in underactuated planar bipeds. *Autonomous Robots*, 27(3):277–290, 2009.

[67] Anne E. Martin, David C. Post, and James P. Schmiedeler. Design and experimental implementation of a hybrid zero dynamics-based controller for planar bipeds with curved feet. *The International Journal of Robotics Research*, 33(7):988–1005, 2014.

[68] H. Zhao, J. Horn, J. Reher, V. Paredes, and A. D. Ames. A hybrid systems and optimization-based control approach to realizing multi-contact locomotion on transfemoral prostheses. In *IEEE Conference on Decision and Control (CDC)*, pages 1607–1612, Dec 2015.

[69] Jacob Reher, Eric A Cousineau, Ayonga Hereid, Christian M Hubicki, and Aaron D Ames. Realizing dynamic and efficient bipedal locomotion on the humanoid robot durus. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1794–1801. IEEE, 2016.

[70] J. Reher, E. A. Cousineau, A. Hereid, C. M. Hubicki, and A. D. Ames. Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS.

In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1794–1801, May 2016.

[71] Brent Griffin and Jessy Grizzle. Nonholonomic virtual constraints for dynamic walking. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4053–4060. IEEE, 2015.

[72] Ayush Agrawal, Omar Harib, Ayonga Hereid, Sylvain Finet, Matthieu Masselin, Laurent Praly, Aaron D. Ames, Koushil Sreenath, and Jessy W. Grizzle. First steps towards translating HZD control of bipedal robots to decentralized control of exoskeletons. *IEEE Access*, 5:9919–9934, 2017.

[73] Thomas Gurriet, Sylvain Finet, Guilhem Boeris, Alexis Duburcq, Ayonga Hereid, Omar Harib, Matthieu Masselin, Jessy Grizzle, and Aaron D Ames. Towards restoring locomotion for paraplegics: Realizing dynamically stable walking on exoskeletons. In *C-ICRA*, pages 2804–2811. IEEE, 2018.

[74] Xingye Da and Jessy Grizzle. Combining trajectory optimization, supervised machine learning, and model structure for mitigating the curse of dimensionality in the control of bipedal robots. *The International Journal of Robotics Research*, 38(9):1063–1097, 2019.

[75] C. Byrnes and A. Isidori. Asymptotic stabilization of nonlinear minimum phase systems. *IEEE Transactions on Automatic Control*, 376:1122–37, 1991.

[76] E. R. Westervelt and J. W. Grizzle. Design of asymptotically stable walking for a 5-link planar biped walker via optimization. In *Proc. of the 2002 IEEE International Conference on Robotics and Automation, Washington, D.C.*, pages 3117–22, 2002.

[77] Ioannis Poulakakis and Jessy W Grizzle. The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper. *IEEE Transactions on Automatic Control*, 54(8):1779–1793, 2009.

[78] Ayonga Hereid, Shishir Kolathaya, Mikhail S. Jones, Johnathan Van Why, Jonathan W. Hurst, and Aaron D. Ames. Dynamic multi-domain bipedal walking with ATRIAS through SLIP based human-inspired control. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*, HSCC '14, pages 263–272, New York, NY, USA, 2014. ACM.

[79] Yu-Ming Chen and Michael Posa. Optimal reduced-order modeling of bipedal locomotion. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8753–8760. IEEE, 2020.

[80] A. Sano and J. Furusho. Realization of natural dynamic walking using the angular momentum information. In *Proc. of the 1990 IEEE International Conference on Robotics and Automation, Cincinnati, OH*, pages 1476–81, 1990.

[81] JW Grizzle, Claude H Moog, and Christine Chevallereau. Nonlinear control of mechanical systems with an unactuated cyclic variable. *IEEE Transactions on Automatic Control*, 50(5):559–576, 2005.

[82] Morteza Azad and Roy Featherstone. Angular momentum based controller for balancing an inverted double pendulum. In *Romansy 19–Robot Design, Dynamics and Control*, pages 251–258. Springer, 2013.

[83] Matthew J Powell and Aaron D Ames. Mechanics-based control of underactuated 3d robotic walking: Dynamic gait generation under torque constraints. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 555–560. IEEE, 2016.

[84] Yukai Gong and Jessy Grizzle. One-step ahead prediction of angular momentum about the contact point for control of bipedal locomotion: Validation in a lip-inspired controller. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2832–2838. IEEE, 2021.

[85] Xiaobin Xiong, Jenna Reher, and Aaron D Ames. Global position control on underactuated bipedal robots: Step-to-step dynamics approximation for step planning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2825–2831. IEEE, 2021.

[86] Kaveh Akbari Hamed, Nasser Sadati, William A Gruver, and Guy A Dumont. Stabilization of periodic orbits for planar walking with noninstantaneous double-support phase. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(3):685–706, 2011.

[87] Wen-Loong Ma, Kaveh Akbari Hamed, and Aaron D Ames. First steps towards full model based motion planning and control of quadrupeds: A hybrid zero dynamics approach. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5498–5503. IEEE, 2019.

[88] Jacob P Reher, Ayonga Hereid, Shishir Kolathaya, Christian M Hubicki, and Aaron D Ames. Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot durus. *Proc. 12th Adv. Robot. Algorithmic Found. Robot*, 13:400–415, 2020.

[89] Takumi Kamioka, Hiroyuki Kaneko, Toru Takenaka, and Takahide Yoshiike. Simultaneous optimization of zmp and footsteps based on the analytical solution of divergent component of motion. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1763–1770, 2018.

[90] J. H. Choi and J. W. Grizzle. Planar bipedal walking with foot rotation. In *Proceedings of the American Control Conference*, pages 4909–4916, Portland, OR, 2005.

[91] Victor Paredes and Ayonga Hereid. Dynamic locomotion of a lower-limb exoskeleton through virtual constraints based zmp regulation. In *Dynamic Systems and Control Conference*, volume 84270, page V001T14A001. American Society of Mechanical Engineers, 2020.

[92] Y. Hurmuzlu and T.H. Chang. Rigid body collisions of a special class of planar kinematic chains. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5):964–971, 1992.

[93] Y. Hurmuzlu. Dynamics of bipedal gait - part 1: objective functions and the contact event of a planar five-link biped. *Journal of Applied Mechanics*, 60:331–336, June 1993.

[94] J. W. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE Transactions on Automatic Control*, 46:51–64, January 2001.

[95] Shuuji Kajita and Kazuo Tani. Study of dynamic biped locomotion on rugged terrain-theory and basic experiment. In *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 741–746. IEEE, 1991.

[96] Omar Harib, Ayonga Hereid, Ayush Agrawal, Thomas Gurriet, Sylvain Finet, Guilhem Boeris, Alexis Duburcq, M. Eva Mungai, Matthieu Masselin, Aaron D. Ames, Koushil Sreenath, and Jessy Grizzle. Feedback control of an exoskeleton for paraplegics: Toward robustly stable hands-free dynamic walking. *arXiv preprint arXiv:1802.08322 [cs.RO]*, 2018.

[97] Jerry E Pratt and Russ Tedrake. Velocity-based stability margins for fast bipedal walking. In *Fast Motions in Biomechanics and Robotics*, pages 299–324. Springer, 2006.

[98] Hongkai Dai and Russ Tedrake. Planning robust walking motion on uneven terrain via convex optimization. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, pages 579–586. IEEE, 2016.

[99] Andreas Hofmann, Marko Popovic, and Hugh Herr. Exploiting angular momentum to enhance bipedal center-of-mass control. In *2009 IEEE International Conference on Robotics and Automation*, pages 4423–4429. IEEE, 2009.

[100] Ambarish Goswami and Vinutha Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 3785–3790. IEEE, 2004.

[101] Matthew J Powell, Ayonga Hereid, and Aaron D Ames. Speed regulation in 3d robotic walking through motion transitions between human-inspired partial hybrid zero dynamics. In *2013 IEEE international conference on robotics and automation*, pages 4803–4810. IEEE, 2013.

[102] Jessy W Grizzle and Christine Chevallereau. Virtual constraints and hybrid zero dynamics for realizing underactuated bipedal locomotion. In Bruno Siciliano and Oussama Khatib, editors, *Springer handbook of robotics*. Springer, 2016.

[103] Hamid Sadeghian, Christian Ott, Gianluca Garofalo, and Gordon Cheng. Passivity-based control of underactuated biped robots within hybrid zero dynamics approach. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4096–4101. IEEE, 2017.

[104] B. Morris and J. W. Grizzle. Hybrid invariant manifolds in systems with impulse effects with application to periodic locomotion in bipedal robots. *IEEE Transactions on Automatic Control*, 54(8):1751–1764, August 2009.

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

[106] A. Ames, K. Galloway, J. W. Grizzle, and K. Sreeenath. Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, 2014.

[107] William Yang and Michael Posa. Impact invariant control with applications to bipedal locomotion. *arXiv preprint arXiv:2103.06907*, 2021.

[108] Giuseppe Viola. Control of underactuated mechanical systems via passivity-based and geometric techniques. ., 2008.

[109] Eric R Westervelt, Jessy W Grizzle, and C Canudas De Wit. Switching and pi control of walking motions of planar biped walkers. *IEEE Transactions on automatic control*, 48(2):308–312, 2003.

[110] Shishir Kolathaya and Aaron D Ames. Input-to-state safety with control barrier functions. *IEEE control systems letters*, 3(1):108–113, 2018.

[111] Sushant Veer, Ioannis Poulakakis, et al. Input-to-state stability of periodic orbits of systems with impulse effects via poincaré analysis. *IEEE Transactions on Automatic Control*, 64(11):4583–4598, 2019.

[112] Yukai Gong. Cassie Angular Momentum Controller. `https://github.com/UMich-BipedLab/Cassie_Controller_AngularMomentum`.

[113] M. H. Raibert. Hopping in legged systems—modeling and simulation for the two-dimensional one-legged case. *IEEE Transactions on Systems, Man and Cybernetics*, 14(3):451–63, June 1984.

[114] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture Point: A Step toward Humanoid Push Recovery. *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006.

[115] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J. Huang, and J. Grizzle. Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway. In *2019 American Control Conference (ACC)*, pages 4559–4566, 2019.

[116] Hereid Ayonga. Fast Robot Optimization and Simulation Toolkit(FROST. `https://ayonga.github.io/frost-dev/`.

[117] Ross Hartley, Maani Ghaffari, Ryan M Eustice, and Jessy W Grizzle. Contact-aided invariant extended kalman filtering for robot state estimation. *The International Journal of Robotics Research*, 39(4):402–430, 2020.

[118] Ross Hartley. Invariant-EKF. `https://github.com/RossHartley/invariant-ekf`.

[119] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[120] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle. Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs. *IEEE Access*, 3:323–332, 2015.

[121] Christian Ott. *Cartesian impedance control of redundant and flexible-joint robots*. Springer, 2008.

[122] Shuuji Kajita, Takashi Nagasaki, Kenji Kaneko, Kazuhito Yokoi, and Kazuo Tanie. A running controller of humanoid biped hrp-2lr. In *Proceedings of the 2005 IEEE international Conference on Robotics and Automation*, pages 616–622. IEEE, 2005.

[123] Patrick M Wensing and David E Orin. High-speed humanoid running through control with a 3d-slip model. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5134–5140. IEEE, 2013.

[124] Koushil Sreenath, Hae-Won Park, Ioannis Poulakakis, and Jessy W Grizzle. Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel. *The International Journal of Robotics Research*, 32(3):324–345, 2013.

[125] Wen-Loong Ma, Ayonga Hereid, Christian M Hubicki, and Aaron D Ames. Efficient hzd gait generation for three-dimensional underactuated humanoid running. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5819–5825. IEEE, 2016.

[126] Grant Gibson, Oluwami Dosunmu-Ogunbi, Yukai Gong, and Jessy Grizzle. Terrain-aware foot placement for bipedal locomotion combining model predictive control, virtual constraints, and the alip. *arXiv preprint arXiv:2109.14862*, 2021.

[127] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[128] Casadi. `https://github.com/casadi/casadi`.

[129] Toru Takenaka, Takashi Matsumoto, and Takahide Yoshiike. Real time motion generation and control for biped robot-1 st report: Walking gait pattern generation. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1084–1091. IEEE, 2009.

[130] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus. Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid. *The International Journal of Robotics Research*, 31(10):1117–1133, August 2012.