Dynamic Bipedal Locomotion: From Hybrid Zero Dynamics to Control Lyapunov Functions via Experimentally Realizable Methods

Thesis by
Jenna Reher

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy in Mechanical Engineering

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Defended February 22, 2021

Jenna Reher
ORCID: 0000-0002-8297-3847

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Professor Aaron Ames, for his mentorship, for giving me the opportunity to work on the incredible projects shown both within this thesis and elsewhere, for giving me the space and time to shape the narrative of my research, and for his professional support as I navigated a massive life transition in the middle of graduate school. I would also like to thank my committee members, Professors Joel Burdick, Soon-Jo Chung, and Richard Murray, for taking time out of their busy schedules to provide valuable suggestions and advice. An additional thank you to Lanny Smoot, my mentor during the time that I spent at Disney Research.

I would like to thank all of the current and previous members of the AMBER lab, for all of the discussions, friendships, and adventures that created a constructive and inspiring lab atmosphere to be a part of. I would specifically like to thank everyone that I collaborated with throughout my graduate career (in no particular order): Wenloong Ma, Ayonga Hereid, Eric Ambrose, Shishir Kolathaya, Huihua Zhao, Rachel Gehlhar, Claudia Kann, Christian Hubicki, Noel Csomay-Shanklin, Dave Christensen, Bobby Bristow, Victor Parades, Jonathan Horn, and Eric Cousineau.

Finally, I am grateful to my family for their unrelenting love and support through the many life changes and moves that I experienced over the past seven years. Especially my parents, who have always encouraged me to pursue my dreams.

ABSTRACT

Robotic bipedal locomotion has become a rapidly growing field of research as humans increasingly look to augment their natural environments with intelligent machines. In order for these robotic systems to navigate the often unstructured environments of the world and perform tasks, they must first have the capability to dynamically, reliably, and efficiently locomote. Due to the inherently hybrid and underactuated nature of dynamic bipedal walking, the greatest experimental successes in the field have often been achieved by considering all aspects of the problem; with explicit consideration of the interplay between modeling, trajectory planning, and feedback control.

The methodology and developments presented in this thesis begin with the modeling and design of dynamic walking gaits on bipedal robots through hybrid zero dynamics (HZD), a mathematical framework that utilizes hybrid system models coupled with nonlinear controllers that results in stable locomotion. This will form the first half of the thesis, and will be used to develop a solid foundation of HZD trajectory optimization tools and algorithms for efficient synthesis of accurate hybrid motion plans for locomotion on two underactuated and compliant 3D bipeds. While HZD and the associated trajectory optimization are an existing framework, the resulting behaviors shown in these preliminary experiments will extend the limits of what HZD has demonstrated is possible thus far in the literature. Specifically, the core results of this thesis demonstrate the first experimental multi-contact humanoid walking with HZD on the DURUS robot and then through the first compliant HZD motion library for walking over a continuum of walking speeds on the Cassie robot.

On the theoretical front, a novel formulation of an optimization-based control framework is introduced that couples convergence constraints from control Lyapunov functions (CLF)s with desirable formulations existing in other areas of the bipedal locomotion field that have proven successful in practice, such as inverse dynamics control and quadratic programming approaches. The theoretical analysis and experimental validation of this controller thus forms the second half of this thesis. First, a theoretical analysis is developed which demonstrates several useful properties of the approach for tuning and implementation, and the stability of the controller for HZD locomotion is proven. This is then extended to a relaxed version of the CLF controller, which removes a convergence inequality constraint in lieu of a conservative CLF cost within a quadratic program to achieve tracking. It is then explored how this new CLF formulation can fully leverage the planned HZD walking gaits to achieve the

target performance on physical hardware. Towards this goal, an experimental implementation of the CLF controller is derived for the Cassie robot, with the resulting experiments demonstrating the first successful realization of a CLF controller for a 3D biped on hardware in the literature. The accuracy of the robot model and synthesized HZD motion library allow the real-time control implementation to regularize the CLF optimization cost about the nominal walking gait. This drives the controller to choose smooth input torques and anticipated spring torques, as well as regulate an optimal distribution of feasible ground reaction forces on hardware while reliably tracking the planned virtual constraints. These final results demonstrate how each component of this thesis were brought together to form an effective end-to-end implementation of a nonlinear control framework for underactuated locomotion on a bipedal robot through modeling, trajectory optimization, and then ultimately real-time control.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] J. Reher and A. D. Ames, "Control Lyapunov functions for compliant hybrid zero dynamic walking," *IEEE Transactions on Robotics*, 2021, In Preparation.
J.R. contributed to the conception of the project, to the performing of the theoretical analysis, software development, implementation, and to the writing of the manuscript.

[2] J. Reher, A. Hereid, E. Cousineau, C. Hubicki, P. Birkmeyer, S. Morfey, Z. Shivers, C. Salisbury, and A. D. Ames, "The humanoid robot DURUS: How hybrid system models, novel electromechanical design and nonlinear control realized new levels of humanoid efficiency," *IEEE Control Systems Magazine*, 2021, In Preparation.
J.R. contributed to the locomotion model, experimental implementation, and software used in the later work shown in this article, assisted in the early experimental results shown, wrote portions of the article, and compiled writings from co-authors into the final manuscript.

[3] J. Reher and A. D. Ames, "Dynamic walking: Toward agile and efficient bipedal robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, 2020. DOI: 10.1146/annurev-control-071020-045021.
J.R. wrote this review article in collaboration with and under the guidance of Dr. Aaron Ames.

[4] ——, "Inverse dynamics control of compliant hybrid zero dynamic walking," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
J.R. contributed to the conception of the project, to the performing of the theoretical analysis, software development, implementation, and to the writing of the manuscript.

[5] J. Reher, N. Csomay-Shanklin, D. L. Christensen, B. Bristow, A. D. Ames, and L. Smoot, "Passive dynamic balancing and walking in actuated environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020. DOI: 10.1109/ICRA40945.2020.9197400.
J.R. contributed to the locomotion model and controller, the experimental implementation and software, and to the writing of the manuscript.

[6] X. Xiong, J. Reher, and A. Ames, "Global position control on underactuated bipedal robots: Step-to-step dynamics approximation for step planning," *arXiv preprint arXiv:2011.06050*, 2020.
J.R. contributed to the experimental validation which was performed on hardware, to the creation of the submission video, and to the editing of the manuscript.

[7] J. Reher, C. Kann, and A. D. Ames, "An inverse dynamics approach to control Lyapunov functions," in *2020 American Control Conference (ACC)*, IEEE,

2020. DOI: 10.23919/ACC45564.2020.9147342.
J.R. contributed to the conception of the project, to the performing of the theoretical analysis, software development, implementation, and to the writing of the manuscript.

[8] J. Reher, W.-L. Ma, and A. D. Ames, "Dynamic walking with compliance on a Cassie bipedal robot," in *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 2589–2595. DOI: 10.23919/ECC.2019.8796090.
J.R. contributed to the conception of the project, worked in parallel with Wel-Loong Ma on the trajectory optimization and robot modeling, developed the control software, and contributed to the experimental implementation and writing of the manuscript.

[9] R. Gehlhar, J. Reher, and A. D. Ames, "Control of separable subsystems with application to prostheses," *arXiv preprint arXiv:1909.03102*, 2019.
J.R. contributed to the early discussions during the project conception, revised the presented theoretical contributions, and assisted in the writing of the manuscript.

[10] S. Kolathaya, J. Reher, A. Hereid, and A. D. Ames, "Input to state stabilizing control Lyapunov functions for robust bipedal robotic locomotion," in *2018 Annual American Control Conference (ACC)*, IEEE, 2018, pp. 2224–2230. DOI: 10.23919/ACC.2018.8430946.
J.R. contributed to the development of the simulation cases used in analysis, generated figures and video, and assisted in the writing of the manuscript.

[11] S. Kolathaya, J. Reher, and A. D. Ames, "Input to state stability of bipedal walking robots: Application to DURUS," *arXiv preprint arXiv:1801.00618*, 2018.
J.R. contributed to the experimental realization of the theoretical methods on hardware, to the development of simulation cases used in analysis, generated figures and video, and assisted in the writing of the manuscript.

[12] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, IEEE, 2018, pp. 98–106. DOI: 10.1109/ICCPS.2018.00018.
J.R. contributed to the control software used experimentally, assisted with experimental trials, and assisted in preparing the final manuscript.

[13] H. Zhao, J. Horn, J. Reher, V. Paredes, and A. D. Ames, "First steps toward translating robotic walking to prostheses: A nonlinear optimization based control approach," *Autonomous Robots*, vol. 41, no. 3, pp. 725–742, 2017. DOI: 10.1007/s10514-016-9565-1.
J.R. contributed to the software and state estimation used experimentally, assisted with experimental trials, and assisted in preparing the final manuscript.

[14] 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 *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 1794–1801. DOI: 10.1109/ICRA.2016.7487325.

[15] J. Reher, A. Hereid, S. Kolathaya, C. M. Hubicki, and A. D. Ames, "Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot DURUS," in *Twelfth International Workshop on Algorithmic Foundations on Robotics*, 2016. DOI: 10.1007/978-3-030-43089-4_26.

[16] H. Zhao, J. Horn, J. Reher, V. Paredes, and A. D. Ames, "Multicontact locomotion on transfemoral prostheses via hybrid system models and optimization-based control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 502–513, 2016. DOI: 10.1109/TASE.2016.2524528.
J.R. contributed to the software and state estimation used experimentally, assisted with experimental trials, and assisted in preparing the final manuscript.

[17] ——, "A hybrid systems and optimization-based control approach to realizing multi-contact locomotion on transfemoral prostheses," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, IEEE, 2015, pp. 1607–1612. DOI: 10.1109/CDC.2015.7402440.
J.R. contributed to the software and state estimation used experimentally, assisted with experimental trials, and assisted in preparing the final manuscript.

[18] H. Zhao, J. Reher, J. Horn, V. Paredes, and A. D. Ames, "Realization of stair ascent and motion transitions on prostheses utilizing optimization-based control and intent recognition," in *Rehabilitation Robotics (ICORR), 2015 IEEE International Conference on*, IEEE, 2015, pp. 265–270. DOI: 10.1109/ICORR.2015.7281210.
J.R. contributed to the software and state estimation used experimentally, assisted with experimental trials, and assisted in preparing the final manuscript.

[19] ——, "Realization of nonlinear real-time optimization based controllers on self-contained transfemoral prosthesis," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, ACM, 2015, pp. 130–138. DOI: 10.1145/2735960.2735964.
J.R. contributed to the software and state estimation used experimentally, assisted with experimental trials, and assisted in preparing the final manuscript.

[20] ——, "Demonstration of locomotion with the powered prosthesis AMPRO utilizing online optimization-based control," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, ACM, 2015, pp. 305–306. DOI: 10.1145/2728606.2728638.
J.R. contributed to the software and state estimation used experimentally, assisted with experimental trials, and assisted in preparing the final manuscript.

RELEVANT PUBLISHED VIDEO CONTENT

[1] *Dynamic walking on DURUS (aka PROXI) at the 2015 DRC finals.* https://youtu.be/a-R4H8-8074.

[2] *Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot DURUS.* https://youtu.be/1fC7b2LjVW4.

[3] *DURUS outdoor experiments at Georgia Tech.* https://youtu.be/Uh7NOD73L_0.

[4] *Passive dynamic balancing and walking in actuated environments.* https://youtu.be/4IXjD6hlbMg.

[5] *Dynamic walking with compliance on a Cassie bipedal robot.* https://youtu.be/NYooAyAC0kA.

[6] *An inverse dynamics approach to control Lyapunov functions. simulated walking and dynamic crouching with CLFs on Cassie.* https://youtu.be/CI9xv_OcfWw.

[7] *Inverse dynamics control of compliant hybrid zero dynamic walking on Cassie.* https://youtu.be/SvhjPZqSGFI.

[8] *Control Lyapunov functions for compliant hybrid zero dynamic walking.* https://youtu.be/vACSWOF6ap0.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

The realization of human-like capabilities on artificial machines has captured the imagination of humanity for centuries. The earliest attempts to realize this were through purely mechanical means. In 1495, Leonardo da Vinci detailed his *Automa cavaliere*, a primitive humanoid in a knights armor and operated by a number of pulleys and cables. However, these mechanical automatons lacked the ability to apply feedback control and thus the field remained largely dormant until digital computers became broadly available. In 1921, the word "robot" was coined by Czech playwriter Karel Carek, just 40 years before microprocessors were introduced and soon thereafter the field of legged robots began to emerge.

Today, the field of robotic legged locomotion is of special interest to researchers as humans increasingly look to augment their natural environments with intelligent machines. In order for these robots to navigate the unstructured environments of the world and perform tasks, they must have the capability to reliably and efficiently locomote. The first control paradigms for robotic walking used a notion of *static stability* where the vertical projection of the Center of Mass (COM) is contained to the support polygon of the feet, leading to the WABOT 1 robot in the early 1970s at Waseda University [1] and the first active exoskeletons by Vukobratović at the Mihailo Puppin Institute [2]. This static stability criterion was very restrictive, leading to the development of the *Zero Moment Point* criterion [3], [4], which enabled a wider range of robotic locomotion capabilities by generalizing from the COM to the Center of Pressure (COP). Despite this generalization, it still restricts the motion of the robot to be relatively conservative and does not allow for more dynamic motions when compared to the capabilities of biological walkers [5]. Nevertheless, this methodology has been perhaps the most popular methodology to date for realizing robotic locomotion. The method has been applied to various successful humanoid robots such as the HONDA ASIMO robot [6], [7], the HRP series [8]–[10], and HUBO [11].

As the field progressed into the 1980s, it became clear that to achieve truly dynamic locomotion, it was necessary to further exploit the natural nonlinear dynamics of these systems in an energy efficient and stable fashion. In stark contrast to the concept of fully actuated humanoid locomotion, Mark Raibert and the LegLab launched a

series of hopping robots which demonstrated running behaviors and flips [12], [13]. To achieve these behaviors, there was a shift from the conservative walking models encoded by the zero moment point to reduced order models (e.g. the spring loaded inverted pendulum) that ensure dynamic locomotion through the creation of stable periodic orbits [14]. Building upon this core idea, Tad McGeer began development of completely passive walking machines, which would ultimately give rise to the field of *passive dynamic walking* [15]. The downside of this method is that the system has little to no actuation with which it can respond to perturbations or to perform other tasks. However, these breakthroughs were critical in demonstrating that dynamic robotic locomotion was possible on systems which were not fully actuated, and that this underactuation could actually be leveraged to improve their performance.

Despite the advances leading up to the turn of the century, there remained a growing gap between the physical capabilities of robotic systems and the development of controllers to exploit them. This was particularly stark in the area of underactuated walking, where the lack of formal approaches that leverage the intrinsically nonlinear dynamics of locomotion limited the ability to fully exploit the robot's actuation authority. Bipedal locomotion algorithms with analytical guarantees of stability and performance have been developed in a variety of contexts. A few alternative algorithms that offer stability guarantees include *controlled symmetries* [16], [17], *hybrid reduction* [18], [19], and *hybrid zero dynamics (HZD)* [20]–[22]. Of these methods, only controllers based on HZD have been validated experimentally [23]–[28]. HZD was developed in the early 2000s, where a key development was introduced by Jessy Grizzle et al. [29] in which they developed the notion of *virtual constraints*, or holonomic constraints enforced via control rather than a physical mechanism. Enforcing these constraints leads to low-dimensional invariant surfaces, the *zero dynamics surface*, in the continuous phase of the model. These virtual constraints could then be designed such that this surface is *hybrid invariant* - being invariant under both the continuous and discrete dynamics - ultimately leading to the concept of Hybrid Zero Dynamics [30]. The end result is formal guarantees on the generation and stabilization of periodic orbits [31], i.e., walking gaits. HZD has been extended to 3D robots [32]–[36], to embedding simpler models [37]–[39], and templates [40]. This paradigm for control of dynamic underactuated locomotion has pushed boundaries on what is achievable, including fast running [41], [42] and efficient walking [43], [44].

With the goal of expanding the field of dynamic powered locomotion, it becomes prudent to develop models and algorithmic approaches which are capable of stabilizing

and even exploiting the underactuated dynamics of the walking problem. While all of the dynamic walking paradigms introduced throughout this section can generate stable walking motions, their actual implementation requires the deployment of feedback controllers capable of stabilizing the desired motions. In the case of methods which aim to stabilize the fullbody dynamics of a robot, the dynamic walking problem can involve a high level of complexity both in the form of nonlinearities and tightly coupled equations of motion. It is for this reason that historically, model-based controllers for HZD-based locomotion have primarily been available only in simulation or on lower-dimensional and simple robots. While there are many difficulties to consider in the implementation of feedback controllers for these robots, such as model inaccuracies, there has been a particularly prohibitive bottleneck in available approaches to both computationally attractive formulations of the problem and in providing sufficient authority to tune the controller.

1.1 Motivation

As we examine this brief historical outline of key developments to dynamic walking, it can be observed that with each new proposed methodology comes a greater understanding of how to model, plan, and execute increasingly complex behaviors on these robotic systems. Due to the inherently difficult nature of dynamic walking, successes in the field have typically been achieved by considering all aspects of the problem, often with explicit consideration of the interplay between modeling and feedback control (see Fig. 1.1). Specifically, the robotic and locomotive models which are used ultimately inform the planning problem and therefore the resulting behavior. Controllers which can actuate and coordinate the limbs must then be developed which, ideally, provide tracking, convergence, and stability guarantees. On top of these fundamental requirements of stability, it is often desired to achieve both increasingly dynamic and efficient locomotion.

Improving the energetics or dynamic nature of legged locomotion is a fundamentally challenging problem. Within both the gait planning and feedback control steps, there are a large number of configurable parameters, models, and actuation strategies which can be applied. In order to achieve this, it is often necessary to explicitly consider domains of increased underactuation and compliance. Further, a roboticist must consider how to tailor the locomotion plan and execution to fully exploit existing hardware to the greatest extent possible. This sets up several open questions to be addressed such as: How do the locomotion models influence the degree to which the walking problem is governed by the natural dynamics? What is a tolerable level

of underactuation in the control problem? How can a feedback controller be implemented on actual hardware to achieve the performance which is expected in plans generated offline? The field of dynamic locomotion will have to confront these issues in order to both expand the available behaviors and improve the resulting motions.

In this thesis, the interplay between modeling, motion planning, and trajectory regulation can be used to achieve dynamic and efficient locomotion on actual hardware platforms is examined. The focus of this work is then on the study of dynamic locomotion problems which are inherently *never* statically stable; in that they seek to achieve stable periodic walking. The methodology and developments presented herein begin with the design of dynamic and efficient walking gaits on bipedal robots through HZD, a mathematical framework that utilizes hybrid systems models coupled with nonlinear controllers that provably result in stable locomotion for both a fixed gait and a continuum of walking speeds. It is then explored how new formulations of feedback controllers can fully leverage these designed gaits to achieve the target performance on physical hardware. Therefore, the overall control framework presented allows for the full utilization of the mechanical designs for both DURUS and Cassie, including: efficient cycloidal gearboxes and compliant elements in the legs for absorbing impacts at foot-strike.

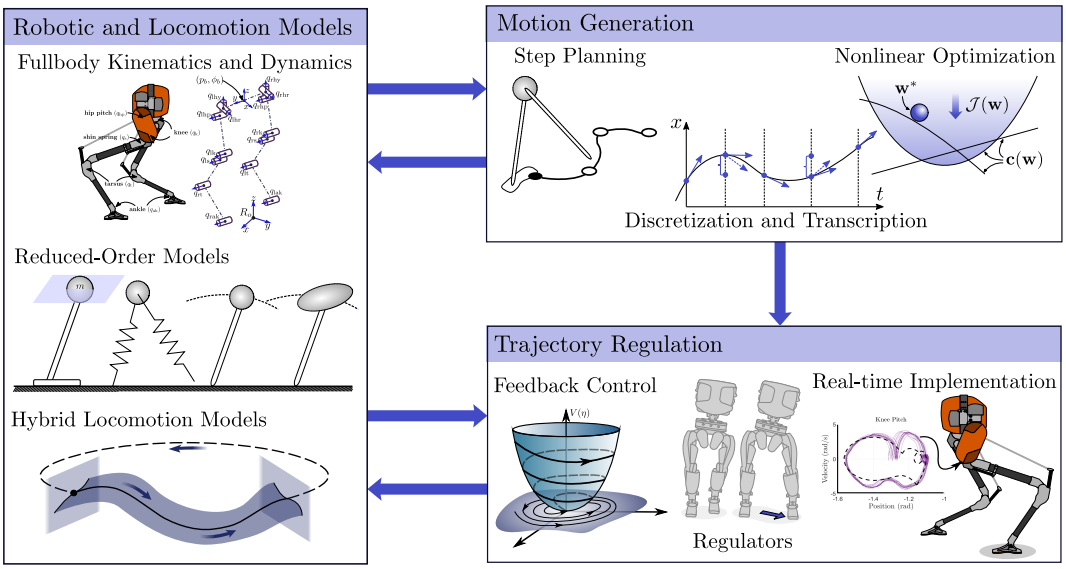

Figure 1.1: Dynamic walking is a complex behavior, requiring control designers and roboticists to simultaneously consider: robotic models, the transcription of locomotion into a motion planning problem, and the coordination and actuation of the system via control laws. Depicted here is the interconnection of these components, which provides an outline for how the broader ideas behind this thesis are connected.

1.2 Approaches to Locomotion in the Literature

Bipedal robots are high degree of freedom (DOF) systems composed of numerous linkages and actuators while subjected to intermittent and sometimes impulsive contact with the environment in order to move. This results in a highly coupled and highly nonlinear mechanical model of the system. It is due to these factors that the locomotion problem is fundamentally different than traditional approaches for fixed-base robots. Because of the nature of the problem, virtually all approaches to realize dynamic walking must first decide upon a locomotion model which will dictate available options for both trajectory planning and feedback control. The literature on bipedal locomotion modeling can largely be divided into two categories:

- *Reduced order models.* At the core of dynamic walking is the idea of reduced order models. These are either hierarchical—representing behaviors on simple models, e.g., inverted pendula and compass bipeds—or formally determined—low-dimensional systems rendered invariant by controllers, e.g., HZD.
- *Full order nonlinear dynamics.* Bipedal robots are inherently nonlinear with hybrid dynamical behaviors. These full order dynamics must be accounted for, either through assumptions thereon that yield reduced order models, through nonlinear controllers, or via optimization algorithms.

1.2.1 Linear Inverted Pendulum and the Zero Moment Point

In this section, the basic aspects of the *Zero Moment Point* (ZMP) and how it has been used in linear inverted pendulum models (LIPM) of locomotion are described. The concept of the ZMP is identical to the center of pressure (COP), and was originally introduced through a series of observations on the stability of anthropomorphic walkers by Vukobrativič in the early 1970s [3], [45]. The primary interpretation of the ZMP is: *the point on the ground at which the reaction forces between the robot's contacts and the ground produce no horizontal moment.* Consider a robot standing in single-support, with a finite number of contact points (p_i) that constrain the foot to be flat. As shown in Fig. 1.2(a), the resultant forces will consist of normal (λ_n) and tangential components (λ_t). The ZMP is then computed as:

$$p_{\text{ZMP}} := \frac{\sum_{i=1}^{N} p_i \lambda_{i,n}}{\sum_{i=1}^{N} \lambda_{i,n}}. \tag{1.1}$$

This led to perhaps the most commonly used dynamic stability margin [46]–[50], referred to as the *ZMP criterion*, which states that a movement is stable so long as

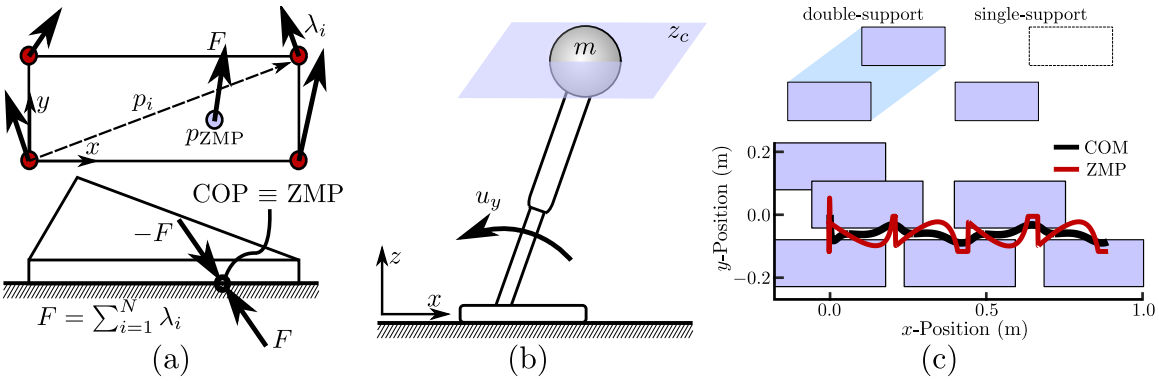

Figure 1.2: A depiction of the principles and modeling assumptions of the LIPM approach: (a) Visualization of the ZMP, where the foot is "dynamically balanced" if the resultant force F is within the support polygon. (b) LIPM with a telescoping leg and actuated ankle to control the robot along a horizontal surface. (c) Support polygon and an example of a planned ZMP trajectory.

the ZMP remains within the convex hull of the contact points (also known as the *support polygon*). This notion is conservative, and controlling these motions typically require the robot to remain fully actuated, with position controlled joints and load cells in the feet for control of the ZMP.

The ZMP criterion has been tied extensively to the linear inverted pendulum model (LIPM) in order to considerably simplify the trajectory design process, as the ZMP can be written explicitly in terms of the COM dynamics [51]. This has led to many researchers to consider a Newton-Euler representation of the *centroidal dynamics*:

$$m(\ddot{c} + g) = \sum_i \lambda_i, \quad \dot{L} = \sum_i (p_i - c) \times \lambda_i, \tag{1.2}$$

with c the COM position, $L = \sum_k (x_k - c) \times m_k \dot{x}_k + \mathbf{I}_k \omega_k$ the angular momentum, g gravitational acceleration, λ_i the contact forces, p_i is each contact force position, $\dot{x}_k$, ω_k the linear and angular velocities on the k-th linkage, m_k, $\mathbf{I}_k$ are the masses and inertia tensors, and m the total mass of the robot. If we constrain the motion of a fully actuated inverted pendulum with a massless telescoping leg such that the COM moves along a horizontal (x,y) plane, then we obtain a simple linear expression for the dynamics of the robot. An example of the LIPM is visualized in Fig. 1.2(b). The dynamics of the LIPM at a given height of the constraining plane, z_c, is given by:

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{m z_c} u_y, \qquad \ddot{y} = \frac{g}{z_c} y + \frac{1}{m z_c} u_x, \tag{1.3}$$

where m is the mass of the robot, g is the acceleration of gravity, and u_x, u_y are the torques about the x and y axes of the attachment to the ground, i.e., the ankle.

The ZMP location on the ground can also be directly written in terms of the LIPM dynamics as:

$$p^x_{\text{ZMP}} = x - \frac{z_c}{g}\ddot{x}, \qquad p^y_{\text{ZMP}} = y - \frac{z_c}{g}\ddot{y}. \tag{1.4}$$

Intuitive visual representations of the LIPM have been frequently discussed as a cart-table system [4], where the cart-table lies on a base with a geometry corresponding to the support polygon of the robot and indicates the acceptable region of the ZMP at which the table will not topple over [52]. The use of this LIPM and cart-table representation means that the design of a walking behavior can be essentially reduced to an inverse kinematics problem, where the primary planning is done on the ZMP trajectory. Fig. 1.2(c) shows an example ZMP trajectory for several forward steps. The ZMP trajectory for this walking is simply planned so that the ZMP always stays within the support polygon. This approach is largely applied to humanoids, where some of the most successful ZMP walkers to date are the WABIAN robots [1], HRP series [8], [10], Johnnie [53], and HUBO [11].

1.2.2 Capturability and Nonlinear Inverted Pendulum Models

Rather than characterize the stability of walking based on the ZMP, Pratt [54] and Hof [55] independently introduced the idea of a *Capture Point* (CP), referred to as the "extrapolated center of mass" (XCOM) by Hof. The CP can be intuitively described as the point on the ground onto which the robot has to step to come to a complete rest, shown in Fig. 1.3(b). In canonical examples of the CP methods, the overall walking motions of the robot are planned and controlled based on the *(instantaneous) capture point* (ICP) dynamics. In this case, the COM of the robot is constrained to move at a constant height along a horizontal plane, and thus uses a LIP representation of the robotic system. Due to the force balance associated with the constant height assumption, we know the forces in Equation (1.2) explicitly:

$$\sum_i \lambda_i = \frac{c - r_{\text{CMP}}}{z} mg \tag{1.5}$$

where r_{CMP} is location of the centroidal moment pivot (CMP) [56]. This greatly simplifies the dynamics of the problem, for which it was shown in [57] that for the compound variable $r^{x,y}_{ic} = c + \sqrt{\frac{z_c}{g_z}}\dot{c}$, the unstable portion of the resulting system dynamics (along the horizontal direction) can be written as:

$$\dot{r}^{x,y}_{ic} = \sqrt{\frac{g_z}{z_c}}(r^{x,y}_{ic} - r^{x,y}_{CMP}), \tag{1.6}$$

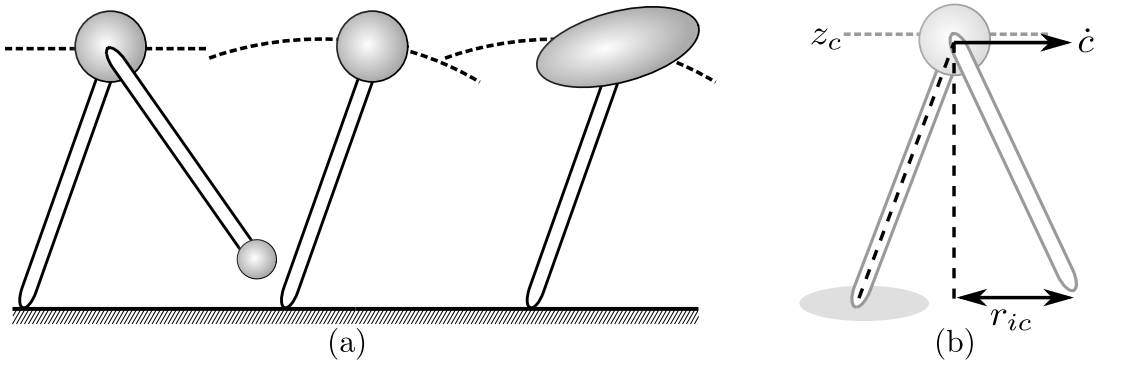

Figure 1.3: Representative nonlinear inverted pendulum models. (a) A depiction of several variations on inverted pendulum models, which attempt to expand the possible behaviors of the robot by accounting for more of the body inertia or by releasing the constrained motion of the hip. (b) A depiction of the capture point for a LIP walking robot.

where $\dot{r}_{ic}^{x,y}$ is the horizontal location of the ICP. The main consideration of the locomotion process is then to ensure that the feet are placed such that $r_{ic}^{x,y}$ lies within the support polygon:

$$r_{ic}^{x,y} \in \text{conv}\{p_i^{x,y}\}. \tag{1.7}$$

Satisfying this condition means that the COM will converge to the CP and come to a rest. Despite this intuitive representation of stability, the LIPM walking simplifications come with a steep cost due to the stringent requirements on the motion and actuation of the robot. On the other hand, it is precisely these characteristics which make the model most suitable for performing complex multi-objective tasks which include manipulation during intermittent conservative motions. The maturity and reliability of the LIPM made it prevalent in the walking controllers used at the DARPA Robotics Challenge [58]–[60].

In an attempt to overcome the issues associated with the strict assumptions of the LIPM, researchers have introduced variations of more complex pendulum models as illustrated in Figure 1.3(a). Perhaps the largest constraint on the motion of LIPM walking is the constant center of mass height assumption, leading to the development of a nonlinear inverted pendulum with variable mass height [61]. To better account for the inertia of a swinging leg, the addition of a mass at the swing foot was proposed and termed the Gravity Compensated LIPM [62]. Additionally, when a large robot or humanoid is performing dynamic walking behaviors, nontrivial angular momenta can result. One of the most commonly used models throughout the literature to address this problem is to add a flywheel to the inverted pendulum, which can be used to

represent the rotational inertia of the robot body as it moves [63]. A flywheel representation of the robot has gained recognition as a convenient representation of the robotic system particularly for CP control [54], [64]. Various pendulum models have been widely used in analysis of push recovery and balance [65]–[68]. The usefulness of the CP approach has also demonstrated walking behaviors successfully on hardware [57], [69], and was famously used on Honda's ASIMO robot [70], [71].

1.2.3 Spring Loaded Inverted Pendulum

Classic work by Raibert on hopping and running robots in the 1980s demonstrated the efficacy of using compliant models in locomotion through the development of a planar hopper which could bound at a speed of 1 m/s [12] and a 3D hopper which could achieve running without a planarizing boom [13]. These early successes drove researchers to investigate a Spring Loaded Inverted Pendulum (SLIP) representation of bipedal robots, shown in Fig. 1.4(a). The SLIP model provides a low-dimensional representation of locomotion which draws inspiration from biological studies on animal locomotion [72], [73]. The SLIP is particularly attractive due to its inherent efficiency and robustness to ground height variations.

In order to use this model to synthesize controllers for actual robots, the control objectives are typically decomposed into three components: (1) achieving a particular footstrike location to regulate forward speed, (2) injecting energy either through passive compliance or motors to regulate the vertical height of the CoM, and (3) regulating the posture of the robot. One then designs the walking and running mo-

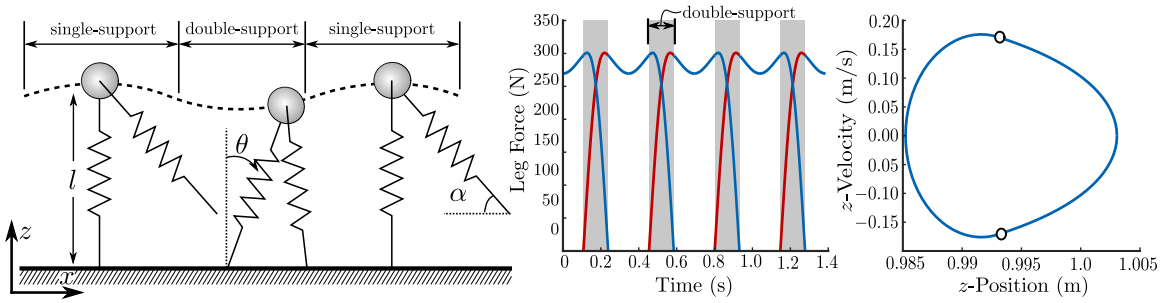

Figure 1.4: An example of the SLIP model and the emergent behavior of the walking task. (a) The SLIP model, with the mass concentrated at the hip and virtual compliant legs. (b) The resulting ground reaction forces during walking, the "double hump" profile is observed throughout walking behaviors in biological walkers. (c) A periodic orbit for the vertical COM position and velocity, noting that the lack of impact means that there is no discontinuity at footstrike.

tions with SLIP models and compensates for model mismatch or disturbances with well-tuned foot placement style controllers [74]–[78]. To this end, the dynamics of the SLIP are derived by assuming that the mass of the robot is concentrated at the hip with virtual springy legs:

$$\begin{aligned} 0 &= m\ddot{l} - ml\dot{\theta}^2 + mg\sin(\theta) + F_{\text{slip}} \\ 0 &= m[l^2\ddot{\theta} + 2l\dot{l}\dot{\theta}] + mgl\sin(\theta), \end{aligned} \tag{1.8}$$

where l is the stance leg length, θ is the stance leg angle, and F_{slip} is the force arising from the spring compression. One of the signature characteristics of this model is the "double hump" profile of the reaction forces, shown in Fig. 1.4(b), described by the force interactions observed in biological walkers [73]. A key contribution introduced by the SLIP community is the handling of underactuated behaviors, with many of the corresponding robots having point-feet and flight phases of motion. Finding a stable gait thereon does not rely on the quasi-static assumptions used for the fully actuated pendulum walkers of the preceding sections—instead focusing on stable cyclic locomotion. Dynamic stability is defined based on a constraint on the periodicity of the walking. To achieve forward walking, the initial states of the robot and the angle of attack α for the swing leg are chosen to yield a periodic gait; see Fig. 1.4(c). It is important to note that since the legs are massless, impacts are not considered, and the resulting orbit will be closed with no instantaneous jumps in the velocity.

The SLIP representation of walking has been primarily used for legged robots which have springs or series-elastic actuators (SEAs). Some of the earliest inclusions of compliant hardware on bipedal robots was with spring flamingo and spring turkey [79]. Later, the COMAN robot included passive compliance to reduce energy consumption during walking [80], and the Valkyrie robot from NASA was the first full-scale humanoid robot to heavily use SEAs [81]. Using inspiration from the SLIP morphology, Hurst designed the planar humanoid robot MABEL [25] and the 3D bipedal robot ATRIAS [82], [83] to include series elastic actuation and thus return energy through impacts and shield the motors from impact forces at footstrike. One of the latest robots in this series, the Cassie biped (described in Sec. 4.2) also mechanically approximates SLIP design principles [84]. Several running robots have specifically considered SLIP model principles in their mechanical design such as the ARL Monopod II [85], the CMU Bowleg Hopper [86], and the Keneken hopper [87].

1.2.4 Step Planning with Linear and Reduced-Order Models

Often for the simplest models of walking, such as traditional ZMP and LIPM versions of CP, the linear dynamics of the restricted system yield straightforward approaches to planning the motion of the COM. The walking characterized by these linear models often implicitly satisfy quasi-static stability assumptions, ultimately allowing a control designer to decouple the high-level step planner and low-level balance controllers [88]. In this vein, Kajita [52] introduced the jerk of the COM as an input controlled by a discrete LQR controller with preview action [89] to plan ZMP trajectories for predefined footsteps. However, predefining the motions of the ZMP or footholds is not always necessary or desirable.

If planners for these simple models could instead be performed online, then the robot may be able to mitigate issues related to reactivity. Weiber [90] proposed using linear trajectory-free *model predictive control* (MPC) as a method for explicitly handling the constraints imposed by the ZMP approach of Sec. 1.2.1 while continuously re-evaluating the walking path. Stephens [91] presented the use of MPC for push recovery and stepping on the SARCOS humanoid, which could be extended to obtain walking behaviors. The example shown in Fig. 1.2(c) visualizes the result of this MPC approach applied to LIPM robotic walking. It has also been shown how optimization and model predictive control can extend the notions of capture point to viable regions on which the biped can step [92], or how push recovery can be planned over a horizon of multiple steps [57]. Despite the ability of these planners to adapt online, they cannot handle the discrete dynamics associated with footstrike, and demand near-zero impact forces [93]. This rules out the nontrivial impacts which are naturally associated with dynamic walking. It is also difficult to provide a priori guarantees on whether any given reduced-order plan is feasible to execute on the full-order dynamics. Such methods typically use inverse kinematics [94], or inverse dynamics [95] sometimes in an operational-space formulation [96] to compute the full-order control inputs at each instant. Solving such near-term inverse problems does not imply that future inverse problems in the trajectory will be feasible, requiring additional planning [97], [98].

1.2.5 Passive Dynamic Walking

Some of the first work to study hybrid systems for the purposes of synthesizing walking were within the field of *passive dynamic walking*, which focused largely on how elegant mechanical design can enable stable walking on simple machines. Tad McGeer [15], [99] introduced several passive walking robots that could ambulate down small

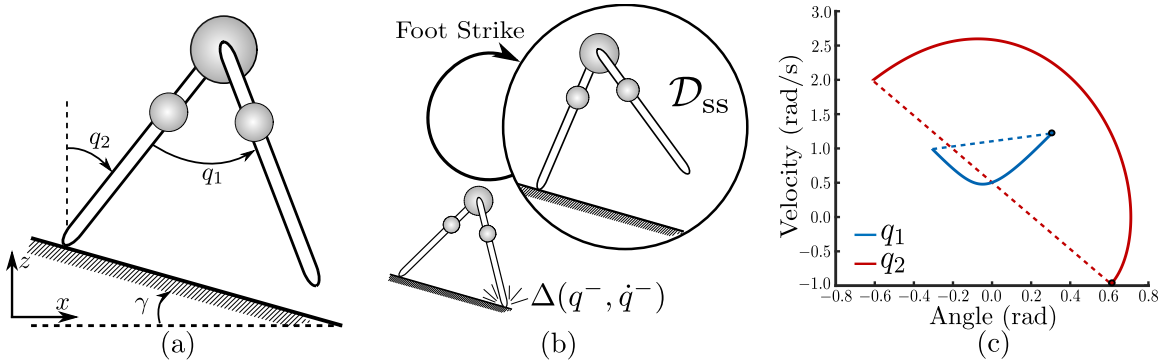

Figure 1.5: A canonical example of passive dynamic walking is the compass biped. (a) An illustration of the biped and its configuration on the slope. (b) Directed graph of the corresponding hybrid dynamical system, with only one continuous domain and one discrete transition. (c) A closed limit cycle implies stable walking, shown is the limit cycle for the biped walking down a 5 degree slope.

declines when started from a reasonable initial condition. While these early bipeds were completely passive and relied on gravity, several bipeds were built to demonstrate that simple actuators could substitute for gravitational power and compensate for disturbances. Small electric actuators were used for the Cornell walkers [100]–[102] and the MIT learning biped [103], [104], while the Delft biped instead used a pneumatic actuator at the hip [105], [106]. Controlled symmetries [16] and geometric reduction [18] has been used to extend these ideas to actuated robots and 3D walking. It has also been shown how the use of actuated environments can excite walking on passive robots [107]. Because of the care taken in mechanical design, these robots could all operate without sophisticated real-time calculations—though at the cost of diminished control authority.

The governing equations of motion for passive dynamic robots are nonlinear, and correspond to the continuous full-order dynamics rather than using an approximate (or reduced order) model. Further, an inherent feature of dynamic walking is that the robot is moving quickly through the environment. This means that the resulting motions cannot be slow enough for the feet to approach the ground with negligible velocity; impacts with the ground, therefore, become an important consideration. The locomotion problem can then be considered *hybrid* [108], [109], meaning it consists of both continuous and discrete nonlinear dynamics where the key element that determines the behavior is a directed cycle of continuous domains.

A canonical example of passive dynamic walking is an unactuated compass biped walking down a slope of angle γ [110], shown in Fig. 1.5(a). This robot consists of two

kneeless legs each with a point-mass and a third mass at the hip. The directed cycle for the biped consists only of a single-support domain, with transition occurring at footstrike (shown in Fig. 1.5(b)). The periodic nature of the stable walking behavior is best summarized by the phase portrait given in Fig. 1.5(c), where there are discrete jumps occurring at impact.

1.2.6 Hybrid Zero Dynamics

The method of *hybrid zero dynamics* (HZD) leverages nonlinear feedback control design to induce stable locomotion on underactuated robots. Jessy Grizzle et al. [20], [29], [111] introduced the concept and developed a set of tools which are grounded in nonlinear control theory to deal formally with the nonlinear and hybrid nature of dynamic walking (cf. the textbook [30]). The basis of the HZD approach is the restriction of the full-order dynamics of the robot to a lower-dimensional attractive and invariant subset of its state space, the *zero dynamics surface*, via *outputs* that characterize this surface. If these outputs are driven to zero, then the closed-loop dynamics of the robot is described by a lower-dimensional dynamical system that can be "shaped" to obtain stability. As was the case for uncontrolled hybrid models generalizing hybrid dynamical systems, a hybrid control system describes an actuated walking robot, leading to the notion of hybrid zero dynamics.

The primary consideration which governs the overall locomotion problem is the specification of a directed cycle for the underlying hybrid (control) system. Because HZD incorporates feedback control, significantly more complex motions are possible, such as multidomain walking. The controlled compass walker [110] is presented in Fig. 1.6

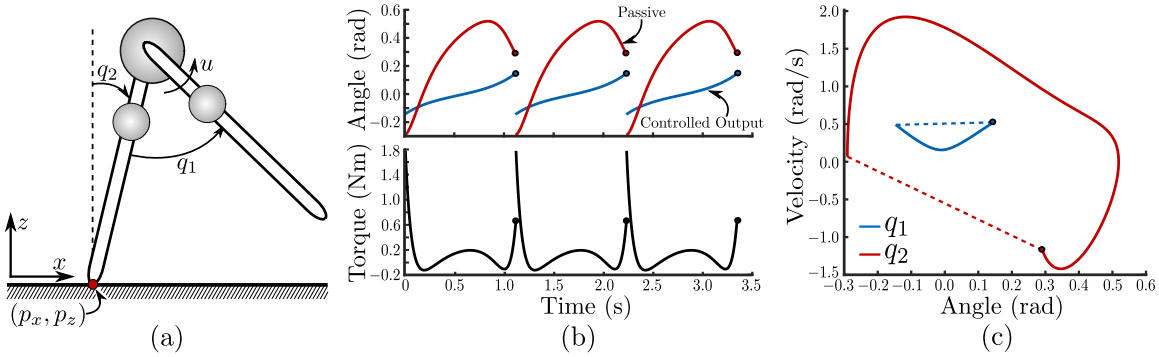

Figure 1.6: An example of HZD-based control for a compass biped on flat ground. (a) The robotic configuration. (b) Joint trajectories and torques over three steps of stable walking. (c) The walking exhibits a stable limit cycle, with discrete jumps occurring at impact.

to provide a comparison to the passive dynamic walking. In this example, it can be seen that torques applied at the hip are used to control the motion, while the robot walks with a stable limit cycle on flat ground.

In the context of robotic implementations, HZD has proven successful in realizing a wide variety of dynamic behaviors. Many of the early uses of the method were on point-footed robots which were restricted to the sagittal plane. The first robot used to study HZD was the Rabbit biped [112], followed later by MABEL [113] and AMBER 1 [114]. The ability of (P)HZD to handle multidomain behaviors led to its use on more complex planar bipedal robots such as ATRIAS [115], [116], AMBER 2, [117] and AMBER 3M [118]. New challenges appeared while extending the method of HZD from planarized robots to 3D robots, which exhibit additional degrees of underactuation. Control of fully actuated humanoids was demonstrated on a small-scale example with a NAO robot [119] via PHZD, while point-footed 3D walking with HZD was first shown at the University of Michigan with the MARLO biped [120]. At the DARPA Robotics Challenge, the humanoid DURUS was featured in an efficiency walk-off [43] where it demonstrated the first sustained humanoid HZD walking—over five hours continuously. DURUS went on to exhibit the most efficient walking on a humanoid to date, while performing human-like multicontact behaviors and managing significant underactuation [44]. The method has been extended to powered prosthetic walking [121]–[123] and to exoskeletons which can walk for patients with paraplegia [124], [125]. The use of springs in locomotion has also proven useful in the development of dynamic walking behaviors, though it presents additional challenges both mathematically and in practice. The notion of compliant hybrid zero dynamics was introduced in the late 2000s [126], and was later expanded upon to obtain compliant robotic running [41]. One of the latest robots to successfully demonstrate stable HZD walking is the Cassie biped, with the robot exhibiting underactuated feet and passive springs in the legs. Dynamic walking on Cassie has been successfully realized on hardware both by planning under the assumption of sufficient rigidity in the legs to ignore compliant elements [127], and for walking which considers the passive compliance in the zero dynamics [128].

1.3 Contributions

While the hybrid zero dynamics (HZD) framework embraces the use of planning and formal control for synthesizing walking gaits in a manner which can exploit the fullbody (and therefore passive) dynamics of the system, it has only recently become possible to actually use it for complex 3D robots. This was largely attributable to the series of new methodologies, algorithms, and optimization developments discussed earlier in this chapter, which has led to more efficient algorithms and optimization problems that can handle the problem's inherent computational complexity. However, most of these developments have been primarily focused on improvements to offline gait synthesis, and there still exists a significant gap in the available methodologies existing in the literature for realizing model- and optimization-based control of these behaviors on hardware. To address this gap, this thesis focuses first on developing a solid foundation for HZD gait synthesis on two 3D underactuated bipeds which will then be used to motivate a new formulation of control Lyapunov functions (CLF)s that are amenable to implementation on robotic systems. The final result of the thesis is then to combine these elements into an experimental study, which uses a parameterization of a compliant HZD motion library to realize a CLF on hardware.

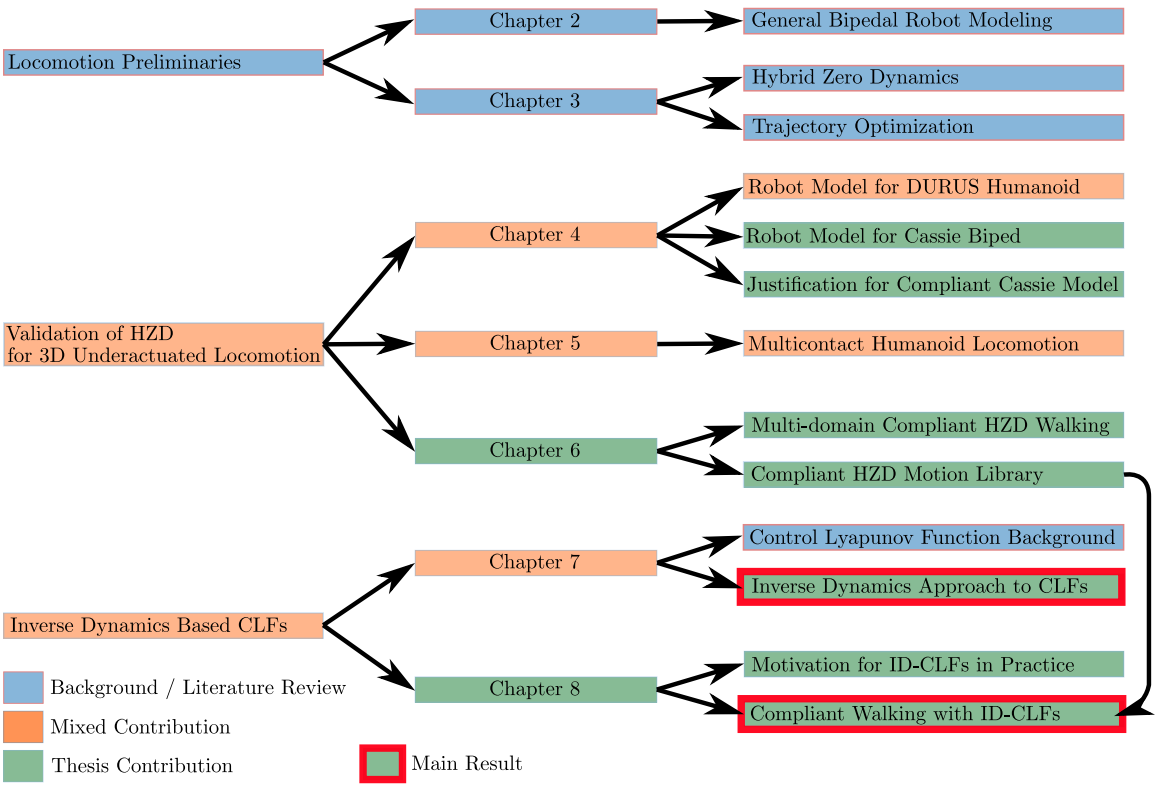

Figure 1.7: Outline and flow of this thesis, with contribution areas highlighted.

The key results of this thesis are summarized below:

- The development of the *first and only* multi-contact humanoid walking developed with HZD to date, which is also the *most efficient* reported humanoid walking in the literature to date.

- The development of a *compliant locomotion model* for the heavily *underactuated 3D Cassie biped* is introduced and then shown to accurately capture the passive dynamics of compliant walking over a variety of walking speeds experimentally. The experiments further the motivation of this thesis on *leveraging fullbody robot models in optimization*, and show how *model-based controllers can be developed* using a parameterization of the trajectory optimization results for each gait.

- A new inverse dynamics-based formulation of the control Lyapunov function controller (ID-CLF-QP) for robotic systems is developed and then theoretically analyzed. Additional discussion shows several of the benefits of this approach, which introduces additional decision variables to the traditional (CLF-QP) in order to pose an inverse dynamics problem with a Lypaunov convergence constraint. This combination allows for a simple affine representation with respect to the robot torques, constraint forces, and accelerations for a variety of constraints and costs.

- A relaxed variant of the CLF controller which was developed in this thesis, termed the (ID-CLF-QP$^+$), is implemented on hardware. The overall approach used a parameterization of the motion library developed in the HZD study of walking on Cassie to allow for the CLF controller to be implemented with smooth torques and within a reasonable computation time. The resulting experiments demonstrate the *first successful realization of a CLF controller on a 3D biped* in the literature.

1.4 Brief Description of Chapters

Chapter 2 describes general modeling concepts for locomotion on bipedal robots. This begins with an introduction of the continuous and discrete models of motion, resulting in a hybrid representation of the locomotion problem.

Chapter 3 presents the formal constructions underlying the hybrid zero dynamics framework. With the models developed in Chap. 2, we introduce the main goal of the HZD discussion: a nonlinear trajectory optimization problem that results in a closed-loop plan for stable walking.

Chapter 4 discusses the hardware and robot models specific to the two robots considered in this thesis: DURUS and more importantly, Cassie. Further, we provide a brief motivation for considering the full leg kinematics on Cassie, including the passive compliant elements present in the leg.

Chapter 5 illustrates how the HZD framework can be leveraged to obtain complex, dynamic, and efficient locomotion on the underactuated and compliant humanoid robot, DURUS. These preliminary results motivate the subsequent work on Cassie in Chap. 6, where the algorithms and software from this chapter are further developed.

Chapter 6 demonstrates the success of the HZD framework in capturing the passive dynamics of locomotion on the compliant Cassie model, while rendering stable walking. Further, a compliant HZD motion library is developed, and the accuracy of the robot model is validated using an inverse dynamics controller on hardware.

Chapter 7 *presents the main theoretical contribution of this thesis*: a novel formulation of the existing control Lyapunov function based quadratic program for robotic systems. The main result of this chapter is an optimization-based controller that couples convergence constraints from CLFs with desirable formulations existing in other areas of the bipedal locomotion field that have proven successful in practice, such as inverse dynamics control and quadratic programming approaches.

Chapter 8 *presents the main experimental contribution of this thesis*: an implementation of the (ID-CLF-QP) controller developed in Chap. 7, which is applied to the compliant model we create for Cassie in Sec. 4.2, and used to track the compliant HZD motion library which is found in Sec. 6.2 over a variety of walking speeds. The results demonstrate the effectiveness of the overall approach, showing that it accurately captures the passive dynamics of the planned walking, with balanced and smooth tracking of the virtual constraints with the anticipated force and acceleration profiles from optimization, and with evaluation times that make it tractable for reliable real-time implementation at 1 kHz.

Chapter 2

MODELING OF BIPEDAL ROBOTS

In this chapter, a general description of the continuous and discrete dynamics which govern legged robotic locomotion is described, followed by more specific descriptions of the DURUS and Cassie robots. In classical nonlinear control design, analysis is typically performed on a dynamical system of the control-affine form:

$$\dot{x} = f(x) + g(x)u, \tag{2.1}$$

where $x \in X \subset \mathbb{R}^n$ is the set of controllable states, $u \in U \subset \mathbb{R}^m$ is the control input, and the mappings $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are assumed to be locally Lipschitz continuous functions of x. While this is a more general approach to representing the dynamics of feedback control systems, the dynamics of most bipedal robots takes on a more specific form which will be useful in later sections.

All of the robotic walking which this thesis considers will be 3D locomotion, meaning that the robot is unconstrained in its motion through the world except via contacts at the feet. In addition, an important aspect which will be considered is passive compliance in the leg structures of the robots. This serves not only the purpose of shielding motors and gearboxes from large impacts, but the compliance can be used to do negative work at impact instead of the actuators and therefore can improve the efficiency of walking if utilized appropriately.

2.1 Bipedal Robots: Floating Base Systems with Contacts

Bipedal robotic platforms are conveniently modeled using a tree-like structure with an ordered collection of rigid linkages. This structure lends itself well to generalization, and thus tools to facilitate the generation of symbolic [129] or algebraic [130] expressions for the kinematics and dynamics of the robot are commonly used. The robot itself must ambulate through a sequence of contact conditions with the environment. Because interactions with the environment are always changing, a convenient method for modeling the system is to construct a representation of the robot in a general position, and then enforcing ground contacts through forces arising from the associated holonomic constraints that are imposed at the feet. This is often referred to as the *floating-base model* of the robot.

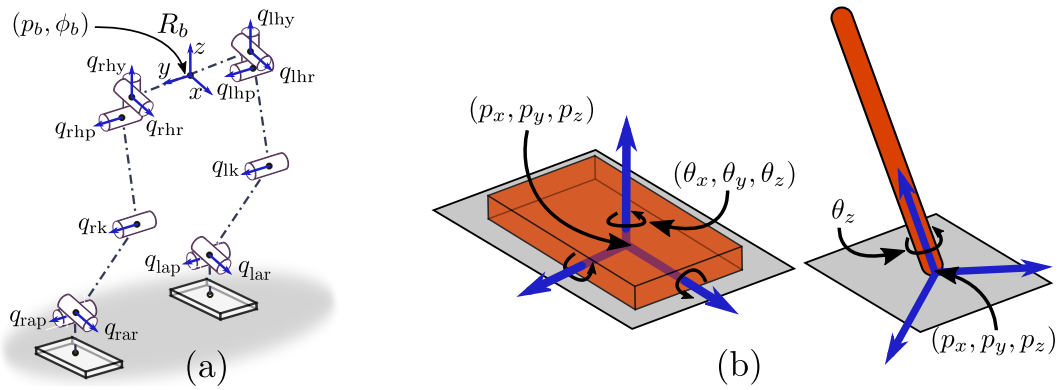

Figure 2.1: A visual demonstration of the robotic configuration and contact constraints which can be applied to a bipedal robot. (a) The floating-base coordinate system for an arbitrary bipedal robot, with a coordinate frame attached to the hip and rotational joints connecting rigid linkages of the body. (b) Contact geometry of the constraints for a flat-foot contact and an underactuated point-foot contact.

2.1.1 The Configuration Space

In order to represent the floating-base, let R_0 be a fixed inertial frame attached to the world and let R_b be a body reference frame rigidly attached to the pelvis of the robot with the origin located at the center of the hip. Then the Cartesian position $p_b \in \mathbb{R}^3$ and orientation $\varphi_b \in SO(3)$ compose the floating base coordinates of frame R_b with respect to R_0. The remaining coordinates which dictate the shape of the actual robot, $q_l \in \mathcal{Q}_l \subset \mathbb{R}^{n_l}$, are the local coordinates representing rotational joint angles and prismatic joint displacements. An image of this floating base coordinate system definition applied to an arbitrary bipedal robot is given in Fig. 2.1(a). Let $T\mathcal{Q}$ be the tangent bundle of $\mathcal{Q}$, then the combined set of coordinates is $q = (p_b^T, \varphi_b^T, q_l)^T \in \mathcal{Q} = \mathbb{R}^3 \times SO(3) \times \mathcal{Q}_l$ with the states $x = (q^T, \dot{q}^T)^T \in T\mathcal{Q} = X$.

2.1.2 Continuous Dynamics

If we continue with the assumption that the robot structure is a rigid collection of linkages, then we can obtain the equations of motion using the method of Lagrange. The Lagrangian for the floating-base system, $\mathcal{L} : T\mathcal{Q} \to \mathbb{R}$ is defined by:

$$\mathcal{L} = \mathcal{K}(q, \dot{q}) - \mathcal{V}(q), \tag{2.2}$$

where $\mathcal{K} : T\mathcal{Q} \to \mathbb{R}$ and $\mathcal{V} : \mathcal{Q} \to \mathbb{R}$ are the kinetic and potential energies of the robotic system, respectively. The robot dynamics can be determined through Lagrange's equations:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \Gamma, \tag{2.3}$$

with Γ the vector of generalized forces acting on the robot, written as:

$$\Gamma = Bu + J_c^T \lambda_c + B_{\text{sp}} \tau_{\text{sp}}(q, \dot{q}). \tag{2.4}$$

The matrices B and B_{sp} are derived from the principle of virtual work and provide a mapping for the actuator torques $u \in U \subset \mathbb{R}^m$ and spring torques $\tau_{\text{sp}} \in \mathbb{R}^{m_{\text{sp}}}$ enter the dynamics, respectively. Finally, external wrenches due to contact constraints, $\lambda_c : T\mathcal{Q} \times U \to \mathbb{R}^{m_c}$, are mapped via the constraint Jacobian $J_c(q)$, which will be derived in Sec. 2.1.3. Because the robots considered in this work are open-chain collections of linkages, the kinetic and potential energies are concisely written as [131]:

$$\mathcal{K}(q, \dot{q}) = \frac{1}{2} \dot{q}^T D(q) \dot{q}, \quad \mathcal{V} = \sum_{i=1}^{n} m_i g h_i, \tag{2.5}$$

where h_i is the center of mass height for the ith link, m_i is the corresponding link mass, g is the gravitational constant, and $D(q)$ is the *manipulator inertia matrix*. In terms of the link Jacobians, J_i, the manipulator inertia matrix is defined as:

$$D(q) = \sum_{i=1}^{n} J_i^T(q) \mathcal{M}_i J_i(q), \tag{2.6}$$

where $\mathcal{M}_i$ is the generalized inertia matrix of the ith link. Following the derivation of Lagrange's equations in (2.3) ultimately leads to a structured set of equations which describe the unpinned dynamics of the robot:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu + J_c(q)^T \lambda_c + B_{\text{sp}} \tau_{\text{sp}}(q, \dot{q}). \tag{2.7}$$

The matrix $C(q, \dot{q})$ contains both Coriolis and centrifugal terms, $G(q)$ is the gravity vector, and these are often grouped into a single term $H(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q)$. These dynamics can also be expressed in the state-space representation of (2.1) as:

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \underbrace{\begin{bmatrix} \dot{q} \\ D^{-1}(q) \left(J_c(q)^T \lambda_c + B_{\text{sp}} \tau_{\text{sp}}(q, \dot{q}) - H(q, \dot{q}) \right) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ D(q)^{-1} B \end{bmatrix}}_{g(x)} u. \tag{2.8}$$

Traditional methods for modeling the dynamics of floating-base systems typically result in the separation of the equations of motion into multiple parts [132]: one arising from the multibody continuous dynamics, and the other imposed via constraints on contacts with the environment. It is due to this that the external wrenches, λ_c, in our models will be governed by an acceleration constraint on the evolution of the dynamics, which is in the next section.

2.1.3 Constrained Dynamics

The fact that the robotic model is derived using a floating-base representation means that as we manipulate the robot, the resulting ground force interaction through the Lagrangian dynamics in Equation (2.7) is critical. In the robotic models used throughout this work, it is assumed that both the robot and the ground are rigid bodies, meaning that the following assumptions should hold: the bodies exhibit no force when they are not in contact and they are subject to nonpenetration, e.g., a point on one body cannot enter below the surface of the other. The resulting forces are then *unilateral* [132], meaning that they can only push and not pull on the ground:

$$\lambda_c \geq 0, \quad \lambda_c^T \eta_c(q) = 0, \quad \eta_c(q) \geq 0, \tag{2.9}$$

where $\eta_c(q)$ is a closure equation representing the configuration of a given constraint, and λ_c is a vector of constraint wrenches corresponding to the forces arising from the constraint. The second constraint, $\lambda_c^T \eta_c(q) = 0$, is an orthogonality condition, and shows that for each possible contact, if the closure is violated and $\eta_c(q) > 0$, then the constraint is no longer enforced and the corresponding normal force should be zero. The equations in (2.9) are referred to as *complimentary conditions* in the literature, and are used frequently for dynamic modeling [133]–[135] and simulation [136], [137] of systems which are subject to intermittent contacts.

Since the robotic locomotion that will be considered in this work will be subjected to a predetermined sequence of constraints, there is an *a priori* specification which can be used rather than detecting and handling the more general case. All of the constraints which will be considered here will be *holonomic constraints*, which will be an explicit constraint on the relation between the position states of the robot, q [138]. More specifically, a holonomic constraint, $\eta_c : \mathcal{Q} \to \mathbb{R}$, is a kinematic constraint of the position or orientation between a contact point and the walking surface. For one contact, the maximum number of constraints is then 6, and can be represented as:

$$\eta_c(q) = \mathbf{S}_c \left[p_c(q)^T, \quad \varphi_c(q)^T \right]^T = \text{constant}, \tag{2.10}$$

where $p_c(q)$ is the Cartesian position of the contact point, $\varphi_c(q)$ is a rotation between contacting bodies, and $\mathbf{S}_c$ is a selection matrix for the applicable constraints. Without loss of generality to (2.9), the constant term on the right hand side can be subtracted to the left. Two of the most common types of foot contact for bipedal walking are shown in Fig. 2.1. On the left, a 6-DOF constraint corresponding to a flat foot in contact with the ground is shown which would correspond to $\mathbf{S}_c = \mathbf{I}_{6\times 6}$, and on the

right a 4-DOF constraint is depicted which restricts a point contact along with an assumption of no yaw and corresponding to $\mathbf{S}_c = [\mathbf{I}_{4\times4}\ \mathbf{0}_{4\times2}]$ if $\varphi_c(q) := [\varphi_c^z, \varphi_c^y, \varphi_c^x]^T$.

We can differentiate (2.10) once to obtain a kinematic velocity constraint:

$$\underbrace{\frac{\partial \eta_c(q)}{\partial q}}_{J_c(q)} \dot{q} = 0, \tag{2.11}$$

and in order to obtain an acceleration constraint, we differentiate once more:

$$J_c(q)\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q} = 0, \tag{2.12}$$

leading to a system of equations where Equations 2.7 and 2.12 couple the accelerations to the inputs and constraint forces. One way of handling the conditions on (2.7) governing contact is to explicitly solve for λ_c and substitute by solving (2.12) for $\ddot{q}$:

$$\begin{aligned}\lambda_c(q,\dot{q},u) = -\left[J_c(q)D^{-1}(q)J^T(q)\right]^{-1}\Big[\dot{J}_c(q,\dot{q})\dot{q} + J_c(q)D^{-1}(q)\left(Bu\right. \\ \left. +B_{\text{sp}}\tau_{\text{sp}} - H(q,\dot{q})\right)\Big].\end{aligned} \tag{2.13}$$

After some rearranging, the augmented equations of motion:

$$D(q)\ddot{q} + \hat{H}(q,\dot{q}) = \hat{B}u + \hat{B}_{\text{sp}}\tau_{\text{sp}} \tag{2.14}$$

now have terms which implicitly satisfy (2.12):

$$\begin{aligned}\hat{H}(q,\dot{q}) &= H(q,\dot{q}) + J_c^T(q)\left[J_c(q)D^{-1}(q)J^T(q)\right]^{-1}\Big[\dot{J}_c(q,\dot{q})\dot{q} + J_c(q)D^{-1}(q)H(q,\dot{q})\Big] \\ \hat{B}_{\text{sp}}(q,\dot{q}) &= B_{\text{sp}} + J_c^T(q)\left[J_c(q)D^{-1}(q)J^T(q)\right]^{-1}J_c(q)D^{-1}(q)B_{\text{sp}} \\ \hat{B}(q,\dot{q}) &= B + J_c^T(q)\left[J_c(q)D^{-1}(q)J^T(q)\right]^{-1}J_c(q)D^{-1}(q)B.\end{aligned} \tag{2.15}$$

However, there is no guarantee that the complimentary conditions (2.9) or any additional feasibility conditions such as friction (e.g. (2.21)) will be satisfied, and the expressions have now become considerably more complex.

Direct Dynamics: Constraint Projection

Many robotic platforms are modeled using closed-loop topologies such as manipulators with workspace constraints [96], [139], [140], parallel robots, [141], or legged robots [142]. The end goal, as was seen in the last section, is to either form a set of coupled differential algebraic equations (DAE) such as (2.7) and (2.12), or an ordinary differential equation (ODE) such as (2.8). Many implementations for both feedback

control and simulation approach this problem by augmenting (2.7) with the solved acceleration as was done in (2.15); however, this approach often performs poorly both for large DOF systems due to repeated inversion of the inertia matrix which magnifies model error, and near singularities as the augmented inertia matrix invertibility relies on the Jacobian rank. Instead, one could consider a linear projection operator [143]:

$$P_c(q) = I - J_c^\dagger(q) J_c(q), \tag{2.16}$$

where $(\cdot)^\dagger$ denotes the pseudoinverse. If we look again at (2.11), we see that it is expressed in the form of the linear operation equation. This means that any admissible velocity must belong to the null space of the constraint Jacobian matrix, i.e. $\dot{q} \in \text{Null}(J_c)$. Thus, we can decompose the constraint velocity and accelerations as:

$$\dot{q}_\perp \equiv (I - P_c(q))\dot{q} = 0 \tag{2.17}$$

$$\ddot{q}_\perp \equiv (I - P_c(q))\ddot{q} = C_c(q, \dot{q})\dot{q} \tag{2.18}$$

with $C_c(q, \dot{q}) = -J_c^\dagger(q)\dot{J}_c(q, \dot{q})$.

Aghili went on to show [143] how this decomposition could be used to pose the *direct dynamics* of a constrained multibody system which produces a unique solution to the acceleration:

$$D_c(q)\ddot{q} + H_c(q, \dot{q}) = B_c(q)u + B_{c,\text{sp}}(q)\tau_{\text{sp}}, \tag{2.19}$$

where $H_c(q, \dot{q}) = P_c(q)H(q, \dot{q}) - C_c(q, \dot{q})\dot{q}$, $B_c(q) = P_c(q)B$, and $D_c(q)$ is called the *constraint inertia matrix*. This constraint inertia matrix is related to the unconstrained inertia matrix, assuming a symmetric matrix, by:

$$D_c(q) := D(q) + P_c(q)D(q) - (P_c(q)D(q))^T. \tag{2.20}$$

Although $D_c(q)$ is not unique[1], since $D(q)$ is invertible, the constraint inertia matrix will also be invertible, meaning the acceleration can be uniquely determined from the constrained dynamics of (2.19).

The form of the dynamics in (2.19) is desirable in some circumstances when a DAE or ODE for the constrained system is desired without having to directly compute the constraint wrenches as in (2.13). Further, it mitigates the downsides of (2.15) in that no inversions of the mass-inertia matrix are necessary in the DAE form and the use of a pseudoinverse is more desirable if the system ever operates near a singularity.

[1]Other valid choices of $D_c(q)$ may include $(P_cD + \alpha(I - P_c))$ for scalar α, $(D + P_cD - DP_c)$, or $(P_cDP_c + (I - P_c)D(I - P_C))$.

Contact Feasibility and Friction

Finally, when designing motions for the robot it is important to also model the real-world limitations to the allowed tangential force before it will break a nonslip condition. Ideally, a classical *Amontons-Coulomb model* of (dry) friction is used to avoid slippage and is represented as a friction constraint [138]. For a friction coefficient μ, the space of valid reaction forces is characterized by the *friction cone*:

$$\mathcal{C} = \left\{ (\lambda_x, \lambda_y, \lambda_z) \in \mathbb{R}^3 \middle| \, \lambda_z \geq 0; \sqrt{\lambda_x^2 + \lambda_y^2} \leq \mu \lambda_z \right\}. \tag{2.21}$$

However, this constraint is nonlinear, and cannot be implemented as a linear constraint. An alternative solution is to use a *pyramidal friction cone*:

$$\mathcal{P} = \left\{ (\lambda_x, \lambda_y, \lambda_z) \in \mathbb{R}^3 \middle| \, \lambda_z \geq 0; |\lambda_x|, |\lambda_y| \leq \frac{\mu}{\sqrt{2}} \lambda_z \right\}. \tag{2.22}$$

This is a more conservative model than the friction cone, but is advantageous in that it is a linear inequality constraint that can be implemented in a linear optimization problem [144]. When a surface is in contact with the outside world, additional constraints are introduced to prevent it from rolling over the contact edge in the form:

$$\begin{aligned} -\frac{l}{2}\lambda_z < &\lambda_{mx} < \frac{l}{2}\lambda_z \\ -\frac{w}{2}\lambda_z < &\lambda_{my} < \frac{w}{2}\lambda_z \end{aligned} \tag{2.23}$$

where l and w are the lengths and widths of the surface [145].

2.2 Discrete Dynamics: Impacts

An inherent feature of dynamic walking is that the robot is moving quickly through the environment. This means that the resulting motions cannot be slow enough for the feet to approach the ground with negligible velocity; impacts with the ground, therefore, become an important consideration in dynamic walking. Formally accounting for nontrivial impacts underlies the basis for hybrid dynamical approaches to modeling locomotion [30], [138], [146]. Impacts during walking typically occur when the nonstance foot strikes the ground. If we integrate the Lagrangian dynamics (Equation (2.7)) over the duration of the impact, we obtain:

$$D(q^-)(\dot{q}^+ - \dot{q}^-) = J_c(q^-)^T F_{\text{imp}}, \tag{2.24}$$

where F_{imp} is the impulsive force and $(q^-, \dot{q}^-)$, $(q^+, \dot{q}^+)$ are the pre and post-impact states. The simplest method for determining F_{imp} is to assume that the impulse is

instantaneous, forming a perfectly plastic impact [108], [109]. Since this contact is rigid and inelastic, when a point makes first contact, velocities must change instantaneously to adhere. This provides a kinematic relation for the post-impact velocity:

$$J_c(q^-)^T \dot{q}^+ = 0. \tag{2.25}$$

Combining Equations (2.24) and (2.25) leads to the overall impact model:

$$\begin{bmatrix} D(q^-) & -J_c(q^-)^T \\ J_c(q^-) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ F_{\text{imp}} \end{bmatrix} = \begin{bmatrix} D(q^-)\dot{q}^- \\ \mathbf{0} \end{bmatrix}, \tag{2.26}$$

where the post-impact velocity, $\dot{q}^+$, can be found using the Schur complement [147]:

$$\dot{q}^+ = \underbrace{\left(I - D(q^-)^{-1} J_c(q^-)^T \left(J_c(q^-) D(q^-)^{-1} J_c(q^-)^T\right)^{-1} J_c(q^-)\right)}_{\Delta^{\dot{q}}(q^-)} \dot{q}^-. \tag{2.27}$$

This is perhaps the simplest and most commonly used impact model leveraged for legged robots. However, the determination of F_{imp} is a complex problem and an open area of research. In real life, impacts are not truly instantaneous and do not always achieve stiction. Situations with multiple impacts can arise [148] leading to Zeno behaviors [149]–[151] or slippage [152], [153].

2.3 Hybrid Dynamical Models

Bipedal robots display both continuous and discrete behaviors, lending themselves naturally to *hybrid systems* models. Continuous evolution of the biped dynamics occurs when there is a fixed number of contact points with the environment, in which we say the robot is on a *vertex*. There is a discrete change in the biped dynamics when the number of contact points with the environment changes, on which we say the robot has reached an *edge*. A bipedal walking gait often consists of one or more different continuous phases followed by discrete events that transition from one phase to another, motivating the use of multi-domain hybrid systems with a specific ordering of phases. This is traditionally described as a *walking cycle*, which is a *directed cycle* with a sequence of continuous domains and edges (changes in contact conditions). A simple example of this is two-domain walking, pictured in Fig. 2.2(c), where the robot traverses a double-support and single-support domain in a repeating order.

Definition 2.3.0.1. *A **directed graph** is a graph described by the tuple $\Gamma = (V, E)$ where V is the set of vertices and $E \in V \times V$ is the set of edges. Because the graph is directed, the edges have a direction associated with them, meaning each edge defines an arc from one vertex to another.*

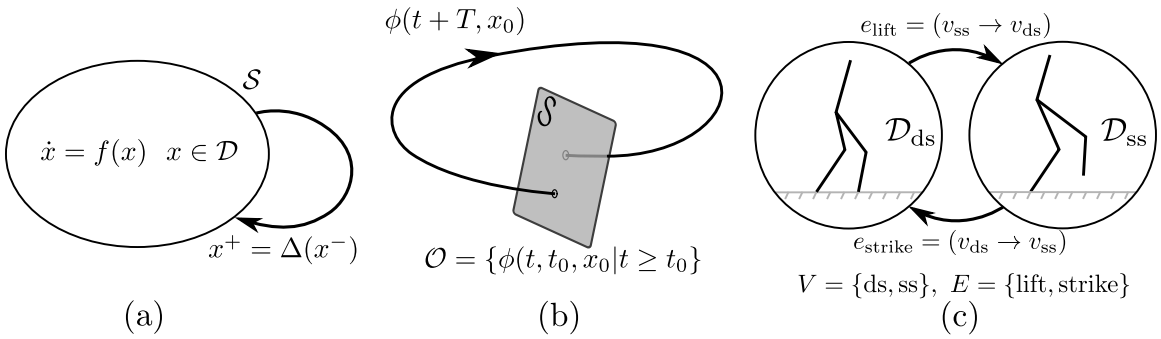

Figure 2.2: A graphical representation of hybrid system model components. (a) A hybrid dynamical system modeled as a system with impulse effects. (b) A periodic orbit on which the flow, $\varphi(t+T, t_0, x_0)$, evolves until it strikes the guard, S. (c) An example of a two-domain directed cycle for a bipedal robot, i.e., discrete phases and transitions that represent the walking pattern of the designated behavior.

Definition 2.3.0.2. *A **directed cycle** is a directed graph $\Gamma = (V, E)$, with a collection of vertices $V = \{v_1, v_2, ..., v_{n_p}\}$, and a set of edges $E = \{e_1 = (v_1 \to v_2), e_2 = (v_2 \to v_3), ..., e_{n_p} = (v_{n_p} \to v_1)$. Put simply, a directed cycle is a directed graph in which each edge is traversed in the same direction.*

Definition 2.3.0.3. *A **hybrid dynamical system**, used to model a walking robot [146], is defined as the tuple:*

$$\mathscr{H} = (\Gamma, \mathcal{D}, S, \Delta, F)$$

- $\Gamma = \{V, E\}$ *is a **directed cycle** specific to the desired walking behavior, with V the set of vertices and E the set of edges, $e = (v_{\text{s}} \to v_{\text{t}}) \in E$ with $v_{\text{s}}, v_{\text{t}} \in V$, in the cycle. Without loss of generality, the notation v_s and v_t has been shortened here to mean a source (s) and target (t) vertex, respectively.*

- $\mathcal{D} = \{\mathcal{D}_v\}_{v \in V}$ *is the set of **domains of admissibility**. Each domain $\mathcal{D}_v$ can be interpreted as the set of physically realistic states of the robot.*

- $S = \{S_e\}_{e \in E}$ *is the set of **guards**, with $S_e \subset \mathcal{D}_{v_{\text{s}}}$ which form the transition points from one domain, $\mathcal{D}_{v_{\text{s}}}$ to the next in the cycle: $\mathcal{D}_{v_{\text{t}}}$.*

- $\Delta = \{\Delta_e\}_{e \in E}$ *is the set of **reset maps**, $\Delta_e : S_e \subset \mathcal{D}_{v_{\text{s}}} \to \mathcal{D}_{v_{\text{t}}}$ from one domain to the next. The reset map gives the post impact state of the robot: $x^+ = \Delta_e(x^-)$.*

- $F = \{f_v\}_{v \in V}$ *is a set of **dynamical systems** where $\dot{x} = f_v(x)$ for coordinates $x \in D_v$, i.e., of the form given in Equation (2.1) with $u = 0$.*

A hybrid system of often alternatively written as a system with impulsive effects:

$$\mathscr{H} \triangleq \begin{cases} \dot{x} = f(x) & \text{if} \quad x \in \mathcal{D} \backslash S \\ x^+ = \Delta(x^-) & \text{if} \quad x^- \in S \end{cases} \tag{2.28}$$

which concisely states that the system evolves according to the dynamic system on the domain $\mathcal{D}$ until the guard or switching surface S is triggered, at which time a discrete jump in the system is applied according to the specified impact equations Δ. This overall concept is illustrated in Fig. 2.2(a), where it is common to represent the continuous domain as a bubble, with the associated reset behavior and direction represented as an arrow.

The continuous dynamics of the system depend on the Lagrangian of the robot model and the holonomic constraints, such as foot contacts with the ground, defined on a given domain. In the case of flat foot walking of a humanoid robot, the Cartesian position and the orientation of the foot that is in contact with the ground remains constant throughout the entire domain until the transition to the next domain occurs.

The discrete dynamics of the system determine the discrete change of robot states at domain transitions. For transitions which do not experience a rigid impact, such as the transition from single-support to double-support, there are no impact or coordinates changes involved, therefore the reset map is described as the identity map: $\Delta_e = I$, with I an identity matrix. Impact typically occurs when the non-stance foot hits the ground. Given the pre-impact states $(q^-, \dot{q}^-)$ on S_e, the post-impact states $(q^+, \dot{q}^+)$ are determined using a reset map Δ_e by assuming a perfectly plastic impact [108], [109]. The configurations of the robot are invariant through the impact, but velocities change due to the changes in holonomic constraints. The reset map, Δ_e, is given as:

$$\begin{bmatrix} q^+ \\ \dot{q}^+ \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{R}(q) & \mathbf{0} \\ \mathbf{0} & \mathcal{R}(q)\Delta_e^{\dot{q}}(q^-) \end{bmatrix}}_{\Delta_e(q,\dot{q})} \begin{bmatrix} q^- \\ \dot{q}^- \end{bmatrix} \tag{2.29}$$

where $\mathcal{R}(q)$ is the relabeling matrix for the coordinates change, and $\Delta_e^{\dot{q}}(q)$ is obtained from the plastic impact equation previously derived in (2.27).

Coordinate Relabeling. In most instances of multi-domain walking, if both legs are separately modeled as "left" and "right," then the reset map $\mathcal{R}(q)$ is simply the identity map. This is due to the fact that there is no update necessary to the configuration of the robot at impact since the leg coordinates remain the same, and the impact model (2.27) is only applicable to the velocity.

However, when studying symmetric walking gaits, it is sometimes useful to specify legs as "stance" and "non-stance" or to specify a configuration reset for a swing leg if there is a predetermined locking of passive elements such as springs. In this case, $\mathcal{R}(q)$ is a relabeling matrix [30], [154], which "flips" the stance and non-stance legs or zeroes specific spring elements. This is most often a linear map [155], [156], though it can be nonlinear, with the only requirement being that $\frac{\partial \mathcal{R}(q)}{\partial q}$ has full rank.

2.4 Periodic Orbits and the Poincarè Map

One can view steady state walking as a periodic motion which is not instantaneously stable, but is stable from step to step [157]—in other words, walking is "controlled falling." For undisturbed steady-state locomotion, this is most often represented by periodic orbits of the hybrid system presented in Definition 2.3.0.3. In a seminal paper on passive dynamic walking [15], McGeer popularized the method of Poincarè to determine the existence and stability of periodic orbits [158] for walking. In this approach, one step is considered to be a mapping from the walker's state at a definite point within a stride to the walker's state at the same point in the next step.

Definition 2.4.0.1. *Let $\mathcal{D} = \mathcal{D}_{v_1} \cup \mathcal{D}_{v_2} ... \cup \mathcal{D}_{v_{n_p}}$ be an open connected submanifold. For the hybrid system, $\mathscr{H}$, we can naturally select a guard, S, as the* ***Poincaré section****, and let $\varphi(t, x_0)$ be the solution to $\dot{x} = f(x)$ with initial condition $x_0 \in \mathcal{D}$.*

- *A solution $\varphi(t, x_0)$ to a dynamical system $\dot{x} = f(x)$ is* ***periodic*** *with finite period $T > 0$ such that $\varphi(t + T, x_0) = \varphi(t, x_0)$ for all $t \in [t_0, \infty)$.*

- *The set $\mathcal{O} \subset \mathcal{D}$ is a* ***hybrid periodic orbit*** *with a* ***fixed point*** *x^* if for some periodic solution $\varphi(t, x_0)$, the following holds $\mathcal{O} = \{\varphi(t, x^*) : 0 \leq t \leq T\}$.*

Stable bipedal robotic locomotion often corresponds to stable periodic orbits in hybrid systems. For robotic walking, the choice of a Poincaré section is governed by a unilateral constraint such as the vertical distance of a contact point above the ground, $h_e(x)$, and coincides with the guard, S_e:

$$S \triangleq \{x \in E : h(x) = 0,\ \dot{h}(x) < 0\} \subset h^{-1}(0) \subset E, \tag{2.30}$$

for which we have a Poincarè map $P : S \to S$ which is a partial function that maps one step to the next as $x_{k+1} = P(x_k)$:

$$P(x)\varphi(T_I(x), \Delta(x)), \tag{2.31}$$

where $T_I : S \to \mathbb{R}_{>0}$ is the *time-to-impact function:*

$$T_I(x) = \min\{t \geq 0 : t, \Delta(x)) \in \mathcal{X}\}, \quad \text{for } x \in \mathcal{X}. \tag{2.32}$$

This periodic phenomenon is visualized in Fig. 2.2(b), where the flow evolves along a line connecting the post-impact state to the pre-impact state through the guard and reset map. The periodic orbit yields a fixed point $x^* = P(x^*)$ with $x^* \in \mathcal{O} \cap S$, and the stability of the orbit is equivalent to the stability of the Poincarè map.

Theorem 2.4.1. *(Method of Poincaré sections [111]) If the hybrid system $\mathscr{H}$ has a periodic orbit $\mathcal{O}$ that is transversal to S, then the following are equivalent:*

1. *x^* is an exponentially stable (respectively, asymptotically stable or Lyapunov stable) fixed point of P;*
2. *$\mathcal{O}$ is an exponentially stable (respectively, asymptotically stable or Lyapunov stable) periodic orbit.*

A periodic orbit $\mathcal{O}$ is *transversal to* S if for all $x^* = \mathcal{O} \cap S_e$ the vector field f_v is not tangent to S_e at the point x^*, i.e. $\frac{\partial H_e(x)}{\partial x} f_{v_s} \neq 0$. A periodic orbit $\mathcal{O}$ is *transversal* if it is transversal to S_e for all $e \in E$, and is *period-one* if its closure intersects S_e at exactly one point for all $e \in E$.

While it is often not possible to analytically compute the Poincaré map, it is possible in almost all circumstances to compute its Jacobian. More specifically, once a fixed point x^* is found, we examine a first order expansion of the Poincarè map:

$$P(x^* + \delta x) \approx x^* + \frac{\partial P}{\partial x}(x^*)\delta x, \tag{2.33}$$

where we can use the derivative of the Poincaré map, $\frac{\partial P}{\partial x}(x^*)$. This is straightforward to evaluate numerically: one can construct a numerical approximation of successive rows by applying small perturbations to each corresponding state and then forward simulate one step to obtain $P(x^* + \delta x)$. The resulting Jacobian can be used to analyze the stability of the discrete update:

$$x_{k+1} = P(x_k), \tag{2.34}$$

with fixed point $x^* = P(x^*)$. The fixed point is exponentially stable if the magnitude of the eigenvalues of $\frac{\partial P}{\partial x}(x^*)$ are less than one [22], [158], [159].

Chapter 3

GAIT SYNTHESIS WITH HYBRID ZERO DYNAMICS

While the constructions introduced in Sec. 2.3 and Sec. 2.4 may adequately characterize how to obtain stable walking for the simple passive dynamic walkers introduced in Sec. 1.2.5, dynamic *actuated* locomotion requires additional considerations on how to shape control inputs to achieve the target behavior. The approach to locomotion that is considered in this thesis is to establish the walking problem as a controlled hybrid and periodic behavior. Though this increases the complexity of the problem relative to simpler models such as the LIPM (Sec. 1.2.1) or the SLIP (Sec. 1.2.3), it allows for high fidelity *closed-loop* trajectory optimization offline. This means that the resulting motion plan has additional stability properties which will make its later implementation (see Chap. 7) more straightforward.

3.1 Hybrid Control System

Recall that in Sec. 2.3, the hybrid dynamical system was introduced. In this section, the controlled version of a hybrid system is introduced. This allows for the consideration of an actuated legged robot within a hybrid framework.

Definition 3.1.0.1. *For the tangent bundle of the robot configuration with (local) coordinates $(q, \dot{q}) \in T\mathcal{Q} \subset \mathbb{R}^{2n}$, the* ***hybrid control system*** *[19] is the tuple:*

$$\mathscr{HC} = (\Gamma, \mathcal{D}, \mathcal{U}, S, \Delta, \mathrm{FG}). \tag{3.1}$$

where Γ, S, and Δ are defined as in Definition 2.3.0.3, and:

- $\mathcal{U} = \{\mathcal{U}_v\}_{v \in V}$ *is the set of admissible control inputs.*
- $\mathcal{D} = \{\mathcal{D}_v\}_{v \in V}$ *is the set of domains of admissibility. Each domain $\mathcal{D}_v \subseteq T\mathcal{Q} \times \mathcal{U}$ which can be interpreted as the set of possible states and controls the robot can assume given the constraints on the feet for the corresponding domain.*
- FG *provides the set of vector fields given by the equation: $\dot{x} = f_v(x) + g_v(x)u$, where $x = (q, \dot{q})$, $u \in \mathcal{U}$. f_v, g_v are defined in each domain by the Euler-Lagrangian dynamics.*

A hybrid dynamical system (Definition 2.3.0.3) is thus simply a hybrid control system with $\mathcal{U} = \emptyset$, meaning that any applicable feedback controllers have been applied, making the system closed-loop. We can now correspondingly expand the definition of a system with impulse effects (see (2.28)) to that of a controlled one:

$$\mathscr{HC} \triangleq \begin{cases} \dot{x} = f(x) + g(x)u & \text{if} \quad x \in \mathcal{D} \backslash S \\ x^+ = \Delta(x^-) & \text{if} \quad x^- \in S \end{cases} \tag{3.2}$$

where in the interest of space, this notation will be adopted where necessary in the rest of this thesis.

3.1.1 Domain of Admissibility

Domains and guards of the hybrid system model can be determined based on admissible states and controls of the system. This construct collects both the feasibility constraints of Sec. 2.1.3 and any additional constraints which must be enforced on a given domain, $\mathcal{D}_v$, for it to be valid. Specifically, one can state constraints such as the friction cone, (2.21), or ZMP foot-roll conditions, (2.23), in terms of a vector of inequalities,

$$\nu_v(q)\lambda_v(q, \dot{q}, u_v) \geq \mathbf{0}, \tag{3.3}$$

where $\nu_v(q)$ depends on the physical parameters of the robot. An additional constraint is that during single-support, the height of the non-stance foot should be above the ground (a requirement of the guard shown in (3.6)), which is a unilateral constraint. This can be modeled as the *unilateral constraint*, $h_v(q) > 0$. There is no unilateral constraint defined on a double-support domain. Combining (3.3) and unilateral constraints (if present) yields the domain of admissibility:

$$\mathcal{D}_v = \{(q, \dot{q}, u_v) \in T\mathcal{Q} \times U_v | A_v(q, \dot{q}, u_v) \geq \mathbf{0}\}, \tag{3.4}$$

where,

$$A_v(q, \dot{q}, u_v) = \begin{bmatrix} \nu_v(q)\lambda_v(q, \dot{q}, u_v) \\ h_v(q) \end{bmatrix}, \tag{3.5}$$

which defines the boundary of the domain manifold.

The guard is then just the boundary of the domain with an additional constraint that the unilateral constraint is decreasing, i.e., the foot is heading in the direction of the ground or the contact forces are approaching zero signaling foot lift:

$$S_e = \{x \in X \mid H_e(x) = 0, \dot{H}_e(x) < 0\}. \tag{3.6}$$

This corresponds to the requirements imposed on the Poincaré section in (2.30).

3.2 The Hybrid Zero Dynamics Framework

In order to obtain closed-loop walking controllers for the robots, we will build on the robotic models and hybrid systems definitions of Chap. 2 and Sec. 3.1. The overall approach will characterize motions through the concepts of virtual constraints and feedback linearization, with the end result being a controller which drives the system to the *zero dynamics* surface and render it invariant to the continuous dynamics. A trajectory optimization problem is then posed, which can shape the surface in order to render it invariant to impacts, resulting in the system having *hybrid zero dynamics*.

3.2.1 Virtual Constraints

At the core of the HZD methodology is the use of *outputs* as control objectives. By approaching the problem in this way, we can encode the evolution of the robot configuration into a set of outputs which can be shaped.

Definition 3.2.0.1. *An **output** for a nonlinear control system* (2.1) *is a differentiable function* $h : X \to \mathbb{R}$. *This can also be written as a vector of* m *functions:*

$$y(x) = \begin{bmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_m(x) \end{bmatrix} \in \mathbb{R}^m.$$

Definition 3.2.0.2. *Analogous to holonomic constraints, **virtual constraints** are defined as a set of outputs that regulate the motion of the robot with a certain desired behavior [155], [156]. The term "virtual" comes from the fact that these constraints are enforced through feedback controllers instead of through physical constraints. Let* $y^a(x)$ *be functions of the generalized coordinates that are to be controlled, i.e., encoding the "actual" behavior of the robot and* $y^d(t, \alpha)$ *be the "desired" behavior where* α *is a matrix of real coefficients that parameterize this behavior:*

$$y(x) := y^a(x) - y^d(\tau(q), \alpha), \tag{3.7}$$

with $\tau(q) : \mathcal{Q} \to \mathbb{R}$ *a **parameterization of time** that is strictly increasing.*

Driving $y \to 0$ results in convergence of the actual outputs to the desired, with $y^a : X \to \mathbb{R}^m$ and $y^d : \mathbb{R} \times \mathbb{R}^a \to \mathbb{R}^m$ smooth functions encoding the underlying behavior. We assume that $y(x)$ has (vector) relative degree r [160]. The notion of relative degree will be useful in Sec. 3.2.2, and will be useful in determining the number of times an output must be time differentiated along the solutions of (2.1) for

our control input to appear. More specifically, the derivatives of the outputs along $f(x)$ and $g(x)$ we obtain:

$$y^{(r)}(x) = \underbrace{L_f^{(r)} f(x)}_{L_f^{(r)} y(x)} + \underbrace{L_g L_f^{(r-1)} g(x)}_{\mathcal{A}(x)} u, \tag{3.8}$$

where $\mathcal{A}(x)$ is called the *decoupling matrix* and is invertible in the case of a (vector) relative degree.

It is often the case in robotic systems that $r = 1$ if $y(x)$ depends on position and velocity and $r = 2$ if it only depends on position, i.e., configuration variables. To motivate later constructions, we consider a mechanical system with configuration space $\mathcal{Q}$, (local) coordinates $q \in \mathcal{Q}$, and states $x = (q^T, \dot{q}^T)^T \in T\mathcal{Q} = X$. Suppose that for (2.1) there is a set of outputs $y(x) = (y_1(q, \dot{q})^T, y_2(q)^T)^T$ of vector relative degree 1 and 2, respectively, on a region of interest; that is for $y_1(q, \dot{q}) \in \mathbb{R}^{m_1}$ and $y_2(q) \in \mathbb{R}^{m_2}$ with $m = m_1 + m_2$, we assume the vector relative degree is 1 for y_1 and the 2 for y_2, i.e., $(1, \dots, 1, 2, \dots, 2)$ with 1's appearing m_1 times and 2's appearing m_2 times. We can then write an output tracking problem for each domain:

$$y_{1,v}(q, \dot{q}, t) = y_{1,v}^a(q, \dot{q}) - v_{d,v} \tag{3.9}$$

$$y_{2,v}(q, t) = y_{2,v}^a(q) - y_{2,v}^d(\tau_v(q), \alpha_v), \tag{3.10}$$

for $v \in V$, where $y_{1,v}$ and $y_{2,v}$ are relative degree 1 and (vector) relative degree 2 by definition, respectively. Additionally, $v_d \in \mathbb{R}$ is the constant "desired" velocity and $y_{2,v}^d(\tau(q, \beta), \alpha_v)$ are the desired outputs that give the desired trajectories of the corresponding actual outputs as dictated by a parameter set α_v and a state-based parameterization of time $\tau(q, \beta)$. The introduction of $\tau(q)$, which has to be monotonic over a gait cycle, is motivated by the desire to create an autonomous controller, which is more robust than non-autonomous controllers [30]. In order to characterize the progression of a walking gait, we then consider the monotonically increasing function:

$$\tau_v(q) = \frac{\theta(q) - \theta^+}{\theta^- - \theta^+}, \tag{3.11}$$

where $\theta : \mathcal{Q} \to \mathbb{R}$ is a phase variable, $\theta^+ = \theta(q^+)$ is its post-impact value, and $\theta^- = \theta(q^-)$ is its pre-impact value. Therefore, $\tau : \mathcal{Q} \to [0, 1]$ from the beginning to end of a domain.

In the context of implementation for robotic platforms, it is useful to select desired outputs which are inherently smooth and computationally easy to work with. One of

the most popular selections is a *Bézier polynomial* [30]:

$$b(\tau, \alpha_v) := \sum_{k=0}^{M} \alpha_v[k] \underbrace{\frac{M!}{k!(M-k)!}\tau^k(q)(1-\tau(q))^{M-k}}_{B_{M,k}}, \tag{3.12}$$

where M is the order of the polynomial. One of the attractive features of the Bézier polynomial is that its derivatives can readily be obtained from simple and computationally attractive calculations [161]. For instance, the first and second derivatives of (3.12) with respect to our phase variable are given by:

$$\frac{\partial b(\tau, \alpha_v)}{\partial \tau} = M \sum_{k=0}^{M-1} \left(\alpha_v[k+1] - \alpha_v[k]\right) B_{M-1,k} \tag{3.13}$$

$$\frac{\partial^2 b(\tau, \alpha_v)}{\partial \tau^2} = M(M-1) \sum_{k=0}^{M-2} \left(\alpha_v[k+2] - 2\alpha_v[k+1] + \alpha_v[k]\right) B_{M-2,k} \tag{3.14}$$

Giving us the desired relative degree 2 Bézier outputs and their derivatives:

$$\dot{y}_2^d(\tau(q), \alpha_v) = \frac{\partial b(\tau, \alpha_v)}{\partial \tau}\frac{\partial \tau(q)}{\partial q}\dot{q} \tag{3.15}$$

$$\ddot{y}_2^d(\tau(q), \alpha_v) = \frac{\partial^2 b(\tau, \alpha_v)}{\partial \tau^2}\left(\frac{\partial \tau(q)}{\partial q}\dot{q}\right)^2 + \frac{\partial b(\tau, \alpha_v)}{\partial \tau}\left(\frac{\partial}{\partial q}\left(\frac{\partial \tau(q)}{\partial q}\dot{q}\right)\dot{q} + \frac{\partial \tau(q)}{\partial q}\ddot{q}\right), \tag{3.16}$$

where in (3.16) we can see that $\ddot{q}$ appears, meaning that we can substitute our dynamics to obtain the derivative along the nonlinear system (2.1) as was shown with the general Lie derivative representation in (3.8). However, Bézier polynomials are not the only possible choices; it has been shown that NURB splines can be useful in characterizing some behaviors [116], and humans appear to follow spring-mass-damper type motions [154], leading to alternative parameterizations.

Remark 1. *The selection of outputs is greatly affected by the holonomic constraints defined on each domain. The guiding principle is that the total number of outputs and holonomic constraints cannot be greater than the total degrees of freedom of the system and the total number of outputs cannot exceed the number of actuators available on the robot. More specifically, given m_v admissible controls and $m_{c,v}$ holonomic constraints, the total number of independent relative degree 2 outputs is given by:*

$$m_{2,v} = \begin{cases} m_v - m_{1,v} & \text{if } m_v \leq n - m_{c,v} \\ m_v - m_{1,v} - m_{c,v} & \text{if } m_v > n - m_{c,v} \end{cases}. \tag{3.17}$$

3.2.2 Feedback Linearization

In order to actually encode the walking behaviors described by the output polynomials introduced in Sec. 3.2.1, we must prescribe a feedback controller which can drive $y(x) \to 0$. Feedback linearization is a common and powerful tool in the control of nonlinear systems. The overall approach transforms a nonlinear system into a linear one given a suitable change of variables and control input [162], [163]. Then, one can obtain convergence guarantees for the linear system. In order to begin this development along the lines of the preview which was shown in (3.8), we must first ensure that a system is feedback linearizable.

Definition 3.2.0.3. *An affine control system is* ***feedback linearizable*** *if there exists a control law $u = K(x, \nu)$, $K : X \times \mathscr{V} \to U$ for $\mathscr{V} \subset \mathbb{R}$, such that:*

$$\dot{x} = f(x) + g(x)K(x,\nu) \qquad \Longrightarrow \qquad y^{(r)}(x) = \nu, \tag{3.18}$$

for the auxiliary control input $\nu \in \mathscr{V} \subset \mathbb{R}$.

Let us begin the derivation of our controller for this system within the context of robotic systems, which are the sole subject of this thesis, by again considering the time derivatives of the outputs (3.9) and (3.10):

$$\begin{aligned}
\dot{y}_1(q,\dot{q}) &= \frac{\partial y_1(q,\dot{q})}{\partial q}\dot{q} + \frac{\partial y_1(q,\dot{q})}{\partial \dot{q}}\ddot{q} \\
&= \underbrace{\frac{\partial y_1(q,\dot{q})}{\partial q}\dot{q} + \frac{\partial y_1(q,\dot{q})}{\partial \dot{q}}\Big[-D^{-1}(q)H(q,\dot{q})\Big]}_{L_f y_1(q,\dot{q})} + \underbrace{\frac{\partial y_1(q,\dot{q})}{\partial \dot{q}}D^{-1}(q)B}_{L_g y_1(q,\dot{q})}\,u, \qquad (3.19)\\
\dot{y}_2(q,\dot{q}) &= \underbrace{\frac{\partial y_2(q)}{\partial q}\dot{q}}_{L_f y_2(q,\dot{q})} \qquad (3.20)\\
\ddot{y}_2(q,\dot{q}) &= \frac{\partial}{\partial q}\left(\frac{\partial y_2(q)}{\partial q}\dot{q}\right)\dot{q} + \frac{\partial y_2(q)}{\partial q}\ddot{q} \\
&= \underbrace{\frac{\partial}{\partial q}\Big(\frac{\partial y(q)}{\partial q}\dot{q}\Big)\dot{q} + \frac{\partial y_2(q)}{\partial q}\Big[-D^{-1}(q)H(q,\dot{q})\Big]}_{L_f^2 y_2(q,\dot{q})} + \underbrace{\frac{\partial y(q)}{\partial q}D^{-1}(q)B}_{L_g L_f y_2(q,\dot{q})}\,u, \qquad (3.21)
\end{aligned}$$

where the spring force term $B_{\text{sp}}\tau_{\text{sp}}$ has been lumped with $H(q,\dot{q})$ in the interest of space, L_f and L_g are the Lie derivatives with respect to the vector fields $f(x)$ and $g(x)$.

For more concise representation, terms are also grouped with a common notation:

$$\begin{bmatrix}\dot{y}_1\\ \ddot{y}_2\end{bmatrix} = \underbrace{\begin{bmatrix}\frac{\partial y_1}{\partial q} & \frac{\partial y_1}{\partial \dot{q}}\\ \frac{\partial}{\partial q}\left(\frac{\partial y_2}{\partial q}\dot{q}\right) & \frac{\partial y_2}{\partial q}\end{bmatrix} f(x)}_{\mathbf{L}_f y(x)} + \underbrace{\begin{bmatrix}\frac{\partial y_1}{\partial \dot{q}}\\ \frac{\partial y_2}{\partial q}\end{bmatrix} g(x)}_{\mathcal{A}(x)} u, \tag{3.22}$$

now in the form of (3.8). It is clear from this derivation that the system (2.1) is feedback linearizable. Further, we can see that the *decoupling matrix*, $\mathcal{A}(x)$, is invertible due to the specific choice of virtual constraints. We can then prescribe the following control law:

$$u(x) = \mathcal{A}^{-1}(x)\Big(- L_f y(x) + \nu\Big) \qquad \Longrightarrow \qquad y^{(r)} = \nu, \tag{3.23}$$

where ν is an auxiliary feedback control value. Assuming that the preliminary feedback (3.23) has been applied to (2.1), we will render a linear system for the output dynamics with coordinates $\eta := (y_1^T, y_2^T, \dot{y}_2^T)^T$:

$$\dot{\eta} = \begin{bmatrix}\dot{y}_1\\ \dot{y}_2\\ \ddot{y}_2\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 0\\ 0 & \mathbf{I}\\ 0 & 0\end{bmatrix}}_{F}\eta + \underbrace{\begin{bmatrix}\mathbf{I} & 0\\ 0 & 0\\ 0 & \mathbf{I}\end{bmatrix}}_{G} v. \tag{3.24}$$

A valid choice of ν which stabilizes this linear system is:

$$\nu = \begin{bmatrix}\dot{y}_1\\ \ddot{y}_2\end{bmatrix} = \begin{bmatrix}-\frac{1}{\varepsilon}K_{\bar{v}}y_1\\ -\frac{1}{\varepsilon^2}K_P y_2 - \frac{1}{\varepsilon}K_D\dot{y}_2\end{bmatrix}, \tag{3.25}$$

where $0 < \varepsilon \leq 1$ is a tunable parameter, and $K_{\bar{v}}$, K_P, $K_D > 0$ are control gains for the relative degree 1 and relative degree 2 output errors, respectively. This can be grouped into the closed-loop linear system:

$$\begin{bmatrix}\dot{y}_1\\ \dot{y}_2\\ \ddot{y}_2\end{bmatrix} = \underbrace{\begin{bmatrix}-\frac{1}{\varepsilon}K_{\bar{v}} & 0 & 0\\ 0 & 0 & \mathbf{I}\\ 0 & -\frac{1}{\varepsilon^2}K_P & -\frac{1}{\varepsilon}K_D\end{bmatrix}}_{F_{\text{cl}}}\begin{bmatrix}y_1\\ y_2\\ \dot{y}_2\end{bmatrix}. \tag{3.26}$$

Since F_{cl} is Hurwitz by definition (meaning that $\text{Re}(\text{eig}(F_{\text{cl}})) < 0$), the resulting linear dynamics is exponentially stable. In addition, the control parameter ε forces the system to converge at a rate governed by ε. This will be useful in later sections, where we note that the selection of $0 < \varepsilon < 1$ creates *rapidly exponential stability*, and $\varepsilon = 1$ instead renders *exponential stability*.

3.2.3 Zero Dynamics

The application of the feedback controller (3.23) to each domain of the nonlinear system (2.1) with the auxiliary control choice of (3.25) can be synthesized for virtual constraints $y_v(q) = y_v^a(q) - y_v^d(\tau(q,t),\alpha_v)$ associated with each domain $\mathcal{D}_v$, $v \in V$, yielding the nonlinear "closed-loop" system:

$$\dot{x} = f_v^\alpha(x,\alpha_v) = f_v(x) + g_v(x)u^*(x,\alpha_v), \tag{3.27}$$

wherein for this system $y \to 0$ exponentially fast according to the closed-loop response, which is dependent on α_v. The multi-domain hybrid system associated with the closed-loop dynamical system is described as [164]:

$$\mathscr{H}^\alpha \triangleq \begin{cases} \dot{x} = f_v^\alpha(x,\alpha_v) & \text{if} \quad x \in \mathcal{D}_v^\alpha \backslash S_e^\alpha \\ x^+ = \Delta_e(x^-) & \text{if} \quad x^- \in S_e^\alpha \end{cases} \tag{3.28}$$

where f_v^α is a set of dynamical systems defined on $\mathcal{D}_v^\alpha$ for the system (3.27). Moreover, the control law prescribed in (3.23)-(3.25) renders the *zero dynamics* submanifold:

$$\mathcal{Z}_v^\alpha = \{(q,\dot{q}) \in \mathcal{D}_v | y_{1,v} = 0, y_{2,v} = \mathbf{0}, L_f y_{2,v} = \mathbf{0}\} \tag{3.29}$$

invariant over the continuous dynamics of the domain [163]. In other words, any solution that starts in $\mathcal{Z}_v^\alpha$ remains on $\mathcal{Z}_v^\alpha$ until reaching a guard. Care has been taken in (3.29) and the associated hybrid system (3.28) to show the explicit dependence on the parameter set α_v. This implies that the zero dynamics surface can actually be shaped through the selection of these parameters.

3.2.4 Hybrid Zero Dynamics

The surface in Equation (3.29) has been designed without taking into account the hybrid transition maps of (2.29), meaning that there is no guarantee that the resulting walking cycle will be invariant to impact. To enforce impact invariance, the

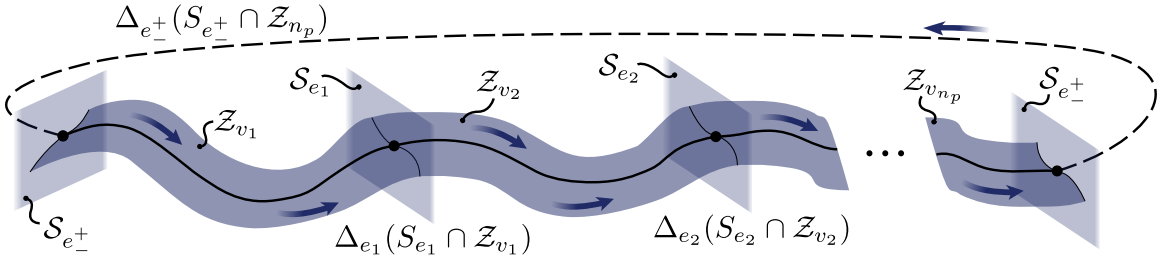

Figure 3.1: A depiction of the multi-domain hybrid system geometry and its associated notation.

desired outputs can be shaped through the parameters α_v in y_v^d such that the walking satisfies a *hybrid invariance* condition which necessarily guarantees that the reduced dimensional manifold is invariant to the impact dynamics due to the discrete change of velocities when reaching an edge.

Definition 3.2.0.4. *At a given edge $e \in E$, the reduced dimensional hybrid system* (3.28) *is impact invariant, and thus has* ***hybrid zero dynamics*** *(HZD) if and only if $\mathcal{Z}_{v_s} \cap S_e \neq \emptyset$ and:*

$$\Delta_e(\mathcal{Z}_{v_s} \cap S_e) \subset \mathcal{Z}_{v_t}, \qquad \forall\ e = (v_s, v_t) \in E, \tag{3.30}$$

where v_s and v_t are the source and target continuous domains mapped by the edge e. A submanifold $\mathcal{Z} \bigcup_{v \in V} \mathcal{Z}_v$ is called hybrid invariant if it is invariant over all domains of continuous dynamics and impact invariant through all discrete dynamics.

The main ideas of hybrid zero dynamics are illustrated in Fig. 3.2, which shows how zeroing the outputs renders a reduced dimensional surface that can be shaped to achieve stable periodic motions. In fact, the HZD condition (3.30) implies stability of the overall system dynamics if the zero dynamics are rendered stable through the continuous domains of the hybrid motion. Additionally, the satisfaction of the invariance condition (3.30) guarantees HZD for the full multi-domain hybrid system.

The overarching goal of these constructions is to provide a framework for the synthesis of dynamic walking gaits. With regards to the actual design of walking motions, shaping the zero dynamics to obtain a stable periodic orbit is perhaps the most challenging task. It can thus be beneficial in the development of HZD controllers or in analysis of the hybrid system to directly deal with coordinates on the zero dynamics manifold rather than within full-order state space (3.27). To construct a restricted hybrid system model, let $z_v \in \mathcal{Z}_v \subset \mathbb{R}^{l_v}$ be the local coordinates of the zero dynamics in a given domain, and let $\eta_v \in \mathcal{D}_v^\alpha \subset \mathbb{R}^{n_v - l_v}$ be the controlled normal states corresponding to our choice of feedback linearized output dynamics in (3.24) and transversal to $\mathcal{Z}_v$. Suppose that there exists local coordinate transformations $\Phi_v^z : \mathcal{D}_v \to \mathcal{Z}_v$ and $\Phi_v^\eta : \mathcal{D}_v \to \mathcal{D}_v^\alpha$ so that $(\eta_v, z_v) = (\Phi_v^\eta(x), \Phi_v^z(x)) := \Phi_v(x)$. With these transformations, we can write our controlled system in *normal form* as:

$$\begin{aligned} \dot{\eta} &= \bar{f}(\eta, z) + \bar{g}(\eta, z)u \\ \dot{z} &= \omega(\eta, z) \end{aligned} \tag{3.31}$$

where because η is defined in terms of x due to our selection of the virtual constraints, y, we can write the dynamics as:

$$\bar{f}(\eta(q,\dot{q}),z(q,\dot{q})) = \begin{bmatrix} \dot{y}(q,\dot{q}) \\ \mathbf{L}_f y(q,\dot{q}) \end{bmatrix}, \quad \bar{g}(\eta(q,\dot{q}),z(q,\dot{q}))) = \begin{bmatrix} 0 \\ \mathcal{A}(q,\dot{q}) \end{bmatrix}. \tag{3.32}$$

The closed-loop dynamics of the η coordinates evolve according to the previously shown linear time-invariant system, $\dot{\eta}_v = F_{\text{cl}}\eta_v$, and describe the dynamics transversal to $\mathcal{Z}_v$. In essence, the feedback linearizing controller and choice of auxiliary feedback control law render the zero dynamics surface attractive and invariant under the continuous dynamics of the system, i.e. $\eta \to 0$ and $\eta(0) = 0$ implies that $\eta(t) \equiv 0 \; \forall \; t \geq 0$. Thus, we can write the zero dynamics as the maximal dynamics compatible with the output equal to zero:

$$\dot{z} = \omega(0,z) := f|^{\alpha}_{\mathcal{Z}_v}(z). \tag{3.33}$$

Then, according to Definition 3.2.0.4, if the zero dynamics has been shaped to achieve impact invariance, the multi-domain hybrid control system has hybrid zero dynamics, $\mathscr{H}|^{\alpha}_{\mathcal{Z}}$. The restricted hybrid dynamical system with impulses can then be written as:

$$\mathscr{H}^{\alpha}|_{\mathcal{Z}} \triangleq \begin{cases} \dot{z} = f^{\alpha}|_{\mathcal{Z}_v}(z) & \text{if} \quad z \in \mathcal{Z}_v^{\alpha} \backslash S_e^{\alpha} \\ z^+ = \Delta_e|_{S_e^{\alpha} \cap \mathcal{Z}_v^{\alpha}}(z^-) & \text{if} \quad z^- \in S_e^{\alpha} \end{cases} \tag{3.34}$$

The stability of hybrid systems, particularly ones which exhibit periodic behaviors, is often determined by the existence and stability of periodic orbits. If the system (3.34) has HZD, then due to the hybrid invariance of $\mathcal{Z}$, there exits a stable hybrid periodic orbit, $\mathcal{O}|_{\mathcal{Z}} \subset \mathcal{Z}$, for the reduced order zero dynamics evolving on $\mathcal{Z}$, i.e., if we are evolving on the restriction dynamics of $f^{\alpha}|_{\mathcal{Z}}(z)$, then $\mathcal{O}|_{\mathcal{Z}}$ is a stable hybrid periodic orbit for the restricted dynamics in (3.34) [30]. More concretely, using the constructions introduced in Sec. 2.4, let $\varphi_t^{f^{\alpha}}|_{\mathcal{Z}}(z_0)$ be the (unique) solution to (3.33) at time $t \geq 0$ with initial condition z_0. For a point $z^* \in S$, we say that $\varphi_t^{f^{\alpha}}|_{\mathcal{Z}}$ is hybrid periodic if there exists a $T > 0$ such that $\varphi_T^{f^{\alpha}}|_{\mathcal{Z}}(\Delta(z^*)) = z^*$. Further, the stability of the resulting hybrid periodic orbit, $\mathcal{O}|_{\mathcal{Z}} = \{\varphi_t^{f^{\alpha}}|_{\mathcal{Z}}(\Delta(z^*)) : 0 \leq t \leq T\}$, can be found by analyzing the stability of the Poincaré map, wherein z^* is a fixed point, as previously presented in (2.33).

On the basis of (3.30), we define the *restricted Poincaré map* $\rho_v = P_v|_{\mathcal{Z}}$, where $\rho_v : S_{v_s} \cap \mathcal{Z}_v \to S_{v_t} \cap \mathcal{Z}_v$. The Poincaré return map for the multi-domain hybrid zero

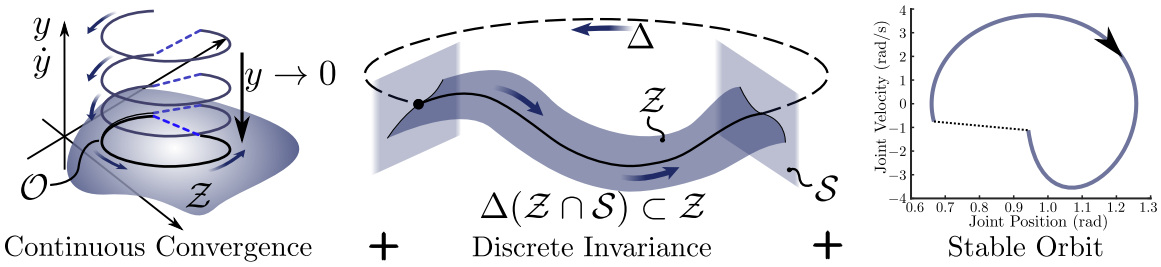

Figure 3.2: Illustration of the key concepts related to hybrid zero dynamics: continuous convergence to a low dimensional zero dynamics surface Z, with an invariance condition: $\Delta(\mathcal{S} \cap \mathcal{Z}) \subset \mathcal{Z}$, and the end goal being a stable periodic walking gait.

dynamics can be defined as the composition of all maps for one gait cycle:

$$\rho := \rho_{v_1} \circ \rho_{v_2} \circ ... \circ \rho_{v_{n_p}}. \tag{3.35}$$

It is shown in [111], [165] that if there exists a feedback control law rendering the invariant manifold sufficiently rapidly attractive, then the existence and stability of the periodic orbit $\mathcal{O}$ of the full order system can be determined from those of the restricted (reduced-dimensional) zero dynamics periodic orbit, $\mathcal{O}|_{\mathcal{Z}}$, through the canonical embedding $\iota : \mathcal{Z} \to \mathcal{D}$. Thus, if the system has HZD and there exists a periodic orbit for $\mathcal{O}|_{\mathcal{Z}}$ of $\mathscr{H}^{\alpha}|_{\mathcal{Z}}$, then the full order system also has a hybrid periodic orbit $\mathcal{O} = \iota(\mathcal{O}|_{\mathcal{Z}})$.

Theorem 3.2.1. *(Orbital Stability of the Hybrid Zero Dynamics) Suppose that $\mathcal{O}|_{\mathcal{Z}}$ is an exponentially stable transverse periodic orbit of the corresponding hybrid zero dynamics system* (3.34)*. Then there exists an $\bar{\varepsilon} > 0$ such that for all $\bar{\varepsilon} > \varepsilon > 0$, the hybrid periodic orbit $\mathcal{O}|_{\mathcal{Z}}$ is locally exponentially stable for the full order hybrid periodic orbit $\mathcal{O} = \iota(\mathcal{O}|_{\mathcal{Z}})$ of the hybrid system $\mathcal{H}^{\alpha}$.*

3.2.5 Partial Hybrid Zero Dynamics

If a walking cycle has been prescribed with relative degree 1, or velocity based outputs, the impact invariance condition of (3.30) is often too strong to be obtained due to the change in velocities caused by the impact map. Therefore, zero dynamics is only enforced on the relative degree 2 outputs, resulting in a *partial zero dynamics* [155], given by:

$$\mathcal{PZ}_v = \{(q, \dot{q}) \in \mathcal{D}_v | y_{2,v} = \mathbf{0}, \dot{y}_{2,v} = \mathbf{0}\}. \tag{3.36}$$

Moreover, we say that the submanifold $\mathcal{PZ}_v$ is impact invariant if and only if there exist a set of parameters $\alpha = \{\alpha_v\}_{v \in V}$ and v_d so that:

$$\Delta_e(\mathcal{PZ}_{v_s} \cap S_e) \subset \mathcal{PZ}_{v_t}, \qquad \forall\, e = (v_s, v_t) \in E. \tag{3.37}$$

A manifold $\mathcal{PZ} = \mathcal{PZ}_{v_s} \cup \mathcal{PZ}_{v_t}$ is thus called hybrid invariant if it is invariant over all domains of continuous dynamics and impact invariant through all discrete dynamics. If a feedback control law renders $\mathcal{PZ}$ hybrid invariant, then the multi-domain hybrid control system has a *partial hybrid zero dynamics* (PHZD), $\mathscr{H}|_{\mathcal{PZ}}$.

3.2.6 Non-autonomous Considerations

Recall how it was previously shown for the autonomous virtual constraint formulation that the derivatives of the outputs took on a feedback linearizable form in (3.8). In the case of a trajectory tracking problem, or in other words the tracking of *time-based desired outputs*, we must separately consider the actual outputs, $y^a(q, \dot{q})$, and desired outputs, $y^d(\tau(t), \alpha)$, as one is a function of state and the other time:

$$y(q,t) = y^a(q) - y^d(\tau(t), \alpha), \qquad \tau(t) = \frac{t - t_0}{t_f - t_0}, \tag{3.38}$$

where the loss of state dependence in $y^d(\tau(t), \alpha)$ is due to $\tau(t)$. For general outputs of relative degree r, the non-autonomous output tracking problem has the derivatives:

$$y^{(r)}(x) = \underbrace{L_f^{(r)} f(x)}_{L_f^{(r)} y^a(x)} - y^{d,(r)}(\tau(t), \alpha) + \underbrace{L_g L_f^{(r-1)} g(x)}_{\mathcal{A}(x)} u, \tag{3.39}$$

where $y^{d,(r)}(\tau(t), \alpha)$ is the rth time derivative of the desired output. We can then feedback linearize the system through the time-based feedback control law:

$$u(x) = \mathcal{A}^{-1}(x)\Big(- L_f^{(r)} y^a(x) + y^{d,(r)}(\tau(t), \alpha) + \nu \Big). \tag{3.40}$$

For our more specific representation of robotic outputs only considering relative degree 1 and 2 outputs, we then select a slightly modified set of normal coordinates:

$$\eta_t := \begin{bmatrix} y_1^a(q, \dot{q}) \\ y_2^a(q) \\ \dot{y}_2^a(q, \dot{q}) \end{bmatrix} - \begin{bmatrix} y_1^d(\tau(t)) \\ y_2^d(\tau(t), \alpha) \\ \dot{y}_2^d(\tau(t), \alpha) \end{bmatrix}, \tag{3.41}$$

from which we can derive a similar linear system to what was shown in (3.24), and obtain a closed-loop stable system through the selection of the auxiliary control, ν.

Recall that for a state-based phase variable $\tau(q)$ in (3.11), the derivatives of our desired outputs in (3.15) and (3.16) were also configuration dependent, leading to their representation in Lie derivative form. For a time-based Bézier output, these derivatives are only a function of polynomial coefficients and time:

$$\dot{y}_2^d(\tau(t), \alpha_v) = \frac{\partial b(\tau, \alpha_v)}{\partial \tau} \dot{\tau}(t) \tag{3.42}$$

$$\ddot{y}_2^d(\tau(t), \alpha_v) = \frac{\partial^2 b(\tau, \alpha_v)}{\partial \tau^2} \dot{\tau}^2(t). \tag{3.43}$$

3.2.7 Output Selection

The designation of outputs is perhaps the area in which a control engineer is given the largest amount of creative freedom when working within the HZD framework. While output selection can directly influence the resulting behaviors of a legged robot, there are no hard and fast rules on what necessarily constitutes a "best" selection. Past work has looked to human motions [146], reduced order models [166], or directly used joint angles to select outputs.

The approach presented here is inspired by the authors' previous work on human-inspired bipedal locomotion [146]. One of the first observations that this method leverages is that the linearized position of the hip, $\delta p_{hip}(q)$, appears to be essentially a linear function of time during forward walking. This feature makes it particularly suitable for selection as a phase variable. On a biped of human-like morphology, this can be written mathematically as:

$$\delta p_{hip}(q) = L_a\theta_{ra} + (L_a + L_c)\theta_{rk} + (L_a + L_c + L_t)\theta_{rh}, \tag{3.44}$$

where L_a, L_c, and L_t denote lengths for the ankle, calf, and thigh linkages of the robot. The linearized forward hip velocity can then be correspondingly picked as the velocity-modulating output:

$$y^a_{1,v}(q, \dot{q}) = \delta \dot{p}_{hip}(q, \dot{q}). \tag{3.45}$$

It is important to note that numerous other selections are possible. Examples of other choices include the position of the center of mass or the pitch angle of the stance ankle.

As the number of holonomic constraints applied to the robot change through the application of various contacts in different domains, so too will the choice of position-modulating outputs. Let m_r denote the number of actuators on the robot, and $n_{\eta,v}$ be the number of holonomic constraints applied in a given domain. We then say that a domain is *Fully-actuated* if $m_r = n - n_{\eta,v}$, *Under-actuated* if $m_r < n - n_{\eta,v}$, and *Over-actuated* if $m_r > n - n_{\eta,v}$. The proper number and configuration of outputs should be selected such that the decoupling matrix $\mathcal{A} = L_g L_f^{(r-1)} y(x)$ has full row-rank. This ultimately means that the Jacobians of the outputs must be linearly independent of one another, and linearly independent with regard to the constraints applied to the robot. Thus for fully- and under-actuated domains, the total number of outputs will then be equal to m_r, while in over-actuated domains, the number of outputs need to be reduced to $n - n_{\eta,v}$.

3.3 Trajectory Optimization

Throughout the previous section, we outlined how the HZD locomotion problem is fundamentally different than traditional approaches to modeling and control of fixed-base robots. It is because of this inherent complexity that virtually all approaches to realize dynamic walking must transcribe the locomotion problem into a motion planner which can handle the various constraints naturally imposed on the problem. While several of the more classical walking paradigms offer simple solutions to conservative walking, there has been a push over the last two decades towards leveraging optimization to obtain increasingly dynamic maneuvers. In this section, it is shown how trajectory optimization can be used to design virtual constraints. This methodology will be essential both to shaping the behavior to achieve desired outcomes, such as efficiency, and to obtain hybrid invariant locomotion.

3.3.1 Background

Open-Loop Optimization for Gait Generation

As a result of the rapid developments within the trajectory optimization community, researchers began to move towards utilizing nonlinear dynamic gait optimizations rather than relying on the constraints imposed by linear modeling assumptions. Using nonlinear optimization, i.e., numerical approaches, to generate stable walking behaviors on bipeds is not a new concept [167], [168], though computational limitations were a considerable hindrance towards generating motions on 3D robots. Computation power finally increased sufficiently throughout the mid 2000's to begin handling 3D dynamic walking behaviors [169]. The simplest application of nonlinear optimization to walking can be found in Sec. 1.2.5, wherein passive dynamic walking relies on the generation of fixed points associated with periodic orbits of a hybrid dynamical system. This naturally lends itself to numerical approaches for the optimization of *open-loop* stable periodic motions [170], since passive dynamic walkers do not have any actuators to consider. The use of open-loop optimization to generate feasible motions for actuated robots are a natural extension of approaches used throughout the field of trajectory optimization, where the planning problem is seen as "decoupled" from the feedback control applied to the actual robot [171] and "approximately optimal" solutions are often sufficient. Further, in recent years, the application of advanced trajectory optimization methods such as *direct collocation* have allowed the optimization of the full body dynamics of (2.7) to be more computationally tractable, sparking a growing interest in considering the fullbody dynamics of robot in the planning problem. For instance, in order to control the open-loop trajectory that results

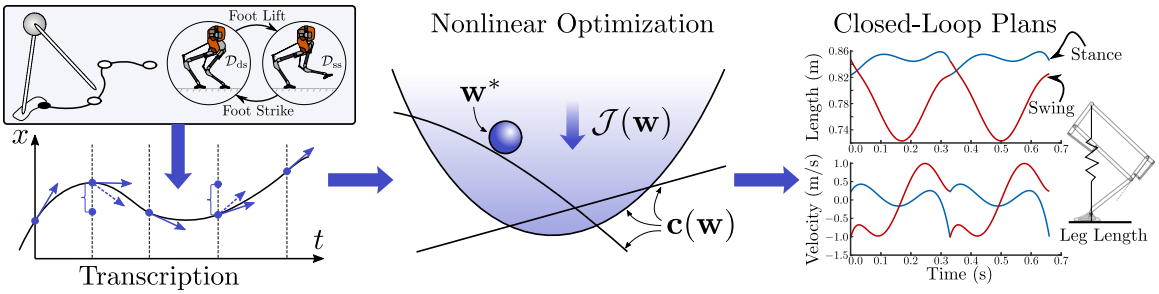

Figure 3.3: A conceptual illustration of how locomotion models must be first transcribed into appropriate representations for use with nonlinear programming approaches in order to yield dynamically stable closed-loop plans for bipedal robots.

from the direct collocation optimization, a classical linear quadratic regulator (LQR) based feedback controller can be constructed to stabilize the resulting trajectory obtained for the constrained dynamical system [172]. In this type of approach, the walking problem can be viewed as generating sequences of footholds for the nonlinear centroidal dynamics given in (1.2) [59], [173] or with respect to the full Lagrangian system given in Equation (2.7) [174]. Complementary Lagrangian systems [136] formed the basis of the approach in [174], which allowed the optimizer to find walking behaviors without a priori enumeration of the type and order of contact events. Open-loop trajectory optimization has also been used for ZMP conditions in a nonlinear fashion [175], considerably improving the conservative walking presented in Sec. 1.2.1.

Closed-Loop Optimization

While the preceding nonlinear optimization approaches do consider the fullbody dynamics of the robot, it is not always desirable to apply feedback controllers to stabilize an approximately optimal open-loop plan. Rather, it is often beneficial to couple the gait generation and controller synthesis problems into a single framework: *closed-loop optimization*. This allows, among other things, for the generation of *provably stable* walking behaviors which simultaneously satisfy the constraints on the system from admissible configurations to torque bounds. This idea forms the basis of designing walking gaits with the HZD method introduced in Sec. 1.2.6, where feedback control is used to generate provable stable periodic orbits. A visual summary of this section is given in Fig. 3.3. By applying these closed-loop feedback strategies in the optimization problem, ambiguous contact sequences are no longer possible [176] and must be prescribed according to the directed cycle which governs the underlying hybrid system. Doing so allows for the enforcement of feasibility constraints, e.g., unilateral contact, in conjunction with the synthesis of controllers that guarantee stability.

3.3.2 Gait Optimization Problem

In the context of HZD methods, with the formal constructions of the zero dynamics (3.29) and hybrid invariance (3.30) defined, the problem of finding stable dynamic walking can be transcribed to a nonlinear programming (NLP) problem of finding a fixed point x^* and set of parameters $\alpha = \{\alpha_v\}_{v \in V}$ parameterizing the virtual constraints of Equation (3.7). The optimization problem is performed over one full step cycle, e.g., footstrike to corresponding footstrike which forms a repeating orbit, with a constraint imposed such that when the discrete impact (2.29) is applied to the terminal state, so it satisfies the hybrid invariance condition of (3.30). It is also critical that the motions respect the limitations of the physical system such as the friction cone (2.21), foot rollover conditions (2.23), actuator limits, and joint limits. These constraints can be directly placed into a NLP problem that can be solved by a standard optimization solver:

$$\begin{aligned} \mathbf{w}(\alpha)^* = \underset{\mathbf{w}(\alpha)}{\operatorname{argmin}} \quad & \mathcal{J}(\mathbf{w}(\alpha)) && \text{(HZD Optimization)} \\ \text{s.t.} \quad & \text{Closed loop dynamics: Equation (3.27)} \\ & \text{HZD condition: Equation (3.30)} \\ & \text{Physical feasibility (e.g. Equation (2.21))} \end{aligned}$$

where $\mathbf{w}(\alpha) \in \mathbb{R}^{N_w}$, with N_w being the total number of optimization variables and here we made the dependence on the parameters, α, that dictate the closed loop dynamics explicit.

In classical HZD implementations, the candidate solutions were found via direct single shooting formulations [29], [155], where the decision variables are the fixed point states x^* and the desired output coefficients α. Because single shooting optimizations are notoriously sensitive to poor initial conditions, multiple shooting methods were also explored [177], with the eventual development of direct collocation formulations [164] which would become the most successful to date. The FROST optimization package [129] was developed based on these successes as an open-source package to transcribe HZD locomotion into a direct collocation problem. In this thesis, the FROST approach to direct collocation and trajectory optimization is the core component to the HZD based optimization success. The robots which will be introduced in Chap. 4 are so complex that a direct single shooting formulation would either take too long to converge or fail completely too often for it to be useful.

Collocation Algorithm

Here, we simply introduce the main idea of the direct collocation optimization [164]. In particular, the solution of each domain, $\mathcal{D}_v$, is discretized based on the time discretization $0 = t_0 < t_1 < t_2 < \cdots < t_{N_v} = T_{I,v}$, assuming $T_{I,v} > 0$ is the time at which the system reaches the guard associated with a given domain. Let x^i and $\dot{x}^i$ be the approximated states and first order derivatives at node i, the defect constraints are defined at each odd node as:

$$\begin{aligned} \zeta_v^i &:= \dot{x}^i - 3(x^{i+1} - x^{i-1})/2\Delta t_v^i + (\dot{x}^{i-1} + \dot{x}^{i+1})/4 = 0, \\ \delta_v^i &:= x^i - (x^{i+1} + x^{i-1})/2 - \Delta t_v^i(\dot{x}^{i-1} - \dot{x}^{i+1})/8 = 0, \end{aligned} \tag{3.46}$$

where $\Delta t_v^i = t_{i+1} - t_{i-1}$ is the time interval. Moreover, the first order derivatives must satisfy the system dynamics, i.e., the restricted hybrid dynamics (3.34).

Constrained Hybrid Dynamics as Differential Algebraic Equations

Recall that reduced dimensional restricted dynamics, i.e., the zero dynamics, is determined via the full order dynamics (2.8) subject to the holonomic constraints and virtual constraints being zero. Thus, the restricted dynamics can be described as the differential algebraic equations:

$$F_v(q, \dot{q}, \ddot{q}, u, \lambda_v, \alpha_v) := \begin{bmatrix} D(q)\ddot{q} + H(q, \dot{q}) - Bu - J_v^T(q)\lambda_v \\ J_v(q)\ddot{q} + \dot{J}_v(q, \dot{q})\dot{q} \\ \dot{\eta}_v(q, \dot{q}, \ddot{q}, \alpha_v) - \nu_v \end{bmatrix} = 0, \tag{3.47}$$

subject to the initial value conditions:

$$h_v(q^+) = \bar{h}_v, \quad J_v(q^+)\dot{q}^+ = 0, \quad y_{2,v}(q^+) = 0, \quad \dot{y}_{2,v}(q^+, \dot{q}^+) = 0, \tag{IVC}$$

where $\bar{h}_v$ is a vector of constants, and $(q^+, \dot{q}^+)$ are the initial state values. This system can be considered as an implicit form that is equivalent to the zero dynamics equation by its definition. It is shown in [176] that this implicit equations yields a much simpler representation of the constrained system due to the fact that we no longer need to invert the inertia matrix $D(q)$ or the decoupling matrix $L_g L_f y_{2,v}$ in (3.23) to solve for the explicit expression of the zero dynamics.

Moreover, the trajectories of the system states of two neighboring domains are connected via the discrete dynamics captured in the reset maps. Specifically, suppose that $(q^+, \dot{q}^+)$ are the post-impact states of a particular domain and $(q^-, \dot{q}^-)$ are the

pre-impact states of the previous domain, then they must satisfy:

$$(q^+, \dot{q}^+) - \Delta_e(q^-, \dot{q}^-) = 0, \tag{3.48}$$

where $e \in E$ corresponds to the transition between these two domains. This constraint together with the initial value constraint in (IVC) guarantee that the hybrid invariant conditions are satisfied, therefore, the solution to the optimization lies in the hybrid zero dynamics manifold given in (3.29) and satisfies the HZD condition (3.30).

Remark 2. *The representation of the dynamics as the implicit DAE system of* (3.47) *no longer requires the inversion of the inertia matrix,* $D(q)$*, to solve the zero dynamics equation* (3.29)*, and is thus a significantly simpler representation of the constrained floating-base system to evaluate from a computational perspective.*

General Direct Collocation Optimization Formulation

In the previous section, we introduced the continuous dynamics and hybrid invariance condition for the continuous dynamical system. If we apply the time-discretization introduced in Equation (3.46), then we must correspondingly apply the time discretization of the constraints on the continuous system as:

$$F_v(q, \dot{q}, \ddot{q}, u, \lambda_v, \alpha_v) = \begin{bmatrix} F_v(q^{(0)}, \dot{q}^{(0)}, \ddot{q}^{(0)}, u^{(0)}, \lambda_v^{(0)}, \alpha_v) \\ F_v(q^{(1)}, \dot{q}^{(1)}, \ddot{q}^{(1)}, u^{(1)}, \lambda_v^{(1)}, \alpha_v) \\ \vdots \\ F_v(q^{(N_v)}, \dot{q}^{(N_v)}, \ddot{q}^{(N_v)}, u^{(N_v)}, \lambda_v^{(N_v)}, \alpha_v) \end{bmatrix} = 0, \tag{3.49}$$

where $i = 0, 1, \ldots, N_v$ indicate the i-th cardinal node in the time discretization for each domain. These continuous dynamics must also satisfy the admissibility constraints at each time step:

$$A_v(q, \dot{q}, \lambda_v) = \begin{bmatrix} A_v(q^{(0)}, \dot{q}^{(0)}, \lambda_v^{(}0)) \\ A_v(q^{(1)}, \dot{q}^{(1)}, \lambda_v^{(}1)) \\ \vdots \\ A_v(q^{(N_v)}, \dot{q}^{(N_v)}, \lambda_v^{(}N_v)) \end{bmatrix} \geq 0. \tag{3.50}$$

Combining the time discretized dynamics with the collocation constraints, discrete dynamics, initial value condition for hybrid invariance, and various variable boundaries, we can now state the more direct collocation form of the general optimization problem which was stated in (HZD Optimization). Let $\mathbf{w}$ be a vector containing

all optimization variables and $\mathbf{c}(\mathbf{w})$ be a vector of constraint functions. The direct collocation HZD optimization is then [164]:

$$\begin{aligned}
\mathbf{w}^* = \underset{\mathbf{w}}{\text{argmin}} \quad & \sum_{v \in V} \mathcal{J}_v(\mathbf{w}) && \text{(HZD Direct Collocation)} \\
\text{s.t.} \quad & \zeta_v^i(\mathbf{w}),\ \delta_v^i(\mathbf{w}) = 0 \qquad i = 1, 3, \ldots, N_v - 1 && \text{(Defects)} \\
& F_v(\mathbf{w}) = 0 && \text{(CL Dynamics)} \\
& A_v(\mathbf{w}) \geq 0 && \text{(Admissibility)} \\
& x_{v_t}^{(0)}(\mathbf{w}) - \Delta_e(x_{v_s}^{(N_{v_s})}(\mathbf{w})) = 0 && \text{(Discrete Dynamics)} \\
& \text{IVC}(\mathbf{w}) && \text{(HZD)} \\
& \mathbf{w}^{\min} \leq \mathbf{w} \leq \mathbf{w}^{\max} && \text{(Bounds)}
\end{aligned}$$

Using this optimization formulation, we can solve for not only the polynomial coefficients which parameterize the stable feedback controller for an optimal walking gait, but due to the discretization and direct collocation formulation, also directly obtain all discrete time-integration values for the robot configuration, generalized accelerations, and constraint forces as optimization variables. This direct collocation algorithm will be used as the general optimization framework for finding walking trajectories for the experimental studies shown in Chap. 5 and Chap. 6.

Cost Functions

The choice of cost function or performance metric, $\mathcal{J}_v$ for use in the optimization is largely a design choice influenced by the individual motion task. Within the field of dynamic walking, this choice most often coincides with torque reduction, mechanical power reduction, or by using the mechanical cost of transport. The simplest of these choices is to minimize the norm of the torque squared:

$$\mathcal{J}_u(\mathbf{w}) := \sum_{v \in V} \sum_{i=0}^{N_v - 1} ||u_v^i||^2. \tag{3.51}$$

However, in some cases simply minimizing overall torque may not always be desirable since it does not always have a direct correlation to overall efficiency. Instead, we can look at the 2-norm sum of the mechanical power, given as

$$P_v^i(\dot{q}, u) = \|(u^i)^T \dot{q}^i\|, \tag{3.52}$$

$$\mathcal{J}_P(\mathbf{w}) := \sum_{v \in V} \sum_{i=0}^{N_v - 1} P_v(\dot{q}, u), \tag{3.53}$$

where u_v^i and $\dot{q}_v^i$ is the computed torques and joint velocities at the i-th cardinal node.

In some cases, it becomes useful to also incentivize forward movement even if it requires more power. For this purpose, we define the cost function of our gait optimization for minimizing the total mechanical cost of transport of the gait [43]:

$$\mathcal{J}_{\text{COT}}(\mathbf{w}) := \frac{1}{mgd} \sum_{v \in V} \sum_{i=0}^{N_v - 1} \left(\frac{\| P_v(\dot{q}^i, u_v^i) \| \cdot \Delta t_i}{T_{I,v}^i} \right), \tag{3.54}$$

where mg is the robot weight, d is the distance traveled, and $P_v(u_v^i, \dot{q}^i)$ is the total power consumed assuming no power-regeneration at each integration node i of the discretized problem passed to the NLP solver [176]. In the context of the direct collocation optimization, the numerical integration in (3.54) is computed with the discrete state and control variables using quadrature rules [176].

Chapter 4

ROBOTIC HARDWARE DESCRIPTIONS: DURUS AND CASSIE

In this chapter, the robot models and hardware for both the DURUS humanoid and the Cassie biped are presented. While their morphology is considerably different, both robots have spring components in their legs to return energy from impact, and have low reduction cycloidal gearboxes. This combination makes both robots particularly well suited for low friction and high efficiency motions. These bipeds will be the primary experimental platforms used in this thesis, and are featured in the walking and control development outlined in Chapters 5, 6, and 8.

4.1 The DURUS Humanoid

Developed by SRI International, the DURUS humanoid was built with the sole purpose of achieving the most efficient walking possible, thereby allowing for extended usage for autonomous battery-powered operation. Reaching this ambitious target required a tight coupling between the electromechanical design of the robot and the control algorithms which would ultimately be realized on the system. This section will give an overview of the key components within the electromechanical design,

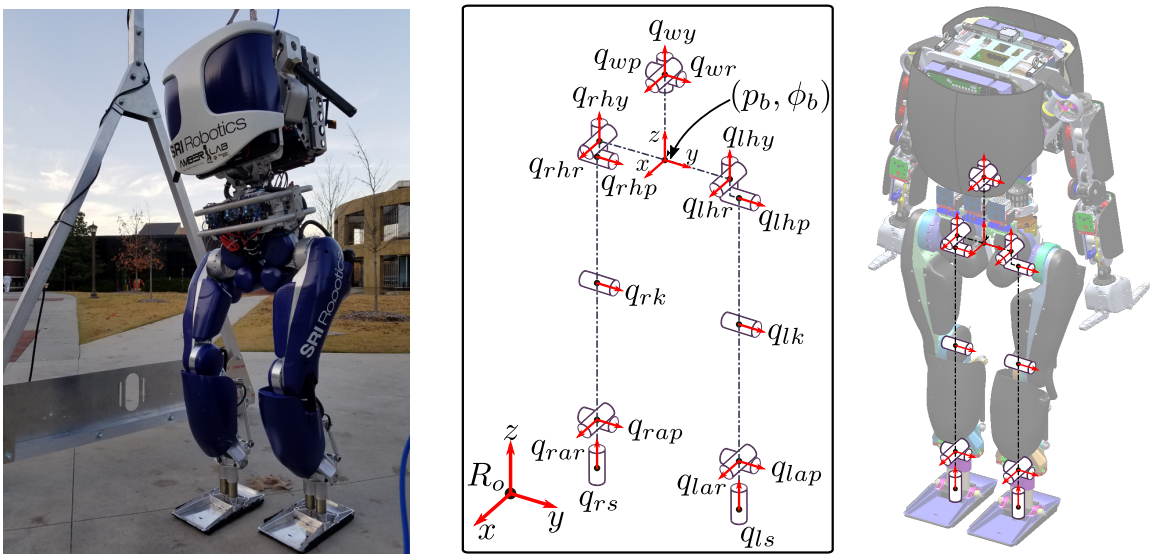

Figure 4.1: The humanoid robot DURUS. (Left) Experimental walking outside at Georgia Tech under a mobile gantry. (Right) The generalized coordinates chosen to represent the configuration of DURUS.

including the actuators, gear reductions, motor controllers, and the overall power delivery system. This design plays an integral role in the walking behaviors achieved on DURUS and—when coupled with the novel electromechanical design and nonlinear control framework—forms the basis for realizing efficient locomotion.

4.1.1 Robot Model

Generalized Coordinates

Assuming R_0 be a fixed world frame and R_b be a body reference frame rigidly attached to the pelvis of the robot with the origin located at the center of the hip, the Cartesian position $p_b = (p_b^x, p_b^y, p_b^z) \in \mathbb{R}^3$ and the orientation $\varphi_b = (\varphi_b^x, \varphi_b^y, \varphi_b^z) \in SO(3)$ of R_b with respect to R_0, respectively, composes the floating base coordinates of the robot. The configuration of the robot body, as illustrated in Fig. 4.1, consists of three kinematic chains: waist joints, $q_w = [q_{wy}, q_{wr}, q_{wp}]^T$, left leg joints, $q_{lleg} = [q_{lhy}, q_{lhr}, q_{lhp}, q_{lk}, q_{lap}, q_{lar}, q_{ls}]^T$, and right leg joints, $q_{rleg} = [q_{rhy}, q_{rhr}, q_{rhp}, q_{rk}, q_{rap}, q_{rar}, q_{rs}]^T$, respectively, where q_{wy}, q_{wr}, and q_{wp} are the waist yaw, roll, and pitch angles, q_{lhy}, q_{lhr}, q_{lhp}, q_{lk}, q_{lap}, q_{lar}, and q_{ls} are the left hip yaw, left hip roll, left hip pitch, left knee pitch, left ankle pitch, left ankle roll angle, and left spring deflection, respectively, and q_{rhy}, q_{rhr}, q_{rhp}, q_{rk}, q_{rap}, q_{rar}, and q_{rs} are the right hip yaw, right hip roll, right hip pitch, right knee pitch, right ankle pitch, right ankle roll angle, and right spring deflection, respectively. Every joint on the robot is actuated with the exception of the floating base, (p_b, φ_b), and the spring elements, q_{ls} and q_{rs}. For simplicity, the generalized coordinates of the robot will be given in terms of stance and non-stance leg angles instead of left and right leg angles. That is, the generalized configuration space $\mathcal{Q}$ is given in the following generalized coordinates:

$$q = (p_b, \varphi_b, q_r) \in \mathcal{Q} = \mathbb{R}^3 \times SO(3) \times \mathcal{Q}_r,$$

where q_r is the coordinates of body configuration space $\mathcal{Q}_r$ determined by $q_r = (q_w, q_{lleg}, q_{rleg}) \in \mathcal{Q}_r$ if the right leg is the stance leg, and $q_r = (q_w, q_{rleg}, q_{lleg}) \in \mathcal{Q}_r$ if the left leg is the stance leg, respectively. When the stance leg switches due to an impact, the coordinates need to be relabeled accordingly. This is done by switching q_{lleg} and q_{rleg} and "flipping" the sign convention for all roll and yaw angles.

Relabeling Matrix

The walking on DURUS in simulation and in optimization is modeled as a symmetric walking gait. This allows for control development to be simplified to only one step rather than two to represent a full stable walking cycle. Following the definition

that was given for the impact model used in our hybrid system representation of locomotion, (2.29), we can use the relabeling matrix $\mathcal{R} : \mathcal{Q} \to \mathcal{Q}$ to swap the left and right leg angles and flip the sign on the corresponding yaw and roll angles as well. The relabelling matrix is a linear map:

$$\mathcal{R} := \begin{bmatrix} \mathcal{R}_b & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{R}_w & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{R}_l \\ \mathbf{0} & \mathbf{0} & \mathcal{R}_l & \mathbf{0} \end{bmatrix}, \tag{4.1}$$

where $\mathcal{R}_b, \mathcal{R}_w$, and $\mathcal{R}_l$ are the individual relabeling matrix for the base, waist, and leg coordinates, given by:

$$\mathcal{R}_b = \text{diag}(1, -1, 1, -1, 1, -1), \tag{4.2}$$

$$\mathcal{R}_w = \text{diag}(-1, -1, 1), \tag{4.3}$$

$$\mathcal{R}_l = \text{diag}(-1, -1, 1, 1, 1 - 1, 1), \tag{4.4}$$

where $\text{diag}(\cdot)$ represents a diagonal matrix.

Robot Dynamics

The continuous dynamics of the robot follows the general derivations presented in Chap. 2, with several modification or additions. We incorporate the reflected motor inertia of the actuators as a decoupled addition to the mass-inertia matrix [178]. In the order of the coordinates defined previously in the robot configuration, the actuated joint reflected inertia are $\mathcal{I}_{m,w} = [1.44662, 1.44662, 1.44662]$ for the waist and $\mathcal{I}_{m,l} = [1.435046, 1.435046, 1.435046, 1.44662, 1.44662]$ for each leg. In addition, the robot has two passive compliant springs at the ankles corresponding to the joints q_{ls} and q_{rs}. Thus, the dynamics evolve according to:

$$\underbrace{D(q) + D_m}_{D^R(q)} + H(q, \dot{q}) = Bu + \kappa(q, \dot{q}) + J_v^T \lambda_v,$$

where $D_m \in \mathbb{R}^{23\times 23}$ is a diagonal matrix with each element corresponding to the reflected rotor inertia through the gearbox on the motor joints which is added to the nominal mass-inertia matrix $D(q)$, and $\kappa(q, \dot{q}) : T\mathcal{Q} \to \mathbb{R}^{23}$ is a vector of spring forces. The springs are modeled with a linear stiffness, $k_s = 38,790$ N/m, and damping, $b_s = 300$ N/m/s, forming a vector of generalized forces at the ankle spring joints:

$$\kappa(q, \dot{q}) = [\mathbf{0}_{1\times 15},\ k_s q_{ls} + b_s \dot{q}_{ls},\ \mathbf{0}_{1\times 6},\ k_s q_{rs} + b_s \dot{q}_{rs}]^T, \tag{4.5}$$

where $\mathbf{0}_{\square\times\square}$ is matrix of all zeros.

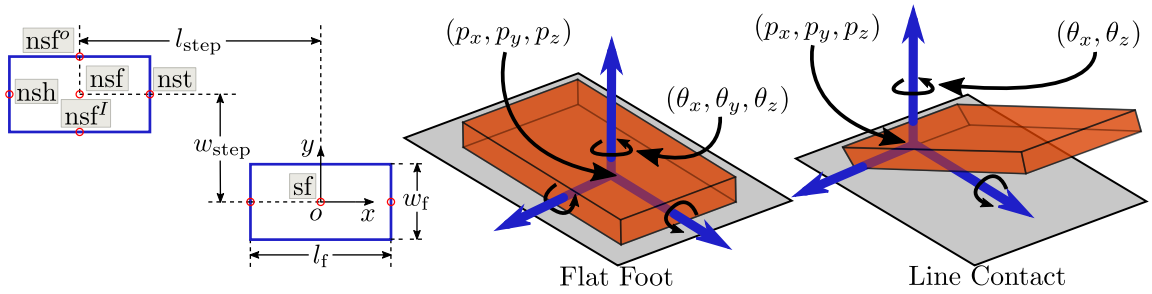

Figure 4.2: Illustration of the geometry used for defining the foot on DURUS. Points of contact for the toe and heel are applied depending on the domain of the robot, and can be applied in any combination. For example, if both *nst* and *nsh* are in contact, it would render the stance foot flat on the ground. On the right, a heel roll and flat foot contact condition are depicted.

Contact Geometry

Because the feet of DURUS are rectangular, we permit two kinds of contact during walking: a *flat-foot constraint* or a *line constraint*. This is illustrated for DURUS' feet in Fig. 4.2, with the set of available contacts being $\mathcal{C} = \{nsh, nst, sth, stt\}$ (swing heel, swing toe, stance heel, stance toe). What determines the holonomic constraints applied to a given domain is the set of contact points currently in fixed contact with the ground. The foot geometry forms a rectangle, where the length of the contacting geometry is $l_{\text{f}} = \text{st} - \text{sh}$ and the width is $w_{\text{f}} = \text{sf}^I - \text{sf}^o$, with the origin of the foot at the center.

Assuming three non-collinear points of contact, the foot can be modeled as a flat plane. When we have established a flat-footed contact, the position and orientation of the plane with respect to the ground will create a 6 degree-of-freedom (DOF) closure constraint in (2.10) ($m_h = 6$). This means that the selection matrix for the constraint is $\mathbf{S}_{c,\text{flat}} = \mathbf{I}_{6\times 6}$.

In the other two cases, we allow for the feet on DURUS to roll over on the toe or heel. When this is occurring, there is a 4 DOF closure constraint imposed with:

$$\mathbf{S}_{c,\text{line}} = [\mathbf{I}_{5\times 4},\ \mathbf{0}_{5\times 1},\ \mathbf{I}_{5\times 1}], \tag{4.6}$$

which demonstrates that there is a 3 DOF constraint on the position of the heel with a corresponding orientation constraint on the roll and yaw, while the pitch is completely unconstrained (meaning that $\mathbf{I}_{5\times 4}$ is an upper diagonal identity with a row of zeroes in the final row and $\mathbf{I}_{5\times 1}$ is a column of zeros with a 1 on the final entry).

Figure 4.3: Final ankle design which was used on DURUS. In particular, two passive springs in the ankle were utilized below a two degree of actuation ankle. The result is compliant interaction between the robot and the ground with rigidly controlled linkages above.

4.1.2 Electromechanical Specifications

We begin by giving an overview of the key elements of the novel electromechanical design that were utilized on DURUS. These key aspects include actuators, joint design including the selection and use of cycloid gearboxes, motor controllers, and power management systems. We give an overview of the main elements that were ultimately combined to yield the final design of the robot.

Compliant Ankles

The morphology of DURUS, and specifically the role of passive-compliant elements, directly impacts the ability of the control methodology to achieve dynamic and efficient locomotion and, conversely, the control scheme utilized—and the fact that it can exploit underactuated degrees of freedom—can inform the mechanical design; that is, there a direct coupling between control algorithms and mechanical design.

The end result of the design procedure was passive springs in the ankles of DURUS, shown in Fig. 4.3, which are used to absorb ground impacts and to store and release energy during the walking cycle, reducing the energy input required. These springs have a linear action and, by adjusting the ankle angle, can be positioned to act in an optimal direction upon touchdown. This makes the spring behavior predictable and easily matched with the control model. The small amount of friction in the linear bushings causes some lost energy, but the resulting damping greatly improves controllability of the system. To reduce rebound and energy loss during ground

impacts, the unsprung mass is kept as low as possible.

In addition to the specific leg morphology of DURUS, the concrete design approach for the leg was to keep the inertia about the hip axis as low as possible to reduce the energy consumption during swing phase. To this end, the motors and other heavy components in the leg were positioned as proximally as possible and the mass of the foot was kept to a minimum. The number of components inside the moving leg is kept to a minimum, with the hip yaw and hip roll actuators positioned up inside the pelvis. There is always a trade-off between weight and stiffness, but the control techniques used on DURUS were able to handle some compliance in the structure of the robot (due to the fact that, as will be discussed later, they utilize the full-order dynamics of the robot). This means that we were able to design a relatively lightweight structure capable of fast motion with low energy consumption.

Actuators and Joints

The sagittal joints of the legs of DURUS, where high-power is required, use a custom transmission design developed by SRI. A low-cost BLDC outrunner motor is connected via a chain drive to a cycloidal transmission which drives the link output, shown in Fig. 4.4. Typical existing cycloids use bushings, are up to 80% efficient, and are non-backdriveable. The mechanisms of such cycloid drives have been analyzed in [179], [180]. The SRI cycloid works on a similar principle, but uses needle roller bearings at the interfaces of all contacting surfaces, thereby reducing friction and resulting in a transmission whose efficiency is measured to be 97%. The gear ratio for the cycloid was selected to be $15:1$, while the chain drive between the motor and the gearbox increases the overall gear ratio to $56:1$. The inclusion of the chain drive is an efficient way of increasing the gear ratio, reducing the necessary size and weight of the cycloid. The motor is a Rotomax 50cc outrunner motor with peak power 5.3kW, chosen for its low cost and high torque density. Joint torque sensing is achieved using a strain-gauge based torque sensor from ME Systeme. A 21-bit absolute encoder is included on the link output and a relative optical encoder measures the speed of the rotor. Mechanical stops are also included within the joint housing. The total weight of the joint is 2.7 kg, the peak torque is 250 Nm, and the maximum joint acceleration exceeds 130 rad/s^2.

For the joints in DURUS with lower power requirements, such as the waist and ankles, a smaller, lighter-weight actuator was designed. In particular, it is smaller than the cycloid joint, weighs only 1.1 kg, and has a peak torque of 150 Nm. This design uses

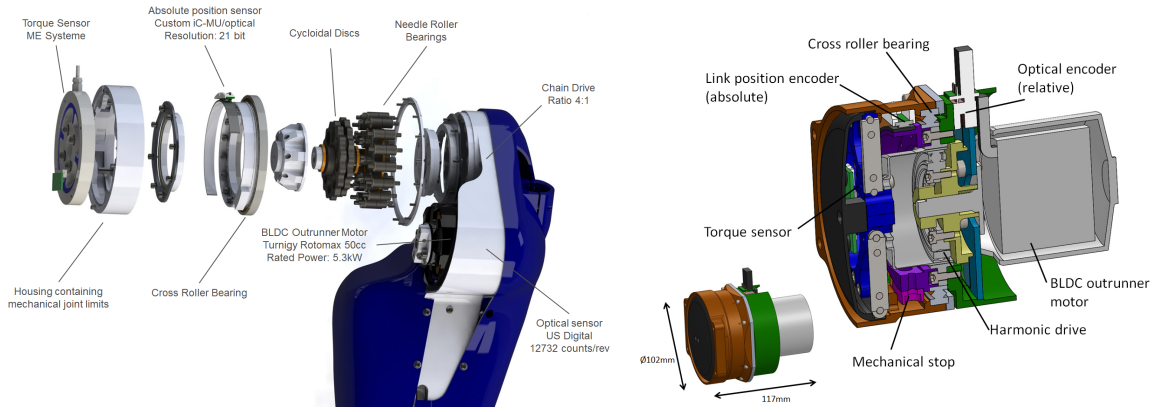

Figure 4.4: DURUS had two actuated joint types: a *high power* (left), and *low power* (right) joint. The high power design used a custom cycloidal gearbox, which is shown in an exploded view. For joints which did not require large torques a compact harmonic drive was coupled with a smaller BLDC motor, the coupled geometry is shown in a cutaway profile.

an inexpensive Aerodrive F3A outrunner motor with peak power 2.7 kW, connected directly to a Harmonic Drive speed reducer with gear ratio of $100:1$. This actuator includes mechanical stops, a 20-bit absolute encoder on the link output, and a custom torque sensor which uses low-cost strain gauge based sensing. The stacked motor and gearbox configuration is shown in Fig. 4.4.

To ultimately realize dynamic and efficient locomotion on DURUS, precision in control implementation is required at every level of the hardware. Therefore, an essential component in the process of realizing locomotion is a motor controller which can accurately track desired trajectories. In particular, desired trajectories will be derived at the high-level via nonlinear controllers, and ultimately realized at the embedded level via high precision position control. Therefore, accurate tracking at the joint level is a necessary component of successful implementation—the motor controllers were specifically designed to achieve this objective. High-fidelity tracking at the joint level is crucial to incorporating the formal walking methods successfully. Good tracking ensures that the physical hardware is as close as possible to the robot model and reduces the stabilizing effort necessary from the more heuristic regulator stabilization methods. These motor controllers tracked joint positions with an overall *rms* error of 0.005 rad and a peak error of 0.026 rad. Therefore, the motor controllers and actuators achieve the desired goal of under 0.01 rad tracking error, allowing for the implementation of the formal controllers that will be discussed in subsequent sections.

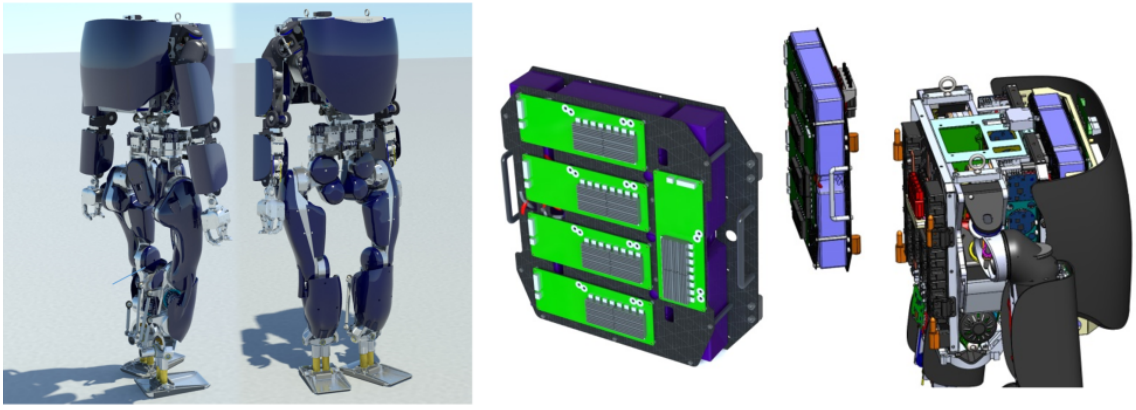

Figure 4.5: Battery design and placement on DURUS. The torso of DURUS, shown on the right, contains mounts for two battery packs along with the central computer for the robot.

On-Board Power and Processing

DURUS is powered by two on-board battery packs. Each pack weighs 10.2 Kg and has a capacity of 1.1 KWh, making a total of 2.2 KWh. The batteries are low cost Li-Po units, purchased from a hobby store, but analysis showed these units to have excellent specific energy density, very close to that of high-end customized units. Each battery pack contains ten of these units encased in an insulating fiberglass enclosure. The power safety circuit of DURUS enables wireless and manual control of power distributed to the system. The concept is implemented by combining solid-state relays and fuses to provide a dual level of protection. A multi-drive wireless emergency stop (ESTOP) transceiver drives a bank of relays enabling wireless cutoff of each appendage of the robot. The wireless receiver also responds to manual ESTOP button control. The batteries design used on DURUS is pictured in Fig. 4.5, along with an illustration of the battery placement. In addition to the battery shown, which is mounted in the back of the torso, an additional battery can be mounted beneath the chest cover. DURUS also had onboard processing (also contained in the torso under the back torso battery mount), in the form of a Linux computer running a Simulink Real-Time binary to communicate with the motor controllers over an EtherCAT network card.

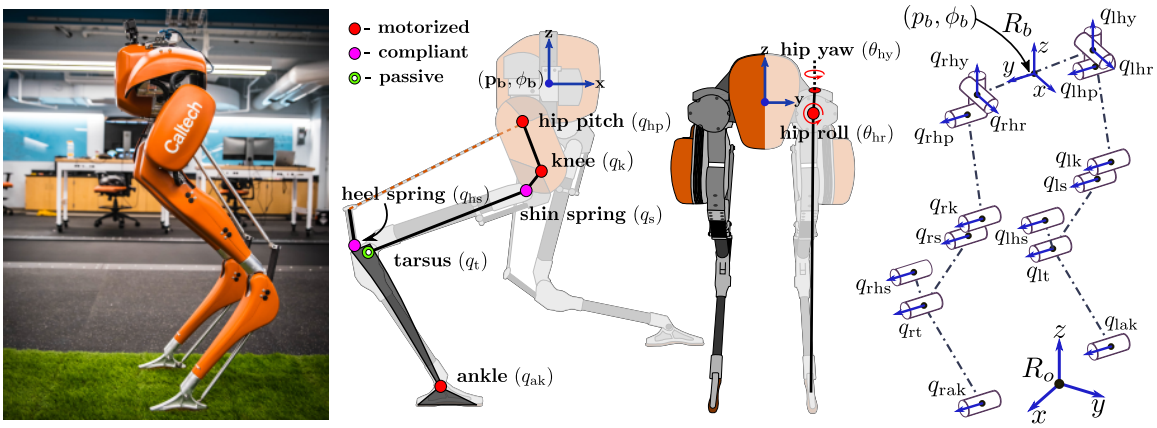

Figure 4.6: The Cassie bipedal robot. (Left) Experimental walking outside at Caltech. (Right) The generalized coordinates chosen to represent the configuration of Cassie.

4.2 The Cassie Biped

The Caltech Cassie biped was purchased from Agility Robotics[1]. The design of the robot encompasses the physical attributes of the spring loaded inverted pendulum (SLIP) model dynamics [84]. The primary characteristic is a pair of light-weight legs with a heavy torso so that the system is approximated by a point-mass with virtual springy legs. On Cassie, a *compliant multi-link mechanism* is used to transfer power from higher to lower limbs without allocating the actuators' weight onto the lower limbs, and effectively acts as a pair of springy legs.

4.2.1 Robot Model

Generalized Coordinates

Assuming R_0 to be a fixed world frame and R_b to be a body reference frame rigidly attached to the pelvis of the robot with the origin located at the center of the hip, the Cartesian position $p_b = (p_b^x, p_b^y, p_b^z) \in \mathbb{R}^3$ and the orientation $\varphi_b = (\varphi_b^x, \varphi_b^y, \varphi_b^z) \in SO(3)$ of R_b with respect to R_0, respectively, composes the floating base coordinates of the robot. The configuration of the robot body, as illustrated in Fig. 4.6, consists of two kinematic chains: left leg joints, $q_{lleg} = [q_{lhr}, q_{lhy}, q_{lhp}, q_{lk}, q_{ls}, q_{lt}, q_{lhs}, q_{lak}]^T$, and right leg joints, $q_{rleg} = [q_{rhr}, q_{rhy}, q_{rhp}, q_{rk}, q_{rs}, q_{rt}, q_{rhs}, q_{rak}]^T$, respectively, where q_{lhy}, q_{lhr}, q_{lhp}, q_{lk}, q_{ls}, q_{lt}, q_{lhs}, and q_{lak} are the left hip roll, left hip yaw, left hip pitch, left knee pitch, left shin spring, left tarsus pitch, left heel spring, and left ankle pitch, respectively, and q_{rhr}, q_{rhy}, q_{rhp}, q_{rk}, q_{rs}, q_{rt}, q_{rhs}, and q_{rak} are the right hip roll, right hip yaw, right hip pitch, right knee pitch, right shin spring, right tarsus pitch, right

[1] `https://www.agilityrobotics.com/`

heel spring, and right ankle pitch, respectively. The actuated joints are symmetric for both legs, shown in Fig. 4.6 as a red joint, and correspond to $[q_{hr}, q_{hy}, q_{hp}, q_k$, and q_{ak} totaling 10 actuated joints on the overall model. As will be shown in the robot dynamics, there are also four passive compliant springs at the q_s and q_{hs} joints on each leg. For simplicity, the generalized coordinates of the robot will be given in terms of stance and non-stance leg angles instead of left and right leg angles. That is, the generalized configuration space $\mathcal{Q}$ is given in the following generalized coordinates:

$$q = (p_b, \varphi_b, q_r) \in \mathcal{Q} = \mathbb{R}^3 \times SO(3) \times \mathcal{Q}_r,$$

where q_r is the coordinates of body configuration space $\mathcal{Q}_r$ determined by $q_r = (q_{lleg}, q_{rleg}) \in \mathcal{Q}_r$. When the stance leg switches due to an impact (this is only done for the walking presented in Sec. 6.1, and not in Sec. 6.2 or Chap. 8), the coordinates need to be relabeled accordingly. This is done by switching q_{lleg} and q_{rleg} and "flipping" the sign convention for all roll and yaw angles.

Relabeling Matrix

In Chap. 6, we will consider two different styles of walking, which will coincide with two different reset maps. The first is a spring reset map for an asymmetric, or period-2, walking gait. This means that no leg swapping is necessary as left and right are explicitly modelled in their own individual stance phases. However, in this model there is an assumption of a rigid swing leg and instantaneous double-support domain. Because of this, the reset map is not simply an identity matrix, but rather a nonlinear mapping which zeroes the spring deflections in the leg. In the second case, walking will by considered as symmetric, meaning we designate the gait cycle under the assumption that right and left stance are the same. For this locomotion model, we do not zero the spring deflections in the swing leg, but do need to use a linear mapping to relabel the left and right leg coordinates from step to step.

Asymmetric Spring Reset Map: Because the legs are compliant, they may not necessarily leave the ground with the springs at exactly their neutral angle. For the majority of the walking that we consider, the swing leg is assumed to be sufficiently rigid to model the springs as a holonomic constraint while in the air. Thus, we must define a spring reset function, which can be applied as part of our relabeling matrix to zero the springs. While simply setting the spring values to zero is sufficient for those joints, we must solve for a nontrivial value on the tarsus. While we can solve explicitly for the inverse kinematics given the pre-impact configuration, the explicit

solution is too long to write concisely. Specifically, the inverse kinematics problem solves the multi-bar (zero spring deflection) closure constraint:

$$\begin{aligned}\bar{\gamma}_{\text{tar}}(q) :=& 0.028794 + 0.118906\cos(q_{\text{k}}) - 0.112216\cos(q_{\text{t}}) \\ &- 0.0280613\cos(q_{\text{k}} + q_{\text{t}}) - 0.0161784\sin(q_{\text{k}}) - 0.0425142\sin(q_{\text{t}}) \\ &- 0.00647928\sin(q_{\text{k}} + q_{\text{t}}) = 0. \end{aligned} \tag{4.7}$$

The inverse kinematics solution for the neutral tarsus angle is then denoted $\bar{q}_{\text{t}}(q_{\text{k}}) := f_{\bar{\gamma}_{\text{tar}}}(q_{\text{k}})$. Using (4.7), we can solve for the post-impact tarsus joint, q_t^+, given the pre-impact values for q_k^- and assuming $q_s^+ \to 0$, $q_{hs}^+ \to 0$. The inverse kinematics solution will be denoted as $q_t^+ = \Delta_{\text{tar}}^{\bar{\gamma}}(q^-)$.

The walking on Cassie in simulation and in optimization is most generally represented as a period two walking cycle, meaning that the cycle repeats after the left and right legs have both been through a stance phase. This is because on Cassie, we explicitly consider lateral walking, which cannot be represented as a symmetric walking gait. Thus, for our asymmetric walking model, the reset map $\mathcal{R} : \mathcal{Q} \to \mathcal{Q}$ becomes:

$$\mathcal{R}_{\text{asym}}(q^-) := \left(\mathcal{R}_{b,\text{asym}}(q^-), \mathcal{R}_{l,\text{asym}}(q^-), \mathcal{R}_{l,\text{asym}}(q^-)\right)^T, \tag{4.8}$$

where $\mathcal{R}_{b,\text{asym}}(q^-) := \mathbf{I}_{6\times 6} q_{\text{b}}$ and $\mathcal{R}_{l,\text{asym}}(q^-)$ is a nonlinear function which is mostly the identity mapping combined with the spring zeroing inverse kinematics applied at the tarsus:

$$\mathcal{R}_{l,\text{asym}}(q^-) := \left(q_{\text{hr}}, q_{\text{hy}}, q_{\text{hp}}, q_{\text{kp}}, 0, \Delta_{\text{tar}}^{\bar{\gamma}}(q^-), 0, q_{\text{tp}}\right)^T, \tag{4.9}$$

and it can be seen that the zero entries correspond to the shin and heel spring indices. The velocity spring reset map is then a linear function which coincides with the Jacobian of the position mapping:

$$\dot{\mathcal{R}}_{\text{asym}} := \frac{\partial \mathcal{R}_{\text{asym}}(q^-)}{\partial q}, \tag{4.10}$$

which results in an overall impact and relabelling mapping that slightly differs from the one previously introduced in (2.29):

$$\Delta(q^-, \dot{q}^-)_{\text{asym}} := \begin{bmatrix} q^+ \\ \dot{q}^+ \end{bmatrix} = \begin{bmatrix} \mathcal{R}_{\text{asym}}(q^-) \\ \dot{\mathcal{R}}_{\text{asym}}(q^-)\Delta^{\dot{q}}(q^-)\dot{q}^- \end{bmatrix}. \tag{4.11}$$

Symmetric Relabelling Matrix: While the relabelling presented in (4.8) is sufficient for asymmetric walking gaits, in Sec. 6.1 we do explicitly consider symmetric

walking due to the complexity of adding an additional double-support domain. Thus, a symmetric relabeling matrix can also be used to swap the left and right leg angles and flip the sign on the corresponding yaw and roll angles as well. When we use the symmetric walking assumption, we also allow the swing leg to remain compliant, meaning we do not need to zero spring elements in the reset as was done in (4.8). The symmetric relabelling matrix is a linear map:

$$\mathcal{R}_{\text{symm}} := \begin{bmatrix} \mathcal{R}_{b,\text{symm}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{R}_{l,\text{symm}} \\ \mathbf{0} & \mathcal{R}_{l,\text{symm}} & \mathbf{0} \end{bmatrix}, \tag{4.12}$$

where $\mathcal{R}_b$,and $\mathcal{R}_l$ are the individual relabeling matrix for the base and leg coordinates:

$$\mathcal{R}_{b,\text{symm}} = \text{diag}\left(1, -1, 1, -1, 1, -1\right), \tag{4.13}$$

$$\mathcal{R}_{l,\text{symm}} = \text{diag}\left(-1, -1, 1, 1, 1, 1, 1, 1\right). \tag{4.14}$$

Robot Dynamics

The continuous dynamics of the robot follows the general derivations presented in Chap. 2, with several modifications or additions. We incorporate the reflected motor inertia of the actuators as a decoupled addition to the mass-inertia matrix [178]. In the order of the coordinates defined previously in the robot configuration, the actuated joint reflected inertia are $\mathcal{I}_{m,l} = [1.435046, 1.435046, 1.435046, 1.44662, 1.44662]$ for each leg. In addition, the robot has two springs at the ankles corresponding to the joints q_{ls} and q_{rs}. Thus, the dynamics evolve according to:

$$\underbrace{D(q) + D_m}_{D^R(q)} + H(q, \dot{q}) = Bu + \kappa(q, \dot{q}) + J_v^T \lambda_v,$$

where $D_m \in \mathbb{R}^{22\times 22}$ is a diagonal matrix with each element corresponding to the reflected rotor inertia through the gearbox on the motor joints which is added to the nominal mass-inertia matrix $D(q)$, and $\kappa(q, \dot{q}) : T\mathcal{Q} \to \mathbb{R}^{22}$ is a vector of spring forces. The springs are modeled with a linear stiffness, $k_s = 2,300$ and $k_{hs} = 2,000$ N/m, and damping, $b_s = 4.4$ and $b_{hs} = 4$ N/m/s, forming a vector of torsional generalized forces at the spring pivots:

$$\kappa(q, \dot{q}) = [\mathbf{0}_{1\times 10},\ k_s q_{\text{ls}} + b_s \dot{q}_{\text{ls}},\ 0,\ k_{\text{hs}} q_{\text{lhs}} + b_{\text{hs}} \dot{q}_{\text{lhs}}, \tag{4.15}$$

$$\mathbf{0}_{1\times 5},\ k_s q_{\text{rs}} + b_s \dot{q}_{\text{rs}},\ 0,\ k_{\text{hs}} q_{\text{rhs}} + b_{\text{hs}} \dot{q}_{\text{rhs}},\ 0]^T, \tag{4.16}$$

where $\mathbf{0}_{\square\times\square}$ is matrix of all zeros.

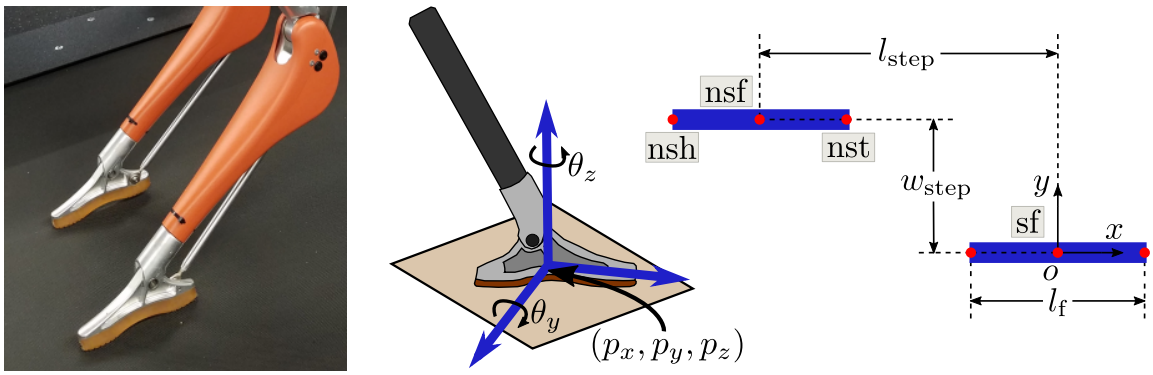

Figure 4.7: Illustration of the dimensions used for defining the foot and contact geometry on Cassie. We only consider each foot as either in contact or not, with both *nst* and *nsh* touching the ground. Enforcement of this contact constraint gives a line contact along the length of the foot.

Contact Geometry

The feet on the Cassie robot are effectively a *line contact*, illustrated in Fig. 4.7, with the set of available contact points being $\mathcal{C} = \{nsh, nst, sth, stt\}$ (swing heel, swing toe, stance heel, stance toe). The foot geometry forms a line, where the length of the contacting geometry is $l_{\text{f}} = \text{st} - \text{sh}$ with the origin of the foot at the center.

When we have established contact, the position and orientation of the plane with respect to the ground will create a 5 degree-of-freedom (DOF) closure constraint in (2.10) ($m_h = 5$). This means that the selection matrix for the constraint is:

$$\mathbf{S}_{c,\text{line}} = [\mathbf{I}_{5\times5}, \mathbf{0}_{5\times1}], \tag{4.17}$$

which demonstrates that there is a 3 DOF constraint on the position of the foot with an orientation constraint on the roll and pitch, while the yaw is unconstrained.

4.2.2 Electromechanical Specifications

This section will provide an overview of the key elements of the electromechanical specifications on the Cassie bipedal robot. The primary aspects of the robot which will be relevant to later experimental work includes a description of the springs in the leg, gearbox and motor specifications, the computing available for control development, and modeling which was done to allow for a mathematical representation of the multi-bar leg mechanism. All hardware was designed and built by Agility Robotics[2]., while the multi-bar leg modeling and software for networking the control computers were created specific to this work.

[2]The Agility Robotics GitHub: https://github.com/agilityrobotics/cassie-doc

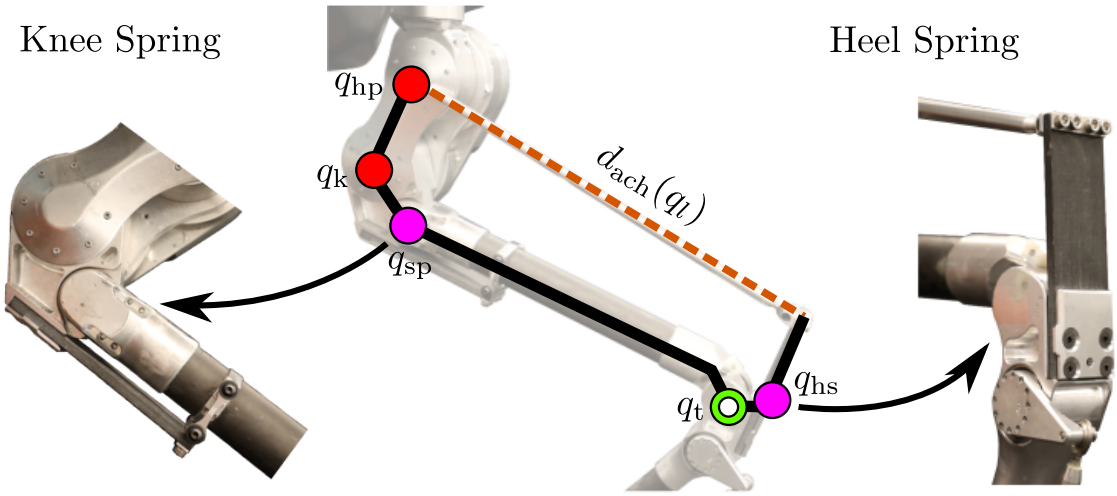

Figure 4.8: A blown-up view of the multi-bar leg mechanism on Cassie. Also shown are the individual fiberglass leaf springs which provide torsional compliance at the shin pitch and heel spring coordinates.

Fiberglass Springs

The Cassie robot has a compliant leg morphology, pictured in Fig. 4.8, effectively forming a 6-bar mechanism with 2 fiberglass leaf springs. In analysis that was introduced by the robot designers [84], [181], [182], it was shown that while this particular morphology does not reduce peak energy consumption for steady-state walking, it does provide exceptionally desirable properties in disturbance rejection, power requirements, and efficiency. The leg design of Cassie is the logical result of these findings, with *opposing springs at the heel and shin* and actuators at the knee and hip pitch which can provide torque transmission to propel walking.

As opposed to the prismatic spring action of DURUS, which was always normal to the foot configuration, the spring action on Cassie acts along the sagittal plane of the leg mechanism in both the radial and tangential directions. This means, if we exponentially zero a virtual constraint in the leg length and leg angle directions, there will be an associated passive compliance along both the leg angle and length, rather than just the axial length.

A significant effort was made in this thesis to fully leverage this compliant leg structure by modeling the passive compliance at q_{sp} and q_{hs} as underactuated coordinates which enter the zero dynamics. These springs will also be used explicitly in feedback controllers which are used on hardware in Chap. 8, allowing the control methodology to use the spring torques for tracking rather than assuming that they allow

instantaneous torque transfer in the leg as would be for a rigid mechanism.

Actuators and Gearboxes

Most of the information on the actual actuators and gearboxes is proprietary information maintained by Agility Robotics. However, we can describe the general components. Each leg is fitted with three separate sets of grouped actuators and gearboxes; the hip roll and hip yaw joints have an intermediate reduction of $25:1$ through a highly efficient custom cycloidal gearbox. This allows for minimal friction losses and highly transparent torque transmission. As the most dynamic and important joints for force control, the hip pitch and knee joints have a lower reduction of $16:1$ with a cycloidal gearbox. Finally, the foot pitch actuators, which are used minimally to perform tasks such as pointing the feet in the air and during standing, have a harmonic gearbox with a reduction of $50:1$. The harmonic gearbox has a significant amount of friction in comparison to the cycloids as would be expected. The end result is a low-reduction and easily backdriveable set of joints that makes model-based control significantly easier to accomplish. This fact will be very important in the implementations provided in Sec. 6.2 and Chap. 8.

Loop Constraints

As previously mentioned, the compliant constrained mechanism which makes up the leg structure on Cassie was meticulously designed to provide symmetric axial compliance which accurately mimics the SLIP model [84]. Because this structure is essentially a constrained 6-bar mechanism, it affects how we obtain manipulator Jacobians for the system. We hereby derive the constrained manipulation Jacobian for the leg. Let $r \in \mathbb{R}^3$ be the position of an end-effector with respect to the robot's center of mass. This can be obtained by $r = f_{\text{FK}}(q)$, where $f_{\text{FK}}(q) : \mathcal{Q} \to \mathbb{R}^3$ is the forward kinematics and $q \in \mathbb{R}^n$ is the vector of generalized coordinates in the unconstrained configuration space. In the case of the Cassie biped, the end effector is chosen as the ankle when we are solving for the estimated contact force, since the foot is a passive joint and only serves to add an additional coordinate to the expressions.

The general methodology to derive the constrained forward kinematics of a robotic manipulator is to open the mechanism loop, propagate the kinematics along the branches, and add kinematic constraints to close the loop. We partition the configuration coordinates into active ($\theta_a \in \mathbb{R}^{n_a}$) and passive ($\theta_p \in \mathbb{R}^{n_p}$) joints, with $n = n_a + n_p$. Note that we define the joints connected by a motor or spring as active.

This means for each leg, $n_a = 7, n_p = 1$. Next, we apply a kinematic constraint to the leg. Specifically, the heel spring is attached to the rear of the tarsus linkage, with its end constrained via a pushrod affixed to the hip pitch linkage. The pushrod attachment is a holonomic constraint applied between the hip and heel spring connectors:

$$\eta_{\text{4bar}}(q_l) := d_{\text{ach}}(q_l) - 0.5012 = 0, \tag{4.18}$$

where the attachment distance $d_{\text{ach}}(q_l) \in \mathbb{R}$ is obtained via the forward kinematics from one connector to the other. For each leg, this constraint is only a function of the knee, shin, tarsus, and heel spring joints. The forward kinematics representing the achilles rod constraint is given by the closure constraint:

$$\begin{aligned}
\eta_{\text{4bar}}(q_l) = &- 0.0245852 - 0.00127968\cos(q_{\text{hs}}) + 0.0145632\cos(q_{\text{k}}) + 0.0546589\cos(q_{\text{s}}) \\
&+ 0.104342\cos(q_{\text{k}} + q_{\text{s}}) - 0.0001308\cos(q_{\text{hs}} - q_{\text{t}}) - 0.0543322\sin(q_{\text{hs}} + q_{\text{t}}) \\
&- 0.0000349\cos(q_{\text{hs}} - q_{\text{k}} - q_{\text{s}} - q_{\text{t}}) - 0.088198\cos(q_{\text{hs}} + q_{\text{t}}) \\
&- 0.0071611\cos(q_{\text{hs}} + q_{\text{s}} + q_{\text{t}}) - 0.003046\cos(q_{\text{k}} + q_{\text{s}} + q_{\text{t}}) \\
&- 0.0249808\cos(q_{\text{hs}} + q_{\text{k}} + q_{\text{s}} + q_{\text{t}}) - 0.013847\sin(q_{\text{hs}} + q_{\text{k}} + q_{\text{s}} + q_{\text{t}}) \\
&- 0.0113784\sin(q_{\text{k}}) + 0.0387967\sin(q_{\text{s}}) - 0.0048\sin(q_{\text{k}} + q_{\text{s}}) \\
&- 0.0000897\sin(q_{\text{hs}} - q_{\text{t}}) - 0.00002636\sin(q_{\text{hs}} - q_{\text{k}} - q_{\text{s}} - q_{\text{t}}) \\
&+ 0.0260913\sin(q_{\text{t}}) - 0.0000281\cos(q_{\text{hs}} - q_{\text{s}} - q_{\text{t}}) \\
&+ 0.00250917\sin(q_{\text{s}} + q_{\text{t}}) - 0.0168716\sin(q_{\text{hs}} + q_{\text{s}} + q_{\text{t}}) \\
&+ 0.00734167\sin(q_{\text{k}} + q_{\text{s}} + q_{\text{t}}) + 0.00844\sin(q_{\text{hs}}) = 0.
\end{aligned} \tag{4.19}$$

Further, we can write the end effector and constraint velocities as:

$$\begin{aligned}
\dot{r} &= \frac{\partial r(q_l)}{\partial q_l} = J_{ee}(q_l)\dot{q}_l \\
\dot{\Gamma}_f &= \frac{\partial \Gamma_f(q_l)}{\partial q_l} = J_c(q_l)\dot{q}_l.
\end{aligned}$$

We then partition the Jacobians J_c and J_{ee} into active and passive components to obtain the system of equations:

$$\begin{cases}
0 &= J_{c,a}\dot{\theta}_a + J_{c,p}\dot{\theta}_p, \\
\dot{r} &= J_{ee,a}\dot{\theta}_a + J_{ee,p}\dot{\theta}_p,
\end{cases} \tag{4.20}$$

where $J_{c,a} \in \mathbb{R}^{n_p \times n_a}$, $J_{c,p} \in \mathbb{R}^{n_p \times n_p}$, $J_{ee,a} \in \mathbb{R}^{3 \times n_a}$, and $J_{ee,p} \in \mathbb{R}^{3 \times n_p}$. Because we have one passive joint per constraint (see the tarsus joint in Fig. 4.6), $J_{c,p}$ is invertible.

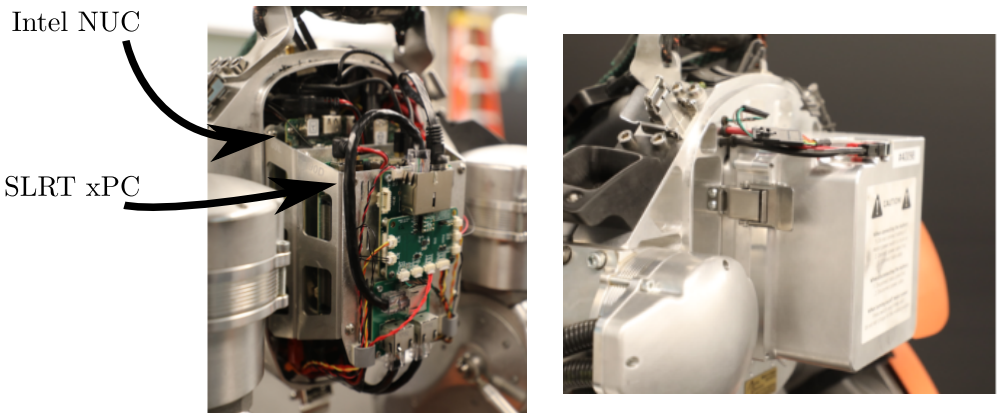

Figure 4.9: The onboard computers and battery pack contained within the pelvis of the Cassie robot. On the back of the pelvis are two computers, one Intel NUC and one Simulink Real-Time target PC. Clamped to the front of the pelvis, effectively forming the face of the robot, is the Li-ion battery pack.

We can then calculate the passive joint velocities from the active as $\dot{\theta}_p = -J_{c,p}^{-1} J_{c,a} \dot{\theta}_a$. The end effector velocity became:

$$\dot{r} = \underbrace{\left(J_{ee,a} - J_{ee,p} J_{c,p}^{-1} J_{c,a}\right)}_{\bar{J}} \dot{\theta}_a, \tag{4.21}$$

where $\bar{J} \in \mathbb{R}^{3 \times n_a}$ is the constrained end effector Jacobian.

On-Board Power and Processing

Cassie is able to operate completely autonomously, with compact onboard power and computing both contained in the pelvis. The battery, pictured on the right in Fig. 4.9 is an approximately 4.75 kg lithium-ion pack enclosed in an aluminum shell that clamps onto the front face of the robot. This accounts for nearly 40% of the total mass of the pelvis, which is the most massive grouped linkage in the entire model.

There are two computers available for use, one Intel NUC and one Simulink Real-Time target PC. The Simulink xPC is linked to Elmo gold twitter motor controllers over an EtherCAT bus and also networked to the NUC via an Ethernet cable and UDP interface. This allows for a hard real-time interface to the critical low-level motor controllers, while exposing a Linux-based PC for more complex control and estimation development. While some existing infrastructure is provided for the xPC from Agility Robotics to facilitate the low-level hardware communication, the Intel NUC was provided as a blank computer, and we were responsible for developing all software development and infrastructure which was implemented.

4.3 Justification of the Compliant Cassie Model

The inclusion of model detail in feedback controllers for underactuated bipedal locomotion can be used to accurately capture the underactuated dynamics of the system in a manner which also facilitates accurate tracking of planned motions. However, implementing model-based planning and control methods on physical systems is typically non-trivial due to the inherent model inaccuracy, dynamically changing contact constraints, and possibly conflicting objectives for the robot which naturally arise in locomotion. It is due to these challenges that bipedal robots which exhibit simultaneously robust, efficient, and agile motions are rare in practice [183].

One area of model-based planning and control which is particularly difficult to directly address is passive compliance in locomotion. Some of the earliest inclusions of compliant hardware on bipedal robots was with spring flamingo and spring turkey [79], ERNIE [184], and more recently MABEL [25] and ATRIAS [83]. One of the latest robots to exhibit compliant leg structures is the Cassie biped (shown in Fig. 4.6), which is the experimental platform that we consider in this work.

In several works from other groups on Cassie [127], [185], it was shown how a rigid model of the robot could be used to generate stable walking behaviors. This model is shown on the right in Fig. 4.10, where the heel spring is removed, the shin spring is fixed at zero deflection, and the tarsus angle is purely a function of the knee angle as a result of a holonomic constraint which is imposed to close the undeflected four-bar

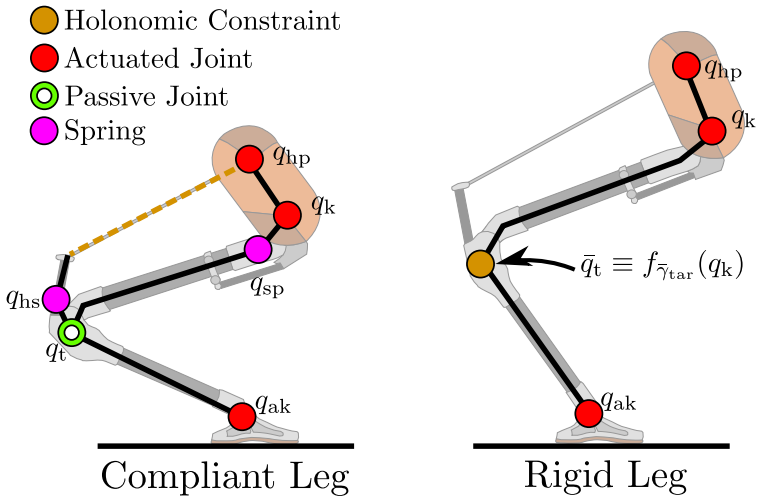

Figure 4.10: An illustration of the model differences between a compliant and rigid representation of the Cassie leg. On the left, a 22 DOF compliant leg with two springs and a passive tarsus is shown, while on the right, a 16 DOF rigid leg with no compliant elements or passive joints.

linkage (see Sec. 4.2). One of the primary reasons that a more constrained model may be considered is that it requires fewer degrees of freedom and the compliant mechanism not only increases local stiffness of the nonlinear dynamics, but also induces modelling uncertainties for the springy joints. Despite these initial difficulties that spring dynamics may bring, the kinematics of the leg mechanism (detailed extensively in Sec. 4.2) clearly have a nontrivial amount of compliance when in contact with the world. Locomotion built on simple models demand the controller to compensate the uncertainties caused by the gap between the assumed reduced model dynamics and the full body dynamics. This can sometimes be an infeasible task and it is not always clear how to coordinate the full-order system to behave as a low-dimensional pendulum while respecting physical limits. It is due to this fact that the translation of rigid plans to the actual hardware often require some form of heuristic tuning in the form of leg angle offsets or additional feedforward torques to overcome the discrepancy in the model which arises from the trajectory planning in order to implement.

In this section, we will discuss the advantages that using a compliant leg model may provide. Cassie was designed to encompass several of the primary physical attributes of the SLIP model [84] dynamics. The primary idea is to have a pair of light-weight legs with a heavy torso so that the actual system can be approximated by a point-mass with virtual springy legs (see Fig. 4.8). On Cassie, a *compliant multi-link mechanism* is used to transfer power from higher to lower limbs without allocating the actuators' major weight onto the lower limbs, and effectively acts as a pair of springy legs, with the compliant model specifically considered shown on the left in Fig. 4.10. Each leg has two passive compliant springs, with the multi-bar linkage closed via a holonomic constraint representing the connection that the achilles rod provides from the hip pivot to the end of the heel spring. The primary differences between the rigid and compliant models can be summarized briefly as:

- **Rigid model** assumes that all four leaf springs are rigid linkages, which yields kinematic approximations such as the trivial geometry relation $\Gamma_s(q) := \theta_{\mathrm{k}} - \theta_{\mathrm{t}} - 13^\circ \equiv 0$ for the multi-link structure. The constrained rigid model with no contact is 16 DOF.

- **Compliant model** instead treats the rotational joint of the leaf spring linkage as a torsional joint, with stiffness and damping effects. In addition, the distance between the hip and end of the heel spring remains a constant (as shown by the dash line in Fig. 4.10). This geometry relation can be described as a holonomic constraint: $\Gamma_f(q) \equiv 0$. The full compliant model with no contact is 22 DOF.

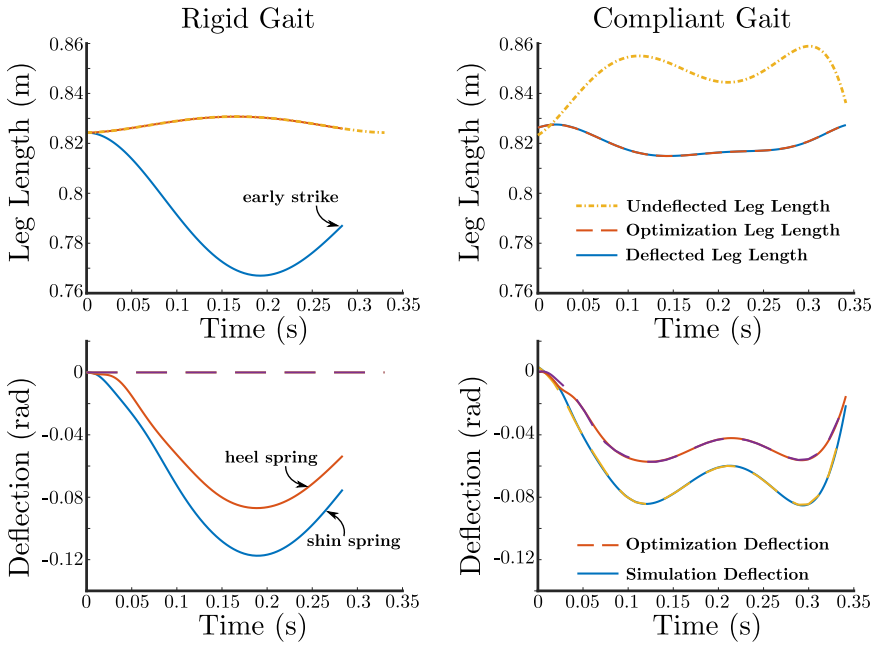

Figure 4.11: A comparison of the rigid model and compliant model for Cassie implemented in an accurate simulation environment. On the left, the rigid gait has not anticipated passive compliance, and thus drops and strikes the ground early. On the right, the compliant motion has an accurate plan for the shin and heel springs, meaning the neutral leg length output offsets to accommodate for the leg deflection.

A Comparison in Optimization and Simulation. As a first point of comparison, an HZD optimization was performed to find a stepping in place gait for both the compliant and rigid models of Cassie [128]. These were run with separate domain structures, where the compliant gait had both a double-support and single-support domain while the rigid only had a single support as the rigid leg necessitates an instantaneous transfer from one leg to the next. The optimization was run on a Ubuntu-based computer with an i7-6820HQ CPU @2.70GHz and 16GB RAM. The resulting computational performance is summarized in Table 4.1, where we see several expected results. Specifically, the rigid model was approximately 10 times faster to converge, and took approximately 3 times less iterations. However, we should note that HZD optimizations are typically performed offline, and approximately 12 minutes is still a reasonably fast convergence time (see the times given in Chap. 5 for convergence times on DURUS, which has 23 DOF). In addition, later work on Cassie [186] (see Sec. 6.2) improved the convergence time of the compliant model to an average of 264 seconds, which is actually less than the rigid model here.

The obvious advantage is illustrated in simulation through Fig. 4.11, where the rigid model and compliant model were both placed into an accurate simulation environ-

Table 4.1: Computation performance on a Ubuntu-based computer with an i7-6820HQ CPU @2.70GHz and 16GB RAM.

	simple model	full model
# of iterations	275	773
time of IPOPT (s)	20	153
time of evaluation (s)	78	755

ment of the robot. Because the rigid model has no planning for the passive degrees of freedom that the true system experiences, the leg length does not account for the vertical leg deflection causing an early strike. Reality differs from the rigid model's dynamics, inducing additional forces on the other foot and pushing the actual dynamics away from its designed behaviors. On the right, the difference between the neutral leg length (corresponding to the outputs which are derived in Chap. 6) clearly anticipates this deflection and increases to keep the torso at the desired walking height. For these same reasons, to make a rigid model based controller (or trajectory) work successfully in reality, one needs to design controllers which treat the compliant dynamics as uncertainty to overcome such dynamical gaps. While we do not wish to overfit physical reality, sufficiently large uncertainty caused by known modelling error could result in poor controller design. We argue that a large portion of this particular trade-off can be compensated through the inclusion of the compliance which exists in in the physical model.

Advantages in a Gait Library and Model-based Control Framework. In Sec. 6.2, we develop a gait library for compliant walking on Cassie at a variety of speeds. This not only provides a user with a set of outputs which have planned for the passive compliance, such as the one just shown in Fig. 4.11, but it can also provide additional information useful in control design. Rather than simply track the output Bézier polynomials purely through a model-free PD control law, we would ideally have some feedforward information on how the dynamics should evolve through time. In other work on passive compliant walking robots [41] simply adding the torque into the control law provided sufficient torque to obtain reasonable tracking. However, many bipedal implementations outside of HZD also rely on inverse dynamics [142] and thus some parameterization of the generalized accelerations, $\ddot{q}$, from step to step [187]. We will also make use of the generalized accelerations for HZD in Sec. 6.2 with inverse dynamics for tracking interpolated gaits from a motion library. Additionally, in Chap. 7 a model-based control Lyapunov function approach is introduced which uses a quadratic program to track gaits in a pointwise-optimal fashion. The actual

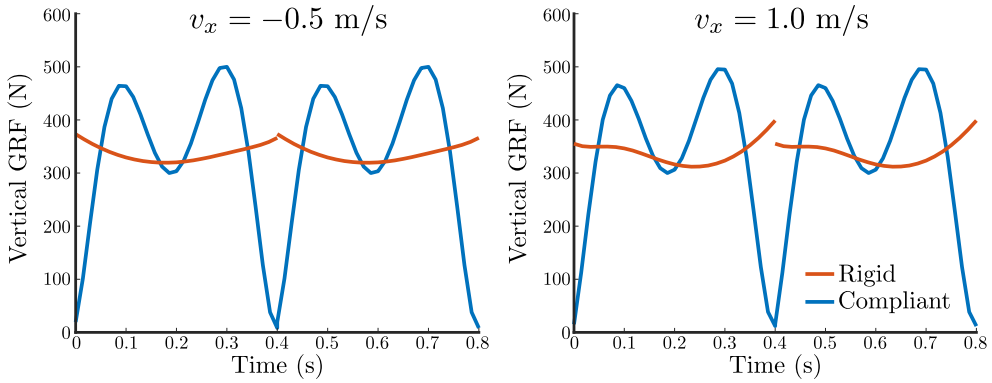

Figure 4.12: The vertical ground reaction forces compared for compliant and rigid walking on Cassie.

implementation of this controller in Chap. 8 requires highly accurate regularization terms for the torque, ground reaction forces, and accelerations on top of the compliant output Bézier polynomial coefficients.

With the aim of developing feedforward and regularization terms for feedback control development, we would next like to investigate some of the characteristics of the compliant motion library found in Sec. 6.2, and compare them to a rigid collection of gaits. The implementation of the optimization for the compliant gait library is outlined in Sec. 6.2, where here we have imposed identical constraints and an identical cost for the rigid optimization (with the exception of the added holonomic constraints on the springs).

As was previously shown, the actual model of the Cassie robot naturally behaved similarly to the SLIP model. The SLIP model is also an emergent behavior of the contact forces shown in Fig. 4.12, where we have plotted the ground reaction forces for a gait for backwards walking and forward walking. Recall in Sec. 1.2.3 that one of the defining characteristics of the SLIP model is the "double-hump" force profile of the biped while walking. This is clearly shown for the compliant model, with the ground reaction force having a fairly continuous load transfer from foot to foot. Because Cassie has such low leg inertia, the post-impact vertical force is not large as the motor torque is not instantaneously transmitted through the springs to the ground. In comparison, the rigid model has an almost constant vertical ground reaction force, meaning that this profile is not an accurate representation of the compliant leg, as it assumes that the force will instantaneously spike to approximately 375 N due to the holonomic constraints imposed on the springs.

One of the most important characteristics of a controller with regards to implementa-

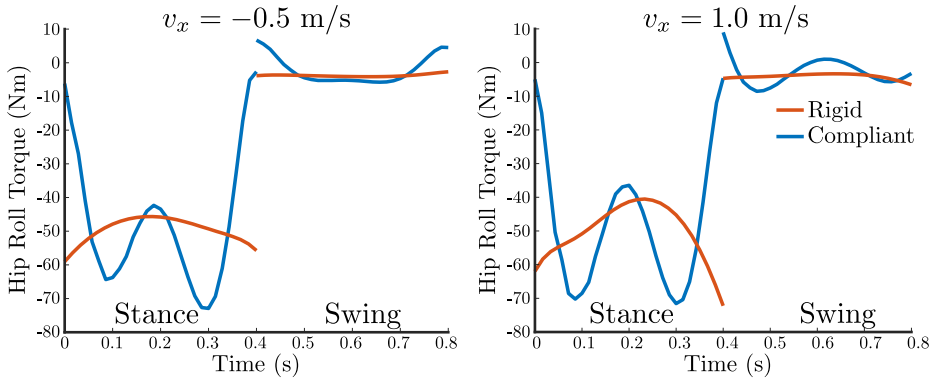

Figure 4.13: Torque at the hip roll joint compared for the rigid and compliant models of Cassie.

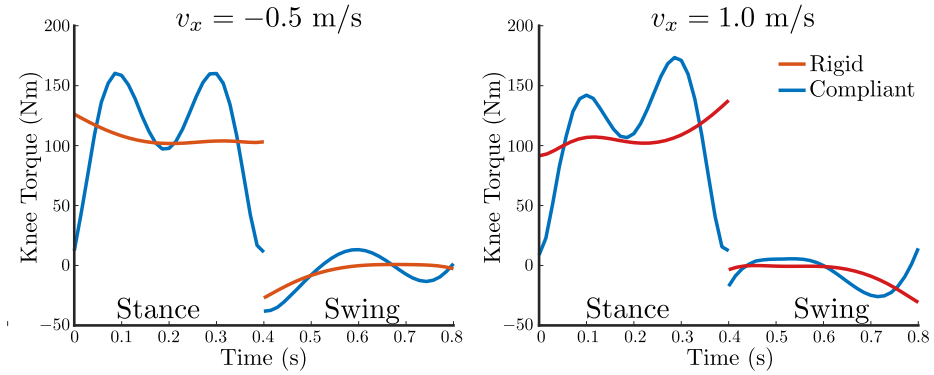

Figure 4.14: Torque at the knee joint compared for the rigid and compliant models of Cassie. Because the knee directly corresponds to the leg length, the emergent torque is very similar to the vertical reaction force.

tion on hardware is smooth torque profiles. Actual actuators cannot instantaneously change torque due to internal dynamics and large discontinuities can lead to instability, vibrations, and even damage to the hardware. On Cassie, the two highest torque joints are the knee and hip roll joints. If we directly compare first the knee in Fig. 4.14 and the hip roll in Fig. 4.13, we can see a similar phenomenon in the torque as was seen with the constraint forces. The most important motivation for using the parameterized torque from the compliant model is the relatively low torque at the beginning and end of a step, with a very small jump at impact. This means that if this torque is used as a feedforward term directly, it will not cause any problems due to discontinuity. In comparison, the rigid model clearly does not consider the passive compliance during stance, and undergoes a massive discontinuity at impact.

Another important consideration for the stability of walking on hardware is the impulsive force arising from impact. In comparison to other robots, Cassie has a relatively

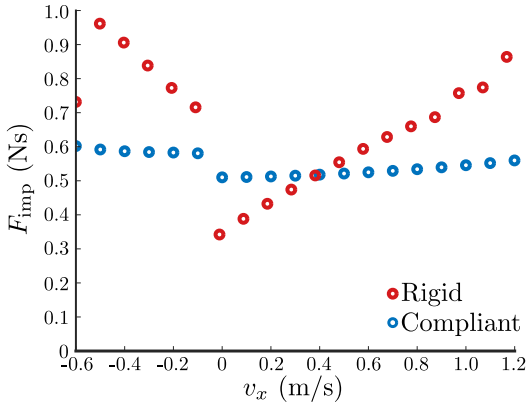

Figure 4.15: Impulsive force at footstrike for sagittal walking with rigid and compliant gait libraries on Cassie.

small impulse as the distal limbs of the robot are all low mass carbon-fiber tubes and the impulsive force is related to the impacting mass. Recall the derivation of the post-impact velocity derived in (2.26). Instead, we can simultaneously solve for the post-impact velocity $\dot{q}^+$ and impulse F_{imp}:

$$\begin{bmatrix} \dot{q}^+ \\ F_{\text{imp}} \end{bmatrix} = \begin{bmatrix} D(q^-) & -J_c(q^-)^T \\ J_c(q^-) & \mathbf{0} \end{bmatrix}^\dagger \begin{bmatrix} D(q^-)\dot{q}^- \\ \mathbf{0} \end{bmatrix}, \tag{4.22}$$

by simply taking the appropriate pseudo-inverse. We compare the impulsive force at impact for the rigid and compliant walking gaits in each library along their sagittal speeds in Fig. 4.15. Here we can see that the compliant model has a very consistent impact force, while the rigid gait varies roughly proportional to speed. Near zero the rigid gait actually performs slightly better, while at the larger speeds in the library exhibits impulsive forces roughly double the compliant gait. Walking is typically the most difficult to stabilize at higher speeds than while stepping in place, meaning that we would like to keep the impulsive force low at these extremes in the library.

Chapter 5

EXPERIMENTAL STUDY: EFFICIENT LOCOMOTION ON DURUS

This chapter will present a set of experiments which were largely performed at Georgia Tech and published in 2016, along with a manuscript currently in preparation:

[1] J. Reher, A. Hereid, S. Kolathaya, *et al.*, "Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot DURUS," in *Twelfth International Workshop on Algorithmic Foundations on Robotics*, 2016.

[2] J. Reher, A. Hereid, E. Cousineau, *et al.*, "The humanoid robot DURUS: How hybrid system models, novel electromechanical design and nonlinear control realized new levels of humanoid efficiency," *IEEE Control Systems Magazine*, 2021, In Preparation.

The walking experiments shown feature the humanoid robot, DURUS, locomoting with a stable human-like HZD gait in 3D on a treadmill, and were motivated largely by improving the walking efficiency of previous work on DURUS presented in [43], [176] and demonstrated at the DARPA robotics challenge. This work was highly collaborative, with co-authors Ayonga Hereid, Shishir Kolathaya, and Christian Hubicki all contributing various aspects under the mentorship of Dr. Aaron Ames. In addition, Eric Cousineau, Eric Ambrose, Stephen Morfey, Paul Birkmeyer, Zachary Shivers, and many more made contributions on the hardware and software infrastructure for DURUS, which were foundational to the work that went into the multi-contact walking. As it pertains to this thesis, the contributions, and the motivation for the inclusion of these experiments are:

- The primary contribution to this work was on the physical implementation, software and algorithm development, and in interfacing with Ayonga Hereid on developing constraints and tuning which were necessary in our model and trajectory optimization to obtain the final walking trajectories.
- The resulting motions are the *first and only* multi-contact humanoid walking developed with HZD to date, and is the *most efficient* reported humanoid walking in the literature to date.
- These experiments demonstrate the *advantages to leveraging fullbody models* of robots, specifically *through underactuation and compliance* within the HZD framework to achieve increasingly *dynamic and efficient* walking.

5.1 Motivation

Preliminary Work with DURUS

Through the formal and heuristic walking methods presented at the beginning of this thesis, combined with novel electromechanical design, the DURUS humanoid was developed as a joint effort between SRI, the AMBER Lab at Texas A&M, and at times the Dynamic Robotics Lab at OSU. This large-scale effort culminated in exceptionally efficient locomotion showcased at the DARPA Robot Endurance Test in 2015. This was a sub-competition which took place during the DARPA Robotics Challenge in Pomona, CA. The competition was funded with the explicit purpose of bettering the humanoid efficiency available on full-scale disaster relief humanoid robots. In a walk-off of sorts available to the public (see Fig. 5.2), the SRI International robot DURUS and the Sandia National Labs robot [189], STEPPR, walked side-by-side on treadmills while displaying efficiency metrics of roughly an order of magnitude more efficient than the walking taking place just steps away at the main event.

Figure 5.1: The competition space inhabited by both DURUS and STEPPR [189] at the DARPA Robot Endurance Test.

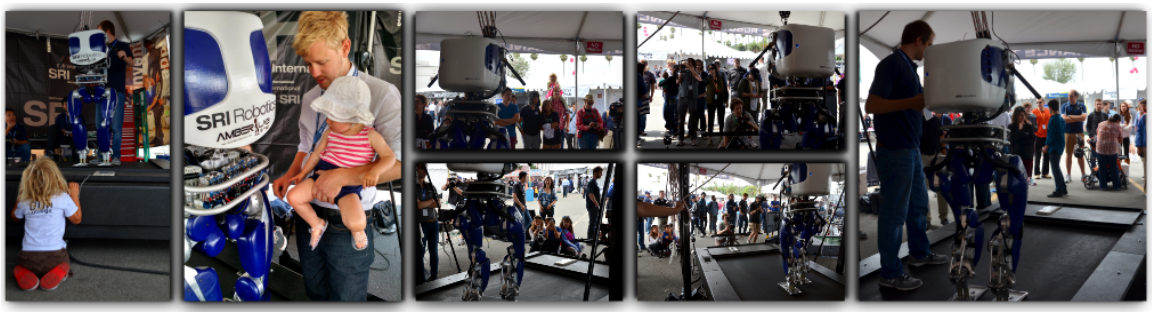

Figure 5.2: Shown are an array of photos which show DURUS walking at the DRC Expo. The robot walked on a large treadmill while the general public could approach and view the experiment as it took place.

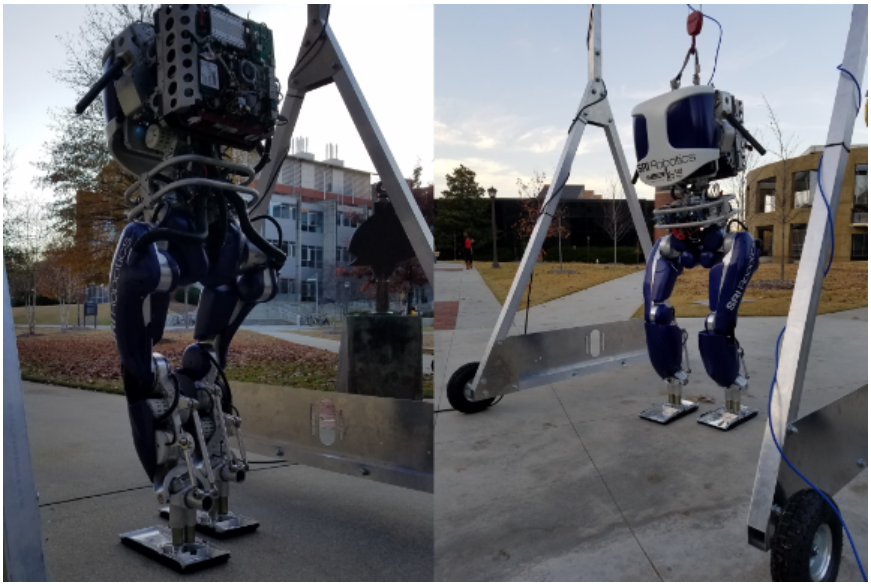

Figure 5.3: Photos of the experiment performed with DURUS traversing the sidewalks of Georgia Tech's campus.

The Robot Endurance Test at the DARPA Robotics Challenge (DRC) took place over two days, during which DURUS exhibited continuous walking over large distances with an efficient cost of transport of approximately 1.6 [43]. The most impressive of these walking runs took place during the second day, during which DURUS walked on two batteries and exceeded walking distances of 3.8 km. Walking data was periodically recorded in ten minute segments while demonstrating the robot to the public. As DURUS walked at the event, a diagnostic feed which displayed real-time efficiency data was facing the public which broadcast distance walked, cost of transport, and power usage. In Fig. 5.1, the walk-off space is shown with STEPPR and DURUS on their respective treadmills and the diagnostic screens[1].

After this event, DURUS was taken with the AMBER Lab to Georgia Tech. There, the existing walking controllers were improved, and new controllers were developed, which will form the basis for this chapter. At Georgia Tech, it was also shown that DURUS could operate without walking solely on a treadmill surface. To perform an outdoor experiment, a mobile gantry was constructed which could catch the robot in the case of a fall, shown in Fig. 5.3. The robot was then taken around Georgia Tech's campus and made to traverse a variety of sidewalks. A video of these experiments[2] is provided, where it can be seen that the robot was able to walk unassisted over a variety of mostly flat areas.

[1]DURUS walking at the DARPA Robotics Challenge: `https://youtu.be/a-R4H8-8074`
[2]DURUS takes a walk around Georgia Tech: `https://youtu.be/Uh7N0D73L_0`

Motivating Multicontact Locomotion Models

Biological bipeds, such as humans, demonstrate walking patterns which are efficient, agile, fast, and robust to a degree not yet attainable by robotic systems. While humans and other biological bipeds can perform these motions with relative ease, translation of these capabilities to 3D humanoid systems is fraught with complexities in the form of nonlinearities, modeling errors, and high degrees of freedom which must be coordinated. The array of behaviors which these bipeds demonstrate is vast: there are bipeds that are efficient but slow [5], fast but inefficient [190], agile but not robust [7], and robust but not fast [91], but none that demonstrates all of these attributes. Human walking consists of several phases, described in this work as domains during which a biped's feet interact with the environment through various contact points. With the goal bridging this gap in natural and efficient locomotion on robots, it is advantageous to develop algorithmic approaches capable of exploiting the natural dynamics of the robot. While some researchers argue that robotics is currently limited by physical hardware capabilities [191], a lack of fundamental knowledge in the area has yet to be bridged as well. Robotic walking presents a wide range of mathematical and algorithmic challenges that provides a fertile proving ground for addressing these gaps. In an attempt to generate more human-like walking patterns, multi-contact methods have been implemented which allowed for longer walking strides and increased energy efficiency through heel and toe contact conditions [192], [193]. One

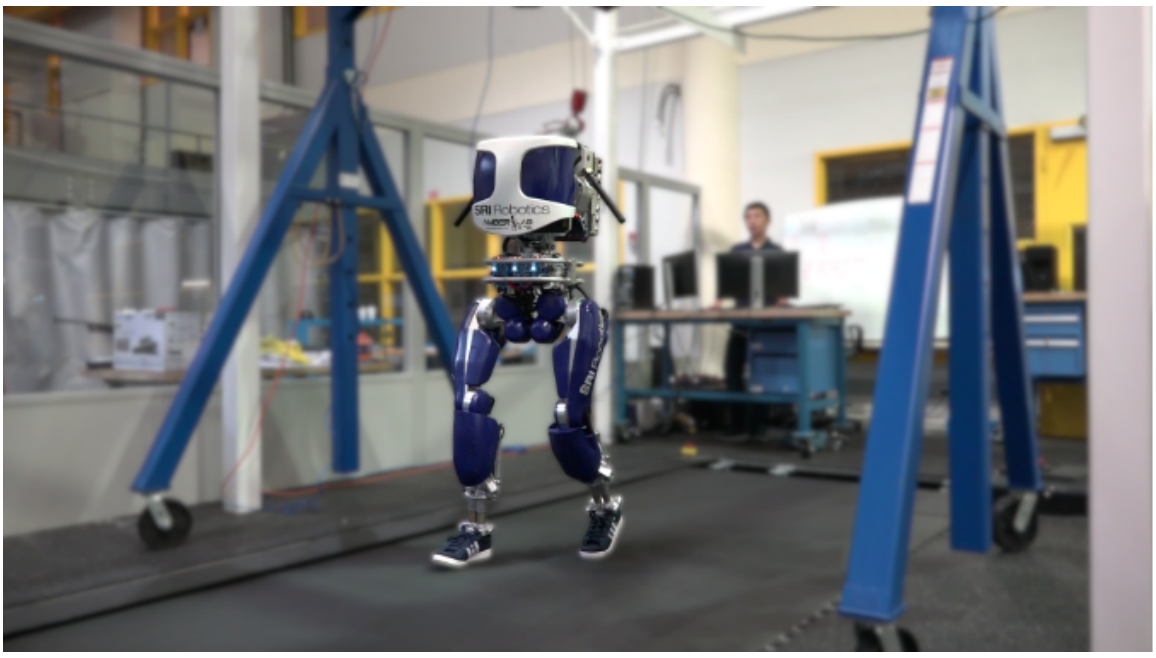

Figure 5.4: The humanoid robot, DURUS, walking heel-to-toe experimentally.

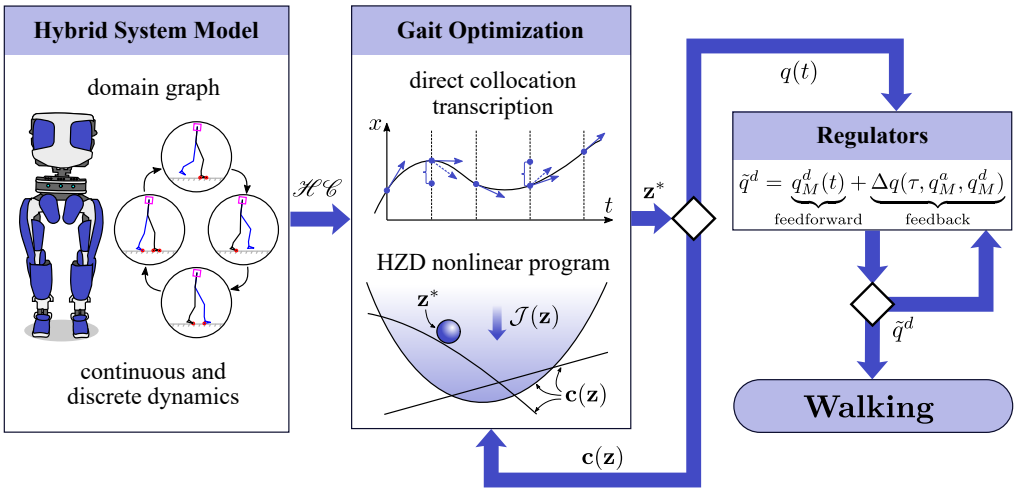

Figure 5.5: The "meta-algorithm" followed to achieve walking control on hardware for the DURUS humanoid, with decision points (diamonds) representing iteration based on the results of robot experiments.

difficulty with this approach is that its inherent assumptions prevent it from utilizing the natural forward momentum of the robot in a manner similar to humans.

The goal of this section is thus to provide a foundation upon which HZD based multi-contact walking behaviors can be formally generated and then experimentally realized on humanoid robots. With this goal in mind, we begin with a discussion of human walking patterns and their relation to the domain structure hybrid model of humanoid walking in Sec. 5.2. The optimization method, including the cost function and constraints necessary to arrive at an experimentally realizable gait, are presented in Sec. 5.3. The experimental methods and results along with the discrete feedback compensation algorithms used for experimental stabilization are presented in Sec. 5.4, in which a mean cost of transport over 200 steps is shown as 1.02, the lowest electrical cost of transport yet reported on a 3D humanoid robot. An analysis of the overall performance for the multi-contact gait is then presented in Sec. 5.5. Each section can be thought of as a component within the overall "meta-algorithm" that synthesizes stable walking controllers for hardware, illustrated in Fig. 5.5. In addition, a photo of DURUS walking with the controllers developed in this chapter is shown in Fig. 5.4.

5.2 Multi-domain Human Locomotion Model

In studies of human locomotion, multi-contact behaviors have been found to be essential in reducing joint torques and increasing walking speeds [195]. In this work, a

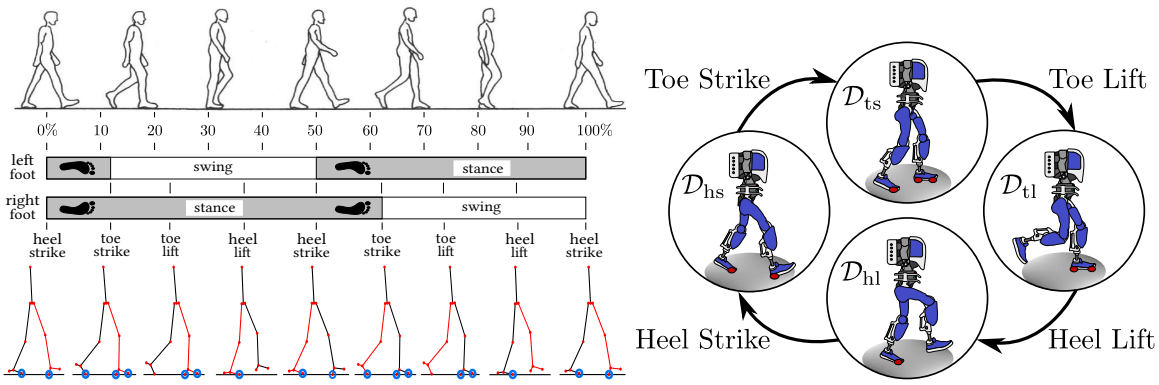

Figure 5.6: Multi-contact locomotion diagram of a typical human gait cycle [194] (left) with a domain breakdown of two steps of one subject based on the changes of heel and toe contact condition. Blue circles represent one specific point in contact with the walking surface. The corresponding directed cycle of four domains is also shown (right), where red circles show the active foot contact points used in our formulation.

walking gait for a 3D humanoid robot is designed with a hybrid domain breakdown matching that of the temporal domain pattern observed in natural human walking motions [155]. From Fig. 5.6, it can be observed that human walking has at least four distinct phases: heel strike (hs) when the swinging foot strikes the ground, toe strike (ts) when the toe of the foot goes down and the legs switch, toe lift (tl) when the other foot takes off the ground and becomes the swinging foot and finally heel lift (hl) when the stance heel goes up with the stance toe being the only contact point with the ground. Other work by authors in the AMBER Lab [156] detailed the implementation of multi-contact locomotion on two 2D robots, in which three domains were used to represent locomotion, with the removal of the domain corresponding to the toe liftoff before swing (ts). While this domain is relatively short in comparison to the overall gait cycle, the inclusion of this phase allows for walking which is more closely aligned with human locomotion and permits longer steps.

The behavior which is ultimately realized on hardware is directly influenced by how the hybrid system is posed. In this work, we consider a dynamic walking motion, which includes more features of human dynamic locomotion such as heel strike and toe push-off with narrow shoe-clad feet. This section details the hybrid graphs, or templates, for how the domain structures are organized, along with the output selections used to parameterize the motions ultimately found through the optimization framework. A periodic human-like walking gait described by a directed cycle consist-

ing of four vertices and edges is prescribed, and represented by a graph $\Gamma = (V, E)$:

$$\begin{aligned} V &= \{ts, tl, hl, hs\}, \\ E &= \{ts \to tl, tl \to hl, hl \to hs, hs \to ts\}, \end{aligned} \tag{5.1}$$

where each vertex represents a continuous domain and each edge corresponds to a transition between these domains, as shown in Fig. 5.6.

5.2.1 Holonomic Constraints

Recall from the DURUS model development in Sec. 4.1 that we allow for two modes of contact for each foot, a flat-foot contact and line contact representing the robot pivoting about the toe or heel. In accordance with the walking pictured in Fig. 5.6, we can prescribe the holonomic constraints applied to each of the respective domains which are then based on the prescribed contact conditions:

$$\eta_{ts}(q) := (p_{sf}, \varphi_{sf}^{xyz}, p_{nst}, \varphi_{nst}^{xz}) \in \mathbb{R}^{11} \tag{5.2}$$

$$\eta_{tl}(q) := (p_{sf}, \varphi_{sf}^{xyz}) \in \mathbb{R}^{6}, \tag{5.3}$$

$$\eta_{hl}(q) := (p_{st}, \varphi_{st}^{xz}) \in \mathbb{R}^{5}, \tag{5.4}$$

$$\eta_{hs}(q) := (p_{st}, \varphi_{st}^{xz}, p_{nsh}, \varphi_{nsh}^{xz}) \in \mathbb{R}^{10}, \tag{5.5}$$

where p_{sf}, p_{st}, p_{nsh}, and p_{nst} are the positions of the stance foot (center), stance toe, non-stance heel, and non-stance toe.

5.2.2 Domains

The continuous domains of walking are determined by the limiting conditions on the ground reaction wrenches and unilateral constraints, as was introduced in Chap. 3, and outlined in the associated discussion on constraint feasibility associated with holonomic foot constraints in Sec. 2.1.3. During the (ts) domain, the stance foot is flat on the ground with a line contact constraining the non-stance toe, meaning we will enforce both friction (2.22) and foot rollover (2.23) in both the pitch and roll on the stance foot while only enforcing friction and lateral rollover on the swing foot:

$$A_{ts}(q, \dot{q}, u) = \begin{bmatrix} \{\lambda_{sf}^{z}, \lambda_{nst}^{z}\} \\ \frac{\mu}{\sqrt{2}}\lambda_{sf}^{z} - \{|\lambda_{sf}^{x}|, |\lambda_{sf}^{y}|\} \\ \frac{\mu}{\sqrt{2}}\lambda_{nst}^{z} - \{|\lambda_{nst}^{x}|, |\lambda_{nst}^{y}|\} \\ \frac{w_f}{2}\lambda_{sf}^{z} - \{|\lambda_{sf}^{mx}|, |\lambda_{sf}^{my}|\} \\ \frac{w_f}{2}\lambda_{nst}^{z} - |\lambda_{nst}^{mx}| \\ p_{nsh}^{z} \end{bmatrix} \geq 0, \tag{5.6}$$

where w_f and l_f are defined for the contact geometry presented in Fig. 4.2, and p^z_{nsh} indicates that the non-stance heel must not go below the ground. Next, the (tl) domain is simply single-support with the stance foot flat on the ground, with (hl) taking place when the robot is in single-support with the foot pitching on the toe:

$$A_{tl}(q,\dot{q},u) = \begin{bmatrix} \lambda^z_{sf} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{sf} - \{|\lambda^x_{sf}|, |\lambda^y_{sf}|\} \\ \frac{w_f}{2}\lambda^z_{sf} - \{|\lambda^{mx}_{sf}|, |\lambda^{my}_{sf}|\} \\ p^z_{nsh}, p^z_{nst} \end{bmatrix} \geq 0, \tag{5.7}$$

$$A_{hl}(q,\dot{q},u) = \begin{bmatrix} \lambda^z_{st} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{st} - \{|\lambda^x_{st}|, |\lambda^y_{st}|\} \\ \frac{w_f}{2}\lambda^z_{st} - \{|\lambda^{mx}_{st}||\} \\ p^z_{sh}, p^z_{nsh}, p^z_{nst} \end{bmatrix} \geq 0. \tag{5.8}$$

Finally, in the (hs) domain, both the stance toe and swing heel are in contact:

$$A_{hs}(q,\dot{q},u) = \begin{bmatrix} \{\lambda^z_{st}, \lambda^z_{nsh}\} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{st} - \{|\lambda^x_{st}|, |\lambda^y_{st}|\} \\ \frac{w_f}{2}\lambda^z_{st} - \{|\lambda^{mx}_{st}||\} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{nsh} - \{|\lambda^x_{nsh}|, |\lambda^y_{nsh}|\} \\ \frac{w_f}{2}\lambda^z_{nsh} - \{|\lambda^{mx}_{nsh}||\} \\ p^z_{sh}, p^z_{nst} \end{bmatrix} \geq 0. \tag{5.9}$$

5.2.3 Guards and Reset Maps

We follow the definitions given in (3.6) to prescribe the guards for each transition:

$$H_{\text{ts}\to\text{tl}}(q,\dot{q},u) := \lambda^{fz}_{nst}(q,\dot{q},u), \tag{5.10}$$

$$H_{\text{tl}\to\text{hl}}(q,\dot{q},u) := \lambda^{fz}_{sh}(q,\dot{q},u), \tag{5.11}$$

$$H_{\text{hl}\to\text{hs}}(q,\dot{q},u) := p^z_{nsh}(q), \tag{5.12}$$

$$H_{\text{hs}\to\text{ts}}(q,\dot{q},u) := p^z_{nst}(q), \tag{5.13}$$

where it can be seen that for the $H_{\text{ts}\to\text{tl}}$ and $H_{\text{tl}\to\text{hl}}$ guards, a domain transition occurs when one of the contact points leaves the ground, corresponding to a loss of normal force. Thus, there is no impact associated with these transitions, and the reset map (2.29) will be an identity map. When the final guard is triggered, we apply the relabeling matrix associated with the symmetric walking assumption given in (4.1), which will switch the right and left legs before entering the (ts) domain. Both the $H_{\text{hl}\to\text{hs}}$ and $H_{\text{hs}\to\text{ts}}$ are associated with an impact, and will enforce stiction of the new contact point through the instantaneous impact equation derived in (2.27).

5.2.4 Virtual Constraint Selection

A control engineer has a large amount of freedom in the selection of outputs to be controlled. In this work, the choice of virtual constraints is inspired by previous work on human-inspired locomotion [155], which uses data from human walking to guide output selection. Several of the core ideas behind the process we follow in the output selection process for walking are given in Sec. 3.2.7.

All possible relative degree 2 output combinations that we made available for our two walking behaviors on DURUS are visualized in Fig. 5.7, where the specific joint indexing can be referenced in Fig. 4.1 and $(p^x_\square(q), p^y_\square(q), p^z_\square(q))$ are the Cartesian positions of a point indicated by the subscript, as illustrated in Fig. 4.2. In addition, the expressions for each output are provided in Table 5.1. We consider positional differences between points on the feet for yaw, pitch, and roll rather than an Euler-angle representation. The reason for picking these outputs over Euler angles is that their expressions do not contain inverse trigonometric functions, which can sometimes cause problems in the optimization.

The output selection for the multicontact walking varies depending on the domain for both the velocity and position-modulating outputs because of the many changing holonomic constraints between domains. Due to the nature of the underactuation

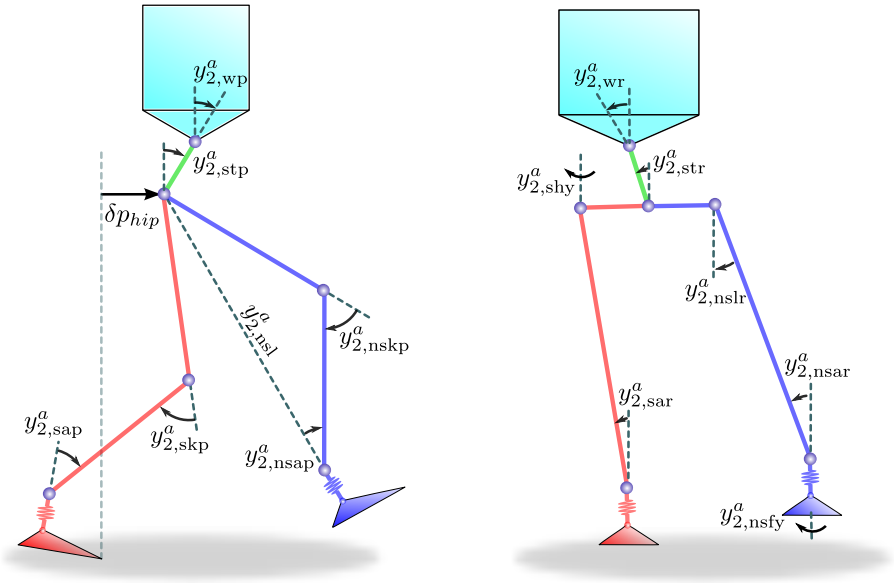

Figure 5.7: A visualization of the outputs which can be selected for DURUS on a given domain. Pictured is the sagittal plane (left) consisting of mostly forward moving and pitch related outputs, and the frontal plane (right) which is primarily associated with the lateral and roll motions of the robot.

Table 5.1: Mathematical expressions for each output made available for selection on DURUS, assuming the right leg is stance.

Stance ankle pitch:	$y^a_{2,\text{sap}}$	$:= \theta_{ra}$
Stance torso pitch:	$y^a_{2,\text{stp}}$	$:= -\theta_{ra} - \theta_{rk} - \theta_{rh}$
Stance ankle roll:	$y^a_{2,\text{sar}}$	$:= \varphi_{ra}$
Stance torso roll:	$y^a_{2,\text{str}}$	$:= -\varphi_{ra} - \varphi_{rh}$
Stance knee pitch:	$y^a_{2,\text{skp}}$	$:= \theta_{rk}$
Non-stance leg roll:	$y^a_{2,\text{nslr}}$	$:= \varphi_{ra} - \varphi_{la}$
Stance hip yaw:	$y^a_{2,\text{shy}}$	$:= \psi_{rh}$
Non-stance foot roll:	$y^a_{2,\text{nsfr}}$	$:= p^z_{nsf^{\text{I}}}(q) - p^z_{nsf^{\text{O}}}(q)$
Waist pitch:	$y^a_{2,\text{wp}}$	$:= \theta_w$
Non-stance foot pitch:	$y^a_{2,\text{nsfp}}$	$:= p^z_{nst}(q) - p^z_{nsh}(q)$
Waist roll:	$y^a_{2,\text{wr}}$	$:= \varphi_w$
Non-stance foot yaw:	$y^a_{2,\text{nsfy}}$	$:= p^y_{nst}(q) - p^y_{nsh}(q)$
Waist yaw:	$y^a_{2,\text{wy}}$	$:= \psi_w$
Non-stance slope:	$y^a_{2,\text{nsl}}$	$:= -\theta_{ra} - \theta_{rk} - \theta_{rh} + \frac{L_c}{L_c+L_t}\theta_{lk} + \theta_{lh}$
Non-stance knee pitch:	$y^a_{2,\text{nskp}}$	$:= \theta_{lk}$

present in $\mathcal{D}_{\text{hl}}$ and $\mathcal{D}_{\text{hs}}$, these domains do not use the velocity-modulating output (3.45), and it is only applied to the other two domains. The position-modulating outputs for each domain are:

$$\begin{aligned}
\mathscr{O}_{\text{ts}} &= \{y_{\text{skp}}, y_{\text{stp}}, y_{\text{sar}}, y_{\text{str}}, y_{\text{shy}}, y_{\text{wr}}, y_{\text{wp}}, y_{\text{wy}}, y_{\text{nskp}}\}, \\
\mathscr{O}_{\text{tl}} &= \mathscr{O}_{\text{ts}} \cup \{y_{\text{nsl}}, y_{\text{nsap}}, y_{\text{nslr}}, y_{\text{nsfr}}, y_{\text{nsfy}}\}, \\
\mathscr{O}_{\text{hl}} &= \mathscr{O}_{\text{tl}} \cup \{y_{\text{sap}}\}, \\
\mathscr{O}_{\text{hs}} &= \mathscr{O}_{\text{ts}} \cup \{y_{\text{sap}}, y_{\text{nsap}}\},
\end{aligned}$$

where it is important to note that where possible, we maintain output selections across domains. This allows us to find one smooth desired polynomial which spans all applicable domains in our optimization routine that follows [155], [156].

5.3 Trajectory Optimization

For multi-contact gait optimization on DURUS, the overall problem is posed as was originally introduced in Sec. 3.3.2, with the specific hybrid system model and virtual constraints outlined in Sec. 5.2. With the goal of optimizing for the most efficient walking possible, we select the mechanical cost of transport (3.54) as the cost function.

As the primary component in designing these walking gaits, an efficient implementation of the optimization is crucial to the rapid design of highly complex walking gaits. The optimization is implemented in MATLAB using the software package IPOPT[3] with linear solver ma57 on a laptop computer with an Intel Core i7-3820QM processor

[3]The IPOPT package: `https://projects.coin-or.org/Ipopt`

(2.7 GHz) and 12 GB of RAM. The number of cardinal nodes are picked as 10, 15, 20, and 12 for the toe-strike, toe-lift, heel-strike, and toestrike domains, respectively. In combining all of the various HZD optimization components, we arrive at an NLP with $21,309$ variables, $22,721$ constraints, and a Jacobian sparsity of 0.05%. Typical computation times for the multi-contact behavior in this work is 647 seconds over 418 iterations.

In addition to the constraints defined in Sec. 3.3.2, other physical constraints can be easily added into $\mathbf{c}$ to ensure that the resulting gaits are feasible on hardware. For example, torque bounds, joint velocity limits and angle limits, etc., can be imposed directly as the boundary value of corresponding optimization variables in $\mathbf{w}^{\min}$ or $\mathbf{w}^{\max}$. Typically, these boundary conditions are determined by the limitation of the robot hardware. It is important to know that, however, only enforcing these basic physical constraints does not guarantee an implementable stable walking gait. Hence, the method lends itself naturally to the addition of physical constraints based on actual hardware considerations for the physical hardware. Using this approach, the following constraints are added to the gait optimization and are configured specifically to provide favorable conditions for experimental walking.

Torso Movement: The optimization tends to find energetically minimal walking gaits in which the torso inertia is used similarly to arm-swing to counter moments generated by the swinging legs. When implemented experimentally, gaits with particularly large torso swing tend to worsen unwanted contact conditions, such as loss of foot contacts or early striking. This can be prevented by constraining the torso movement in the gait design. Let $\varphi_{tor}(q) : \mathcal{Q} \to \mathbb{R}^3$ be the three dimensional orientations of the upper torso link, be restricted within a small range $[\varphi_{tor}^{\min}, \varphi_{tor}^{\max}]$, i.e.:

$$\varphi_{tor}^{\min} \leq \varphi_{tor}(q) \leq \varphi_{tor}^{\max}. \tag{5.14}$$

Impact Velocity: If the swing foot impacts the ground too hard, it can destabilize the robot. Therefore, we constrain the impact velocities of the heel to be within a reasonable range. Let $v_x^{\max}, v_y^{\max}, v_z^{\max} > 0$ be the maximum allowable impact velocities in x, y, and z directions, respectively, then the swing heel velocities $\dot{h}_{swh}(q^-, \dot{q}^-)$ should satisfy:

$$|\dot{h}_{swh}^x(q^-, \dot{q}^-)| \leq v_x^{\max}, \;\; |\dot{h}_{swh}^y(q^-, \dot{q}^-)| \leq v_y^{\max}, \;\; |\dot{h}_{swh}^z(q^-, \dot{q}^-)| \leq v_z^{\max}, \tag{5.15}$$

where $(q^-, \dot{q}^-) \in \mathcal{D}_{hl} \cap S_{hl \to hs}$.

Swing Leg Roll: Due to the existence of unmeasured compliance in the mechanical system, the swing leg can strike the stance leg if they are not separated enough. The separation of legs can be expressed as the difference between stance and swing hip roll angles. Assuming the right leg is the stance leg, then the following constraint should be enforced:

$$\varphi^{\min} \leq \varphi_{rh} - \varphi_{lh} \leq \varphi^{\max}, \tag{5.16}$$

where φ_{rh} and φ_{lh} are the right and left hip roll angles, and $\varphi^{\max} > \varphi^{\min} \geq 0$ are the maximum and minimum allowable leg separation angles.

Ground Reaction Wrench Constraints: In Chap. 2, we model the ground-foot contact as holonomic constraints. However, these constraints are unilateral in essence. Thus the ground reaction wrenches resulting from the contact conditions cannot be infinitely large. The limitations of ground reaction wrenches are often described as the Zero Moment Point (ZMP) constraints, which are discussed thoroughly in [138]. In particular, we enforce the ZMP constraints only during the single support domain $\mathcal{D}_{tl}$ when the stance foot is flat on the ground. In addition, we also constrain the yaw reaction moment of the stance foot, λ_{sf}^{mz}, to be reasonably small:

$$\|\lambda_{sf}^{mz}\| \leq \lambda^{\max}, \tag{5.17}$$

where $\lambda^{\max}$ is the maximum acceptable yaw reaction moment.

5.4 Feedback Control and Implementation

While the prior sections detail how to formulate and find stable walking behaviors which can be applied to our humanoid in simulation, it is not sufficient to simply try to place the trajectories and feedback linearizing controllers on hardware. Specifically, model uncertainty and computational limitations mean that we will not be able to always achieve the perfect tracking properties enjoyed by our simulated robot. Additional considerations must be taken to design controllers which can track the desired behaviors as closely as possible, and to add trajectory level feedback to keep the system stable through disturbances. In the previous section, we discussed the optimization and constraint tuning procedures for obtaining periodic walking motion plans. These trajectories are then slightly modified in their implementation through the use of heuristic *regulators*, shown at a high-level in Fig. 5.8, which help overcome uncertainties due to model mismatch and tracking errors.

5.4.1 Control Architecture

Due to the large effort placed into the formal generation of walking motion offline, the high-level control code actually applied directly on hardware for walking was relatively simple. The hardware implementation of behaviors in this work is realized through the time-based tracking of the optimized motions on embedded level position controllers, with several state feedback terms in the form of trajectory perturbations:

$$\tilde{q}^d = \underbrace{q_0^d(t)}_{\text{feedforward}} + \underbrace{\Delta q^d(\tau, \varphi_b)}_{\text{feedback}} \tag{5.18}$$

where q_0^d is the feedforward trajectory taken directly from simulation of the optimization result, τ is a state based parameterization of time, α_v is a parameter set found through optimization, and $\Delta q^d(\tau, \theta)$ is a trajectory perturbation induced by an additional feedback control layer which we term a *regulator*. This concept is shown in Fig. 5.8, where regulators are superimposed on top of the nominal trajectories as a feedback element to robustify the walking behavior. Once this has been computed, the model coordinates can be transformed back into hardware coordinates for tracking. The implementation in software followed that of [178]: each joint on the robot had a corresponding microcontroller and communicated with a Simulink-generated real-time process via EtherLab using EtherCAT.

Desired joint trajectories travel through two stages before reaching the low-level embedded control. First, desired motor positions are sent via shared memory from the

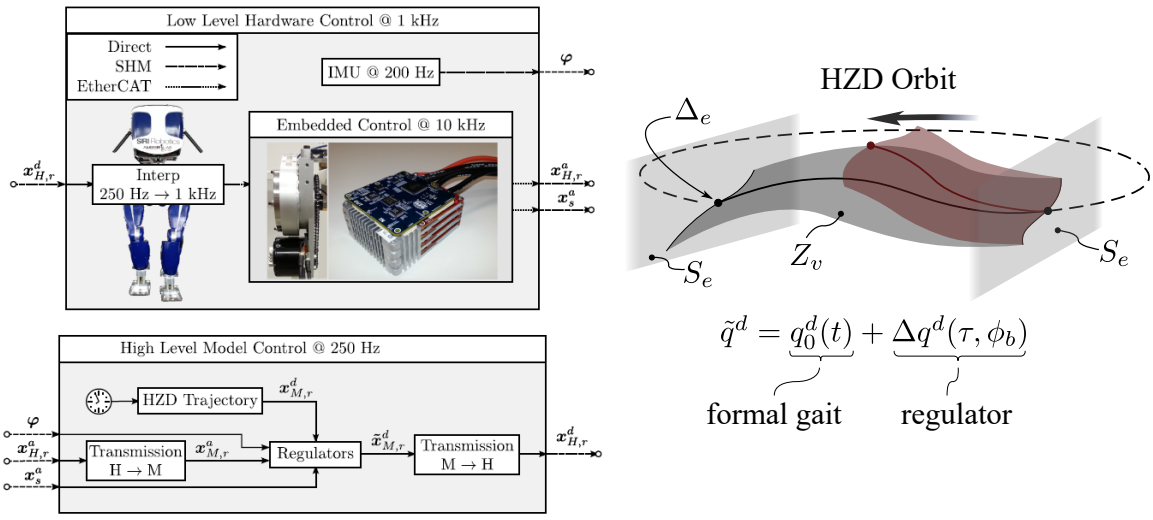

Figure 5.8: Visualizations of the algorithm and control implementation on DURUS. (Left) High-level diagram of the software control components integrated with the embedded electronics. (Right) A visualization of the stabilizing feedback control perturbing trajectories in an attempt to drive the system back onto the HZD surface.

high level process to the Simulink-generated process responsible for communicating with the joint microcontrollers. At this stage, the desired trajectory is interpolated from 250 Hz to 1 kHz and sent to the embedded level via EtherCAT. These components, with their respective loop rates, are shown in Fig. 5.8. Incremental encoders and absolute encoders were both available on each joint, but the incremental encoders were the only sensors actively used in the joint-level feedback. Additionally, a YEI 3-space IMU was used to determine the global orientation of the torso which was used in the feedback regulators.

Formal Gait. To effectively implement the walking gait produced from the optimization problem on hardware, the desired joint and angular velocities of the robot in each iteration must be found. To obtain a set of time-based trajectories for playback on the physical hardware, DURUS is simulated using the feedback linearizing controller in (3.23) [155] and the parameter set obtained from the optimization. The joint trajectories of the stable walking in simulation are recorded and stored as a set of time-based positions and velocities for tracking as:

$$q_0^d(t, \alpha_v) = q_{sim}^a(c_{\mathrm{t}} t, \alpha_v) \tag{5.19}$$

$$\dot{q}_0^d(t, \alpha_v) = \dot{q}_{sim}^a(c_{\mathrm{t}} t, \alpha_v) \tag{5.20}$$

where c_{t} is a scaling constant which is used to allow for the walking trajectories to be sped up or slowed down when implemented on hardware as the q_0^d term in Fig. 5.8.

Regulators for Stabilization. While the nominal trajectories are generated to satisfy dynamics and physical constraints of the ideal system, it was very evident in experimental trials that some form of minor feedback would be crucial to stabilizing the robot for sustained periods of walking. The authors adopted a regulator design similar to those of [83], [196], but with a focus on local position control instead of foot placement or modifying how the outputs are formulated in the trajectory design. Given the actual and desired objectives for a given regulator, perturbations to the nominal trajectory, Δq_d, were computed and then superimposed to yield the final desired trajectory. The primary function of these trajectory perturbations is to smoothly stabilize the robot laterally and to steer the robot.

Each leg is assigned both a stance and swing blending factor for the associated stance and nonstance hips, $s_{i,k} \in [0, 1]$, and is dependent on the contact conditions of the associated leg, $i \in \{stance, swing\}$. This blending prevents large jumps in the commanded position that can occur through transitions between domains. The discrete logic of how the blending factor evolves is implemented using the phase variable,

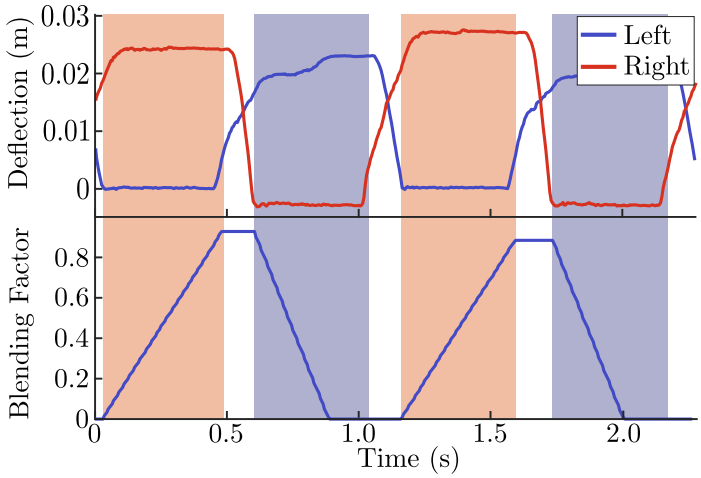

Figure 5.9: An example of how the regulator control modes are blended during periodic flat-footed walking on hardware. Pictured is data recorded from the left hip roll swing blending factor, with the spring deflections triggering classification of contact. Left single support is shaded as blue, while right single support is red, regions without shading are double-support, in which no change is made to the blending factor.

(3.11). Throughout domains in which only one foot is in contact with the ground, $\{\mathcal{D}_{tl}, \mathcal{D}_{hl}\}$, the swing leg blending factor $s_{ns,k}$ was increased for the swing leg according to $s_{ns,k} = s_{ns,k} + \Delta s$ and the associated stance leg blending factor is correspondingly decreased as $s_{st,k} = s_{st,k} - \Delta s_{st,k}$, where $\Delta s = c_{fb}(\tau_k - \tau_{k-1})$ and $c_{fb} \geq 1$ is a tunable acceleration factor. These blending factors are either increased according to $s_{i,k} = s_{i,k} + \Delta s_{i,k}$ where $\Delta s = (\tau_k - \tau_{k-1})$, or decreased according to $s = s - c_{fb}\Delta s_k$ where $c_{fb} \geq 1$ is an acceleration factor such that the applied regulator action blends out faster than the duration of the entire domain. During domains with double-support, $\{\mathcal{D}_{hs}, \mathcal{D}_{ts}\}$, all blending factors are held constant, such that each of the legs do not oppose the motion of the other.

The implementation of the blending factor and how it behaved with regard to the spring based switching played a critical role in the behavior that the regulators induced. During the single support phases of a step, the non-stance regulator would begin to blend into action. Throughout double support phases, no deceleration is applied to the factor. Once a new leg has become stance, the blending factor is taken away, forcing the regulator control action to return to zero, the stance foot is the point of contact with the ground and will turn or roll the body slightly as it returns to the feedforward trajectory.

Switching of the legs and the regulator modes is handled by spring deflection. The discrete modes of the controller are shown in Fig. 5.9, with the red and blue shaded

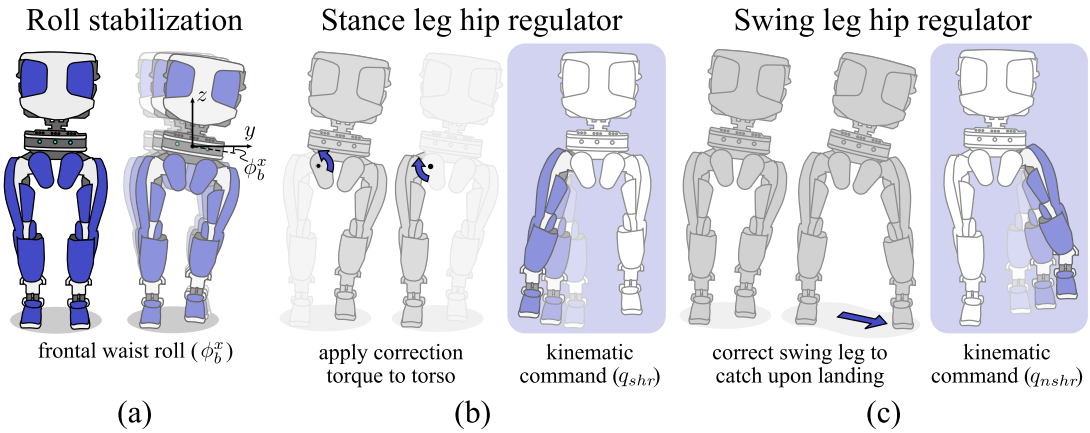

Figure 5.10: Pictured is an illustration of the regulator response to (a) an excessive frontal waist roll. During stance, (b) a counter-rotating torque on the torso is desired to correct the torso roll (left), so a kinematic command is given to the stance leg to adjust the abduction/adduction angle (right). During swing, (c) the regulator widens the strike stance between the two legs (left) by kinematically adjusting the swing leg abduction/adduction angle (right).

regions representing the portions of the gait that the right and left legs are stance, respectively. The effects of the blending factor are seen here, where an angle change is allowed to blend in as the swing phase progresses, then blend out after double support. A physical phenomenon that manifested in the regulator response was the slight hardware asymmetry in the legs of the robot. Throughout the walking experiments, the robot was noticeably less stable to the right in the roll direction. The robot also had a tendency to steer to the left. The result of these asymmetries can be seen in the regulator response as compensating with more roll action in the right leg and a steering action opposing a left turn (Fig. 5.11).

Roll Regulator: Roll regulation is provided for two scenarios in which it can compensate for rolling to the outside of the stance leg or towards the swing leg, pictured in Fig. 5.10. Each regulator performs motion in one direction; adduction for the stance leg and abduction for the swing. Specifically, the main stabilizing action of the swing roll regulator is to abduct the hip joint of the swing leg to change the foot striking position of the leg while the stance roll regulator adjusts the hip angle to correct for excessive lateral torso lean. The regulator action for each leg is then:

$$\Delta q_{i,k}^d = -s_{i,k} K_i (y^a - y^d), \tag{5.21}$$

where $\Delta q_{i,k}^d$ are the angle abduction and adduction angles added to the trajectories as regulation, $y^a := \varphi_b^{x,a}$, and $y^d := \varphi_b^{x,d}$ are the measured waist roll and time based

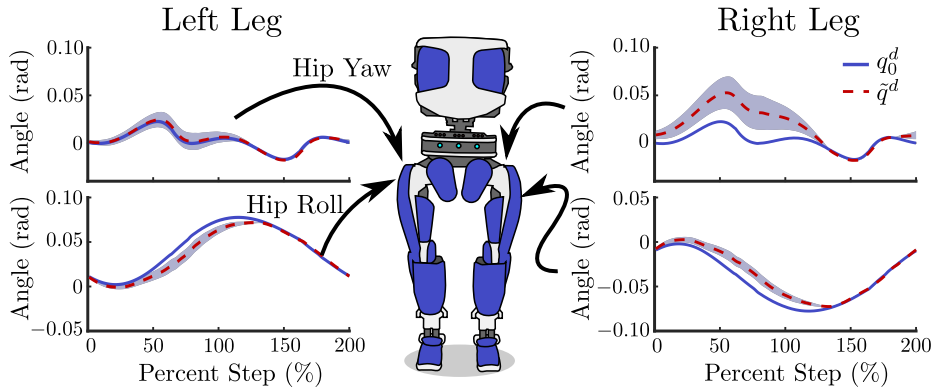

Figure 5.11: Regulator changes to formal walking trajectories during flat-footed walking. Shaded area is one standard deviation of regulated trajectory

waist roll recorded in simulation, and K_i are the tunable nonstance hip and stance hip proportional gains. In order to arrive at a steady state walking gait, the robot is typically brought up with gains of zero. The gains are iteratively increased and decreased while a researcher stabilizes the robot until the appropriate regulator action is achieved to permit steady state walking.

Yaw Regulator: The yaw regulator incorporates user input from a joystick as the desired objective, takes zero as the actual heading, and then modulates the hip yaw joints. The user input from the joystick was used instead of a magnetometer heading as environmental interference was sufficient to make the data unusable in direction-keeping while walking on a treadmill. The desired effect of this regulator on the physical system is to yaw the hip joint while the leg is in swing phase to change the orientation at which the foot will strike the ground. After double support, the hip yaw is blended away with the foot planted, turning DURUS about the stance leg and into the desired direction.

Unmodeled Compliance Compensation. A prevalent problem with humanoid hardware is unmeasured compliance in the system. Throughout the walking behaviors presented in this work, DURUS exhibited unmeasured compliance in its hips with deflections reaching over 0.1 radians. This has also been cited as an issue on early versions of the DRC ATLAS [58], in which a linear compliance assumption is introduced to augment the measured angles fed to a fullbody estimator. A similar approach is used here with the primary difference that the compensator directly adjusts the desired joint configuration via a position command:

$$q_j^{comp} = q_{M,j}^d + u_j/K_j, \tag{5.22}$$

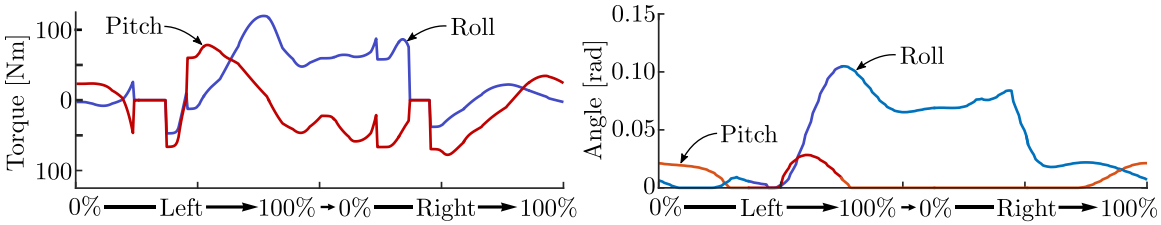

Figure 5.12: The anticipated torque computed on the robot (Left) using the time-based trajectories over two steps of multicontact walking, with the angle compensation applied directly to the hip roll and pitch joints (Right) over the corresponding steps.

where $q_j \in \mathcal{Q}$ are the corresponding joints, q_j^{comp} is the preprocessed joint angle to be passed to the controller, $q_{M,j}^d$ is the nominal joint angle, u_j is the feedback linearizing torque computed at each time step from simulation, and K_j is the stiffness coefficient which has been measured for each joint. The compliance parameters, measured with a force gauge and caliper in units of Nm/rad, for the joints on DURUS were found to be $K_{lhp} = 1284$, $K_{lhr} = 900$, $K_{rhp} = 1124$, and $K_{rhr} = 815$. The anticipated torque at the joint is computed online and the values q_j^{comp} are displayed in Fig. 5.12.

Ankle Inverse Kinematics. While performing multi-contact walking, the heel contact constraint is very important, particularly as it impacts the ground and transitions between the domains $\mathcal{D}_{hl}$ and $\mathcal{D}_{hs}$. If the robot strikes the ground with a foot configuration which does not have the heel parallel to the ground, then it will be thrown off balance. To ensure the holonomic constraint at this transition is satisfied, we implement an inverse kinematics solver. Angles for the motors controlling the push-rod transmission in the swing foot are solved with a Newton-Raphson method and ensure the swing heel strikes the ground evenly. The solver's objective ensures that the swing foot strikes the ground evenly, with both the inner and outer edges of the heel at the same time. The solver is only active on the swing leg through the domains $\mathcal{D}_{tl}, \mathcal{D}_{hl}$, and once the leg switches to stance, the modified configuration is slowly blended away with a linearly decreasing constant.

5.5 Results and Conclusions

This section discusses the experimental results which were obtained through the implementation of the walking gait which was found for the multi-contact motion primitive. The limit cycles achieved experimentally on DURUS shown in Fig. 5.14 exhibit a closed behavior, indicating that the multi-contact walking behaviors are stable. It is clear from the hip roll limit cycle that this is the joint most heavily augmented by the feedback regulators. The resulting walking is also shown as a series of walking

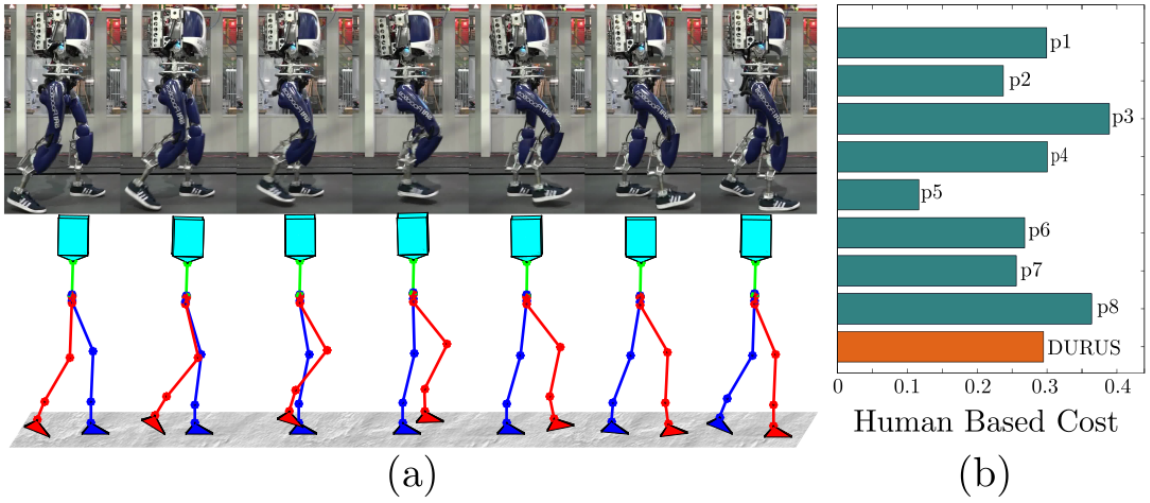

Figure 5.13: Visualization of the human-like motion on hardware. (a) Tiles of DURUS walking seen at an angle from front. (b) Human-based cost of DURUS compared to eight healthy human subjects.

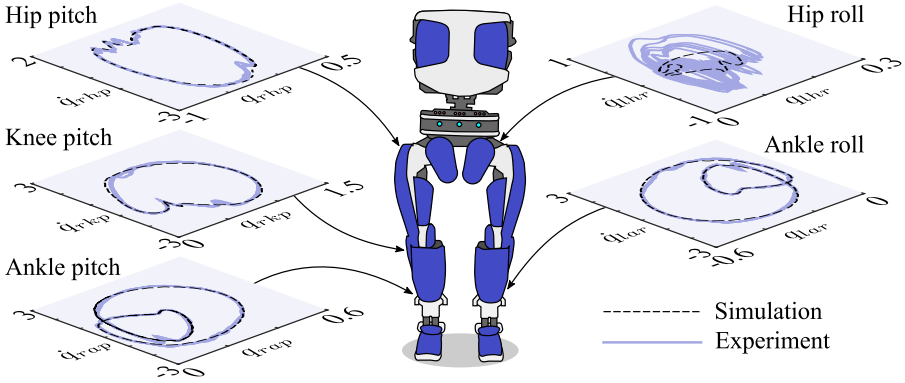

Figure 5.14: Pictured is the trace of continuous walking limit cycles over 10 steps (solid) compared to the nominal trajectory simulation (dashed).

tiles in Fig. 5.13(a), where the experimental movements are time-synchronized to the simulated walking motions.

The experimentally implemented multicontact walking gait ambulated with a forward velocity of 0.60 m/s and a stride length of 0.39. Energy efficiency is used throughout these results as a metric in which to evaluate the mechanical design and control implementation. Specifically, we compare the walking controllers using the specific cost of electrical transport. Note that this is different from the cost function previously used in (3.54), as it includes all electrical power consumed. The specific cost of

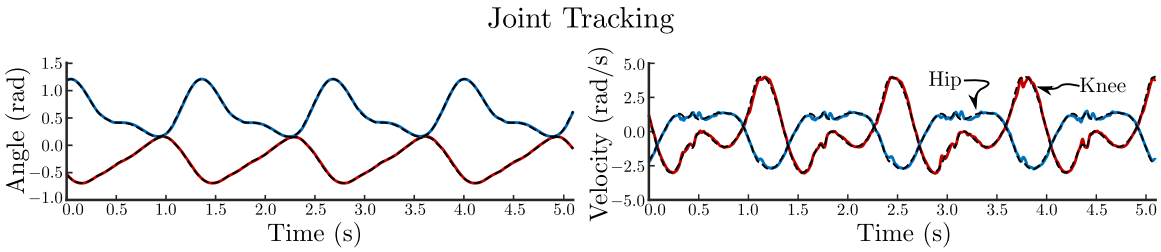

Figure 5.15: Experimental results for the use of PD control of joint tracking for the DURUS hip and knee during multicontact walking.

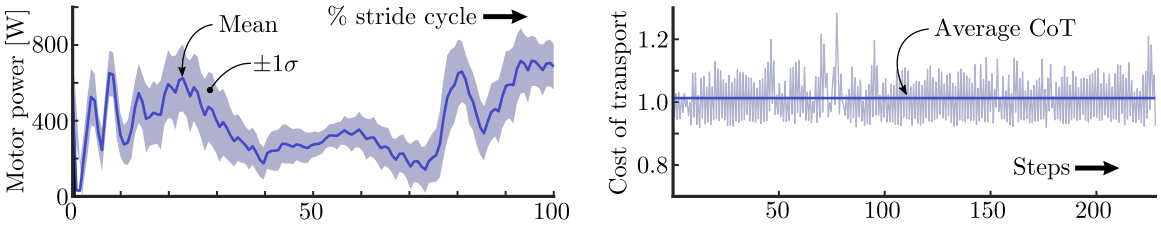

Figure 5.16: The specific cost of transport and mean motor power consumed per step over 200 steps of continuous walking.

electrical transport $c_{et,i}$ for each step i is computed according to [197] as:

$$c_{et,i} = \frac{1}{mgd_i} \int_{t_i^+}^{t_i^-} P_{el} + \sum_{j=1}^{15} I_j(t) V_j(t) dt, \tag{5.23}$$

where P_{el} is the average logic power consumed by the on-board computer and motor controllers, d_i is the x-position traveled by the non-stance foot of the robot through the i^{th} step, and $I_j(t)$ and $V_j(t)$ are the currents and voltage recorded for the j^{th} motor. The mean total power consumed over all 15 actuators for 200 steps along with the cost of transport per step can be seen in Fig. 5.16. These results indicate that the mean CoT for DURUS during steady-state multi-contact locomotion is $\bar{c}_{et} = 1.02$, which is 37% more efficient than experimental results obtained on DURUS for flat-footed walking [43].

The reported electrical cost of transport for several robots is summarized in Table 5.2, from which we observe the robots utilizing passive elements, small motors, or anthropomorphic designs to leverage energy savings demonstrating the lowest energy expenses (Cornell Ranger and Biped). Additionally, robots employing HZD to achieve locomotion exhibit efficient locomotion (AMBER 1 and 2D-DURUS), although these are restricted to walking in a 2D plane. The closest efficiency numbers come from ATRIAS—yet this robot is not humanoid in nature. Therefore, in the category of full-scale bipedal humanoid robots (e.g., ATLAS and ASIMO), the electrical cost of transport on DURUS is the lowest reported to date.

Table 5.2: Comparison of reported specific cost of electrical transport for walking behaviors on several bipedal platforms.

Name	$\bar{c}_{et}$	m (kg)	v (m/s)
Human	0.2	·	·
Cornell Ranger [199]	0.19	9.9	·
Cornell Biped [197]	0.2	13	0.4
2D-DURUS [178]	0.63	31.5	0.68
AMBER 1 [114]	1.88	3.3	0.44
ATRIAS [83]	1.13	62	0.85
ASIMO [197]	3.23	52	0.44
ATLAS [199]	5	102	2
DURUS (Flat foot)	1.61	79.5	0.3
· (Multicontact)	1.02		0.6

Table 5.3: Comparison of domain durations in humans versus the walking on DURUS.

Domain	Human	DURUS
$\mathcal{D}_{ts}$	6%	4.6%
$\mathcal{D}_{tl}$	59%	52.4%
$\mathcal{D}_{hl}$	18%	29.6%
$\mathcal{D}_{hs}$	17%	13.4%

Since this methodology begins with analysis and utilization of a domain sequencing mirroring that of humans, we would like to better determine whether the approach generates behaviors in line with that of nominal human walking. To provide a more quantitative measure of "human-likeness" of DURUS' walking, the *human-based cost* of eight human subjects[4] and DURUS are computed with respect to the nominal human domain cycle presented in Table 5.3. We define a *walking cycle* as a pair (γ, l) with $l = (V, E)$ the graph presented in Fig. 5.6 and $\gamma : l \to \mathbb{R}^{|V|}$ is a function such that $\gamma(v) \geq 0$ and $\sum_{v \in V} \gamma(v) = 1$. The human-based cost for the multi-contact walking gaits can be found as the *cut distance* between the the nominal human cycle (γ^*, l^*) and the optimized cycle (γ_r, l_r).

We then view γ^* and γ_r as functions on $V^* \cup V_r$ by letting $\gamma^*(i) \equiv 0$ if $i \in V_r \backslash V^*$ and $\gamma_r(j) \equiv 0$ if $j \in V^* \backslash V_r$. The cut distance is then computed as in [198] by:

$$d(\gamma^*, l^*, \gamma_r, l_r) = \max_{I,J \subset V^* \cup V_r} \Bigg| \sum_{i \in I, j \in J} (\gamma^*(i)\gamma^*(j)\beta^*(i,j) - \gamma_r(i)\gamma_r(j)\beta_r(i,j)) \Bigg| + \sum_{k \in V^* \cup V_r} |\gamma^*(k) - \gamma_r(k)|, \tag{5.24}$$

where $\beta^*(i,j) = 1$ for all edges $(i,j) \in E^*$ and $\beta_r(i,j) = 1$ for all edges $(i,j) \in E_r$. The eight healthy subjects feature human-based costs ranging from 0.12 to 0.36 in which DURUS has a cost of 0.30. A comparison of the human-based costs for each of these subjects and DURUS is pictured in Fig. 5.13(b).

[4]The human walking cycles analyzed are derived from the dataset presented in [198].

Chapter 6

EXPERIMENTAL STUDY: COMPLIANT WALKING ON CASSIE

The experiments presented in this chapter are the result of research performed at Caltech from 2017 through the end of 2020, which resulted in two publications specifically on the modeling and optimization of compliant HZD locomotion for Cassie:

[1] J. Reher, W.-L. Ma, and A. D. Ames, "Dynamic walking with compliance on a Cassie bipedal robot," in *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 2589–2595.

[2] J. Reher and A. D. Ames, "Inverse dynamics control of compliant hybrid zero dynamic walking," *arXiv preprint arXiv:2010.09047*, 2020.

This chapter contains two sections; the first details preliminary work on a compliant multi-domain locomotion model for Cassie which includes a double-support domain, and the second outlines the development and implementation of library of compliant walking gaits. Both sections draw heavily from the compliant Cassie model which was developed and motivated in Chap. 4.

This work was performed with a much smaller team than the previously presented DURUS work in Chap. 5, with coauthor Wen-loong Ma collaborating closely on the development of the trajectory optimization for the first publication. The optimization and implementation in the second publication was primarily a sole effort, which built on some of the existing tools from Chap. 5 and the first publication on Cassie. As it pertains to this thesis, its contributions, and the motivation for the inclusion of these experiments are:

- The contributions focused on the conception of the project, a collaborative effort on the early modeling and optimization and an individual effort on the motion library, theoretical discussion, implementation, software, and experiments.
- The development of a *compliant locomotion model* for a heavily *underactuated 3D biped* is introduced and shown to provide desirable properties over the existing methods in the literature.
- The experiments further the argument of this thesis on *leveraging fullbody robot models in optimization*, and show how *model-based controllers can leverage* a parameterization of the resulting HZD trajectories.

6.1 Two-domain Compliant Walking

6.1.1 Motivation

One of the objectives of this thesis is to motivate the use of fullbody dynamics, including compliance and underactuation, to synthesize dynamic and efficient locomotion. As was previously discussed in Chap. 1, a significant subset of the bipedal robotics literature mitigates the complexity of humanoids and bipeds by viewing walking as a problem wherein the real world dynamics are assumed to be governed by the evolution of a simpler system, such as a LIP models (Linear Inverted Pendulum [52], [200]), SLIP models (Spring Loaded Inverted Pendulum [83]), and the ZMP (Zero Moment Point [4]). These methods can reduce computational complexity for fast planning and experimental success. Despite its viability in practical implementation, this local representation of the system can limit the agility of behaviors and compromise energy efficiency, and may require addition optimization to ensure viable walking [201], [202].

In contrast to these approaches, this section will explore a preliminary implementation of HZD walking on Cassie that mirrors the two-domain compliant hybrid structure of early work on DURUS (see the flat-footed walking in Sec. 5.1). Specifically, when there is compliance within a leg structure during locomotion, there will inevitably be a nontrivial domain in which both legs are in contact with the ground (double-support). This additional domain can be leveraged in optimization to provide for a smooth load transfer from stance to non-stance, and to improve stability.

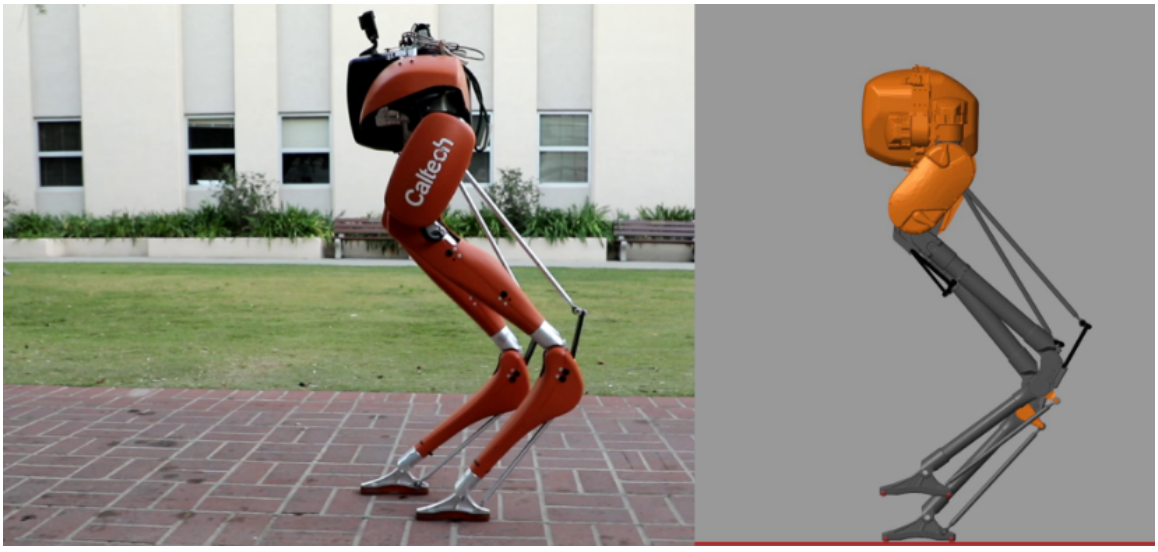

Figure 6.1: Cassie walking outside, shown in double-support, and in a Simscape Multibody simulation with a multi-domain and compliant gait.

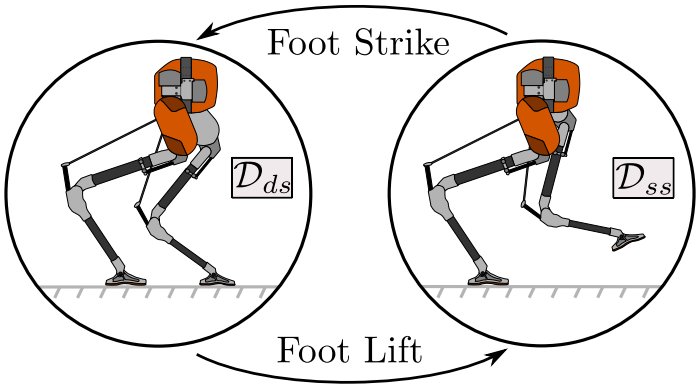

Figure 6.2: The directed graph of walking dynamics, on the left is double support domain $\mathcal{D}_{\text{ds}}$ and on the right is the single support domain $\mathcal{D}_{\text{ss}}$.

The primary point that will be made is that the use of a *high-fidelity HZD trajectory optimization can lead to a straightforward control implementation* on the robot as the *optimization result encodes a significant amount of control information*. Because HZD synthesizes a closed-loop locomotion plan in the optimization which was presented in Sec. 3.3.2, one of the simplest ways to enforce the behavior is to enforce high-gain PD control of the output polynomials. If these are sufficiently zeroed, then the natural dynamics of the system should behave in a stable manner according to the hybrid invariance condition (3.30). The implementation here will thus demonstrate that the optimization can accurately capture the passive compliance of a highly complex robot, and will be used as a starting point for the development of more advanced trajectory optimization strategies in Sec. 6.2 and as a foundation for implementable model-based controllers in Chap. 8.

6.1.2 Multi-domain Locomotion Model

We structured the dynamics of walking on Cassie in a multi-domain and hybrid fashion. A directed cycle, $\Gamma = (V, E)$, is specified for the system:

$$\begin{aligned} V &= \{ds, ss\}, \\ E &= \{ds \to ss, ss \to ds\}, \end{aligned} \tag{6.1}$$

where each vertex represents a continuous domain and each edge corresponds to a transition between these domains, as shown in Fig. 6.2. Specifically, walking on Cassie involves two continuous domains — a *double-support* domain, $\mathcal{D}_{\text{ds}}$ and *single-support* domain, $\mathcal{D}_{\text{ss}}$, which are connected by two state dependent events — lift and impact. In addition, we consider the contact dynamics as a set of holonomic constraints $\Gamma_v(q) \equiv 0$. For the double support phase $v = \text{ds}$, both feet remain in

static contact with the ground, and for the single support phase $v = \text{ss}$ we only constrain the stance foot's contact dynamics. Much like the walking presented in Chap. 5, the two domain walking considered on Cassie will be symmetric, meaning that we only need to consider one stance leg in the gait development.

Holonomic Constraints

Recall from the Cassie model development in Sec. 4.2 that we allow for only one mode of contact for each foot, corresponding to line contact representing the robot pivoting along the roll direction of the foot. In accordance with double- and single-support indicating either one or both feet in contact with the ground, we define the the holonomic constraints as:

$$\eta_{ds}(q) := (p_{sf}, \varphi^{yz}_{sf}, p_{nsf}, \varphi^{yz}_{nst}, \eta_{4\text{bar}}) \in \mathbb{R}^{12} \tag{6.2}$$

$$\eta_{ss}(q) := (p_{sf}, \varphi^{yz}_{sf}, \eta_{4\text{bar}}) \in \mathbb{R}^{7}, \tag{6.3}$$

where p_{sf} and p_{nsf} are the positions of the stance foot and non-stance foot, pictured in Fig. 4.7, and it can be seen that the closure of the compliant multi-bar leg is enforced through the holonomic constraint (4.18). In the walking of this section, both the swing and stance legs remain compliant, as can be seen by the lack of an additional holonomic constraint on the swing spring deflections in (6.3).

Domains of Admissibility

The continuous domains of walking are determined by the limiting conditions on the ground reaction wrenches and unilateral constraints, as was introduced in Chap. 3, and outlined in the associated discussion on constraint feasibility associated with holonomic foot constraints in Sec. 2.1.3. During the (ds) domain, the stance foot is in contact with the ground as a line contact constraining the length of the foot, meaning we will enforce both friction (2.22) and foot rollover (2.23) in the pitch direction on both the stance and non-stance foot:

$$A_{ds}(q, \dot{q}, u) = \begin{bmatrix} \{\lambda^z_{sf}, \lambda^z_{sf}\} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{sf} - \{|\lambda^x_{sf}|, |\lambda^y_{sf}|\} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{nsf} - \{|\lambda^x_{nsf}|, |\lambda^y_{nsf}|\} \\ \frac{l_f}{2}\lambda^z_{sf} - \{|\lambda^{my}_{sf}|\} \\ \frac{l_f}{2}\lambda^z_{nsf} - |\lambda^{my}_{nsf}| \end{bmatrix} \geq 0, \tag{6.4}$$

where l_f is defined for the contact geometry presented in Fig. 4.7. Next, the (ss) domain is simply single-support:

$$A_{ss}(q,\dot{q},u) = \begin{bmatrix} \lambda^z_{sf} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{sf} - \{|\lambda^x_{sf}|, |\lambda^y_{sf}|\} \\ \frac{w_f}{2}\lambda^z_{sf} - \{|\lambda^{my}_{sf}|\} \\ p^z_{nsf} \end{bmatrix} \geq 0, \tag{6.5}$$

where p^z_{nsf} indicates that the non-stance foot must not go below the ground for the domain to be valid.

Guards and Reset Maps

We follow the definitions given in (3.6) to prescribe the guards for each domain transition:

$$H_{\text{ds}\to\text{ss}}(q,\dot{q},u) := \lambda^{fz}_{nsf}(q,\dot{q},u), \tag{6.6}$$

$$H_{\text{ss}\to\text{ds}}(q,\dot{q},u) := p^z_{nsf}(q), \tag{6.7}$$

where it can be seen that for the $H_{\text{ts}\to\text{tl}}$ guard, the domain transition occurs when one of the swing foot leaves the ground, corresponding to a loss of normal force. Thus, there is no impact associated with this transition, and the reset map (2.29) will be an identity map. When the final guard is triggered, we apply the relabeling matrix associated with the symmetric walking assumption given in (4.12), which will switch the right and left legs before entering the (ds) domain. The $H_{\text{ss}\to\text{ds}}$ guard is associated with an impact, and will enforce stiction of the new contact point through the instantaneous impact equation derived in (2.27).

Virtual Constraint Selection

As was outlined in Sec. 3.2.7, the choice of virtual constraints within the HZD framework is largely a design choice, and has some influence over the resulting behavior. Since the Cassie robot was designed to behave mechanically similar to a SLIP, we define outputs based on the undeflected Cartesian positions of the legs by zeroing the spring deflections. By formulating the outputs in this way, the passive dynamics of the system will contain the dynamics associated with the compliant elements [41].

Due to the nature of the underactuation, and of the relatively small nature of the foot on the robot, we do not include a relative degree 1 output. Instead, the stance foot actuator is constrained to produce zero torque and assumed to be passive, resulting

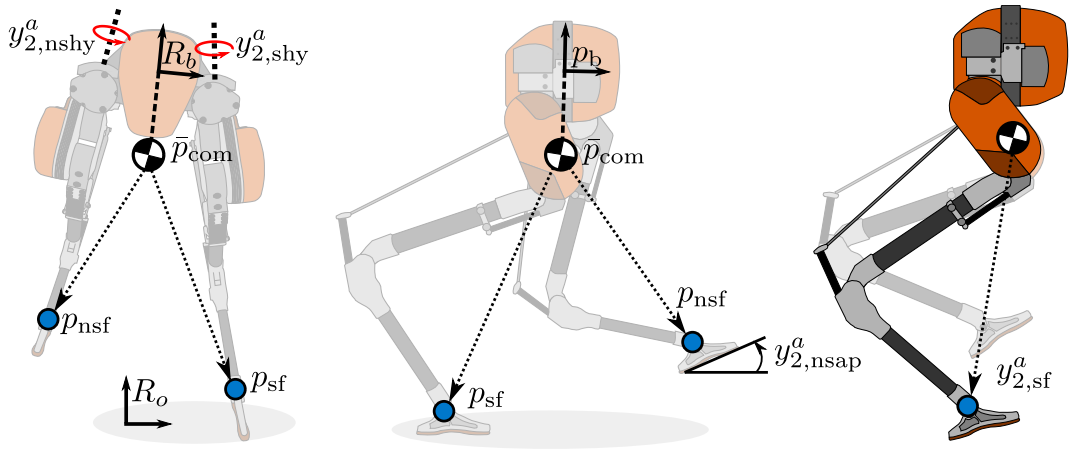

Figure 6.3: A visualization of the outputs which can be selected for Cassie in this section. Pictured are the definitions for the average center of mass location and Cartesian foot locations. On the right is an illustration of the difference between the actual foot position and the neutral leg configuration, for which the neutral (meaning undeflected) positions are used as actual outputs.

in the robot having access to 8 actuators during double-support and 9 actuators in single-support. Following the natural representation of Cassie as pendulum-like, we first describe the "average" center of mass pictured in Fig. 6.3:

$$\bar{p}_{\text{com}}(q) = p_b + R(\varphi_b)[0,\ 0,\ -0.125]^T, \tag{6.8}$$

where $R(\varphi_b)$ is the rotation matrix associated with the floating base. This position was found as an approximate value by moving the robot's legs through a variety of configurations and observing the location of the center of mass.

We then define all of the relative degree 2 actual outputs which are made available to walking on Cassie as:

$$\begin{aligned}
y^a_{2,\text{sf}} &= \bar{p}_{\text{com}}(q_{\text{b}}) - p_{\text{sf}}(q_{\text{b}}, \bar{q}_l) && \text{(stance foot positions)}\\
y^a_{2,\text{nsf}} &= \bar{p}_{\text{com}}(q_{\text{b}}) - p_{\text{nsf}}(q_{\text{b}}, \bar{q}_l) && \text{(swing foot positions)}\\
y^a_{2,\text{shy}} &= q_{\text{shy}} && \text{(stance hip yaw)}\\
y^a_{2,\text{nshy}} &= q_{\text{nshy}} && \text{(swing hip yaw)}\\
y^a_{2,\text{nsap}} &= \varphi^y(q_{\text{b}}, \bar{q}_l) && \text{(swing foot pitch)}
\end{aligned}$$

where p_{sf} and $\varphi^y(q_{\text{b}})$ are the stance ankle Cartesian position and swing foot pitch, and the expressions for the various leg configurations are shown derived for the "neutral," or uncompressed, leg through the coordinate substitution $\bar{q}_l \in \{\mathcal{Q}_l \mid q_{\text{sp}} = 0, q_{\text{hs}} = 0, q_{\text{t}} = 13^o - q_{\text{k}}\}$. It should be noted that while the undeflected leg outputs are

computed with the rough approximation given by $q_{\mathrm{t}} = 13^o - q_{\mathrm{k}}$ rather than the true neutral configuration of (4.7), the compliant passive dynamics will still be encoded for the full-order model of the compliant leg structure given in (4.18) through the optimization. The result is to have a simple set of control objectives for derivation of feedback controllers, while allowing the nonlinear optimization framework to do the more complex work of embedding the spring dynamics.

The position-modulating outputs selected for control in each domain are:

$$\mathscr{O}_{\mathrm{ds}} = \{y_{2,\mathrm{sf}}, y_{2,\mathrm{nsf}}, y_{2,\mathrm{shy}}, y_{2,\mathrm{nshy}}\}, \tag{6.9}$$

$$\mathscr{O}_{\mathrm{ds}} = \mathscr{O}_{\mathrm{ds}} \cup \{y_{2,\mathrm{nsap}}\}, \tag{6.10}$$

where we maintain all outputs except for the swing foot pitch across both domains. In order to simplify the control implementation on hardware and in optimization, we enforce the selection of one smooth desired polynomial which spans both domains for all applicable outputs (all except the swing foot). This is possible because the guard $H_{\mathrm{ds}\to\mathrm{ss}}$ is associated with a trivial impact map $\Delta = \mathbf{I}$.

6.1.3 Trajectory Optimization

Following the derivations of Chap. 3, assume that we have zero dynamics in the form $\dot{z} = f|_{\mathcal{Z}_v^\alpha(z)}$ for the full-order dynamical system of Cassie presented in this section and Sec. 4.2. This leaves us with an optimization problem, wherein we must find a set of Bézier parameters α^* to shape the zero dynamics in order to achieve hybrid invariance Sec. 1.2.6. This parameter search is solved with a *direct collocation* method, see Sec. 3.3.2, coupled with the IPOPT solver [203].

As a proof of concept, we only design a single gait for *stepping in place* by constraining the average forward and lateral velocity over a step to be 0. Corresponding to the goal of simply finding a stepping in place gait, the objective function which was first chosen is to minimize the pelvis velocity:

$$\mathcal{J}_v(\mathbf{w}) = \sum_i \dot{p}_b(i)^T \dot{p}_b(i), \qquad i \in \{1, 2, ...M\}, \tag{6.11}$$

where $i \in \{1, 2, 3...M\}$ with M the total number of nodes for each domain. While this choice is a particularly naïve one, as typically one would generally like to minimize torque or some other performance metric, for simply stepping in place the resulting controller was well behaved.

In fact, adding additional cost metrics proved to generate walking gaits which had larger torso movements or attempted to more aggressively use spring compression in

order to reduce energy consumption or torque, especially since Cassie has almost no influence over its angular momentum due to its mass distribution. While these energy reduction techniques and more dynamic motions are what we ultimately desire, the simple heuristics and PD feedback control approach shown in Sec. 6.1.4 did not offer high enough fidelity tracking to actually leverage them on hardware. Expanding the available control methodologies will be the subject of Sec. 6.2, and then ultimately to one of the main results of this thesis in Chap. 8. In these later sections, model-based approaches will be developed which, coupled with the compliant model in optimization, produce accurate feedforward and feedback controllers for more dynamic and efficient locomotion over a range of walking speeds.

The optimization problem was solved on an Ubuntu-based computer with an i7-6820HQ CPU @2.70GHz and 16GB RAM with linear solver ma57. The number of cardinal nodes were picked as 8 for double-support and 18 in single-support. The gait which was selected for use on hardware converged in 755 seconds and underwent 773 iterations. Additional constraints were added to the nominal problem, including the double-support domain terminal time, $T_{ds} \in (0.025, 0.3)$ seconds, single-support domain terminal time, $T_{ss} \in (0.3, 0.45)$ seconds, a narrower set of joint limits to encourage convergence, an impact velocity bound $\dot{p}^z_{\text{nsf}}(T_{ss}) \in (-0.02, -0.10)$ m/s, step width bound $|p^y_{\text{sf}} - p^y_{\text{nsf}}| \in (0.2, 0.25)$ m, vertical center of mass height $p^z_{\text{b}} > 0.9$, and swing foot clearance $p^z_{\text{nsf}}(\frac{T_{ss}}{2} + T_{ds}) > 0.09$ m.

6.1.4 Experimental Implementation

The estimation and control routines are deployed in Simulink Real-Time, and run on a real-time target machine on the robot. In order to adhere to the strict timing requirements of the system, we run the estimation and control routines with concurrently executed multithreading. The estimation routine runs at 500 Hz, while the control thread runs at 2 kHz. A block diagram of the software structure on the robot is shown in Fig. 6.4. To facilitate testing before actually running controllers on the physical hardware, a Simscape Multibody simulation of the robot provided by Agility Robotics[1] was modified to implement our control algorithm (pictured on the right in Fig. 6.1). This was then used to tune controller parameters before implementation which are directly used on the physical robot for performing the experiments.

In this section, the individual components to first the estimation and then feedback control components of the real-time implementation are discussed. The feedback con-

[1] https://github.com/agilityrobotics/agility-cassie-doc

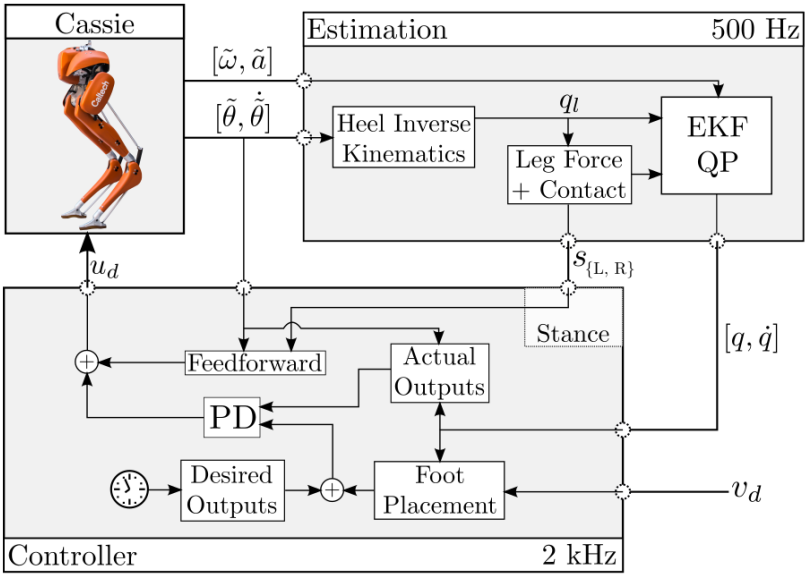

Figure 6.4: Control and estimation diagram for two-domain locomotion on Cassie. The estimation and controller blocks are separate threads running in parallel on the robot's real-time PC at 500 Hz and 2 kHz, respectively. The current controller domain is triggered via the $s_{\{\mathrm{L,\,R}\}}$ contact classifier.

trol method in this section is relatively simple, consisting of a heuristic feedforward term for gravity compensation, and a PD controller which can approximately zero the virtual constraints. In order to stabilize the walking from step-to-step, a foot placement routine was also used to augment the Bézier polynomial coefficients to change the footstrike locations. In the estimation thread, we first run inverse kinematics to solve for the heel spring deflections based on the encoder measurements and the full kinematics given in (4.19). This spring deflection is then used to obtain a quasi-static contact force estimate and then use this to classify contact. These first two components are common to this implementation and the gait library implementation in Sec. 6.2, and are discussed in [128]. At the time of this implementation, a separate velocity estimation scheme was used, which will be introduced next.

Estimation of floating base coordinates

Control for walking robots typically relies on knowledge of the full 6 DOF floating base pose and velocities. However, the proprioceptive sensing typically included on these robots cannot directly measure these states and they must be estimated. To do this, we choose a set of states which capture the floating base coordinates while providing implicit measurements through the full body kinematics. Specifically, the

estimator state[2] is chosen as $x = [R, p, v, b_a, b_\omega, c_i]^T$, where $p \in \mathbb{R}^3$ is the position of the CoM, v is its linear velocity, $R \in SO(3)$ is the rotation describing the orientation of the floating base in the world, $b_a \in \mathbb{R}^3$ is the accelerometer bias, $b_\omega \in \mathbb{R}^3$ is the gyroscope bias, and $c_i, \in \mathbb{R}^3$ is the i-th contact location. The estimator presented here is primarily drawn from [204], from which we combine measurements on the contact velocities.

The Cassie biped is equipped with 14 rotary encoders and a VectorNav VN-100 IMU, from which we will utilize the 3-axis accelerometer and gyroscope. The raw accelerometer and gyroscope data, $\tilde{a}$ and $\tilde{\omega}$, are subject to the additive noise w_a, w_ω as well as a random-walk bias with noise w_{ba}, $w_{b\omega}$. We then can obtain the expected values of acceleration and angular velocity at the center of mass as:

$$\begin{aligned} a &= R(\tilde{a} - b_a - w_a) + g \\ \omega &= \tilde{\omega} - b_\omega - w_\omega. \end{aligned}$$

The encoders provide access to the corresponding joint angle measurements $\tilde{\theta}$ and their velocities $\dot{\tilde{\theta}}$, which are used to compute implicit measurements of the leg kinematics. Specifically, we can obtain the position c_i and velocity $\dot{c}_i$ of the ith foot ($i = \{\text{L, R}\}$) as:

$$\begin{aligned} c_i &= p + R \cdot \left(f_{\text{FK},i}(\tilde{\theta})\right) - n_c \\ \dot{c}_i &= v + R \cdot \left(\omega^\times f_{\text{FK},i}(\tilde{\theta}) + J(\tilde{\theta})\dot{\tilde{\theta}}\right) - n_{\dot{c}} = 0, \end{aligned}$$

where $\omega^\times$ denotes the cross product matrix of the angular velocity and the noise $v = \left[n_c^T, n_{\dot{c}}^T\right]^T \sim \mathcal{N}(0, R)$. We do not consider the feet of the robot in the filter, and treat the ankle pivot as the contact. The discrete Gaussian noise terms n_c and $n_{\dot{c}}$ incorporate sources of noise, including measurement and modeling uncertainty.

The estimator dynamics is given by its position and orientation along with their associated velocities. We can track the state of the contact location to provide a relative location of the contact-to-floating base. If no contact is detected, the associated covariance is set to a large value. The discrete-time dynamics of the floating base are

[2]This should not be confused with the walking dynamics state x in (3.27).

given by:

$$\hat{x}_k^- = f_{\text{est}}(x_{k-1}^+) = \begin{pmatrix} R_{k-1}^+ \Lambda(\omega \Delta t) \\ p_{k-1}^+ + v_{k-1}^+ \Delta t + a\frac{1}{2}\Delta t^2 \\ v_{k-1}^+ + a\Delta t \\ b_{a,k-1}^+ + w_{k,ba} \\ b_{\omega,k-1}^+ + w_{k,b\omega} \\ c_{i,k-1}^+ + w_{k,c_i} \end{pmatrix}, \tag{6.12}$$

where $\Lambda(\omega \Delta t)$ is an incremental rotation matrix [204]:

$$\begin{aligned} \Lambda(\alpha) &:= \exp(\alpha^\times) \\ &= I + \frac{\sin(||\alpha||)\alpha^\times}{||\alpha||} + \frac{(1 - \cos(||\alpha||))(\alpha^\times)^2}{||\alpha||^2}, \end{aligned} \tag{6.13}$$

with $\|\cdot\|$ the Euclidean norm and $w_k \sim \mathcal{N}(0, Q)$. If the robot has established contact with the ground, we can then produce an implicit measurement of the foot position and velocity relative to the floating base through the forward kinematics. We then have the measurement and corresponding prediction models:

$$z_k = \begin{bmatrix} f_{FK,i}(\tilde{\theta}) \\ \tilde{\omega}^\times f_{\text{FK},i}(\tilde{\theta}) + J(\tilde{\theta})\dot{\tilde{\theta}} \end{bmatrix}, \; h(\hat{x}_k^-) = \begin{bmatrix} (R_k^-)^T(c_{i,k}^- - p_k^-) \\ -(R_k^-)^T v_k^- \end{bmatrix}.$$

The filter presented thus far utilizes additive noise on a constant foot contact prediction to allow for foot slippage during stance. However, there are certain scenarios in which we may want the state estimate to satisfy some physical constraints on contact during stance. Recent work on estimation for legged robots [205] exploits symmetries naturally present in the model to provide additional convergence guarantees. An estimation scheme using full-body dynamics with assumed knowledge of the contact surface has been used in a mixed integer Quadratic Program (QP) [206] for handling contacts, and others have looked at predicting covariance values for contact velocities through contact force [207]. In our work, we maintain that the estimate should satisfy the heuristic inequality:

$$\begin{bmatrix} -a/\bar{F}_i \\ -a/\bar{F}_i \\ -b/\bar{F}_i \end{bmatrix} \leq c_{i,k}^+ - c_{i,k-1}^+ \leq \begin{bmatrix} a/\bar{F}_i \\ a/\bar{F}_i \\ 0 \end{bmatrix}, \tag{6.14}$$

where a and b are positive tunable scalar values, and $\bar{F}_i = ||F_{\text{grf},i}||$. The primary function of this heuristic constraint is to disallow vertical positional drift in the contact

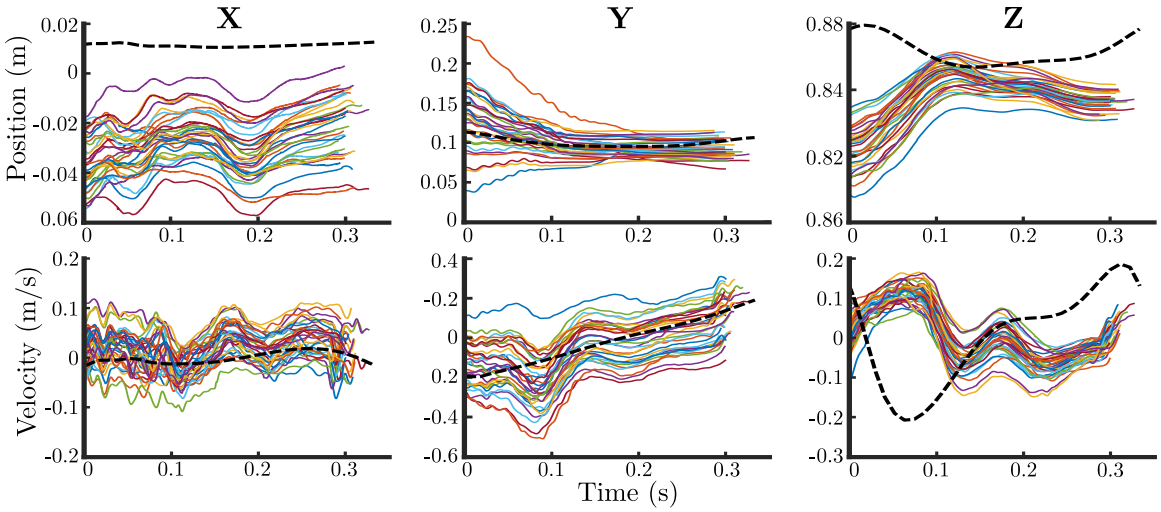

Figure 6.5: Center of mass positions (left three) and velocities (right three) with respect to the stance foot over 10 seconds of stepping in place on hardware (solid) versus the optimization result (dashed).

estimate. We proceed with the standard EKF recursion:

$$F_k = \left.\frac{\partial f_{\text{est}}}{\partial x}\right|_{x_{k-1}^+}, \; G_k = \left.\frac{\partial f_{\text{est}}}{\partial w}\right|_{x_{k-1}^+}, \; H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k^-}$$
$$P_k^- = F_k P_{k-1}^+ F_k^T + G_k Q_k G_k^T, \tag{6.15}$$

also computing $\hat{x}_k^-$ via (6.12) and $y_k = z_k - h(\hat{x}_k^-)$. The measurement update with the contact constraint can then be implemented as a QP with a linear heuristic admissibility constraint:

$$x_k^+ = \underset{x \in \mathbb{R}^{18}}{\text{argmin}} \;\; \|x - \hat{x}_k^-\|^2_{(P_k^-)^{-1}} + \|y_k - H_k(x - \hat{x}_k^-)\|^2_{(R_k)^{-1}}$$
$$\text{s.t.} \;\; \text{Contact constraint: } Eq.(6.14)$$

where $\|v\|_{\mathcal{A}} := \sqrt{v^T \mathcal{A} v}$ is the Mahalanobis norm, and the posteriori error covariance is updated as:

$$P_k^+ = P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^-. \tag{6.16}$$

The QP is solved using a custom MATLAB port of the static memory implementation of QPOASES [208], which is autocoded for implementation on hardware using Simulink Coder to allow for hotstarting. The resulting linear velocities will be used in the next section to stabilize the walking.

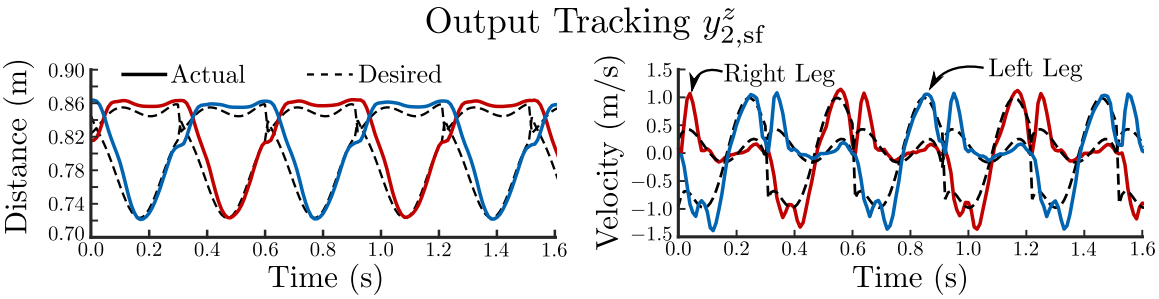

Figure 6.6: Experimental results for the use of PD control for tracking walking on hardware. Plotted is the vertical Cartesian position output tracking on Cassie while walking with the two-domain compliant HZD gait [128].

Virtual Constraint Feedback Controller: PD + Feedforward

The simplest control scheme for determining motor torques is Proportional-Derivative (PD) control [209]. The strongest argument for using this approach is the sheer simplicity in its implementation and the intuitive physical meaning with respect to tuning. As such, we formulate our virtual constraint tracking problem as a task-space PD controller with a gravity compensation term:

$$u = -Y(q)^{-1}\left[K_{\text{p}}(y^a_{v,2}(q) - y^d_{v,2}(t,\alpha)) + K_{\text{d}}(\dot{y}^a_{v,2}(q,\dot{q}) - \dot{y}^d_{v,2}(t,\alpha))\right] + \sum_{i\in\{R,L\}} s_i \bar{J}^T_{i,m} M g,$$

where $Y(q) = \partial y^a_{(v,2)}/\partial q$ is the output Jacobian, and $y^d_{v,2}(t,\alpha))$ indicates that in implementation, we are parameterizing each step using a time based phase variable that begins at the start of double-support and reaches 1 at the end of single-support, T_{ss}. The PD gain matrices are given by $K_{\text{p}}, K_{\text{d}}$, with their values provided in Table 6.1. A blending term is also used for the feedforward term, $s_i \in [0,1]$, such that $s_L + s_R = 1$, $\bar{J}_{i,m}$ are the rows of (4.21) for a given leg corresponding to the motors, $M = 33.32$ kg is the total mass of the robot, and $g = [0,\ 0,\ -9.81]^T$ is the gravitational constant. Note that s_i is used to transition the approximate gravity compensation to the alternating stance legs [83]. For underactuated dynamic walkers whose motions have been planned with virtual constraints, simply tracking the outputs with a well-tuned PD controller is sometimes sufficient to achieve walking on hardware [44], [113], [128], [210], [211], and even running [42]. This is because the outputs *implicitly* encode the dynamic behavior and stability constraints, even if it requires different torques on the actual robot. In addition, because dynamic behaviors are often rendered stable through this behavioral encoding while satisfying appropriate physical constraints, almost all passive dynamic and HZD walkers to date do not include load cells in the feet as feedback control of these quantities is not necessary for stability.

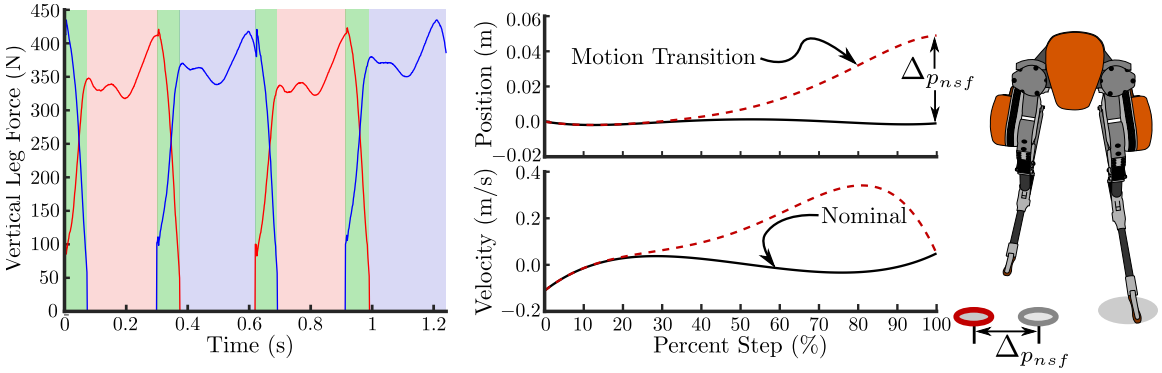

Figure 6.7: Vertical leg forces as measured on hardware over four steps of typical stepping in place for the left (red) and right (blue) legs. Contact classification is shown as shaded regions, with double-support in green.

Directly implementing this controller with trajectory obtained from the NLP (HZD Optimization) using time-based outputs, as in the PD controller in this section, can at best result in a marginally stable locomotion for experiments. Motivated by this, a discrete PD controller to augment the footstrike locations during locomotion is implemented as:

$$\Delta p_{nsf} = \tilde{K}_p(\bar{v}_k - v_{\text{ref}}) + \tilde{K}_d(\bar{v}_k - \bar{v}_{k-1}), \tag{6.17}$$

where the average velocity of the current step $\bar{v}_k$ and previous step $\bar{v}_{k-1}$ are computed directly from the floating base estimator. The reference velocity v_{ref} is obtained from the average velocity over the first half of the desired walking cycle, and can be perturbed to command forward or lateral velocities to the robot. This regulator-type controller is largely inspired by early work of [12], and has been successfully implemented on similar legged systems [83], [211]. In addition, because the output values are computed based on a Bézier polynomial, the update value Δp_{nsf} can directly augment the last two parameters of the corresponding output polynomials. We employ a motion transition method [212] to update the trajectory which results in a smooth tracking and preservation of the desired impact velocity.

Table 6.1: Controller parameters used on the physical robot.

Parameter	Value
K_P	$\{600;\ 600;\ 600;\ 500;\ 600;\ 600;\ 600;\ 500;\ 100\}$
K_D	$\{10;\ 10;\ 10;\ 5;\ 10;\ 10;\ 10;\ 5;\ 4\}$
$\tilde{K}_p$	0.35
$\tilde{K}_d$	0.07

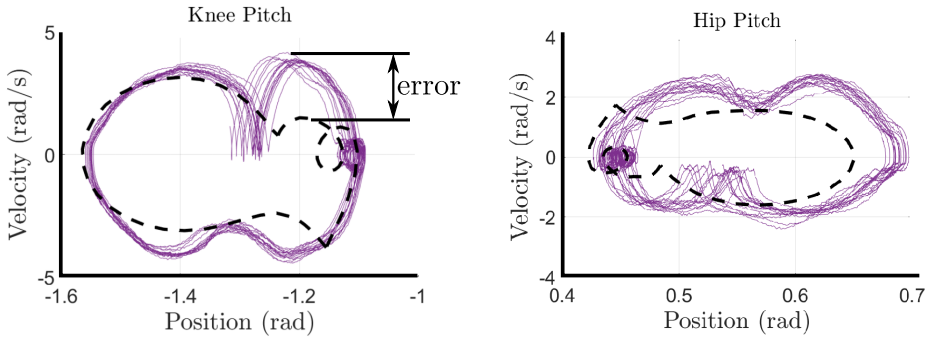

Figure 6.8: Limit cycles for the right knee and hip over 10 seconds of stepping in place on hardware (solid) versus the nominal cycle (dashed).

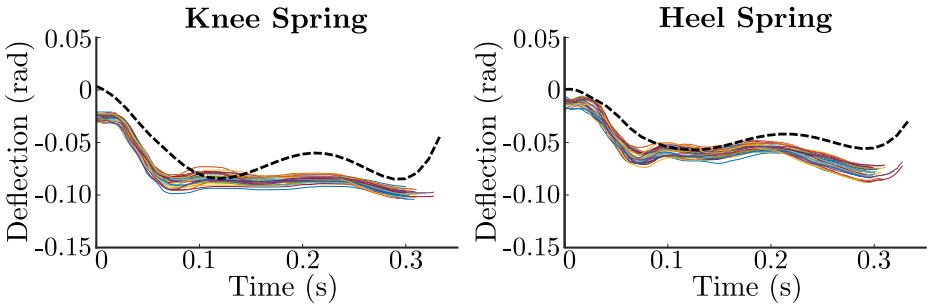

Figure 6.9: Deflections of the stance knee and heel springs over 10 seconds of walking on hardware (solid) versus the optimization result (dashed).

6.1.5 Results and Conclusions

The controller and algorithms presented in this section were implemented on Cassie experimentally, with the result being stable walking on hardware. In Fig. 6.9, we compared the spring deflection between the actual experiments and designed behavior from optimization. The gait design accurately captured the robotic compliance, with the knee and heel spring deflections on physical hardware matching closely with the nominal result. Because this matched closely to the planned compliance, minimal tuning was then required to implement a simple output tracking PD controller. The COM kinematics are shown in Fig. 6.5, with the primary difference appearing in the vertical direction, likely due to the gravity compensation pushing on the ground inconsistent with the designed motions. Additionally, limit cycles for the knee and hip pitch joints are shown in Fig. 6.8 to illustrate stable walking.

The Cassie biped poses a unique challenge due to its compliant mechanism and highly underactuated nature of the dynamics. In order to leverage these components in experiments, we constructed a hybrid model for walking dynamics based on a rigid model (simple model) and compliant model (full model). A comparison of these

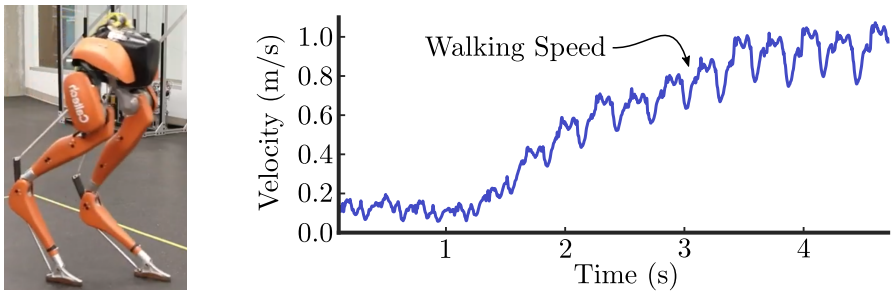

Figure 6.10: Cassie walking at 1m/s in the AMBER Lab using the two-domain gait.

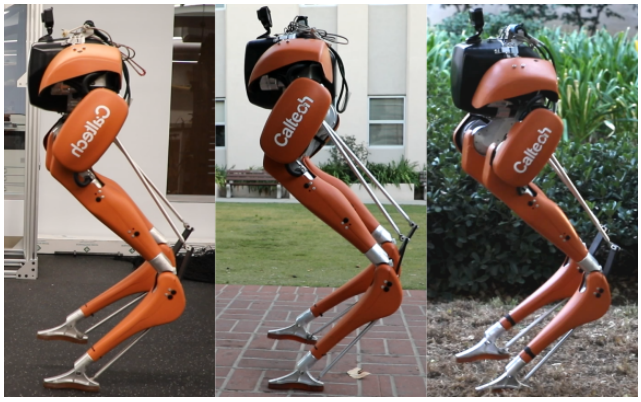

Figure 6.11: Cassie walking indoors, outdoors, and on rough terrain.

two models with regards to computation performance and simulation suggested two directions: ignoring the compliance and designing controllers which are robust to the mismatch, and using a more complex model which designs locomotive behaviors encoding the compliant behavior. We then posed an optimization problem to design gaits for the 22 DOF compliant robot and present an algorithmic approach to estimate and control the hardware. The result is that Cassie walks with experiment-level robustness in various environments: indoor and outdoor (see snapshots in Fig. 6.11 and the associated video[3]). The two-domain walking was later shown to be capable of walking at speeds exceeding 1 m/s, shown in Fig. 6.10 and as a video[4].

These preliminary results demonstrate that the HZD trajectory optimization can accurately model and synthesize closed-loop control plans for the underactuated and compliant model of Cassie outlined in Sec. 4.2. The optimization which was presented, and the successful algorithmic components of the experimental implementation, will be used in later sections to obtain additional feedback control parameters which enable the use of model-based and optimization-based real-time control development.

[3]Experimental two-domain compliant walking on Cassie: `https://youtu.be/WQupor989dc`
[4]Cassie walking at 1m/s: `https://youtu.be/CgKEnQvKaKY`

6.2 Compliant Motion Library for Walking

6.2.1 Motivation

In the previous experimental study in Sec. 6.1, it was shown that the HZD methodology can accurately capture the passive dynamics of the high-DOF and underactuated model of Cassie which was introduced in Sec. 4.2. With the aim of creating a set of tools which can enable a wider range of walking motions on Cassie, this section will develop a *motion library* of compliant walking gaits over a range of speeds in both the coronal and sagittal directions. While the single gait introduced in Sec. 6.1 was able to locomote over a range of speeds, there was still a significant tracking error at times. This is largely due to the fact that as the walking deviated further from 0 m/s in average speed, the walking relied more heavily on heuristics for stability.

The first contribution of this section is a motion library of walking behaviors for Cassie that leverage its full-body dynamics including its compliance. While motion libraries for sagittal motions under the assumption of sufficient rigidity have been successfully realized on Cassie [127] and on other robots [213], all of the existing literature within the HZD field ignores the robot's inherent compliance. This work, therefore, is the *first to combine motion libraries that consider the full-body dynamics* of Cassie together *with compliance*. To generate this library, we utilize the framework of Hybrid Zero Dynamics (HZD) [30] which has demonstrated success in developing controllers for highly underactuated walking behaviors, as discussed extensively in Chap. 5 and Sec. 6.1. This section is also the first example of using an HZD optimization to *extract additional information on the closed-loop plan* generated offline, *including generalized*

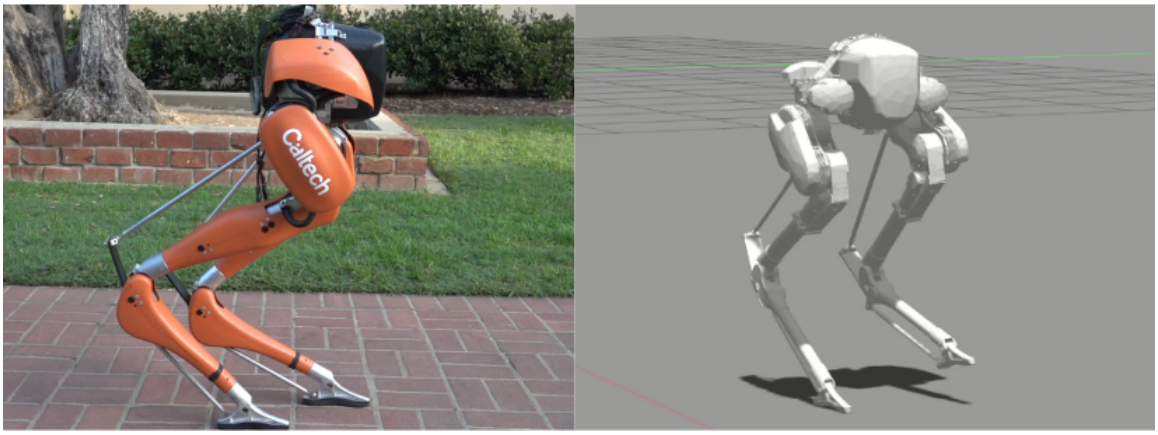

Figure 6.12: Cassie walking outdoors and in a Gazebo environment, using an identical inverse dynamics controller built on the parameterized results from a motion library of compliant walking gaits.

accelerations and velocity profiles. While this is common practice in many robotic applications, literature on real-time control development in the HZD walking field has only extracted Bézier polynomial coefficients [30], [44], [127], [154], and in some cases the torques [41], ignoring these terms and opting for high-gain output feedback of the output polynomials.

The second contribution of this section is the *synthesis of model-based controllers able to realize motion libraries* in a manner that *exploits the compliance and underactuated dynamics* for which they were generated. One of the limiting assumptions which can affect HZD behaviors is that their formal stability guarantees are tied to exponential tracking of motion objectives or outputs. This can lead to the use of high-gain PD feedback controllers when applied to hardware to obtain periodic stability. Instead, it would be more desirable to achieve tracking with some inherent control compliance to perturbations and unknown terrain. In this direction, model-based control in the form of inverse dynamics [143], [214], control Lyapunov functions [31], or some combination of the two [186] may offer more desirable control properties. However, implementing model-based planning and control methods on physical systems is typically non-trivial due to the inherent model inaccuracy, dynamically changing contact constraints, and possibly conflicting objectives for the robot which naturally arise in locomotion. In this work, we consider an analytical solution to the floating-base inverse dynamics problem, extending the work in [142]. It is shown how we can use the additional information extracted from the optimization to generate feedforward torques that respect the active constraints on the robot. The *end result is a model-based controller* that is able to *leverage the compliance* of Cassie *for all motion primitives* in the library. This is demonstrated experimentally on Cassie, with the result being stable walking on hardware.

6.2.2 Locomotion Model

A bipedal walking gait consists of one or more different continuous phases followed by discrete events that transition from one phase to another, motivating the use of a hybrid system formulation with a specific ordering of phases. This is traditionally described as a walking cycle, which is a directed cycle with a sequence of continuous domains (continuous dynamics) and edges (changes in contact conditions). We structured the dynamics of walking on Cassie in a multi-domain and hybrid fashion.

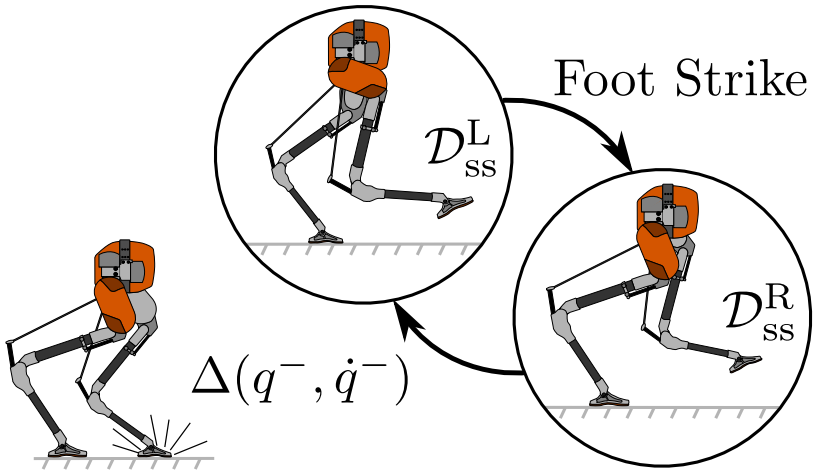

Figure 6.13: The directed graph of walking used in this paper is shown, where we view walking on Cassie as consisting of two single-support domains with a compliant stance leg, and rigidly stiff swing leg.

A directed cycle, $\Gamma = (V, E)$, is specified for the system:

$$\begin{aligned} V &= \{ss^{\mathrm{R}}, ss^{\mathrm{L}}\}, \\ E &= \{ss^{\mathrm{R}} \to ss^{\mathrm{L}}, ss^{\mathrm{L}} \to ss^{\mathrm{R}}\}, \end{aligned} \tag{6.18}$$

where each vertex represents a continuous domain and each edge corresponds to a transition between these domains, as shown in Fig. 6.13. Specifically, walking on Cassie in this section is considered as a period-two cycle of alternating *single-support*, which are connected by the state dependent event of impact. This means that as opposed to Sec. 6.1, the *double-support* domain, $\mathcal{D}_{\mathrm{ds}}$, here is *instantaneous*. While the inclusion of a double-support domain is the most physically accurate representation of the locomotion on Cassie, it adds a significant amount of additional variables and cardinal nodes to the optimization problem and makes the feedback control approaches we will later derive unnecessarily complex for this study. Instead, the assumption of a rigid swing leg allows for us to develop gaits that ignore this additional domain, as we can plan walking which lifts the leg instantaneously on impact.

As previously stated, the walking is considered to be asymmetric, or period-two, meaning that in optimization and in control development, we consider the right and left stance phases as distinctly different. This is primarily done to allow for explicit modeling of lateral walking gaits, which cannot be represented by a symmetric motion.

Holonomic Constraints

It is common practice to model robotic manipulators in branched tree structures. However, on Cassie, a compliant multi-link mechanism is used to transfer power from higher to lower limbs without allocating the actuators' major weight onto the lower limbs, and effectively acts as a pair of springy legs [84]. When the mechanism has a kinematic loop, this is often managed by cutting the loop at one of the joints and enforcing a holonomic constraint at the connection to form the closed-chain manipulator. In the Cassie leg, the heel spring is attached to the rear of the tarsus linkage, with its end constrained via a pushrod affixed to the hip pitch linkage. Also recall from the Cassie model development in Sec. 4.2 that we allow for only one mode of contact for each foot, corresponding to line contact representing the robot pivoting along the roll direction of the foot. In accordance with single-support indicating either one or both feet in contact with the ground, we define the holonomic constraints as:

$$\eta_{ss}^{\mathrm{R}}(q) := (p_{rf}, \varphi_{rf}^{yz}, \eta_{4\mathrm{bar}}, q_{lsp}, q_{lhs}) \in \mathbb{R}^9, \tag{6.19}$$

$$\eta_{ss}^{\mathrm{L}}(q) := (p_{lf}, \varphi_{lf}^{yz}, \eta_{4\mathrm{bar}}, q_{rsp}, q_{rhs}) \in \mathbb{R}^9, \tag{6.20}$$

where $p_{\square f}$ and $q_{\square sp}, q_{\square hs}$ are the positions of the stance foot (center) and the swing springs, pictured in Fig. 4.7. The closure of the compliant multi-bar leg is enforced through the holonomic constraint (4.18), with the swing leg assumed to be sufficiently rigid to enforce the springs as an additional holonomic constraint. This rigid assumption on the swing leg simplifies both the optimization and control implementations [41], and *makes the dynamics less numerically stiff.*

Domains of Admissibility

The continuous domains of walking are determined by the limiting conditions on the ground reaction wrenches and unilateral constraints, as was introduced in Chap. 3, and outlined in the associated discussion on constraint feasibility associated with holonomic foot constraints in Sec. 2.1.3. Here, the domains of admissibility for each single-support domain follows the derivation already presented in Sec. 6.1:

$$A_{ss}^{\{\mathrm{R,L}\}}(q, \dot{q}, u) = \begin{bmatrix} \lambda_{sf}^{z} \\ \frac{\mu}{\sqrt{2}}\lambda_{sf}^{z} - \{|\lambda_{sf}^{x}|, |\lambda_{sf}^{y}|\} \\ \frac{w_f}{2}\lambda_{sf}^{z} - \{|\lambda_{sf}^{my}|\} \\ p_{nsf}^{z} \end{bmatrix} \geq 0, \tag{6.21}$$

where p^z_{nsf} indicates that the non-stance foot must not go below the ground for the domain to be valid and l_f is defined for the contact geometry presented in Fig. 4.7.

Guards and Reset Maps

We follow the definitions given in (3.6) to prescribe the guards for each domain transition:

$$H_{\mathrm{ss^R}\to\mathrm{ss^L}}(q,\dot{q},u) := p^z_{lf}(q), \tag{6.22}$$

$$H_{\mathrm{ss^L}\to\mathrm{ss^R}}(q,\dot{q},u) := p^z_{rf}(q), \tag{6.23}$$

where it can be seen that for each guard, we apply the relabeling matrix associated with the asymmetric walking assumption given in (4.8), which does not swap the left and right legs, but instead zeroes the compliance in the post-impact swing leg. These guards are also associated with an impact, and will enforce stiction of the new contact point and update the velocity of the tarsus to satisfy the rigid post-impact velocity through the instantaneous impact equation derived in (2.27). The associated reset map, Δ, is thus given as the update [109], [138]:

$$\Delta(q^-,\dot{q}^-) := \begin{bmatrix} q^+ \\ \dot{q}^+ \end{bmatrix} = \begin{bmatrix} \mathcal{R}_{\mathrm{asym}}(q^-) \\ \dot{\mathcal{R}}_{\mathrm{asym}}(q^-, \Delta^{\dot{q}}(q^-)\dot{q}^-) \end{bmatrix}, \tag{6.24}$$

where q^- and q^+ denote the pre- and post-impact configuration, $\mathcal{R}_{\mathrm{asym}}(q^-)$ the spring reset matrix (4.8), $\Delta^{\dot{q}}(q)$ is obtained from the plastic impact equation derived in (2.27), and $\dot{\mathcal{R}}_{\mathrm{asym}}(q^-, \Delta^{\dot{q}}(q^-)\dot{q}^-)$ applies the velocity reset matrix to correct the tarsus velocity (4.10). As in [41] and shown in Sec. 4.2, the full transition map thus accounts for the assumption that the spring on the swing leg is at its rest deflection, and is otherwise identity as the positions remain constant when transitioning domains.

Virtual Constraint Selection

Since the Cassie robot was designed to behave mechanically similar to a SLIP, we define outputs based on the undeflected leg length and leg angles, rather than Cartesian positions as was done in Sec. 6.1. One of the benefits of this approach is that the expressions are far simpler, and can be easily decomposed into their respective pitch and roll directions when using the orientation of the floating base, rather than as a function involving a rotation matrix. A 6th-order Bézier polynomial is chosen for the desired outputs, for which α is a matrix of real coefficients that parameterize the curve according to (3.12). We then define all of the relative degree 2 actual outputs

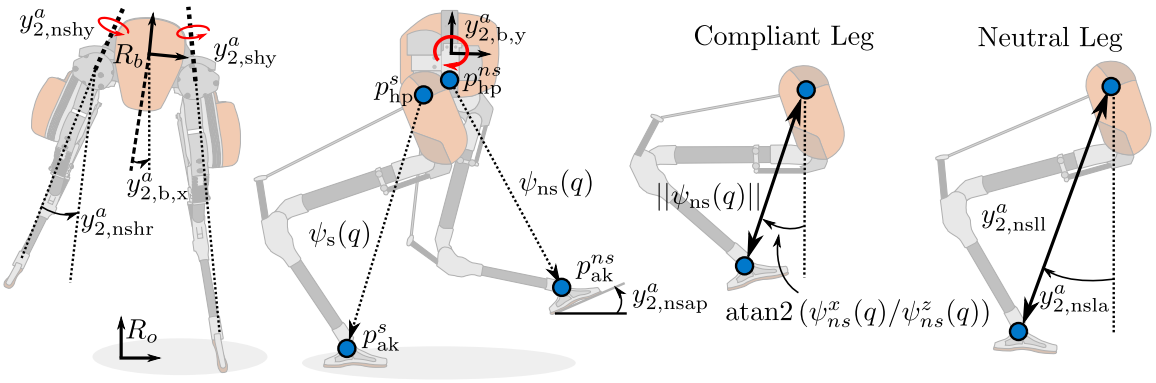

Figure 6.14: A visualization of the outputs which can be selected for Cassie in this section. Pictured are the definitions for the average center of mass location and Cartesian foot locations. On the right is an illustration of the difference between the actual leg length and angle versus the neutral leg configuration, for which the neutral (meaning undeflected) positions are used as actual outputs.

which are made available to walking on Cassie as:

$$
\begin{aligned}
y^a_{2,\text{b},x} &= \varphi^x && \text{(pelvis roll)}\\
y^a_{2,\text{b},y} &= \varphi^y && \text{(pelvis pitch)}\\
y^a_{2,\text{sll}} &= ||\psi_s(\bar{q}_l)||_2 && \text{(stance leg length)}\\
y^a_{2,\text{nsll}} &= ||\psi_{ns}(\bar{q}_l)||_2 && \text{(swing leg length)}\\
y^a_{2,\text{nsla}} &= \text{atan2}\,(\psi^x_{ns}(\bar{q}_l)/\psi^z_{ns}(\bar{q}_l)) && \text{(swing leg pitch)}\\
y^a_{2,\text{nshr}} &= q_{\text{nshr}} && \text{(swing hip roll)}\\
y^a_{2,\text{shy}} &= q_{\text{shy}} && \text{(stance hip yaw)}\\
y^a_{2,\text{nshy}} &= q_{\text{nshy}} && \text{(swing hip yaw)}\\
y^a_{2,\text{nsap}} &= \varphi^y(q_\text{b}, \bar{q}_l) && \text{(swing foot pitch)}
\end{aligned}
$$

where $\varphi^y(\theta_{\text{tp}})$ is the swing foot pitch angle and,

$$\psi^{\{s,ns\}}(q) = p^{\{s,ns\}}_{\text{hp}}(q) - p^{\{s,ns\}}_{\text{ak}}(q), \tag{6.25}$$

is the expression for the distance between the hip pitch and ankle pitch joints from the forward kinematics, and because the foot geometry is quite small, we leave the stance foot passive and do not define any outputs using this coordinate. Because we use the neutral leg length, we can remove the tarsus and spring coordinates from the expressions using the substitution rule $\bar{q}_l \in \{\mathcal{Q}_l \mid q_{\text{sp}} = 0, q_{\text{hs}} = 0, q_{\text{t}} = 13^o - q_{\text{k}}\}$,

leaving us with a concise set of expressions for the leg length and leg angle outputs:

$$y^a_{2,\mathrm{ll}} = 0.727226\sqrt{1.002271 + \cos(q_\mathrm{k}) - 0.0352921\sin(q_\mathrm{k})} \tag{6.26}$$

$$y^a_{2,\mathrm{nsla}} = \mathrm{atan2}\left(\frac{-0.052805(\cos(q_\mathrm{hp}) + 1.276576\cos(q_\mathrm{hp}+q_k)}{0.526541(\cos(q_\mathrm{hp}) + 0.940933\cos(q_\mathrm{hp}+q_\mathrm{k})} \cdots \right.$$
$$\left. \cdots \frac{+9.971365\sin(q_\mathrm{hp}) + 9.38239\sin(q_\mathrm{hp}+q_k))}{-0.100287\sin(q_\mathrm{hp}) - 0.128024\sin(q_\mathrm{hp}+q_\mathrm{k}))}\right). \tag{6.27}$$

The full geometry of the relevant expressions, along with the output definitions, are illustrated in Fig. 6.14. By formulating the outputs in this way, the passive dynamics of the system will contain the additional dynamics associated with the compliant elements [41]. As a practical matter, this is also important as directly controlling the compliance in the leg is a significantly more difficult problem to achieve. It should be noted that while the undeflected leg outputs are computed with the rough approximation given by $q_\mathrm{t} = 13^o - q_\mathrm{k}$ rather than the true neutral configuration of (4.7), the compliant passive dynamics will still be encoded for the full-order model of the compliant leg structure given in (4.18) through the optimization. The result is to have a simple set of control objectives for derivation of feedback controllers, while allowing the nonlinear optimization framework to do the more complex work of embedding the spring dynamics.

The position-modulating outputs selected for control in each domain are:

$$\mathscr{O}_{\mathrm{ss}}^{\{\mathrm{R,L}\}} = \{y_{2,\mathrm{b},x}, y_{2,\mathrm{b},y}, y_{2,\mathrm{sll}}, y_{2,\mathrm{nsll}}, y_{2,\mathrm{nsla}}, y_{2,\mathrm{nshr}}, y_{2,\mathrm{shy}}, y_{2,\mathrm{nshy}}, y_{2,\mathrm{nsap}}\}, \tag{6.28}$$

for which the outputs are identical in the respective left or right stance domain, with the expressions simply having the appropriate left or right leg coordinates swapped.

6.2.3 Gait Library Optimization

This section details the trajectory optimization approach used to design a collection of trajectories which can be implemented to obtain compliant walking behaviors on the physical system in both the sagittal and coronal directions. While each HZD optimization problem determines one stable walking orbit, it has been shown that one can expand the range of motions a robot can perform through systematic optimization to build libraries of walking parameters [211]. Reinforcement learning has also been used to handle robust transitions for different speeds or unknown terrain height disturbances [213].

In order to obtain stable walking gaits for individual speeds, the desired evolution of the outputs (3.7) must be designed such that we can satisfy hybrid invariance with

a *periodic orbit repeating every two steps*. This is also combined with the physical limitations of the hardware. A nonlinear trajectory optimization problem is formed to solve this problem. Building off of the design of single walking gaits in Chap. 5 and Sec. 6.1, we would now like to design a variety of walking speeds for which the robot can operate [127], [215]. To accomplish this, a library of walking gaits at sagittal speeds of $v_x \in [-0.6, 1.2]$ m/s and coronal speeds of $v_y \in [-0.4, 0.4]$ m/s are generated in a grid of 0.1 m/s intervals, resulting in 171 individual optimization problems to be solved. The impact-to-impact duration of each step is fixed at 0.4 seconds for ease of implementation.

Each hybrid optimization was performed over the two domains, $\mathcal{D}_{\text{SS}}^{\{\text{R,L}\}}$, with a constraint imposed such that when the discrete impact (2.29) is applied to the terminal state of each domain enforcing satisfaction of the hybrid invariance condition (3.30), and that there is a periodic orbit repeating from right-impact to right-impact. It is critical that the motions respect the limitations of the physical system. In order to address this constraints for the friction cone, actuator limits, and joint limits are imposed following the implementation in Sec. 3.3.2 and the admissibility constraints outlined earlier in this section (6.21). While these constraints alone ensure invariance and satisfaction of the physical constraints, additional constraints on the behavior were tuned for implementation on hardware, such as the swing foot velocity, impact foot configuration, step symmetry when $v_y = 0$, and others outlined in Table 6.2.

In order to minimize torque and to center the floating base orientation movement around the origin, the following cost function was minimized:

$$\mathcal{J}(\mathbf{w}) := \int_{t=0}^{tf} \left(c_u |u|^2 + c_\varphi |\varphi_b|^2 \right) dt, \tag{6.29}$$

where $\mathbf{w} \in \mathbb{R}^{N_w}$ with N_w being the total number of optimization variables and $c_\square$ are

Table 6.2: Optimization constraints and parameters

Step duration	$= 0.4$	sec
Average step velocity, $\bar{v}_{x,y}$	$= v_{x,y}$	m/s
Pelvis height, p_z	≥ 0.80	m
Mid-step foot clearance, p^z_{nsf}	≥ 0.14	m
Vertical impact velocity, $\dot{p}^z_{sw}$	$\in (-0.40, -0.10)$	m/s
Step width, $p^y_{lf} - p^y_{lf}$	$\in (0.14, 0.35)$	m
Swing foot pitch, $\varphi^y(q)$	$= 0$	rad
Friction cone, μ	< 0.6	

weights applied to the respective terms. For the library presented in this work, the weights used were $c_u = 0.0001$ and $c_\varphi = (20, 1, 30)$.

An optimization package, FROST [129], was used to transcribe the constraints and cost of each of the 171 gaits into a nonlinear programming (NLP) problem that is passed to the nonlinear optimization solver, IPOPT [203]. Each optimization was then solved through the C-FROST interface [164] on a laptop with an Intel Core i7-6820 HQ CPU @ 2.7 GHz with 16 GB RAM, and consisted of 8418 variables with 4502 equality and 5880 inequality constraints. Using each gait as an initial guess to warm-start the next speed in the library, the average number of iterations per run was 199 with an average total evaluation time of 263.8 seconds, and an average objective value of $\mathcal{J}(\mathbf{w}^*) = 4.12$.

6.2.4 Parameterizing Optimization Results

The controller implemented on hardware needs both the feedback control objectives, defined by $y^d(\alpha, t)$, and acceleration information $\ddot{q}^*$ from the optimal path to complete the planned motions. The feedback parameters, α, already concisely parameterize the feedback control, and are placed in a large matrix which can be used in a bilinear interpolation routine. A subset of the resulting library of output parameters are shown in Fig. 6.15, where the leg length, leg pitch, and hip roll outputs are visualized over various walking speeds alongside a corresponding motion in simulation. An interesting characteristic which can be immediately seen is that the stance leg length

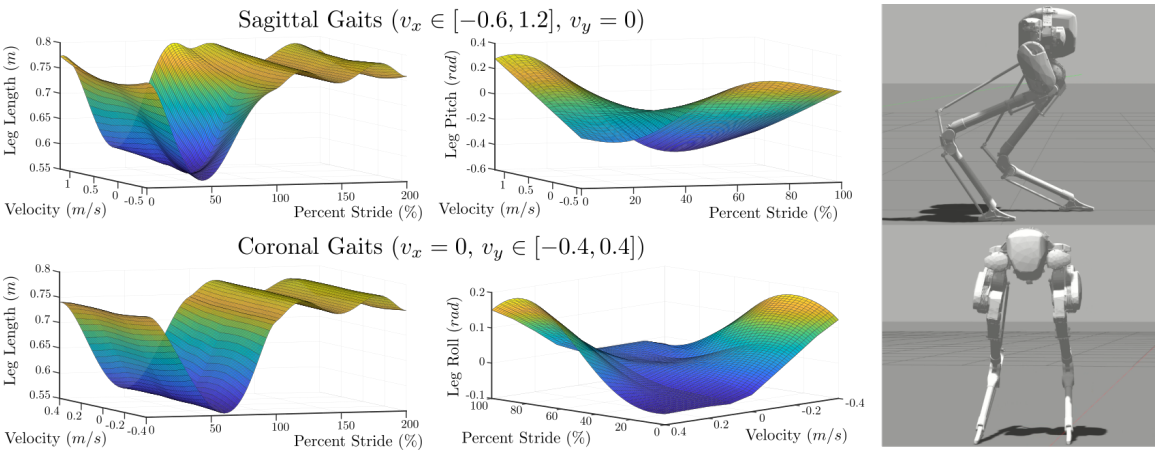

Figure 6.15: Contour plots of the swing leg length, leg angle, and leg roll outputs over the library speeds in the sagittal and coronal directions, showing the forward and reverse sweep of the leg as it tracks the motions. Also shown is the corresponding motion executed in a Gazebo simulation.

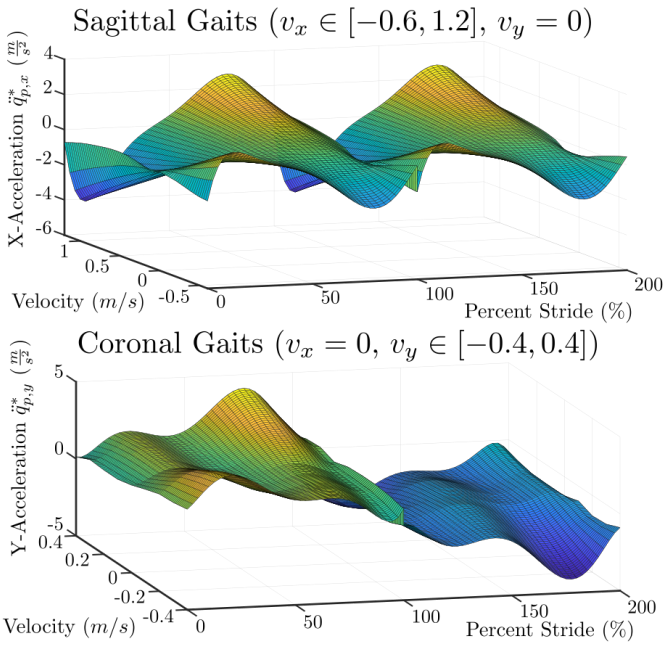

Figure 6.16: The contours of the floating base x and y accelerations which are obtained from the trajectory optimization problem. These contribute to the control action of a feedforward controller which is dynamically consistent with the contact constraints of the underactuated robotic walking.

output has a 'double-hump' shape, which is a result of the planned motions taking advantage of the spring deflection in the leg to minimize torque (6.29). This will be further exploited through the ground reaction forces in the optimization-based control strategy presented in Chap. 8.

Generalized accelerations $\ddot{q}^*$ are also extracted directly from the optimization variables, $\mathbf{w}$, as time-series data from each step. To allow for easier implementation, regression is performed on each curve to obtain the parameters for a 6th order Bézier polynomial (i.e. $M = 5$ in (3.12)), corresponding to the same order as the output polynomials. They can then be stacked with the α parameters in the same bilinear interpolation routine for code efficiency. Plots of the accelerations for the floating base x and y coordinates are visualized in Fig. 6.16. Finally, the floating-base position $p^*_{x,y}$ and velocity $v^*_{x,y}$ relative to the stance foot is also extracted in the same manner as $\ddot{q}^*$. This is shown on the velocity plots in Fig. 6.18 as the dashed lines and serves as a reference velocity for regulation.

To begin the interpolation process, we collect the target average speeds into a sorted,

searchable matrix:

$$\bar{v} := \begin{bmatrix} v^{(1,1)} & v^{(1,2)} & \dots & v^{(1,n_{vy})} \\ v^{(2,1)} & v^{(2,2)} & \dots & v^{(2,n_{vy})} \\ \vdots & \vdots & \ddots & \vdots \\ v^{(n_{vx},1)} & v^{(n_{vx},2)} & \dots & v^{(n_{vx},n_{vy})} \end{bmatrix} \tag{6.30}$$

where each row and column is sorted in ascending order from lowest speed to highest in the associated x or y direction, and each gait has been optimized on an evenly spaced grid of speeds. Given a desired speed, we can then search this matrix for the indices corresponding to the adjacent values to the target.

The bilinear interpolation routine is built on the assumption that each parameterized variable from the optimization is a rectangular matrix, $\alpha_\square \in \mathbb{R}^{M+1,n_{\alpha_\square}}$, where $n_{\alpha_\square}$ is the dimension of the variable in question (i.e. $n_{\alpha_{\ddot{q}}} = 22$ for Cassie as $n = 22$). In order to make the problem easier to manage, we first flatten each parameter matrix, and then concatenate them into a single parameter array:

$$\beta^{(i,j)} := (\bar{\alpha}_y, \bar{\alpha}_p, \bar{\alpha}_v, \bar{\alpha}_{\ddot{q}}), \tag{6.31}$$

where (i, j) corresponds to an index which matches its place in the velocity array (6.30). Each flattened array is then organized into a matrix of arrays:

$$\mathcal{B} := \begin{bmatrix} \beta^{(1,1)} & \beta^{(1,2)} & \dots & \beta^{(1,n_{vy})} \\ \beta^{(2,1)} & \beta^{(2,2)} & \dots & \beta^{(2,n_{vy})} \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{(n_{vx},1)} & \beta^{(n_{vx},2)} & \dots & \beta^{(n_{vx},n_{vy})} \end{bmatrix}. \tag{6.32}$$

In order to find the set of gait parameters for a given speed, we then need to perform the interpolation. Specifically, we first interpolate in the sagittal direction using the indices found in our prior search for the desired v_x and x_y place within (6.30):

$$p(v_x, v_y^{(j)}) = \frac{v_x^{(i+1)} - v_x}{v_x^{(i+1)} - v_x^{(i)}} \beta^{(i,j)} + \frac{v_x - v_x^{(i)}}{v_x^{(i+1)} - v_x^{(i)}} \beta^{(i+1,j)} \tag{6.33}$$

$$p(v_x, v_y^{(j+1)}) = \frac{v_x^{(i+1)} - v_x}{v_x^{(i+1)} - v_x^{(i)}} \beta^{(i,j+1)} + \frac{v_x - v_x^{(i)}}{v_x^{(i+1)} - v_x^{(i)}} \beta^{(i+1,j+1)}, \tag{6.34}$$

where we can then also interpolate in the coronal direction to obtain the desired interpolated parameters:

$$p(v_x, v_y) = \frac{v_y^{(j+1)} - v_y}{v_y^{(j+1)} - v_y^{(j)}} p(v_x, v_y^{(j)}) + \frac{v_y - v_y^{(j)}}{v_y^{(j+1)} - v_y^{(j)}} p(v_x, v_y^{(j+1)}). \tag{6.35}$$

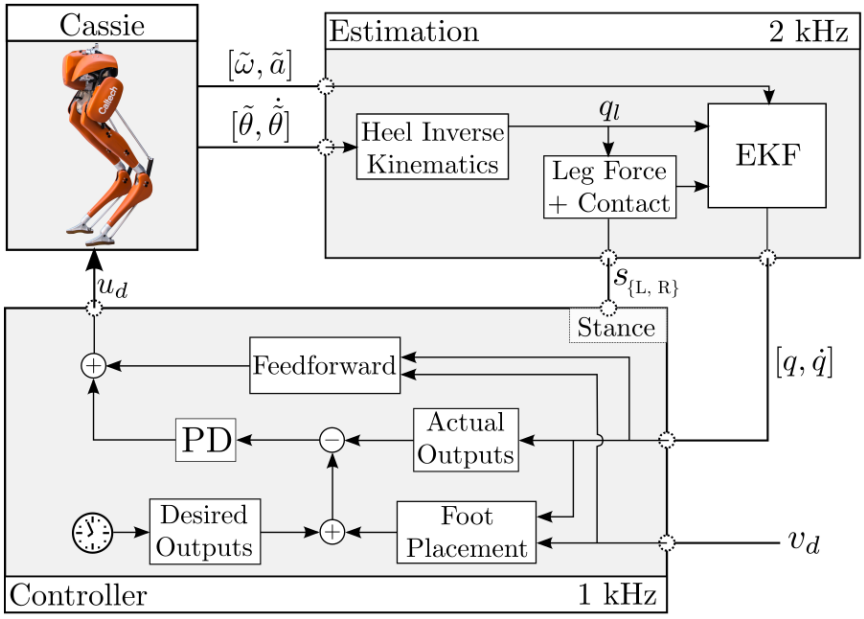

Figure 6.17: The control and estimation diagram for implementation of the compliant gait library on Cassie. The estimation and controller ROS nodes run in separate threads on dedicated cores of the Intel NUC CPU at 2 kHz and 1 kHz, respectively.

6.2.5 Experimental Implementation

The software infrastructure for feedback control and estimation used on hardware were significantly upgraded in this implementation compared to the preliminary result in Sec. 6.1. Specifically, the control code is implemented in C++ on the Intel NUC computer which comes installed in the Cassie torso, on which we added a PREEMPT_RT kernel. The software runs on two ROS nodes: one which communicates to the Simulink Real-Time xPC over UDP to send torques and receive sensor data and to perform estimation, and a second which runs the controllers. Each node is given a separate core on the CPU, and is elevated to real-time priority. The first node runs at 2 kHz and executes contact classification, inverse kinematics to obtain the heel spring deflection, and an EKF for velocity estimation [128], [204]. The second node runs at 1 kHz and receives the estimation and proprioceptive data over ROS. It then runs either the standing controller presented in [186], or executes the inverse dynamics controllers which will be presented and heuristic feedback control elements before communicating the commanded torque over ROS. The overall system architecture for the control and estimation is illustrated in Fig. 6.17.

In this section, the primary components necessary for achieving stable walking on hardware are outlined. This begins with the introduction and derivation of an inverse dynamics controller that draws generalized accelerations from Sec. 6.2.4 (shown in

Fig. 6.16 for the floating-base coordinates). This control approach generates an accurate control torque which satisfies the holonomic constraints imposed on the physical hardware at any given time, and approximately zeroes the virtual constraints. While this inverse dynamics controller effectively forms a feedforward controller, we layer a low-gain PD controller in order to approximately stabilize the zero dynamics. In order to change the footstrike location from step-to-step if the robot is perturbed, a PD controller which is equivalent to the one presented in Sec. 6.1 is introduced, with some additional calculations required due to our change in virtual constraints. The final feedback control component is to add a small amount of PID feedback to the generalized acceleration on the floating base. These augmented accelerations incentivize the inverse dynamics to choose a torque which propels the robot in a desired direction of motion. The estimation routines remain largely unchanged from Sec. 6.1, with the primary difference being that they can now run at 2 kHz due to their implementation in C++ rather than Simulink Coder.

Feedback Control Development: Inverse Dynamics + PD

The HZD optimization framework and subsequent library of walking behaviors creates a continuum of walking trajectories [213], which can be interpolated to obtain walking motions at a range of speeds. In the previous HZD development on Cassie, shown in Sec. 6.1 [128], PD control achieved reasonable tracking of the desired outputs with the passive compliance in the system matching the planned response. However, implementation relied on high-gain PD control with heuristic feedforward terms and trajectory modifications that needed to be hand-tuned to account for the model mismatch. A similar approach was used also by others [127], which demonstrated walking without planning for compliance or velocity terms in the dynamics. A benefit to using inverse dynamics control approaches on robotic systems is that lower gain PD feedback control can be used, while feedforward terms which respect the constrained rigid body dynamics of the system are used to produce most of the control input.

The use of inverse dynamics control for underactuated and floating-base robots is significantly more complex than for fixed-base manipulators. In developing controllers for these systems, there are many considerations to address such as numerical problems due to repeated matrix inversions of the inertia matrix, contact force distribution, and computational efficiency [186]. Here, we follow the approach of Mistry [142], which uses an orthogonal decomposition to compute the inverse dynamics torques in the null-space of the constraints.

The underlying idea of the method is to use a QR decomposition of the constraint Jacobian matrix:

$$J_c(q)^T = Q \begin{bmatrix} R \\ 0 \end{bmatrix}, \tag{6.36}$$

where Q is an orthogonal matrix and R is an upper triangle matrix of rank m_η. The Q in the QR decomposition provides a coordinate transform which separates the system dynamics (2.7) into the constrained and unconstrained components:

$$Q^T \left[D(q)\ddot{q} + H(q, \dot{q}) \right] = Q^T B u + \begin{bmatrix} R \\ 0 \end{bmatrix} \lambda. \tag{6.37}$$

Remark 3. *In* (6.37)*, the spring force term, $B_s \tau_s$, is not present. We originally developed the controller with the spring dynamics and were able to achieve walking. However, the robot would often overreact to perturbations in the spring deflections. Instead, we apply a constraint on the springs, but at their current deflected state. This produces a torque consistent with the deflected leg configuration, but does not react aggressively to spring oscillations.*

Through the use of a selection matrix $S_u = \begin{bmatrix} 0_{(n-k)\times k} & I_{(n-k)\times(n-k)} \end{bmatrix}$, we can obtain the unconstrained system dynamics which are not dependent on the constraints:

$$S_u Q^T \left[D(q)\ddot{q} + H(q, \dot{q}) \right] = S_u Q^T B u. \tag{6.38}$$

Solving for u leads us to the inverse dynamics:

$$u_{ff} = (S_u Q^T B)^\dagger S_u Q^T \left[D(q)\ddot{q}^d + H(q, \dot{q}) \right], \tag{6.39}$$

Table 6.3: Feedback control values used for PD feedback and regulator heuristics for tracking the gait library on Cassie.

Parameter	Value
K_p	$[900, 500, 300, 250, 200, 200, 200, 200, 25]$
K_d	$[12, 6, 4, 6, 5, 6, 4, 4, 2]$
$\tilde{K}_p^{x,y}$	$[0.045, 0.086]$
$\tilde{K}_d^{x,y}$	$[0.20, 0.22]$
$k_p^{x,y}$	$[1.25, 1.90]$
$k_v^{x,y}$	$[0.80, 0.60]$
$k_i^{x,y}$	$[1.90, 0]$

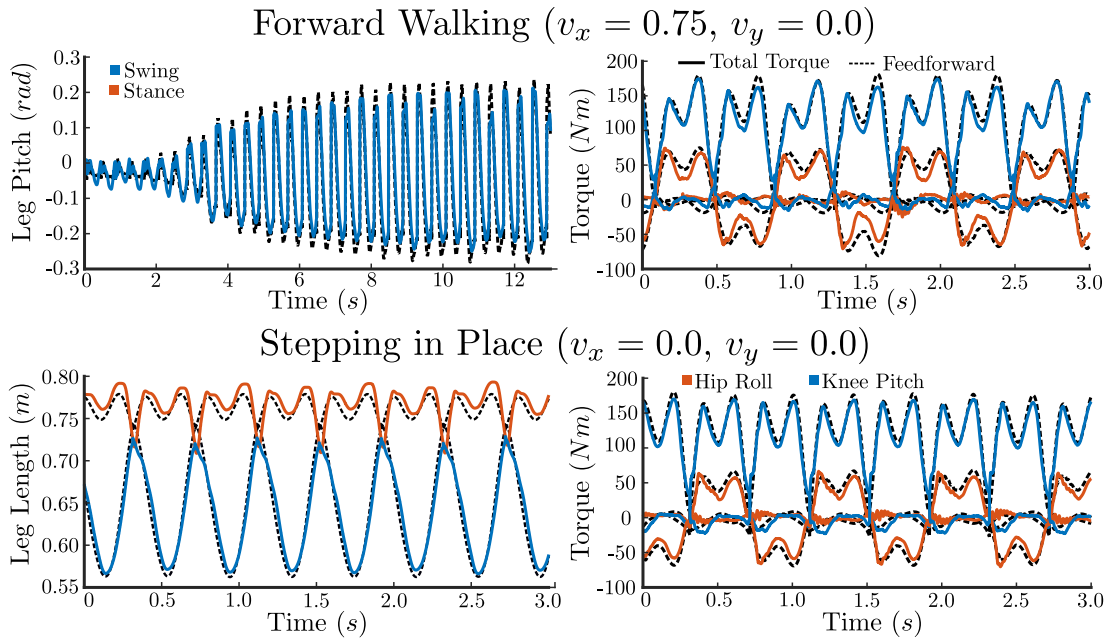

Figure 6.18: Experimental results for walking (top) and stepping in place (bottom). The plots depict the accuracy of the tracking on the leg length and swing leg pitch outputs. On the right is a comparison of the torque from the feedforward controller with the total torque applied for the knee pitch and hip roll joints.

where † represents the appropriate Moore-Penrose inverse and $\ddot{q}^d$ is a target acceleration for the system to follow. In addition to the feedforward term provided by (6.39), a feedback correction is computed as a standard PD controller:

$$u = u_{ff} - \left(\frac{\partial y(t,q)}{\partial q_m}\right)^{\dagger} \left(K_p y(t,q) + K_d \dot{y}(t,q,\dot{q})\right), \tag{6.40}$$

where K_p and K_d are the proportional and derivative gains applied to the output feedback on hardware, with their specific values provided in Table 6.3.

In order to select the parameters which are used for output tracking and for the feedforward term, we use the average velocity of the previous step, $\bar{v}^a_{k-1}$. Thus, our Bèzier polynomial array provides us at each time instant with the nominal acceleration $\ddot{q}^*(t, \bar{v}^a_{k-1})$, outputs $y^d(t, \bar{v}^a_{k-1})$, floating-base positions $p^*_{x,y}(t, \bar{v}^a_{k-1})$, and velocities $v^*_{x,y}(t, \bar{v}^a_{k-1})$. Plots which demonstrate the accuracy of the output tracking, and feedforward torques are shown in Fig. 6.18 for forward walking and stepping in place. The accuracy of the feedforward torques is clearly demonstrated in this result, where the outputs error is much smaller than was previously seen via methods only considering PD in Sec. 6.1. In addition, the feedforward accelerations which come from the bilinear interpolation of the gait library in Sec. 6.2.4 are doing most of the effort in the controller through the inverse dynamics.

Foot Placement

The presented controller required no heuristic tuning on the feedforward terms to achieve tracking of the outputs such as offsets or hand-tuned torque profiles. However, directly implementing this controller with the trajectory obtained from the NLP (HZD Direct Collocation) can at best result in a marginally stable locomotion for experiments as it has no way to trend walking towards an overall target walking speed. Motivated by this, a PD controller is used to find an offset to the footstrike location, and translated to the desired outputs:

$$\Delta := \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \end{bmatrix} = \begin{bmatrix} \tilde{K}_{p,x}(\tilde{v}^a_x - v^d_x) + \tilde{K}_{d,x}(\tilde{v}^a_x - \bar{v}^a_{x,k-1}) \\ \tilde{K}_{p,y}(\tilde{v}^a_y - v^d_y) + \tilde{K}_{d,y}(\tilde{v}^a_y - \bar{v}^a_{y,k-1}) \\ 0 \end{bmatrix}, \tag{6.41}$$

$$\begin{aligned} y^d_{sw,ll} &= ||p^*_{nsf}(y^d) + \Delta||_2, \\ y^d_{lp} &= \sin^{-1}\left(\frac{p^*_x(y^d) + \Delta_x}{y^d_{sw,ll}}\right) - y^d_{b,x}(t,\alpha), \\ y^d_{lr} &= \sin^{-1}\left(\frac{p^*_y(y^d) + \Delta_y}{y^d_{sw,ll}}\right) - y^d_{b,y}(t,\alpha), \end{aligned}$$

where $\tilde{v}^a_{x,y} = \bar{v}^a_{k-1} + \left(v^a_{x,y} - v^*_{x,y}\right)$ is the current step velocity, $v^d_{x,y}$ is the target step velocity from the user joystick, $v^a_{x,y}$ is the instantaneous velocity of the robot relative to the stance foot, and $p^*_{nsf}(y^d) = (p^*_{nsf,x}, p^*_{nsf,y}, p^*_{nsf,z})$ are the nominal Cartesian swing foot positions computed from the desired outputs. This style of regulator is inspired by early work of [12], and has been used widely in the literature.

Acceleration Augmentation

An additional regulator is applied to modify the nominal accelerations of the floating base accelerations, which were pictured in Fig. 6.16. Through the application of a heuristic feedback controller, we can make the robot choose feedforward torques which trend the robot toward the target velocity while satisfying the contact constraints on the system:

$$\ddot{q}^d_{x,y} = \ddot{q}^*_{x,y} + k_p\left(p^a_{x,y}(q) - p^*_{x,y}\right) + k_v(\tilde{v}^a_{x,y} - v^d_{x,y}) + k_i\int_0^t \gamma(\tilde{v}^a_{x,y}(t') - v^d_{x,y}(t'))dt', \tag{6.42}$$

where k_p is a gain affecting the x and y position errors of the pelvis relative to the stance foot, k_v is a gain on the step velocity tracking error, and k_i is a gain on

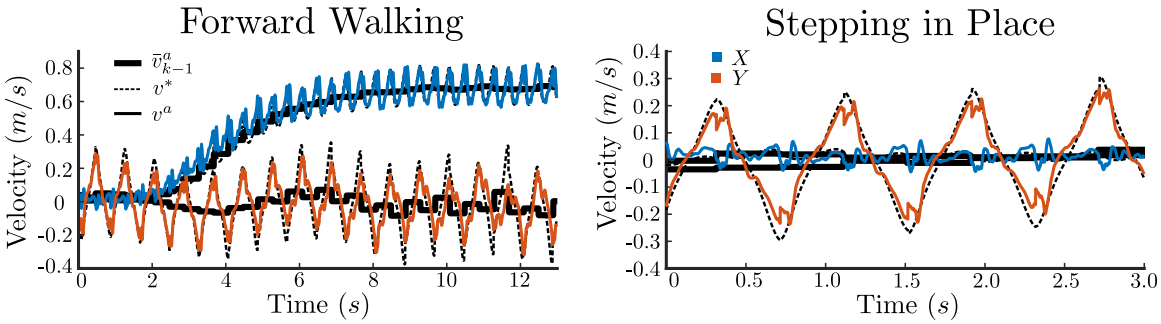

Figure 6.19: Shown is a comparison of the desired velocities from the current gait for the gait library on Cassie, $\bar{v}^a_{k-1}$, compared with the actual velocity of the robot for walking (left) and stepping in place (right).

the accumulated step velocity error with a decay constant of $\gamma = 0.9995$ to avoid integral windup. The gains used on hardware are detailed in Table 6.3. Because the feedforward term (6.39) uses the full actuator matrix B, it does apply some control effort on the stance foot actuator. It is for that reason that the integral term is applied to the x direction, and not in the y.

6.2.6 Results and Conclusions

The Cassie biped poses a unique challenge due to its compliant leg mechanism and the highly underactuated nature of the dynamics. In order to leverage these components in experiments, we constructed a hybrid model for walking dynamics based on a compliant model. A trajectory optimization was then developed to efficiently generate walking trajectories using the method of HZD and was then tracked through an inverse dynamics controller. The resulting experiments show that the optimization can effectively capture the passive dynamics on a highly complex robot, while providing accurate model-based feedforward information.

The controller, simulation, and trajectory optimization are also made available in an open-source repository[5]. These controllers were first tuned in a Gazebo simulation, shown on the right in Fig. 6.15, before implementation on hardware. Cassie was then tested indoors for stepping in place and forward, backward, lateral, and diagonal walking. Cassie was then taken outside to walk over several hills, raised roots, and a brick path. A video of both the simulation and experiments is provided[6], with walking tiles of the robot walking on flat ground and traversing rough terrain shown in Fig. 6.21.

[5]Open-sourced repository for the C++ controller code used in this work on hardware and for use in a Gazebo simulation: `https://github.com/jpreher/cassie_documentation`

[6]Video of simulation and experimental results on Cassie: `https://youtu.be/SvhjPZqSGFI`

The results presented throughout this section demonstrate the *ability of the HZD optimization problem to be successfully extended to a continuous motion library* of walking gaits for the Cassie robot. Further, the hardware controller development which includes an inverse dynamics controller with reliance on the generalized accelerations directly taken from optimization, $\ddot{q}^*$, *demonstrates the benefit of extracting additional model information from the HZD optimization* beyond simply the Bézier polynomial coefficients or torque. While simply commanding the optimal torques as a feedforward term have been done to achieve tracking elsewhere on planar walkers [41], this may not respect the current holonomic constraints imposed on the robot.

Plots demonstrating the accuracy of the output tracking and feedforward torques are shown in Fig. 6.18 for forward walking and stepping in place, which help motivate the desire to implement model-based controllers on hardware with HZD. Additionally, limit cycles of the knee and hip pitch joints are shown in Fig. 6.20, with the closed nature of the cycle demonstrating the stability of the walking. Pertaining to the accuracy of the trajectory optimization and gait library parameterization to characterize the walking behavior, velocity tracking data is shown in Fig. 6.19. These velocity plots show that in combination with tracking a given motion, *effectively rendering the zero dynamics stable*, the combined motion library is able to shift the walking from one orbit to another, *with each planned gait satisfying hybrid invariance*. One

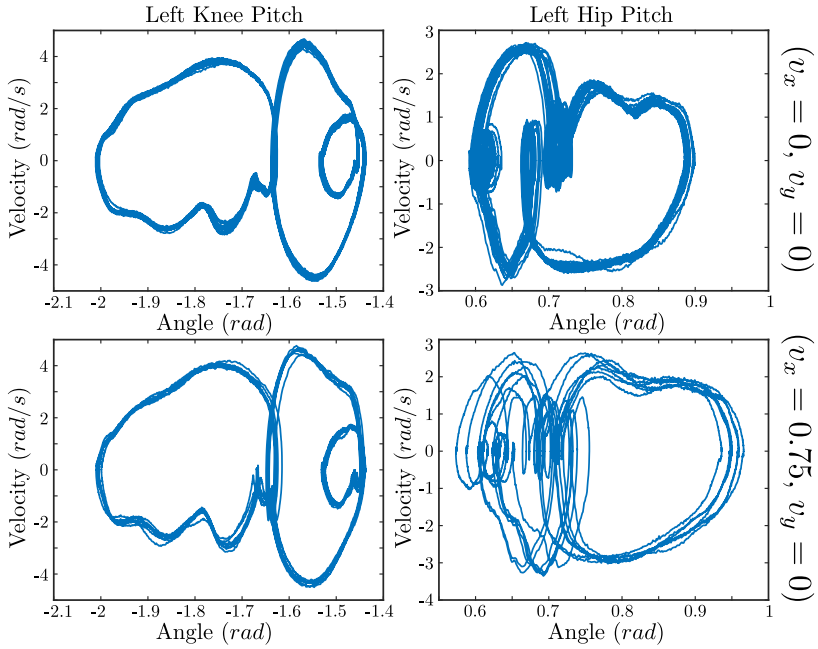

Figure 6.20: Phase portrait of the left knee and hip pitch joints for forward walking and stepping in place.

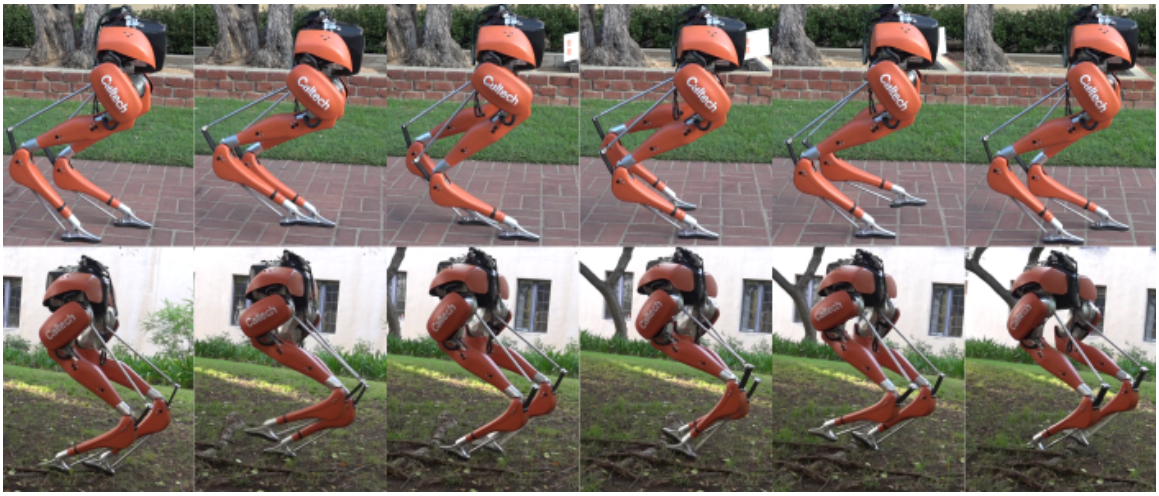

Figure 6.21: Gait tiles of the robotic walking exhibited experimentally on the Caltech Cassie robot. The top half is two steps of flat-ground forward walking outside, while the bottom half shows several steps of the robot walking over a height variation caused by both a shallow grassy slope and exposed roots.

additional facet of this implementation to note is how small the foot placement gains are in Table 6.3 in comparison to those of Sec. 6.1. This is largely attributable to the gait library, which always selects a gait which satisfies hybrid invariance at the current walking speed and thus requires significantly less action from the foot placement controller to stabilize the walking.

On the physical system, a trivial double-support domain is virtually impossible to attain due to the compliance present in both legs. While this assumption simplified the development of controllers and trajectory planning, the consideration of a continuous double-support domain can enhance the stability of the behaviors. Future work on locomotion should focus on reintroducing a nontrivial double-support domain to our walking model. Additionally, the use of an analytical solution to the inverse dynamics cannot account for optimal distributions of contact forces, friction, torque limits, or provide convergence guarantees. Thus, our future controller development is focused on further developing real-time model based controllers for implementation on hardware, such as extending the control Lyapunov function based methods in [186] to walking. These new control developments are introduced in Chap. 7 and Chap. 8.

Chapter 7

CONTROL LYAPUNOV FUNCTIONS FOR ROBOTIC SYSTEMS

In this chapter, control Lyapunov functions (CLFs) are introduced, beginning with a general background for synthesizing and deriving them for nonlinear systems which are feedback linearizable. With this general background introduced in the first half of the chapter, the second half will focus on a novel representation of the CLF for robotic systems, which was developed along two publications:

[1] J. Reher, C. Kann, and A. D. Ames, "An inverse dynamics approach to control Lyapunov functions," in *2020 American Control Conference (ACC)*, IEEE, 2020.

[2] J. Reher and A. D. Ames, "Control Lyapunov functions for compliant hybrid zero dynamic walking," *IEEE Transactions on Robotics*, 2021, In Preparation.

The end result will be a CLF with equivalent convergence properties, but with more desirable traits for actual implementation and tuning on hardware. This will be extended to a relaxed version of the CLF controller, which removes a convergence inequality constraint in lieu of a conservative CLF cost within a quadratic program to achieve tracking. These new developments will then form the basis for the experimental study introduced for Cassie in Chap. 8. As it pertains to this thesis, its primary contributions and the relevance of these developments are:

- Sections 7.1-7.2 are largely literature review, while the theoretical developments presented in Sec. 7.3 form the main theoretical contribution of this thesis.
- A theoretical analysis is shown which demonstrates several useful properties of the approach for tuning and implementation, and the stability of the controller for HZD locomotion is proven.
- The main result of this chapter is an optimization-based control framework that couples convergence constraints from control Lyapunov functions with desirable formulations existing in other areas of the bipedal locomotion field that have proven successful in practice, such as inverse dynamics control and quadratic programming approaches.

7.1 Background: Several Existing Control Methods for Legged Robots

While the methods outlined in Chap. 3 and shown experimentally in Chap. 5-6 demonstrate our ability to synthesize dynamic and efficient locomotion plans for high-DOF nonlinear robots, there is a clear gap in the actual implementation of the walking on hardware compared to the formal tools used in analysis. In almost all HZD approaches, simple PD controllers such as those shown in Chap. 5 and Sec. 6.1 are used to approximately zero the virtual constraints. However, in Sec. 6.2 the argument was made that not only could tracking be improved through the use of model-based controllers, but we can also obtain desirable properties such as less aggressive gains and control compliance to disturbances that satisfy the holonomic constraints imposed on the robot.

As one of the most common tools within nonlinear control, feedback linearization was introduced in Chap. 3 as a way to design stabilizing controllers which cancel the nonlinear dynamics of a problem, rendering a linear one on which we can obtain convergence properties. While this was useful as a conceptual tool in designing the zero dynamics surface, feedback linearization is often not practically implementable on highly complex robots, such as those introduced in Chap. 2 and more specifically Chap. 4. Even if one was able to directly implement these controllers, they often run counter to the goals set forth in Sec. 1.1, due to the fact that the approach does not create controllers which utilize the natural dynamics of the system.

In this section, we will first outline several existing methods which have become popularized for the control of humanoids and bipedal robots. These methods have proven to be implementable on embedded systems and have shown reasonable performance in enabling complex motions and behaviors. Several of these ideas are used within the control Lyapunov function framework background in Sec. 7.2 and will form a basis for several new developments in Sec. 7.3.

Operational Space Control and Inverse Dynamics

Control approaches to legged robots need to enable safe and dexterous movement while satisfying a variety of task objectives and contact constraints. Because the stability of the system is often tied to the simultaneous satisfaction of constraints and accurate objective tracking, high-performance feedback control approaches are most desirable. In addition, while moving through environments which may contain an array of disturbances, it is important that the controller have some control compliance when perturbed, while best satisfying the objectives. Torque control strategies,

especially model-based approaches, have been widely used on robotic manipulators for these reasons. However, until the late 2000's they were not widely used on legged robots because they are often intrinsically underactuated and are subject to varying contact constraints. In the late 2000's and early 2010's, several inverse dynamics approaches were shown to be successful on floating-base legged robots [142], [143], often using a null-space projections. By using a dynamically consistent null-space projection, control solutions are able to compensate for task dynamics and decouple them from constraint dynamics, and is often necessary to obtain analytical expressions for model-based feedback controllers.

Inverse dynamics is a widely used method to approaching controller design for achieving a variety of motions and force interactions. Given a target behavior, the dynamics of the robotic system are inverted to obtain the desired torques. In most formulations, the system dynamics are mapped onto a support-consistent manifold using methods such as the dynamically consistent support null-space [218], linear projection [143], and orthogonal projection [142]. Recall in Sec. 2.1.3 that we had derived a null-space projection operator for finding the direct dynamics as:

$$P_c(q) = I - J_c^{\dagger}(q)J_c(q), \tag{7.1}$$

which allowed us to obtain a representation of the dynamics which implicitly satisfied the holonomic constraint equations (2.12). Given a desired generalized acceleration, $\ddot{q}^d$, the joint torque is calculated simply by pseudo-inversion:

$$u = (P_c(q)B)^{\dagger} P_c(q) \left[D(q)\ddot{q} + H(q,\dot{q})\right]. \tag{7.2}$$

Alternatively, we can use a QR decomposition [219] as was done in Sec. 6.2 on the inverse dynamics controller presented on Cassie. This approach was initially shown to be effective for quadrupedal locomotion [142], where the constraint Jacobian is decomposed as $J_c^T = Q \begin{bmatrix} R^T & 0^T \end{bmatrix}^T$, to obtain the inverse dynamics:

$$u = (S_u Q^T B)^{\dagger} S_u Q^T \left[D(q)\ddot{q}^d + H(q,\dot{q})\right], \tag{7.3}$$

with $S_u = \begin{bmatrix} 0_{(n-k)\times k} & I_{(n-k)\times(n-k)} \end{bmatrix}$ a selection matrix corresponding to the unconstrained dynamics.

When prescribing behaviors in terms of purely task space objectives, this is commonly referred to as the *operational space control* (OSC) [96] formulation, which derives end-effector dynamics for rigid body manipulators [220]. Recall in Sec. 2.1.3 how

we had derived the direct dynamics (2.19) of a constrained robotic system using the orthogonal constraint projection operator $P = I - J_c^{\dagger}(q)J_c(q)$. Additionally, using the fact that a force at an end-effector F is mapped into joint torque via $u = J^T F$, we can obtain the *operational space dynamics* for a constrained system as:

$$\Lambda_c(q)\ddot{x} + \Lambda_c(q)\left(J_c(q)D_c^{-1}(q)P_c(q)H(q,\dot{q}) - \left(\dot{J}(q,\dot{q}) + J_c(q)D_c^{-1}(q)C_c(q,\dot{q})\right)\dot{q}\right) = F, \tag{7.4}$$

where $\Lambda_c = (J_c(q)D_c^{-1}(q)P_c(q)J_c^T(q))^{-1}$. We can then formulate the OSC problem as:

$$u = J^T(q)F + Nu_0, \tag{7.5}$$

where F is defined by (7.4) using a desired end-effector acceleration $\ddot{x}_d$ in place of $\ddot{x}$, $N = I - J^T(q)J^{T\#}(q)$ and:

$$J^{T\#}(q) := (J(q)D_c^{-1}(q)P_c(q)J_c^T(q))^{-1}J_c(q)D_c^{-1}(q)P_c(q). \tag{7.6}$$

Sentis [218], [221], [222] developed several underactuated OSC approaches specifically addressing the underactuation of humanoids, alongside Park and Khatib [223], [224].

Quadratic Programming Approaches to Legged Control

In some cases, there may be tasks of differing priority which cannot be simultaneously satisfied on a robot. For instance, the task of maintaining a foothold is critical, while some manipulation task is not necessary to maintain stability [225]. One way to look at this hierarchy of tasks is as a system of linear equation:

$$A_i(x)\mathcal{X} = b_i(x), \tag{7.7}$$

with the optimization variable $\mathcal{X}$, and problems of the same priority are stacked within each i of A_i and b_i. This can be posed as a least squares problem:

$$\min_{\mathcal{X}} ||A_i(x)\mathcal{X} - b_i(x)||^2, \tag{7.8}$$

where if $A = [A_1^T, A_2^T, \ldots, A_{n_p}^T]^T$ has full column rank, the cost is 0. Hutter [226] showed how this can be solved analytically by iteratively performing a null-space projection on each task. Alternatively, this can be posed as a quadratic program (QP) [227], with the benefit of performing an efficiently solvable optimization subject to linear inequality and equality constraints [187]:

$$\begin{aligned}
\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmin}} \quad & \frac{1}{2}\mathcal{X}^T Q(x)\mathcal{X} + c^T(x)\mathcal{X} && \text{(QP)} \\
\text{s.t.} \quad & A_E(x)\mathcal{X} = b_e(x) && \text{(Linear Equality Constraints)} \\
& C_I(x)\mathcal{X} \geq d_I(x), && \text{(Linear Inequality Constraints)}
\end{aligned}$$

where $\mathcal{X}$ is the unknown vector of optimization variables. The rest of the variables are known vectors and matrices. By construction, we always maintain a positive semidefinite matrix $Q(x)$ which will allow us to solve this problem with standard QP solvers. This is frequently done by formulating the cost as a least squares problem, penalizing distance from $A(x)\mathcal{X} = b(x)$. Thus, $G(x) = A^T(x)A(x)$ and $g(x) = -A^T(x)b(x)$. The matrices $A(x)$ and $b(x)$ are often decomposed into weighted blocks:

$$A(x) = \begin{bmatrix} w_0 A_0^T & w_1 A_1^T & \dots & w_{n_p} A_{n_p}^T \end{bmatrix}^T, \; b(x) = \begin{bmatrix} w_0 b_0^T & w_1 b_1^T & \dots & w_{n_p} b_{n_p}^T \end{bmatrix}^T, \quad (7.9)$$

where w_i is the weight associated with tracking the ith task. Through tuning these weights, the optimization can handle conflicting goals with weighted priorities.

In many recent works, variations of these approaches have been shown to allow for high-level tasks to be encoded with intuitive constraints and costs in optimization based controllers, some examples being [59], [144], [214], [226], [228], [229]. Here we present a minimal implementation of an inverse dynamics controller. The inverse dynamics problem can also be posed using a QP to exploit the fact that the instantaneous dynamics and contact constraints can be expressed linearly with respect to a certain choice of decision variables. Specifically, let us consider the set of optimization variables $\mathcal{X} = [\ddot{q}^T, u^T, \lambda^T]^T \in \mathbb{X}_{\text{ext}} := \mathbb{R}^n \times U \times \mathbb{R}^{m_h}$, which are linear with respect to (2.7) and (2.12):

$$\begin{bmatrix} D(q) & -B & -J_h(q)^T \\ J_h(q) & 0 & 0 \end{bmatrix} \mathcal{X} + \begin{bmatrix} H(q,\dot{q}) \\ \dot{J}_h(q)\dot{q} \end{bmatrix} = 0. \quad (7.10)$$

Also consider a positional objective in the task space of the robot, which can be characterized using: (7.54):

$$J_y(q,\dot{q})\ddot{q} + \dot{J}_y(q,\dot{q})\dot{q} - \ddot{y}_2^* = 0 : \quad (7.11)$$

where $\ddot{y}_2^* = K_P y_2 + K_D \dot{y}_2$ is a PD control law which can be tuned to achieve convergence. In its most basic case, we can combine these elements to pose this QP tracking problem as:

$$\begin{aligned} \mathcal{X}^*(x) = \underset{\mathcal{X} \in \mathbb{X}_{\text{ext}}}{\text{argmin}} \quad & ||J_y(q)\ddot{q} + \dot{J}_y(q,\dot{q})\dot{q} - \ddot{y}^*||^2 + \sigma W(\mathcal{X}) && \text{(ID-QP)} \\ \text{s.t.} \quad & \text{Equation (7.10)} && \text{(System Dynamics)} \\ & u_{\min} \leq u \leq u_{\max} && \text{(Torque Limits)} \\ & \text{Equation (2.22)} && \text{(Friction Pyramid)} \end{aligned}$$

where $W(\mathcal{X})$ is included as a regularization term with a small weight σ such that the problem is well posed, and we have included feasibility constraints such as the friction cone (Equation (2.21)) and torque limits. Although this controller satisfies the contact constraints and yields an approximately optimal solution to tracking task-based objectives, it does not provide formal guarantees with respect to stability. In dynamic walking motions, this becomes an important consideration, wherein impacts and footstrike can destabilize the system, requiring more advanced nonlinear controllers.

7.2 Background: Control Lyapunov Functions

In Sec. 3.2.2, it was shown how feedback linearization could be used to render a linear system which could be stabilized via PD feedback. While this procedure can exponentially stabilize the output dynamics, it does not result in controllers that utilize the natural dynamics of the system, and can result in overly aggressive torques. Instead, we would like to examine output tracking from a Lyapunov perspective, which is to drive a Lyapunov function to decay in order to achieve convergence. Control Lyapunov functions (CLFs), and specifically rapidly exponentially stabilizing control Lyapunov functions (RES-CLFs), were introduced as methods for achieving (rapidly) exponential stability for walking robots [31], [165]. This control approach has the benefit of yielding an entire class of controllers that provably stabilize periodic orbits for hybrid system models of walking robots, and can be realized in a pointwise optimal fashion via optimization-based controllers. The CLF-based controllers presented throughout this section have recently been explored for application on hardware, because much like the optimization controllers of Sec. 7.1, they can be solved in real-time. Experimental results have been shown on MABEL [31], [230] and DURUS-2D [178], with recent results indicating how robust formulations can be used [231]. Additionally, it was implemented at over 5 kHz as an embedded level controller on series elastic actuators [232], indicating possible future uses on explicitly controlling compliant dynamic walking. The use of CLFs has also been used to automatically generate stable walking gaits through SLIP approximations [233], and also to enforce planned motions for reduced order models [234] along with realizing 3D bipedal jumping experimentally on Cassie [235]. The fundamentals presented here will be used in Sec. 7.3 to derive a new form for CLFs specifically tailored for floating-base robotic systems, and are shown to be successful on hardware in Chap. 8.

In the theory of ordinary differential equations, *Lyapunov functions* are scalar functions which can be used to prove the stability of an equilibrium. Let $V : X \to \mathbb{R}$ be a

Lyapunov-candidate-function for the uncontrolled dynamical system $\dot{x} = f(x)$. The derivative of V along its solutions is:

$$\dot{V}(x) = \frac{\partial V(x)}{\partial x} f(x), \tag{7.12}$$

where the stability of an equilibrium point is tied to the negativity of $\dot{V}(x)$.

Definition 7.2.0.1. *(Lyapunov Stability [162]) The equilibrium point $x = 0$ of the autonomous system $\dot{x} = f(x)$ is:*

- *stable if, $V(0) = 0$, $V(x) > 0$ in $X - \{0\}$, and $\dot{V} \leq 0$ in X,*
- *asymptotically stable if, in addition, $\dot{V}(x) < 0$ in $X - \{0\}$.*

In simple terms, Definition 7.2.0.1 states that if V is positive definite and $\dot{V}$ is negative semi-definite, then $x = 0$ is stable, and if $\dot{V}$ is instead negative definite, then $x = 0$ is asymptotically stable.

The concept of control Lyapunov functions is to then extend the Lyapunov stability properties to control affine nonlinear dynamical systems:

$$\dot{x} = f(x) + g(x)u, \tag{7.13}$$

where $x \in X \subseteq \mathbb{R}^n$ is the set of controllable states and $u \in U \subseteq \mathbb{R}^m$ is the control input. The mappings $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are assumed to be locally Lipschitz continuous functions of x. For the control system (7.13), we can propose a control Lyapunov function candidate $V : X \to \mathbb{R}$. As opposed to the case in (7.12), time derivative of $V(x)$ is now a function of our input:

$$\dot{V}(x, u) = \underbrace{\frac{\partial V(x)}{\partial x} f(x)}_{L_f V(x)} + \underbrace{\frac{\partial V(x)}{\partial x} g(x)}_{L_g V(x)} u, \tag{7.14}$$

with $L_f V(x) \in \mathbb{R}$ and $L_g V(x) \in \mathbb{R}^{1 \times m}$.

Definition 7.2.0.2. *(Control Lyapunov Function [236], [237]) For the nonlinear system (7.13), a **control Lyapunov function** is a function $V : X \subseteq \mathbb{R}^n \to \mathbb{R}$ satisfying:*

$$c_1 ||x||^2 \leq V(x) \leq c_2 ||x||^2$$
$$\inf_{u \in U} \left[L_f V(x) + L_g V(x) u \right] \leq -c_3 V(x),$$

for all $x \in X$ and constants $c_1, c_2, c_3 \in \mathbb{R}_{>0}$.

7.2.1 Control Lyapunov Functions for Output Tracking

For all cases relevant to this thesis, we will consider only feedback linearizable systems which attempt to zero a set of outputs with the goal of rendering a stable zero dynamics surface according to the background presented in Chap. 3. Thus, we consider a system which follows the modeling definitions provided in Chap. 2; with configuration space $\mathcal{Q}$, (local) coordinates $q \in \mathcal{Q}$, and states $x = (q^T, \dot{q}^T)^T \in T\mathcal{Q} = X$. This goal of this system is to track desired trajectories of the form:

$$y(x, \alpha) = y^a(x) - y^d(\tau(x), \alpha), \tag{7.15}$$

where $y^a : X \to \mathbb{R}^m$ and $y^d : \mathbb{R} \times \mathbb{R}^a \to \mathbb{R}^m$ are smooth functions encoding the desired behavior. We assume that $y(x)$ has (vector) relative degree r [160]. Suppose, as was described in the construction of the zero dynamics surface in Sec. 3.2.4, that there exists a diffeomorphism $\Phi : X \to Y \times Z$ for $\eta \in Y \subset \mathbb{R}^m$ representing the output dynamics of the original dynamical system, and $z \in \mathcal{Z} \subset \mathbb{R}^l$ as the uncontrolled states, and $x = \Phi^{-1}(\eta, z)$. The controlled system written in general normal form is then:

$$\begin{aligned} \dot{\eta} &= \bar{f}(\eta, z) + \bar{g}(\eta, z)u \\ \dot{z} &= \omega(\eta, z), \end{aligned} \tag{7.16}$$

where $\bar{f}$ and $\bar{g}$ are derived in (3.32). Noting that Definition 7.2.0.2 is valid for the normal form output dynamics in (7.16) with η, the derivative of the CLF along the output dynamics is:

$$\dot{V}(\eta, z, u) = \frac{\partial V(\eta)}{\partial \eta}\left(\bar{f}(\eta, z) + \bar{g}(\eta, z)u\right) = L_{\bar{f}}V(\eta, z) + L_{\bar{g}}(\eta, z)u. \tag{7.17}$$

One of the consequences of Definition 7.2.0.2 is that the existence of a CLF yields a family of controllers which exponentially stabilize η. More specifically, $\eta \to 0$ exponentially for any control value in the family of controllers:

$$K(x) = \{u \in U : L_{\bar{f}}V(\eta, z) + L_{\bar{g}}V(\eta, z)u + c_3 V(x) \leq 0\}. \tag{7.18}$$

In the context of underactuated hybrid systems, which can be represented by the normal form dynamics in (7.16), it can sometimes be necessary to achieve a stronger bound on the convergence rate than the exponential one in Definition 7.2.0.2. This motivates the following definition.

Definition 7.2.0.3. *(RES-CLF [31]) For the system* (7.16)*, a continuously differentiable function* $V_\varepsilon : Y \to \mathbb{R}$ *is said to be a* ***rapidly exponentially stabilizing control Lyapunov function*** *if there exist positive constants* $c_1, c_2, c_3 > 0$ *such that:*

$$c_1||\eta||^2 \leq V_\varepsilon(\eta) \leq \frac{c_2}{\varepsilon^2}||\eta||^2, \tag{7.19}$$

$$\inf_{u\in U}\left[L_{\bar{f}}V_\varepsilon(\eta,z) + L_{\bar{g}}V_\varepsilon(\eta,z)u + \frac{c_3}{\varepsilon}V_\varepsilon(\eta)\right] \leq 0, \tag{7.20}$$

for all $0 < \varepsilon < 1$ *and for all* $(\eta, z) \in Y \times Z$*.*

In the case of a RES-CLF, we can then consider the family of controllers:

$$K_\varepsilon(\eta,z) = \{u \in U : L_{\bar{f}}V(\eta,z) + L_{\bar{g}}V(\eta,z)u + \frac{c_3}{\varepsilon}V(\eta) \leq 0\}, \tag{7.21}$$

consisting of the control values that result in $\dot{V}_\varepsilon(\eta, z, u) \leq -\frac{c_3}{\varepsilon}V_\varepsilon(\eta)$. To see this, note that for $u_\varepsilon(\eta, z) \in K_\varepsilon(\eta)$:

$$\begin{aligned} \dot{V}_\varepsilon(\eta,u^*(\eta,z)) \leq -\frac{c_3}{\varepsilon}V_\varepsilon(\eta) \qquad &\Rightarrow \qquad V(\eta(t)) \leq e^{-\frac{c_3}{\varepsilon}t}V(\eta(0)) \\ &\Rightarrow \qquad \|\eta(t)\| \leq \frac{1}{\varepsilon}\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}e^{-\frac{c_3}{2\varepsilon}t}\|\eta(0)\|. \end{aligned} \tag{7.22}$$

Thus, this gives the set of control values that exponentially stabilize the outputs and we can control the convergence rate via ε. The selection of an appropriate choice for the "best" control value possible leads to the notion of optimization based control with CLFs, which will be addressed in Sec. 7.2.4.

7.2.2 Synthesizing Control Lyapunov Functions

With several of the basic definitions for CLFs in hand, we would now like to address the task of actually synthesizing a CLF for a feedback linearizable system. Suppose that for (7.13) there is a set of outputs $y(x) = (y_1(q,\dot{q})^T, y_2(q)^T)^T$ of vector relative degree 1 and 2, respectively, on a region of interest; that is for $y_1(q,\dot{q}) \in \mathbb{R}^{m_1}$ and $y_2(q) \in \mathbb{R}^{m_2}$ with $m = m_1 + m_2$, we assume that the vector relative degree is 1 for y_1 and the 2 for y_2, i.e., $(1, \ldots, 1, 2, \ldots, 2)$ with 1's appearing m_1 times and 2's appearing m_2 times. We can then write an output tracking objective:

$$y_1(q,\dot{q},t) = y_1^a(q,\dot{q}) - y_1^d(\tau,\alpha) \tag{7.23}$$

$$y_2(q,t) = y_2^a(q) - y_2^d(\tau,\alpha), \tag{7.24}$$

where y^a and y^d are the actual and desired outputs, and τ is some parameterization of time for the desired outputs to evolve on. Taking the respective derivatives of the

outputs results in:

$$\begin{bmatrix}\dot{y}_1\\ \ddot{y}_2\end{bmatrix} = \underbrace{\begin{bmatrix}\frac{\partial y_1}{\partial q} & \frac{\partial y_1}{\partial \dot{q}}\\ \frac{\partial}{\partial q}\left(\frac{\partial y_2}{\partial q}\dot{q}\right) & \frac{\partial y_2}{\partial q}\end{bmatrix} f(x)}_{\mathbf{L}_f y(x)} + \underbrace{\begin{bmatrix}\frac{\partial y_1}{\partial \dot{q}}\\ \frac{\partial y_2}{\partial q}\end{bmatrix} g(x)}_{\mathcal{A}(x)} u, \tag{7.25}$$

now in the form of (3.8). Assuming that the preliminary feedback introduced in (3.23) has been applied to (7.13), we will render a linear system for the output dynamics with coordinates $\eta := (y_1^T, y_2^T, \dot{y}_2^T)^T$:

$$\dot{\eta} = \begin{bmatrix}\dot{y}_1\\ \dot{y}_2\\ \ddot{y}_2\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 0\\ 0 & \mathbf{I}\\ 0 & 0\end{bmatrix}}_{F} \eta + \underbrace{\begin{bmatrix}\mathbf{I} & 0\\ 0 & 0\\ 0 & \mathbf{I}\end{bmatrix}}_{G} \nu. \tag{7.26}$$

We can find the derivative of V with respect to the linear system (7.26) to be:

$$\dot{V}(\eta) = L_F V(\eta) + L_G V(\eta) v, \tag{7.27}$$

where the Lie derivatives of V_ε along the linear output system's dynamics (7.26) are:

$$L_F V(\eta) = \eta^T (F^T P_\varepsilon + P_\varepsilon F)\eta, \tag{7.28}$$

$$L_G V(\eta) = 2\eta^T P_\varepsilon G. \tag{7.29}$$

In order to convert back into a form which can be represented in terms of the control input, we can use the previous relationship between u and ν:

$$\mathcal{A}(x)u + \mathbf{L}_f y(x) = \nu, \tag{7.30}$$

to obtain the CLF derivative in terms of x since η depends on x (via y_1, y_2 and $\dot{y}_2$):

$$\dot{V}(x,u) = \underbrace{L_F V(x) + L_G V(x)\mathbf{L}_f y(x)}_{L_f V(x)} + \underbrace{L_G V(x)\mathcal{A}(x)}_{L_g V(x)} u. \tag{7.31}$$

Continuous-Time Lyapunov Equation Synthesis

It was shown in Chap. 3 how in the context of robotic walking and Hybrid Zero Dynamics, we had applied a feedback linearizing controller with the auxiliary feedback chosen to force the resulting output dynamics (7.26) to evolve according to:

$$\ddot{y} = \begin{bmatrix}\dot{y}_1\\ \ddot{y}_2\end{bmatrix} = \begin{bmatrix}-\frac{1}{\varepsilon}K_{\bar{v}} y_1\\ -\frac{1}{\varepsilon^2}K_P y_2 - \frac{1}{\varepsilon}K_D \dot{y}_2\end{bmatrix} \tag{7.32}$$

where $0 < \varepsilon \leq 1$ is a tunable parameter, and $K_{\bar{v}}$, K_P, K_D are control gains for the relative degree 1 and relative degree 2 output errors, respectively. If we choose the linear state variables:

$$\eta_\varepsilon = \begin{bmatrix} \frac{1}{\varepsilon} y_1 \\ \frac{1}{\varepsilon} y_2 \\ \dot{y}_2 \end{bmatrix} \implies \varepsilon \dot{\eta}_\varepsilon = \underbrace{\begin{bmatrix} -K_{\bar{v}} & 0 & 0 \\ 0 & 0 & \mathbf{I} \\ 0 & -K_P & -K_D \end{bmatrix}}_{F_{\mathrm{cl},\varepsilon}} \eta_\varepsilon(t), \tag{7.33}$$

where $F_{\mathrm{cl},\varepsilon}$ is Hurwitz by definition (meaning that $\mathrm{Re}(\mathrm{eig}(F_{\mathrm{cl},\varepsilon})) < 0$). Performing a change of time scale $\tau = \frac{t}{\varepsilon}$ yields $\frac{d}{d\tau}\eta_\varepsilon(\tau) = F_{\mathrm{cl},\varepsilon}\eta_\varepsilon(\tau)$. Due to the Hurwitz assumption, for any $Q = Q^T > 0$ there exists a $P = T^T > 0$ which satisfies the Continuous-Time Lyapunov Equation (CTLE):

$$F_{\mathrm{cl},\varepsilon}^T P + P F_{\mathrm{cl},\varepsilon} = -Q. \tag{7.34}$$

This yields the Lyapunov function:

$$V(\eta_\varepsilon) = \eta_\varepsilon^T P \eta_\varepsilon \implies \frac{d(\eta_\varepsilon(\tau)}{d\tau} \leq -\underbrace{\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}}_{\gamma} V(\eta_\varepsilon(\tau)), \tag{7.35}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues. It follows from a conversion back to the original coordinates that:

$$||\eta(t)|| \leq \frac{1}{\varepsilon}\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}} e^{-\frac{\gamma}{2\varepsilon}t} ||\eta(0)||. \tag{7.36}$$

Hence the norm of $\eta(t)$ converges to zero exponentially and the selection of $0 < \varepsilon < 1$ creates *rapidly exponential stability*, and $\varepsilon = 1$ instead renders *exponential stability*. We can also view V as a function of ε, with η independent of ε, with the selection of:

$$V_\varepsilon(\eta) = \eta^T \underbrace{\mathbf{I}_\varepsilon P \mathbf{I}_\varepsilon}_{P_\varepsilon} \eta, \quad \text{with } \mathbf{I}_\varepsilon := \begin{bmatrix} \frac{1}{\varepsilon} I & 0 & 0 \\ 0 & \frac{1}{\varepsilon} I & 0 \\ 0 & 0 & I \end{bmatrix}. \tag{7.37}$$

This can ultimately be combined with Definition 7.2.0.2 and Definition 7.2.0.3 by seeing that we can write the set of admissible control inputs for the V_ε that we have just found as (7.21).

Algebraic Riccati Equation Synthesis

Rather than choosing specific control gains for ν, it is actually possible to synthesize CLFs using the continuous time algebraic Riccati equations (CARE):

$$F^TP + PF - PGR^{-1}G^TP + Q = 0, \tag{7.38}$$

for $Q = Q^T > 0$, $R = R^T > 0$, and with solution $P = P^T > 0$. This ultimately follows from the fact that the CTLE is a special case of the CARE, with no input (i.e. $GR^{-1}G^T = 0$). While the CTLE is important in that it shows how to synthesize a Lyapunov equation and show stability, the solution to the CTLE shows pointwise optimality with respect to the linear output dynamics in (7.26). To see this, we can consider the auxiliary feedback controller, $\nu = -R^{-1}G^TP\eta$, where this specific choice is just the infinite-horizon linear quadratic regulator (LQR) that minimizes:

$$\mathcal{J}_{\text{LQR}} = \frac{1}{2}\int_0^\infty \left(\eta^TQ\eta + \nu^TR\nu\right) dt. \tag{7.39}$$

We can then take the optimal cost as a Lyapunov function $V(\eta) = \eta^TP\eta$, and show that $\dot{V}$ is negative semidefinite:

$$\begin{aligned}
\dot{V} &= 2\eta^TP\dot{\eta} \\
&= 2\eta^TP(F - GR^{-1}G^TP)\eta \\
&= 2(\eta^TPF\eta - \eta^TPGR^{-1}G^TP\eta) \\
&= \eta^T(PF + F^TP)\eta - 2\eta^T(PGR^{-1}G^TP)\eta \\
&= -\eta^TQ\eta - x^TPGR^{-1}G^TP\eta,
\end{aligned}$$

where both Q and $PGR^{-1}G^TP$ are positive semidefinite by definition, so $\dot{V}$ is negative semidefinite. In terms of Definition 7.2.0.2, we can show that the P solution to the CARE yields the inequality [31], [238]:

$$\begin{aligned}
\inf_{\nu\in\mathbb{R}^m}\left[L_FV + L_GV\nu + \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}\right] &= \inf_{\nu\in\mathbb{R}^m}\left[\eta^T(F^TP + PF)\eta + 2\eta^TPG\nu + \eta^TQ\eta\right] \\
&= \inf_{\nu\in\mathbb{R}^m}\left[PGR^{-1}G^TP\eta + 2\eta^TPG\nu\right] \\
&= \inf_{\nu\in\mathbb{R}^m}\left[\eta^TPG(R^{-1}G^TP\eta + 2\nu)\right] \\
&\leq 0,
\end{aligned}$$

from the selection of $\nu = -\frac{1}{2}R^{-1}G^TP\eta$ and:

$$\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}V(\eta) \leq \lambda_{\min}(Q)||\eta||^2 \leq \eta^TQ\eta.$$

The same convergence results will follow from the selection of a (rapidly) exponentially stabilizing CLF using P_ε as in (7.37).

7.2.3 Stabilizing Hybrid Zero Dynamics

This section addresses control Lyapunov functions within the context of the Hybrid Zero Dynamics framework introduced in Chap. 3. Starting from the assumption that a system has stable zero dynamics and is shaped in such a way that it has hybrid invariance, meaning that it has hybrid zero dynamics, several conditions on the stability of a periodic orbit are established. Towards this goal, consider a hybrid control system (3.1) now in normal form (7.16):

$$\mathscr{HC} = \begin{cases} \dot{\eta} = \bar{f}(\eta, z) + \bar{g}(\eta, z)u & \text{if} \quad (\eta, z) \in \mathcal{D} \backslash S \\ \dot{z} = \omega(\eta, z) & \\ \eta^+ = \Delta_\eta(\eta^-, z^-) & \text{if} \quad (\eta, z) \in S \\ z^+ = \Delta_z(\eta^-, z^-) & \end{cases} \tag{7.40}$$

where $\eta, z, \bar{f}$, and $\bar{g}$ are defined as in (7.16), Δ_η and Δ_z are locally Lipschitz in their arguments, and the domain and guard are defined according to (3.4) and (3.6) now in terms of the normal coordinates as:

$$\mathcal{D} = \{(\eta, z) \in Y \times \mathcal{Z} \mid H(\eta, z) \geq 0\}, \tag{7.41}$$

$$S = \{(\eta, z) \in Y \times \mathcal{Z} \mid H(\eta, z) = 0, \dot{H}(\eta, z) < 0\} \tag{7.42}$$

where $H(\eta, z) : Y \times \mathcal{Z} \to \mathbb{R}$ is a continuously differentiable function where $L_{\bar{g}} H = 0$. If we now assume that the normal-form hybrid control system has continuous invariance, $\bar{f}(0, z) = 0$, and discrete invariance, $\Delta_x(0, z) = 0$, then (7.40) has hybrid zero dynamics. In other words, we have encoded satisfaction of the invariance condition, $\Delta(\mathcal{Z} \cap S) \subset \mathcal{Z}$, previously given for the full-order system (3.30). Thus the restricted hybrid dynamical system $\mathscr{H}^\alpha|_{\mathcal{Z}}$ in (3.34) can be considered. If we further assume that a RES-CLF V_ε is chosen to obtain a locally Lipschitz control law $u_\varepsilon(\eta, z) \in K_\varepsilon(\eta, z)$

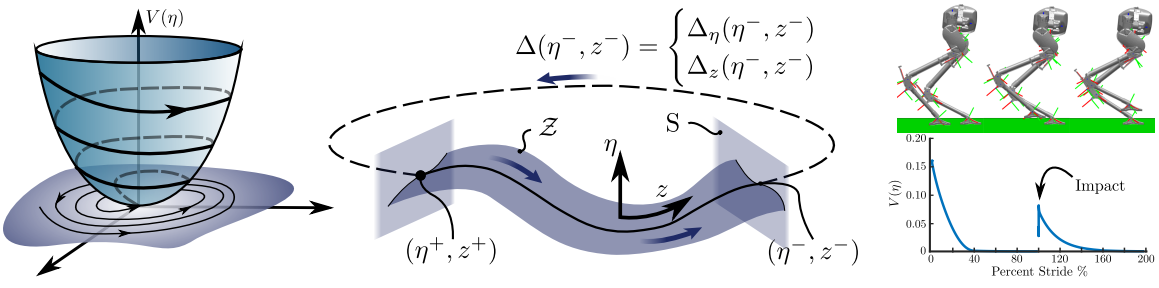

Figure 7.1: A visualization of a CLF driving a Lyapunov function to zero (left), the normal form hybrid zero dynamics representation, and a walking simulation (right). The rapid exponential zeroing of the outputs (Equation (7.22)) is critical to achieve sufficient convergence before impact (right).

that can be applied to the closed-loop hybrid system:

$$\mathscr{H}_\varepsilon = \begin{cases} \dot{\eta} = \bar{f}(\eta, z) + \bar{g}(\eta, z) u_\varepsilon(\eta, z) & \text{if} \quad (\eta, z) \in \mathcal{D} \backslash S \\ \dot{z} = \omega(\eta, z) & \\ \eta^+ = \Delta_\eta(\eta^-, z^-) & \text{if} \quad (\eta, z) \in S \\ z^+ = \Delta_z(\eta^-, z^-) & \end{cases},$$

where $u_\varepsilon(\eta, z) \in K_\varepsilon(\eta)$ implies that $u_\varepsilon(0, z) = 0$ and thus preserves the hybrid zero dynamics $\mathscr{H}^\alpha|_\mathcal{Z}$. Because the hybrid zero dynamics holds, the analysis presented in Sec. 3.2.4 on the stability of periodic orbits also follows [155]. In fact, a stronger statement can be made regarding the stability of the hybrid system.

Theorem 7.2.1. *(RES-CLF and Hybrid Zero Dynamics [31]) Let $\mathcal{O}|_\mathcal{Z}$ be an exponentially stable periodic orbit of the hybrid zero dynamics $\mathscr{H}^\alpha|_\mathcal{Z}$ transverse to $S \cap \mathcal{Z}$ and assume there exists a RES-CLF V_ε for the continuous dynamics* (7.13) *of* (7.40)*. Then there exists an $\bar{\varepsilon} > 0$ such that for all $0 < \varepsilon < \bar{\varepsilon}$ and for all Lipschitz continuous $u_\varepsilon \in K_\varepsilon(\eta, z)$, $\mathcal{O} = \iota_0(\mathcal{O}|_\mathcal{Z})$ is an exponentially stable hybrid periodic orbit of $\mathscr{H}_\varepsilon$.*

The proof of Theorem 7.2.1 can be found at [31], with the primary takeaway being that any RES-CLF controller $u_\varepsilon \in K_\varepsilon(\eta, z)$ results in a stable orbit for the full-order dynamics if one exists in the reduced order dynamics. This general concept is illustrated for robotic walking in Fig. 7.1, where a CLF is shown which renders stable zero dynamics. If the system also has hybrid invariance, then the corresponding periodic orbit is stable, and is in fact exponentially stable, with a robot shown starting from a perturbed initial condition, and driving the jump at impact to zero over time as the robot walks from step to step.

7.2.4 Control Lyapunov Functions as a Quadratic Program

The advantage of (7.18) is that it gives a large set of controllers that can result in stable walking on bipedal robots. That is, for any $u \in K(x)$, the hybrid system model of the walking robot, per the HZD framework introduced in 1.2.6, has a stable periodic gait given a stable periodic orbit in the zero dynamics [31]. This suggests that an optimization-based framework nonlinear controller synthesis is possible, where the inequality:

$$L_f V(x) + L_g V(x) u + \gamma V(\eta) \leq 0, \tag{7.43}$$

is satisfied via a QP-based controller:

$$u^* = \underset{u \in U \subset \mathbb{R}^m}{\operatorname{argmin}} \quad \frac{1}{2} u^T Q(x) u + c^T(x) u \tag{CLF-QP}$$
$$\text{s.t.} \quad L_f V(x) + L_g V(x) u \leq -\gamma V(x) \tag{CLF Convergence}$$

where $Q \in \mathbb{R}^{m \times m}$ is a symmetric positive-definite matrix and $c \in \mathbb{R}^m$ is vector.

The choice of $Q(x)$ and $c(x)$ is important in implementation. Specifically, not all choices will result in Lipschitz continuity of the resulting torque, and selecting costs which are inconsistent with the CLF convergence inequality can cause the controller to "chatter." One common choice is to use the fact that the preliminary feedback control law in the HZD and CLF constructions is feedback linearization to minimize ν in terms of the full-order states rather than u:

$$u^*(x) = \underset{u \in U \subset \mathbb{R}^m}{\operatorname{argmin}} \quad ||\mathcal{A}(x) u + \mathbf{L}_f y(x)||^2$$
$$\text{s.t.} \quad L_f V(x) + L_g V(x) u \leq -\gamma V(x)$$

with $Q(x) = \mathcal{A}^T(x)\mathcal{A}(x)$ and $c^T(x) = 2(\mathbf{L}_f y(x))^T \mathcal{A}(x)$ in terms of the original cost.

For the holonomic constraints to be satisfied in the dynamics (2.8), and thus in the QP constraint (7.43), we must either augment u with λ as an additional decision variable [225], [233], or solve for the generalized force explicitly, assuming the holonomic constraint is satisfied:

$$\lambda = (J_c(q) D^{-1}(q) J_c^T(q))^{-1} \left(J_c(q) D^{-1}(q) (H(q, \dot{q}) - Bu) - \dot{J}_c(q, \dot{q}) \dot{q} \right), \tag{7.44}$$

and substitute back into the expression (2.8) to remove it as a free variable.

Remark 4. *In the special case where the cost function is simply the torque squared and the only constraint is the Lyapunov convergence constraint, the CLF-QP:*

$$u^* = \underset{u \in U \subset \mathbb{R}^m}{\operatorname{argmin}} \quad u^T u$$
$$\text{s.t.} \quad L_f V(x) + L_g V(x) u \leq -\gamma V(x)$$

has an analytical solution given by the ***min-norm controller****:*

$$u = \begin{cases} -\dfrac{(L_f V(x) + \gamma V(x)) L_g V(x)}{L_g V(x) (L_g V(x))^T} & \text{if} \quad L_f V(x) + \gamma V(x) > 0 \\ 0 & \text{if} \quad L_f V(x) + \gamma V(x) \leq 0. \end{cases} \tag{7.45}$$

Convergence Relaxations for Implementation and Feasibility

The optimization formulation of CLFs allows for additional constraints and objectives to be incorporated into the optimization. These constraints can include various things which are important for realization on actual robotic platforms such as torque constraints for input saturation, friction constraints, or unilaterality conditions on contact forces. These constraints naturally fall under the general form introduced in Sec. 7.1 as the inequality:

$$C_I(x)u \leq d_i(x). \tag{7.46}$$

One of the downsides to incorporating additional constraints into the problem is that it may not be possible to satisfy them concurrently with (7.43). This means that a relaxation variable, δ, must be added to penalize violation of (7.43):

$$\begin{aligned} u^* = \underset{u,\delta}{\operatorname{argmin}} \quad & \frac{1}{2}u^T Q(x)u + c^T(x)u + \rho\delta^2 \qquad \text{(CLF-QP-}\delta\text{)} \\ \text{s.t.} \quad & L_f V(x) + L_g V(x)u \leq -\gamma V(x) + \delta \\ & C_I(x)u \leq d_i(x) \end{aligned}$$

where ρ is a large positive weight that penalizes relaxation of the CLF inequality.

In Sec. 7.3, we will develop a series of controllers which necessitate a relaxation of CLF convergence in order to be experimentally realized in Chap. 8. While allowing this violation enables more advanced controllers, it can destroy the formal stability conditions which we have introduced for the hybrid zero dynamics framework. In Sec. 7.3, we therefore provide sufficient conditions under which a relaxed CLF-QP controller can retain exponential stability in the hybrid periodic orbit.

7.3 Control Lyapunov Function-based Inverse Dynamics

Model-based control methods can help enable dynamic and compliant motion of robots while achieving remarkable control accuracy. However, implementing such techniques on floating base robots is non-trivial due to model inaccuracy, underactuation, dynamically changing contact constraints, and possibly conflicting objectives for the robot [225], [239]. Unlike their classical counterparts, optimization-based approaches to handling these control problems allow for the inclusion of physical constraints that the system is subject to [174], [240]. Partially as a consequence of this feature, QP-based controllers have been increasingly used to stabilize real-world systems on complex robotic platforms without the need to algebraically produce a control law or enforce convergence guarantees [187], [214], [228].

These examples, however, typically do not consider periodic notions of stability for highly underactuated systems; systems which often require additional convergence guarantees in order to realize stability. It was shown in [31] that through the use of a RES-CLF, coupled with HZD [30], [138], a wide class of controllers can be designed to create rapidly exponentially convergent hybrid periodic orbits. It was also shown that this class of controllers can be posed as a QP, in which the convergence is enforced via an inequality constraint, forming a CLF-QP [31], [225]. Often, robotic systems cannot produce sufficient convergence to dynamic motions without violating physical constraints. One approach to address this conflict is to relax convergence guarantees, which allows (local) drift in the control objectives to accommodate feasibility. This class of controllers has since been used to achieve dynamic locomotion on robotic systems both in simulation [233], [234], [241], [242] and on hardware [230].

While high level task-space controllers based on inverse dynamics approaches pose similar problems as CLF-QPs, they have traditionally not been formulated in the same way. In implementations of CLF-QPs, the vector fields associated with robotic systems are typically utilized, which involves costly computations. Alternatively, in task-based controllers, the dynamics are an equality constraint. Here, objectives are driven towards their targets through PD controllers in the cost [187]. There have been several connections shown in related research [59], [229], where control Lyapunov functions were included in an inverse dynamics controller via an LQR in the cost. In this section, we aim to repurpose several of the more mature concepts from inverse dynamics-based approaches and demonstrate a more efficient CLF inspired formulation.

Towards this goal, this section will consider only floating-base robotic systems such as

those described in Chap. 2. As they will be used extensively throughout this section, recall that the dynamics of these robotic systems can be formulated using the method of Lagrange, with holonomic constraints [131]:

$$D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_c^T(q)\lambda_c \tag{7.47}$$

$$J_c(q)\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q} = 0, \tag{7.48}$$

where $D(q)$ is the inertia matrix, $H(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q) + F$ is the vector sum for the Coriolis, centrifugal, gravitational, and additional non-conservative generalized forces, B is the actuation matrix, and the Jacobian of the holonomic constraints is $J_c(q) = \partial\eta_c(q)/\partial q$ with its corresponding constraint wrenches $\lambda_c \in \mathbb{R}^{m_c}$. This can be converted to an ODE in the form of (2.1) as:

$$f(x) = \begin{bmatrix} \dot{q} \\ -D^{-1}(q)\left(J_c^T(q)\lambda_c - H(q,\dot{q})\right) \end{bmatrix}, \qquad g(x) = \begin{bmatrix} 0 \\ D(q)^{-1}B \end{bmatrix}. \tag{7.49}$$

The specific form of (7.47) and (7.48) will be leveraged throughout this chapter to enable CLF-QP formulations that are realizable on actual hardware in Chap. 8.

7.3.1 Revisiting Feedback Linearization

Taking inspiration from inverse dynamics approaches, we return to Sec. 3.2.2, where the auxiliary control input, ν, for a feedback linearizing controller is set to equal the second time derivative of the outputs. Suppose that for (7.49), there is a set of outputs $y(x) = (y_1(q,\dot{q})^T, y_2(q)^T)^T$ of vector relative degree 1 and 2, respectively, on a region of interest; that is for $y_1(q,\dot{q}) \in \mathbb{R}^{m_1}$ and $y_2(q) \in \mathbb{R}^{m_2}$ with $m = m_1 + m_2$ we assume the vector relative degree is 1 for y_1 and the 2 for y_2:

$$y_1(q,\dot{q},t) = y_1^a(q,\dot{q}) - y_1^d(\tau(t,q)) \tag{7.50}$$

$$y_2(q,t) = y_2^a(q) - y_2^d(\tau(t,q)), \tag{7.51}$$

where y^a and y^d are the actual and desired outputs, and $\tau(t,q)$ is some parameterization of time for the desired outputs to evolve on. Taking the derivatives of the outputs along $f(x)$ and $g(x)$, we obtain:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial q} & \frac{\partial y_1}{\partial \dot{q}} \\ \frac{\partial}{\partial q}\left(\frac{\partial y_2}{\partial q}\dot{q}\right) & \frac{\partial y_2}{\partial q} \end{bmatrix} f(x)}_{\mathbf{L}_f y(x)} + \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial \dot{q}} \\ \frac{\partial y_2}{\partial q} \end{bmatrix} g(x)}_{\mathcal{A}(x)} u, \tag{7.52}$$

where $\mathcal{A}(x)$ is the decoupling matrix, which is invertible in the case of a (vector) relative degree. This implies that the system (7.49) is feedback linearizable, and we

can then prescribe the following control law:

$$u(x) = \mathcal{A}^{-1}(x)\Big(- \mathbf{L}_f y(x) + \nu\Big) \qquad \Longrightarrow \qquad \begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \nu, \tag{7.53}$$

now in the form of (3.8) where ν is an auxiliary feedback control value.

This can equivalently be done by taking the derivatives of the outputs (3.9) and (3.10) in terms of the system accelerations instead of along the vector fields $f(x)$ and $g(x)$:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial q} \\ \frac{\partial}{\partial q}\left(\frac{\partial y_2}{\partial q}\dot{q}\right) \end{bmatrix}}_{\dot{J}_y} \dot{q} + \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial \dot{q}} \\ \frac{\partial y_2}{\partial q} \end{bmatrix}}_{J_y} \ddot{q}. \tag{7.54}$$

We then return to (7.30) where ν is set to equal the second time derivative of the output. Rather than directly choosing an input, u, we can instead solve for an acceleration, $\ddot{q}$, that generates an equivalent response in the outputs.

Theorem 7.3.1. *For a robotic system with dynamics* (7.47) *and outputs of the form* (3.9) *and* (3.10)*, where $D(q)$ is positive definite (and therefore invertible) and the outputs are independent (i.e., the Jacobian matrix $J_y(q)$ is not rank-deficient), then any controller in the set:*

$$K_{IO}(q, \dot{q}) = \{u \in U : \ddot{q} = J_y^{\dagger}(q)(-\dot{J}_y(q, \dot{q})\dot{q} + \nu)\}, \tag{7.55}$$

elicits the same response in the output dynamics as the feedback linearizing input,

$$u = \mathcal{A}^{-1}(x)(-\mathbf{L}_f y(x) + \nu). \tag{7.56}$$

Proof. Using (7.54), $\ddot{q}$ can be chosen to satisfy

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = J_y(q)\ddot{q} + \dot{J}_y(q, \dot{q})\dot{q} = \nu. \tag{7.57}$$

By constraining:

$$\ddot{q} = J_y^{\dagger}(q)(-\dot{J}_y(q, \dot{q})\dot{q} + \nu), \tag{7.58}$$

where $J_y^{\dagger}$ is a right pseudo inverse of the full rank matrix J_y, with $J_y J_y^{\dagger} = I$, and the outputs evolve as:

$$\begin{aligned} J_y(q)\ddot{q} + \dot{J}_y(q, \dot{q})\dot{q} &= \dot{J}_y\dot{q} + J_y J_y^{\dagger}(-\dot{J}_y\dot{q} + \nu) \\ &= \nu \\ &= \begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \mathbf{L}_f y(x) + \mathcal{A}(x)u. \end{aligned}$$

□

This fact illustrates a significant connection back to the inverse dynamics controllers implemented in Sec. 6.2 and outlined as background in Sec. 7.1. This concept is not new, and the connection between inverse dynamics (or computed torque) for manipulator dynamics and feedback linearization has been extensively studied in the literature [243], along with their connections to underactuated or rank deficient tracking problems [244]. To see this more clearly, note that we can transform the pair of equations in (7.47) and (7.48) into the constrained dynamics (2.14) (or alternatively in the form of (2.19)):

$$D(q)\ddot{q} + \hat{H}(q,\dot{q}) = \hat{B}u, \tag{7.59}$$

which effectively pins the robot and makes the equations of motion no longer dependent on the constraint forces. If we substitute the acceleration found in (7.58) into these dynamics, we can simply perform several steps of algebra to obtain the feedback linearizing controller:

$$\begin{aligned}
D(q)\left(J_y^\dagger(q)(-\dot{J}_y(q,\dot{q}) + \nu)\right) + \hat{H}(q,\dot{q}) &= \hat{B}u \\
J_y^\dagger(-\dot{J}_y(q,\dot{q}) + \nu) &= D^{-1}(q)(\hat{B}u - \hat{H}(q,\dot{q})) \\
\nu &= J_y(q)D^{-1}(q)(\hat{B}u - \hat{H}(q,\dot{q})) + \dot{J}_y(q,\dot{q})\dot{q} \\
&= J_y(q)\ddot{q} + \dot{J}_y(q,\dot{q})\dot{q} \\
&= \mathbf{L}_f y(x) + \mathcal{A}(x)u.
\end{aligned}$$

We can then use this to design an inverse dynamics feedback controller for floating-base manipulator dynamics which has similar convergence properties as were given for feedback linearization in Chap. 3 while directly accounting for the impact that holonomic constraints have on the accelerations. Consider the inverse dynamics control law posed by Aghili [143], where a specific case by Mistry [142] was used on Cassie in Sec. 6.2:

$$P(q)\left(D(q)\ddot{q} + H(q,\dot{q})\right) = P(q)Bu, \tag{7.60}$$

where $P(q) = I - J_c^\dagger(q)J_c(q)$ and the inverse-dynamics control input is:

$$u = (P(q)B)^\dagger P(q)\left(D(q)\ddot{q}_d + H(q,\dot{q})\right). \tag{7.61}$$

Let us now substitute this controller into (7.47), and apply the constraint projection (noting that $P(q)$ is an annihilator for the constraint force, i.e. $P(q)J_c^T(q)\lambda_c = 0$):

$$D(q)\ddot{q} + H(q,\dot{q}) = B\left[(PB)^\dagger P(q)\left(D(q)\ddot{q}_d + H(q,\dot{q})\right)\right] + J_c^T(q)\lambda_c, \tag{7.62}$$

$$P(q)\left(D(q)\ddot{q} + H(q,\dot{q})\right) = P(q)B(PB)^\dagger P(q)\left(D(q)\ddot{q}_d + H(q,\dot{q})\right), \tag{7.63}$$

$$P(q)\left(D(q)\ddot{q} + H(q,\dot{q})\right) = P(q)\left(D(q)\ddot{q}_d + H(q,\dot{q})\right), \tag{7.64}$$

which implies that $\ddot{q} = \ddot{q}_d$ if $\dot{q} \in \mathcal{N}(J_c(q))$ and $\ddot{q}_d$ is chosen such that it implicitly satisfies (7.48). This allows us to effectively use the accelerations as a control input, so long as they are chosen in a manner which is consistent with the holonomic constraints.

As was already shown in Theorem 7.3.1, we can design a desired acceleration to elicit an equivalent response in the output dynamics to the feedback linearizing controller. However, the floating-base inverse dynamics controller shown in (7.61) has an additional requirement on this feedback acceleration with respect to (7.48). We can use the same definition of $P(q)$ in Sec. 2.1.3 for (2.16) to obtain a desired acceleration which is orthogonal to the holonomic constraints:

$$\ddot{q}_y^{\perp} = J_c^{\dagger}(q)\dot{J}_c(q,\dot{q})\dot{q} + (I - J_c^{\dagger}J_c(q))\ddot{q}. \tag{7.65}$$

This then transforms the condition (7.58), and therefore the output dynamics (7.54):

$$\nu = J_y(q)\ddot{q}^{\perp} + \dot{J}_y(q,\dot{q})\dot{q} \tag{7.66}$$

$$= \underbrace{J_y(q)(I - J_c^{\dagger}J_c(q))}_{J_{\bar{y}}^{\perp}(q)}\ddot{q} + \underbrace{\left(\dot{J}(q,\dot{q}) + J_y(q)J_c^{\dagger}(q)\dot{J}_c(q,\dot{q})\right)}_{\dot{J}_{\bar{y}}^{\perp}(q,\dot{q}}\dot{q} \tag{7.67}$$

and assuming that $J_c^{\dagger}(q)$ is invertible (i.e. that the outputs and holonomic constraints are linearly independent and $J_c^{\dagger}(q)$ thus has full row rank), application of Theorem 7.3.1 then gives the set of controllers which elicit an equivalent reaction to feedback linearization written now for the constrained floating-base manipulator dynamics as:

$$K_{IO}^{\perp}(q,\dot{q}) = \{u \in U : \ddot{q} = (J_y^{\perp}(q))^{\dagger}(-\dot{J}_y^{\perp}(q,\dot{q})\dot{q} + \nu)\}. \tag{7.68}$$

We can then design ν according to (3.25):

$$\nu = \begin{bmatrix} -\frac{1}{\varepsilon}K_{\bar{v}}y_1 \\ -\frac{1}{\varepsilon^2}K_P y_2 - \frac{1}{\varepsilon}K_D\dot{y}_2 \end{bmatrix}. \tag{7.69}$$

Because we have already shown that (7.61) renders $\ddot{q} = \ddot{q}_d$, application of Theorem 7.3.1 means that the assertions in Chap. 3 on the exponential stability of the zero dynamics hold for (7.61) with $\ddot{q}$ given by (7.68) and the selection of (7.69) as the auxiliary feedback control law.

This approach has several benefits when applied to complex systems and in implementation on hardware. Perhaps the two largest incentives for using this approach is that the actual controller does not require any inversions of the mass-inertia matrix, and the constraint forces are eliminated through a simple kinematic projection rather than back-solved through a highly model dependent series of calculations (see (2.13)). This not only reduces computational load, but sensitivity to modeling errors.

7.3.2 Inverse Dynamics Quadratic Programs with CLFs

In this section, we return to the concept of a QP which can solve the inverse dynamics problem for a floating-base robot, as was shown in the literature review outlined in Sec. 7.1. Despite the connections shown between inverse dynamics and feedback linearization in Sec. 7.3.1, CLF-based controllers have only been successfully implemented on hardware in planarized and low-dimensional robots with the min-norm analytical solution (7.45) [31], as a very minimal QP [230], or indirectly by simulating the nominal system and tracking the resulting motions via a PD controller [178]. There are several outstanding issues which we suggest may be influencing this lack of successful implementations in the literature.

The first significant difficulty in realizing optimization and model-based controllers, and therefore in implementing CLFs, is in obtaining accurate models for the high-DOF and highly-coupled nonlinear equations of motion. In the earliest implementations of min-norm CLF controllers, significant system identification was necessary in order to construct effective controllers [25]. To mitigate these issues, there has been recent exploration of robust CLF formulations [245], machine learning methods to identify Lyapunov function errors [246], and machine learning to account for model inaccuracy in the continuous dynamics of the robot [247]. While these model discrepancies may be large in some cases, the successes of (ID-QP) controllers on highly complex humanoids as ATLAS [59] shows that it is possible. Thus the aim of the approach presented in this thesis will not directly address model uncertainty and will instead focus on how the formulation of the problem can influence its behavior on real systems.

Definition 7.3.1.1. *Given a set of outputs* (7.15) *for the floating-base robotic control system described by* (7.47) *and* (7.48)*, the* ***inverse dynamics control Lyapunov function quadratic program*** *(ID-CLF-QP) with decision variables* $\mathcal{X} = [\ddot{q}^T, u^T, \lambda^T]^T \in \mathbb{X}_{\text{ext}} := \mathbb{R}^n \times U \times \mathbb{R}^{m_c}$ *is given as:*

$$
\begin{aligned}
\mathcal{X}^* = k^*_{\text{idclf}}(x) = \underset{\mathcal{X} \in \mathbb{X}_{\text{ext}}}{\text{argmin}} \quad & \frac{1}{2}\mathcal{X}^T Q(x)\mathcal{X} + c^T(x)\mathcal{X} && \text{(ID-CLF-QP)} \\
\text{s.t.} \quad & L_F V(x) + L_G V(x)\left(\dot{J}_y(q,\dot{q})\dot{q} + J_y(q)\ddot{q}\right) \leq -\gamma V(x) \\
& D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_c^T(q)\lambda \\
& J_c(q)\ddot{q} + \dot{J}_c(q)\dot{q} = 0
\end{aligned}
$$

with $Q(x) = Q^T(x) > 0$ *and real vector* $c(x) \in \mathbb{R}^{n+m+m_c}$.

Here we have termed the QP with the phrase "inverse dynamics" as it is determining a control input u based on convergence criteria imposed on the generalized accelerations,

$\ddot{q}$, through satisfaction of the equations of motion (7.47) and (7.48). Perhaps the most significant observation of Definition 7.3.1.1 is that we have traded an increased number of decision variables for a set of equality constraints that do not require any matrix inversions in a similar manner to the DAE representation of the dynamics (3.47) used in our direct collocation trajectory optimization. This is particularly relevant to implementation as it has been shown that the condition number of the joint space inertia matrix increases quartically with the length of a kinematic chain [130]. Repeated inversions of this matrix therefore may be an obvious source of numerical stiffness, and can lead to controller degradation on hardware [220]. For complex multi-link robots, such as bipedal robots, these condition numbers are often exceptionally large (for full humanoids sometimes on the order of 10^8). In addition, performing the required inversions for evaluating the vector fields (2.8) are very computationally expensive, and can often violate strict timing requirements when implementing these controllers on hardware.

The main goal of this section is to construct a framework around the (ID-CLF-QP) controller which motivates its use as a stabilizing controller for HZD locomotion. This will build on the concepts outlined for CLFs in Sec. 7.2 to establish exponential stability of the hybrid system. Specifically, let $\mathcal{O}$ be a periodic orbit of the zero dynamics $\dot{z} = \omega(0, z)$ and assume that $\mathcal{O} \subset \mathcal{Z}$ is exponentially stable. Then the following result states that if the resulting controller from (ID-CLF-QP) using the RES-CLF condition (7.74) is locally Lipschitz and unique, then it is possible to stabilize $\mathcal{O}$ in the full-order dynamics.

Main Theorem 7.3.1. *(RES-ID-CLF-QP and HZD) Assume that* (ID-CLF-QP) *is locally Lipschitz and unique for all points in a neighborhood of an exponentially stable periodic orbit, $\mathcal{O}|_{\mathcal{Z}}$, of the hybrid zero dynamics $\mathscr{H}^{\alpha}|_{\mathcal{Z}}$ transverse to $S \cap \mathcal{Z}$. Then for the* (ID-CLF-QP) *controller with choice of RES-CLF, $V_\varepsilon(x)$, $u^*_\varepsilon(x) = k_{\text{idclf},\varepsilon}(x)$, there exists an $\bar{\varepsilon} > 0$ such that for all $0 < \varepsilon < \bar{\varepsilon}$, $\mathcal{O} = \iota_0(\mathcal{O}|_{\mathcal{Z}})$ is an exponentially stable hybrid periodic orbit of $\mathscr{H}_\varepsilon$.*

Before we proceed with a proof for Theorem 7.3.1, we will first establish that the pointwise optimal control action obtained from (ID-CLF-QP) in fact renders stability of the transverse dynamics (7.26) in a similar manner to (CLF-QP). Using this result, we will show that the (ID-CLF-QP) can be transformed into an equivalent (CLF-QP) and then discuss the connection between the ES-CLF and RES-CLF constraint within the (ID-CLF-QP) problem.

Lemma 7.3.1.1. *The pointwise optimal solution* $\mathcal{X}^* = k^*_{\text{idclf}}(x)$ *of* (ID-CLF-QP) *yields a control action within the set of admissible inputs for a CLF given by Definition 7.2.0.2:*

$$k^*_{\text{idclf}}(x) \in K_u(x) = \{u \in \mathbb{R}^m : L_f V(x) + L_g V(x) u \leq -\gamma V(x)\}. \tag{7.70}$$

As a result, if u^* *taken from* $\mathcal{X}^*$ *is locally Lipschitz and if the zero dynamics* $\dot{z} = \omega(0, z)$ *is locally exponentially stable, then* (ID-CLF-QP) *is a locally exponentially stabilizing controller for the closed-loop system in* (3.31)*.*

Proof. Application of Theorem 7.3.1 means that the collection of constraints:

$$\begin{cases} L_F V(x) + L_G V(x) \left(\dot{J}_y(q, \dot{q}) \dot{q} + J_y(q) \ddot{q} \right) \leq -\gamma V(x) \\ D(q)\ddot{q} + H(q, \dot{q}) = Bu + J_c^T(q)\lambda \\ J(q)\ddot{q} + \dot{J}_c(q)\dot{q} = 0 \end{cases}$$

can be rewritten as a single inequality by solving (7.47) and (7.48) for $\ddot{q}$ and substituting into the CLF inequality to obtain:

$$L_f V(x) + L_g V(x) u \leq -\gamma V(x), \tag{7.71}$$

which is the convergence condition required for exponential convergence provided in Definition 7.2.0.2. Because we can analytically show this equivalence, the existing CLF convergence conditions in Definition 7.2.0.2 apply to (ID-CLF-QP). Just as in Sec. 7.2, for any Lipschitz continuous feedback control law $u \in K_u(x)$, the inequalities in Definition 7.2.0.2 imply that the solutions to the closed-loop system:

$$\begin{aligned} \dot{\eta} &= \bar{f}(\eta, z) + \bar{g}(\eta, z) u \\ \dot{z} &= \omega(\eta, z) \end{aligned}$$

satisfy (with $\gamma = c_3$):

$$\begin{aligned} \dot{V}(\eta, u^*(\eta, z)) \leq -c_3 V(\eta) \qquad &\Rightarrow \qquad V(\eta(t)) \leq e^{-c_3 t} V(\eta(0)) \\ &\Rightarrow \qquad \|\eta(t)\| \leq \sqrt{\frac{c_2}{c_1}} e^{-\frac{c_3}{2} t} \|\eta(0)\|. \end{aligned}$$

□

Remark 5. *As was stated in Sec. 7.2 for* (CLF-QP)*, not all choices of the cost terms for* $Q(x)$ *and* $c(x)$ *for* (ID-CLF-QP) *will result in Lipschitz continuity of the resulting QP controller [248], [249]. If they are selected in a way which conflicts with the convergence constraint* $\dot{V}(\eta, u^*(\eta, z)) \leq -\gamma V(\eta)$*, then the input can instantaneously change and create a discontinuity when interfering with the inequality.*

Remark 6. *If there exists a set of parameters α^* that yields a feasible gait for the* (HZD Optimization) *problem, then there exists a family of CLF controllers* (7.21) *that can exponentially stabilize the transverse dynamics. Therefore, by Lemma 7.3.1.1, there also exists a* (ID-CLF-QP) *controller which can stabilize this problem.*

One of the most important consequences of Lemma 7.3.1.1 is that we can pose a (ID-CLF-QP) controller to stabilize the zero dynamics surface during continuous phases of motion for underactuated robotic systems. In fact, we can also show that a (ID-CLF-QP) can be analytically converted into an (CLF-QP).

Lemma 7.3.1.2. *For any given cost $\mathcal{J}_{\text{id}}(x, \mathcal{X}) = \frac{1}{2}\mathcal{X}^T Q_{\text{id}}(x)\mathcal{X} + c_{\text{id}}^T(x)\mathcal{X}$ with $Q_{\text{id}}(x) = Q_{\text{id}}^T(x) > 0$ and real vector $c_{\text{id}}(x) \in \mathbb{R}^{n+m+m_c}$ of* (ID-CLF-QP)*, there exists a cost $\mathcal{J}_{\text{u}}(x, u)$ which is quadratic with respect to u for* (CLF-QP) *such that the problems are analytically equivalent.*

Proof. To begin, we will simply establish a linear transformation between the decision variables and then plug them into $\mathcal{J}_{\text{id}}(x, \mathcal{X})$ to find a cost. We previously showed in (2.13) that the analytical solution to (7.47) and (7.48) for the constraint wrenches is:

$$\lambda_c(q, \dot{q}, \ddot{q}, u) = -\left[J_c(q)D^{-1}(q)J^T(q)\right]^{-1}\left[\dot{J}_c(q, \dot{q})\dot{q} + J_c(q)D^{-1}(q)\left(Bu - H(q, \dot{q})\right)\right].$$

In Sec. 2.1.3, it was shown how the substitution of this constraint force to (7.47) results in the constrained equations of motion:

$$D(q)\ddot{q} + \hat{H}(q, \dot{q}) = \hat{B}(q, \dot{q})u,$$

where we can solve for $\ddot{q}$, and thus form an expression relating u to $\mathcal{X}$:

$$\underbrace{\begin{bmatrix}\ddot{q}\\ u\\ \lambda_c\end{bmatrix}}_{\mathcal{X}} = \underbrace{\begin{bmatrix}D^{-1}\hat{B}\\ I\\ \left[J_c D^{-1} J_c^T\right]^{-1} J_c D^{-1}\hat{B}\end{bmatrix}}_{A_u} u + \underbrace{\begin{bmatrix}-D^{-1}\hat{H}\\ 0\\ \left[J_c D^{-1} J_c^T\right]^{-1}\left(\dot{J}_c\dot{q} - J_c D^{-1} H\right)\end{bmatrix}}_{b_u} \tag{7.72}$$

where dependencies were dropped for the sake of compact presentation. Directly substituting this relation yields a quadratic cost:

$$\begin{aligned}\mathcal{J}_{\text{id}}(x, \mathcal{X}) &= \mathcal{X}^T Q_{\text{id}}\mathcal{X} + 2c_{\text{id}}^T\mathcal{X}\\ &= \left[A_u u + b_u\right]^T Q_{\text{id}}\left[A_u u + b_u\right] + 2c_{\text{id}}^T\left[A_u u + b_u\right]\\ &= u^T A_u^T Q_{\text{id}} A_u u + 2\left[b_u^T Q_{\text{id}} + c_{\text{id}}^T\right]A_u u + \left[b_u^T Q_{\text{id}} b_u + 2c_{\text{id}}^T b_u\right]\\ &=: \mathcal{J}_u(x, u).\end{aligned} \tag{7.73}$$

We have already shown in the proof for Lemma 7.3.1.1 that the constraints for (ID-CLF-QP) analytically reduce to (CLF-QP), which completes the proof. □

Remark 7. *While Lemma 7.3.1.1 demonstrates exponential stability (ES-CLF) for the controller shown in* (ID-CLF-QP)*, an implication of the equivalence shown in Lemma 7.3.1.1 and Lemma 7.3.1.2 is that we can similarly pose a RES-CLF, $V_\varepsilon(x)$ within the* (ID-CLF-QP) *formulation to obtain rapidly exponential stability with bounds given by Definition 7.2.0.3 using the inequality:*

$$L_F V_\varepsilon(x) + L_G V_\varepsilon(x) \left(\dot{J}_y(q,\dot{q})\dot{q} + J_y(q)\ddot{q} \right) \leq -\frac{\gamma}{\varepsilon} V_\varepsilon(x), \tag{7.74}$$

where $V_\varepsilon(x)$ can be constructed according to (7.37)*.*

Proof of Main Theorem 7.3.1:

Using Remark 6, let us begin by posing a (ID-CLF-QP) with a RES-CLF condition (7.74) on the CLF convergence:

$$\begin{aligned} \mathcal{X}_\varepsilon^* = k_{\text{idclf},\varepsilon}^*(x) = \underset{\mathcal{X}\in\mathbb{X}_{\text{ext}}}{\text{argmin}} \quad & \frac{1}{2}\mathcal{X}^T Q_{\text{id}}(x)\mathcal{X} + c_{\text{id}}^T(x)\mathcal{X} \\ \text{s.t.} \quad & L_F V_\varepsilon(x) + L_G V_\varepsilon(x) \left(\dot{J}_y(q,\dot{q})\dot{q} + J_y(q)\ddot{q} \right) \leq -\frac{\gamma}{\varepsilon} V_\varepsilon(x) \\ & D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_c^T(q)\lambda \\ & J_c(q)\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q} = 0. \end{aligned}$$

The primary consequence of Lemma 7.3.1.1 and Lemma 7.3.1.2 is that this problem will render an analytically equivalent control action to the RES-(CLF-QP):

$$\begin{aligned} u_\varepsilon^* = \underset{u\in U\subset\mathbb{R}^m}{\text{argmin}} \quad & \mathcal{J}_u(x,u) \\ \text{s.t.} \quad & L_f V(x) + L_g V_\varepsilon(x) u \leq -\frac{\gamma}{\varepsilon} V_\varepsilon(x). \end{aligned}$$

Thus, the control action belongs to the family of RES-CLF controllers given by:

$$k_{\text{idclf},\varepsilon}^*(x) \in K_\varepsilon(x) = \left\{ u \in \mathbb{R}^m : L_f V(x) + L_g V_\varepsilon(x) u \leq -\frac{\gamma}{\varepsilon} V_\varepsilon(x) \right\}. \tag{7.75}$$

Because the pointwise optimal control action is thus a RES-CLF, if $\mathcal{X}_\varepsilon^*$ is locally Lipschitz and unique, then Theorem 7.2.1 applies to $k_{\text{idclf},\varepsilon}^*(x)$, completing the proof. □

7.3.3 Cost Functions and Gait Parameterization for HZD

One of the clear benefits to (ID-CLF-QP) is that there exists a wide range of costs that can be designed without needing complex expressions, as the decision variables are affine with respect to mostly kinematic matrices in the equations of motion (7.47)-(7.48) and output dynamics (7.26). In other words, while it was shown in Lemma 7.3.1.2 that costs can be shown transformed into a form which is affine to the joint torques, u, they are considerably more complex expressions involving the full system dynamics. This will be illustrated through a discussion of several of the most common costs applied to CLFs in the literature.

Perhaps the simplest cost function that can be applied to (ID-CLF-QP) is to minimize the square of the joint torques for the robot:

$$\mathcal{J}_u(\mathcal{X}) := u^T u, \tag{7.76}$$

for which (ID-CLF-QP) has an analytical solution given by the min-norm controller if it does not contain constraints other than the CLF derivative and dynamics. However, driving the response towards zero torque may not be the most desirable action for a controller to take if we already have some information on the preliminary feedback controller (feedback linearization), which was prescribed in Chap. 3 to render the output dynamics as a linear system:

$$J_{\text{IO}}(x, \mathcal{X}) := ||J_y(q)\ddot{q}_y + \dot{J}_y(q, \dot{q})\dot{q}||^2. \tag{7.77}$$

The equivalence of the expression in $J_{\text{IO}}(x, \mathcal{X})$ to (3.23) can be seen by Theorem 7.3.1.

Perhaps the most important observation that we should make is that in order for (ID-CLF-QP) to be solved uniquely, the Hessian matrix, $Q(x)$, must be positive definite and therefore also full rank. However, the costs posed up until this point each deal with only a subset of the decision variables. One of the most common ways to address this is to regularize the decision variables about a nominal value:

$$J_{\text{reg}}(x, \mathcal{X}) = ||\mathcal{X} - \mathcal{X}^*||^2. \tag{7.78}$$

In fact, if we have solved for a stable walking gait using the HZD methodology outlined in Chap. 3 and using (HZD Direct Collocation), then we have already done the work of obtaining a parameterized piecewise polynomial for $\mathcal{X}^*_\alpha(t)$, which is the nominal acceleration, constraint wrenches, and inputs when the robot is operating on the stable hybrid periodic orbit.

Proposition 7.3.1.1. *(See [250], Chapter 4) Suppose that* $\mathbf{w}(\alpha^*)$ *describes a feasible walking gait solving* (HZD Direct Collocation)*. Then the piecewise polynomial solution* $\varphi^*(t)$ *determined by the NLP solution,* $\{T^*, q^*(t), \dot{q}^*(t), \ddot{q}^*(t), \lambda^*(t), u^*(t)\}$*, is hybrid invariant under the virtual constraint feedback control law* (3.8) *with parameters* α^**, i.e.* $\varphi^*(t) \subset \mathcal{Z}_{\alpha^*}$*.*

Proposition 7.3.1.2. *Consider an* (ID-CLF-QP) *with two cost terms:*

$$\begin{aligned} \mathcal{X}^* = \underset{\mathcal{X}\in\mathbb{X}_{\text{ext}}}{\text{argmin}} \quad & \mathcal{J}_z(x,\mathcal{X}) + ||\mathcal{X} - \mathcal{X}^*_\alpha(t)||^2 \\ s.t. \quad & L_F V_\varepsilon(x) + L_G V_\varepsilon(x)\left(\dot{J}_y(q,\dot{q})\dot{q} + J_y(q)\ddot{q}\right) \leq -\frac{\gamma}{\varepsilon}V_\varepsilon(x) \\ & D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_c^T(q)\lambda \\ & J_c(q)\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q} = 0 \end{aligned}$$

where $\mathcal{J}_z(x,\mathcal{X})$ *is defined in such a way that* $\mathcal{J}_z(\mathcal{X})|_{\mathcal{Z}} \equiv 0$ *(we can see that* (7.77) *is an example of such a cost as* $y(x) \equiv 0$ *when* $\eta(x) \equiv 0$*). Then when the robot is on the zero dynamics surface (i.e.* $\dot{z} = \omega(0,z)$*), the optimal control action is* $\mathcal{X}^* = \mathcal{X}^*_\alpha(t)$*.*

Proof. Proposition 7.3.1.1 means that the solution to (HZD Direct Collocation) lies on the hybrid invariant zero dynamics surface of the corresponding walking gait. Thus, when the robot is on the zero dynamics surface (i.e. $\dot{z} = \omega(0,z)$), the QP constraints vanish:

$$0 \equiv \begin{cases} L_F V_\varepsilon(x) + L_G V_\varepsilon(x)\left(\dot{J}_y(q,\dot{q})\dot{q} + J_y(q)\ddot{q}\right) + \frac{\gamma}{\varepsilon}V_\varepsilon(x) \\ D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_c^T(q)\lambda \\ J_c(q)\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q} = 0 \end{cases}$$

since they are implicitly satisfied on the solution $\varphi^*(t) \subset \mathcal{Z}_{\alpha^*}$ if $\mathbf{w}(\alpha^*)$ is a feasible solution to (HZD Direct Collocation). Further, the cost $\mathcal{J}_z(\mathcal{X})|_{\mathcal{Z}} \equiv 0$ by definition, and thus the optimal control action is given by $\mathcal{X}^* = \mathcal{X}^*_\alpha(t)$. □

Incentivized convergence: One observation of the standard (CLF-QP) is that it does not incentivize faster convergence rates than the chosen γ if control bandwidth is available. This can lead to intermittent triggering of the CLF derivative inequality and sometimes cause discontinuities in the controller. When the outputs are written as in (7.54), the derivative of the Lyapunov function is only in terms of the decision variable $\ddot{q}$ and scalar functions of the states:

$$\dot{V}(x,\mathcal{X}) = L_F V(x) + L_G V(x)\left(\dot{J}_y(q,\dot{q})\dot{q} + J_y(q)\ddot{q}\right). \tag{7.79}$$

However, for a QP the constant terms in the cost do not influence the optimal solution. Therefore, they are dropped and we consider the addition of the $\ddot{q}$-dependent portion of (7.79) to the cost of an (ID-CLF-QP) in order to incentivize convergence:

$$\begin{aligned} \mathcal{X}^* = \underset{\mathcal{X}\in\mathbb{X}_{\text{ext}}}{\text{argmin}} \quad & \frac{1}{2}\mathcal{X}^T Q(x)\mathcal{X} + c^T(x)\mathcal{X} + L_G V(x) J_y(q)\ddot{q} \\ \text{s.t.} \quad & L_F V(x) + L_G V(x)\left(\dot{J}_y(q,\dot{q})\dot{q} + J_y(q)\ddot{q}\right) \leq -\gamma V(x) \\ & D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_c^T(q)\lambda \\ & J(q)\ddot{q} + \dot{J}(q,\dot{q})\dot{q} = 0. \end{aligned} \tag{7.80}$$

7.3.4 Constraint Relaxation

Up until this point in the development of (ID-CLF-QP) controllers, we have considered only the dynamics and CLF derivative constraints applied to the QP. However, in order to implement these controllers on actual hardware, we often require additional constraints that fall under the equality and inequality constraints given in (QP):

$$\begin{aligned} A_E(x)\mathcal{X} &= b_e(x), \\ C_I(x)\mathcal{X} &\geq d_I(x). \end{aligned}$$

For walking robots, these are typically torque limits or admissibility conditions on the constraint forces (see (2.22) and (2.23)). Due to the presence of these constraints, it is not always feasible for the system to simultaneously satisfy physical constraints and converge according to the CLF bound [230]. The accepted way of dealing with this within the literature [230] is to add a relaxation term, δ, to the convergence constraint with an associated weight, ρ. While this was previously illustrated for the (CLF-QP-δ), we can do the same thing for the (ID-CLF-QP) form:

$$\begin{aligned} \mathcal{X}^* = \underset{\mathcal{X}\in\mathbb{X}_{\text{ext}},\delta\in\mathbb{R}}{\text{argmin}} \quad & \frac{1}{2}\mathcal{X}^T Q(x)\mathcal{X} + c^T(x)\mathcal{X} + \rho\delta^2 \qquad \text{(ID-CLF-QP } \delta\text{)} \\ \text{s.t.} \quad & L_F V(x) + L_G V(x)\left(\dot{J}_y(q,\dot{q})\dot{q} + J_y(q)\ddot{q}\right) \leq -\gamma V(x) + \delta \\ & D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_c^T(q)\lambda \\ & J_c(q)\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q} = 0 \\ & A_E(x)\mathcal{X} = b_e(x) \\ & C_I(x)\mathcal{X} \geq d_I(x). \end{aligned}$$

Because we have introduced a weighted relaxation to the inequality that is minimized in the cost, we can actually solve for a cost term which moves the constraint completely

into the cost as an exact penalty function [251]:

$$\mathcal{J}_\delta = \frac{1}{2}\mathcal{X}^T Q(x)\mathcal{X} + c^T(x)\mathcal{X} + \rho||g^+(q,\dot{q},\ddot{q})||, \tag{7.81}$$

where:

$$g(q,\dot{q},\ddot{q}) := L_F V(x) + L_G V(x)\left(\dot{J}_y(q,\dot{q})\dot{q} + J_y(q)\ddot{q}\right) + \gamma V(x)$$
$$g^+(q,\dot{q},\ddot{q}) \triangleq \max(g,0)$$

One of the downsides to using this approach is that the cost term $||g^+(q,\dot{q},\ddot{q})||$ is clearly non-smooth. Instead, we can allow $g(q,\dot{q},\ddot{q})$ to go negative, meaning that the controller will always drive convergence to $V(\eta) \to 0$ even when the inequality (7.43) is not triggered. This will lead us to a differentiable term in the cost that is smooth, though possibly more aggressive in it4s control action. This observation, motivates the removal of the CLF convergence inequality from (7.80) to obtain a relaxed controller:

$$\begin{aligned}
\mathcal{X}^* = &\underset{\mathcal{X}\in\mathbb{X}_{\text{ext}}}{\text{argmin}} \quad \frac{1}{2}\mathcal{X}^T Q(x)\mathcal{X} + c^T(x)\mathcal{X} + \dot{V}(x,\mathcal{X}) && \text{(ID-CLF-QP}^+\text{)}\\
\text{s.t.} \quad & D(q)\ddot{q} + H(q,\dot{q}) = Bu + J^T(q)\lambda\\
& J(q)\ddot{q} + \dot{J}(q,\dot{q})\dot{q} = 0\\
& A_E(x)\mathcal{X} = b_e(x)\\
& C_I(x)\mathcal{X} \geq d_I(x)
\end{aligned}$$

which incentivizes rapid convergence. Further, *whenever it is feasible to do so, this problem will render $\dot{V}$ as negative as possible.*

Remark 8. *While the implementation for* (ID-CLF-QP$^+$) *does not add any non-smooth terms to the cost, it still may be the case that the controller is not Lipschitz continuous depending on the equality and inequality constraints that are applied to the problem. Addressing this problem requires careful consideration of the cost, constraints, and target behavior which is being planned [248].*

Hard and Soft Holonomic Constraints

When implementing on hardware, often holonomic constraints are not satisfied precisely. The analytical solutions presented thus far make the problem more prone to infeasibility. To solve this, we once again look to the robotics community where it has become practice to differentiate between *hard* and *soft* constraints. Hard constraints

such as the holonomic equality constraint equation for the system accelerations (7.48) cannot be violated. Soft constraints, however, refer to an addition to the cost function which penalizes violation of a preferred relationship. These are frequently added to the nominal cost as the norm of a least squares problem:

$$w||A\mathcal{X} - b||^2 \tag{7.82}$$

where $\mathcal{X}$ is as in (ID-QP) and w is a weight which penalizes violation of the equality constraint. On hardware, holonomic constraints for footholds are a common candidate to be implemented as soft constraints:

$$J_c(q)\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q} = 0 \Rightarrow \underbrace{\begin{bmatrix} J_c(q) & 0 & 0 \end{bmatrix}}_{A} \mathcal{X} = -\underbrace{\dot{J}_c(q,\dot{q})\dot{q}}_{b}. \tag{7.83}$$

The formulation of holonomic constraints in this way allows for small violations, which is sometimes necessary in practice where systems can be significantly perturbed. Additional soft constraints that are beneficial for robotic walking include specifying force distributions (weight per foot or in different places on the foot). The benefit of using soft constraints is two-fold; as previously mentioned, it allows for small violations of constraints and it speeds up computation time for the QP.

$$\begin{aligned} \mathcal{X}^* = \underset{\mathcal{X}\in\mathbb{X}_{\text{ext}}}{\text{argmin}} \quad & \frac{1}{2}\mathcal{X}^T Q(x)\mathcal{X} + c^T(x)\mathcal{X} + w_{hol}\left|\left|J_s(q)\ddot{q} + \dot{J}_s(q,\dot{q})\dot{q}\right|\right|^2 \\ \text{s.t.} \quad & D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_s^T\lambda_s \end{aligned}$$

where the notation for $J_c(q)$ is changed to $J_s(q)$ to indicate a soft constraint.

Projection operators for implicit satisfaction of soft constraints: One of the consequences of using a soft constraint for the holonomic constraint equations is that other costs may actively incentivize violation of (7.83). As an example, consider the (ID-QP) trajectory tracking problem:

$$\begin{aligned} \mathcal{X}^* = \underset{\mathcal{X}\in\mathbb{X}_{\text{ext}}}{\text{argmin}} \quad & w_y\left|\left|J_y(q)\ddot{q} + \dot{J}_y(q,\dot{q})\dot{q} + \ddot{y}_d(t,x)\right|\right|^2 + w_{hol}\left|\left|J_s(q)\ddot{q} + \dot{J}_s(q,\dot{q})\dot{q}\right|\right|^2 \\ \text{s.t.} \quad & D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_s^T\lambda_s. \end{aligned}$$

There are cases where the minimizing acceleration for the output feedback term:

$$w_y\left|\left|J_y(q)\ddot{q} + J_y(q,\dot{q})\dot{q} + \ddot{y}_d(t,x)\dot{y}\right|\right|^2,$$

with $\ddot{y}_d(t,x) = \ddot{y}^*(t) - K_\mathrm{p} y(x) - K_\mathrm{d}$ are in direct conflict with:

$$w_{hol}\left|\left|J_s(q)\ddot{q} + \dot{J}_s(q,\dot{q})\dot{q}\right|\right|^2,$$

the error of which will be balanced according to the choice of weights w_{hol} and w_y. This means achieving higher gain feedback may require a user to increase the soft constraint weight until it is effectively an equality constraint, and it loses all the intended desirable properties of a soft constraint.

To resolve this issue, we can use a null space projection, as was detailed extensively in Sec. 7.1 and Sec. 7.3.1, to apply only feedback actions to the robot accelerations that maintain the holonomic constraints. More specifically, accelerations which are orthogonal to the holonomic constraints imposed on the system are:

$$\ddot{q}_y^\perp = \left(I - J_c^\dagger(q)J_c(q)\right)\ddot{q} + J_c^\dagger(q)\dot{J}_c(q,\dot{q})\dot{q}. \tag{7.84}$$

Plugging in the orthogonal acceleration equation and performing some algebra gives an output tracking cost that respects the holonomic constraints:

$$\left|\left|J_y\ddot{q}_y^\perp + \dot{J}_y\dot{q}\right|\right|^2 = \left|\left|\underbrace{J_y(I - J_c^\dagger J_c)}_{J_y^\perp(q)}\ddot{q} + \underbrace{(\dot{J}_y + J_y J_c^\dagger \dot{J}_c)}_{\dot{J}_y^\perp(q,\dot{q})}\dot{q}\right|\right|^2, \tag{7.85}$$

where dependencies were dropped for compactness.

While the previous derivation was shown for the (ID-QP) controller with output feedback in the cost, it can also be applied to the (ID-CLF-QP) and (ID-CLF-QP$^+$) controllers through the substitution of (7.84). As it pertains to (ID-CLF-QP), the convergence inequality becomes:

$$L_F V(x) + L_G V(x)\left(J_y^\perp(q)\ddot{q} + \dot{J}_y^\perp(q,\dot{q})\dot{q}\right) \leq -\gamma V(x). \tag{7.86}$$

With respect to (ID-CLF-QP$^+$), the cost term for the control Lyapunov function derivative becomes:

$$\mathcal{J}_{\dot{V}}^\perp(x,\mathcal{X}) := L_G V(x) J_y(q)\ddot{q}^\perp = L_G V(x) J_y^\perp(q)\ddot{q},$$

where the remainder term $J_y J_c^\dagger \dot{J}_c \dot{q}$ is dropped as does not directly affect the optimal value to the problem.

Chapter 8

REAL-TIME OPTIMAL CONTROL ON CASSIE WITH CONTROL LYAPUNOV FUNCTIONS

The simulations and experiments presented in this chapter has resulted in two publications: first forming the motivation for using ID-CLF-QPs on hardware, and second encompassing the overall approach and application to walking:

[1] J. Reher, C. Kann, and A. D. Ames, "An inverse dynamics approach to control Lyapunov functions," in *2020 American Control Conference (ACC)*, IEEE, 2020.

[2] J. Reher and A. D. Ames, "Control Lyapunov functions for compliant hybrid zero dynamic walking," *IEEE Transactions on Robotics*, 2021, In Preparation.

This chapter is a culmination of both the HZD planning which was presented in Chap. 3 for the compliant model of Cassie which was introduced in Chap. 4 and the feedback controllers which were developed in Chap. 7. First, the (ID-CLF-QP) developed in Chap. 7 is motivated for use on hardware through two examples, one being a simulation study with comparisons drawn to the traditional (CLF-QP) and why alternative formulations may be necessary for hardware, with the second being an implementation of a (ID-CLF-QP$^+$) controller for dynamic crouching on hardware.

With this motivation in hand, the HZD motion library of compliant walking gaits for Cassie obtained in Chap. 6 is further parameterized and then used to develop a real-time (ID-CLF-QP$^+$) implementation which achieves stable walking on hardware. As it pertains to this thesis, the primary contributions, and the relevance of these developments are:

- The content developed in this chapter was primarily an individual effort, with assistance from co-author Claudia Kann in the early developments on simulation and crouching shown in the motivating examples of Sec. 8.1.

- This chapter provides a thorough *experimental validation of the* (ID-CLF-QP) *controller* developed throughout Chap. 7. The resulting experiments in Sec. 8.2.2 demonstrate the accuracy of the parameterized motion plans for walking, and how they can be leveraged for smooth real-time control on hardware.

- The implementation shown in Sec. 8.2.2 demonstrate the *first successful realization of a CLF controller on a 3D bipedal robot.*

8.1 Motivation

This section presents an implementation of an optimization-based controller that leverages the desirable convergence results provided by control Lyapunov functions (see Chap. 7) and combines them with concepts from inverse dynamics-based controllers presented in Sec. 6.2 and Sec. 7.1. This general methodology will first be evaluated in Sec. 8.1.1 through a Matlab simulation for compliant walking on Cassie. Several controllers are evaluated, showing the benefits of the approach which will ultimately be used on hardware. The most consequential observation is that inclusion of a Lyapunov term in the cost helped incentivize the system to converge more rapidly and improved performance with respect to discretization and model inaccuracy/stiffness. Based on these initial simulation findings, an (ID-CLF-QP) (see Sec. 7.3) is developed for the real-time implementation of dynamic crouching on Cassie in Sec. 8.1.2.

The results shown throughout this motivation demonstrate that the control framework can generate smooth torque, force, and acceleration profiles while accurately tracking outputs. Further, the software and algorithms motivated and developed throughout this section and Chap. 6 were critical to forming the necessary infrastructure to implement the first successful 3D walking with CLFs on hardware in Sec. 8.2.

8.1.1 Preliminary Walking Results in Simulation

As the ultimate goal of this thesis is to arrive at a model-based control framework which is realizable on hardware, it is important to evaluate the control concepts developed in Sec. 7.3 in simulation to ensure they are successful before attempting them on hardware. Several walking gaits are found for forward walking behaviors using the HZD framework introduced in Chap. 3 and Chap. 6 which are then parameterized according to Sec. 6.2 so that output polynomials and accelerations are available for use

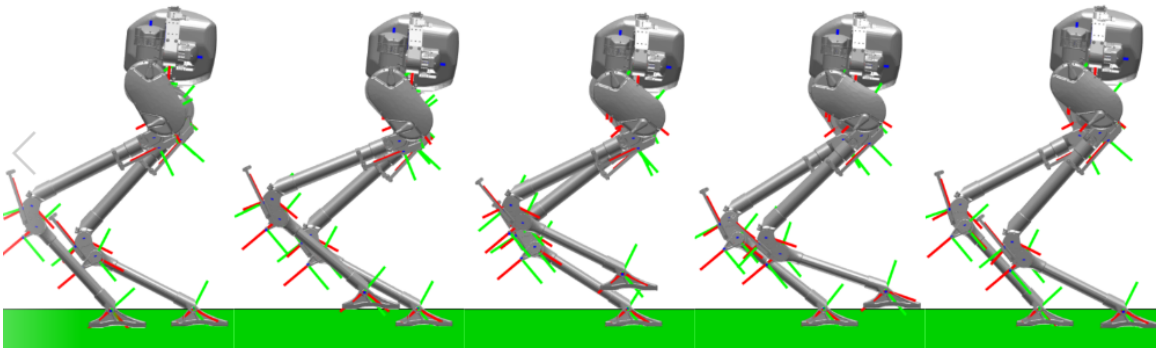

Figure 8.1: Time series motion tiles for simulated walking in Matlab with an ID-CLF-QP controller on a compliant model of Cassie.

within the controllers. Both gaits use a single continuous domain, with the first simulation assuming a rigid and planar version of Cassie and the second simulation using the fully compliant and 3D model. The simulations present a side-by-side comparison of the traditional CLF-QP with the series of controllers introduced in Sec. 7.3.

Planar Walking Simulation

The first gait we consider is designed on a planar, rigid model of Cassie. This is done for the first case as this renders the robot fully actuated if the stance foot is included as an actuator (as opposed to the passive ankle gait design in Chap. 6), and the synthesis of stable controllers for fully actuated walking is considerably more simple than underactuated. For typical walking motions, the forward position of the hip evolves in a roughly linear manner with respect to time [146]. This means that $\delta p_{\text{hip}} \approx p_{\text{hip}} t$ where p_{hip} is the linearized forward hip position with respect to the stance foot. This motivates the state-based parameterization of time:

$$\tau(q) := \frac{\delta p_{\text{hip}}(q) - \delta p_{\text{hip}}(q^+)}{\bar{v}_{\text{hip}}}, \tag{8.1}$$

where $\delta p_{\text{hip}}(q^+)$ is the initial value of the linearized forward hip position and $\bar{v}$ is a parameter selected by the optimization corresponding to the desired velocity.

With the goal of controlling the forward velocity of the robot, we define a relative degree 1 output corresponding to the linearized hip position:

$$y_1^a(q) = \delta \dot{p}_{\text{hip}}(q, \dot{q}) = d\delta p_{\text{hip}}(q, \dot{q})\dot{q}, \qquad y_1^d = \bar{v}_{\text{hip}}. \tag{8.2}$$

A 6th-order Bézier polynomial chosen for the relative degree 2 desired outputs, $y_2^d(\tau, \alpha)$, for which α is a matrix of real coefficients that parameterize the curves according to (3.12). Because we use the rigid leg, we can simply use the full leg kinematics including the tarsus angle (as opposed to the substitutions necessary for a compliant leg in Chap. 6), the relative degree 2 actual outputs are:

$$\begin{aligned}
y^a_{2,\text{b},y} &= \varphi^y && \text{(pelvis pitch)} \\
y^a_{2,\text{sll}} &= ||\psi_s(q)||_2 && \text{(stance leg length)} \\
y^a_{2,\text{nsll}} &= ||\psi_{ns}(q)||_2 && \text{(swing leg length)} \\
y^a_{2,\text{nsla}} &= \text{atan2}\left(\psi^x_{ns}(q)/\psi^z_{ns}(q)\right) && \text{(swing leg pitch)} \\
y^a_{2,\text{nsap}} &= \varphi^y(q) && \text{(swing foot pitch)}
\end{aligned}$$

corresponding to the output selection used in Sec. 6.2 where the yaw and roll coordinates are removed due to the planarizing assumption.

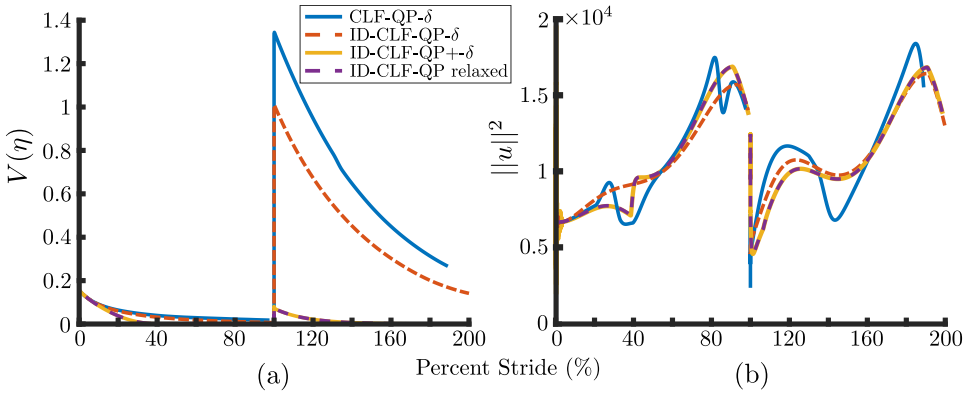

Figure 8.2: Performance of the planar simulated walking gait over two steps, started from rest. Improvement is seen when the Lyapunov term is added to the cost. (a) Lyapunov function, $V(\eta)$ (b) Torque squared, $\|u\|^2$.

Four controllers are then applied to the model; (CLF-QP-δ), (ID-CLF-QP δ), (ID-CLF-QP$^+$ δ), and (ID-CLF-QP$^+$). The simulation is run in Matlab using the ode45 solver with a variable timestep. Each step is simulated until the guard (3.6) is triggered, at which time the integration is terminated and the impact model (4.12) is applied and the resulting post-impact state is used as the initial value for the following step. In Fig. 8.2, the convergence of the Lyapunov function can be seen for a system that is perturbed to start from rest (not started on its periodic orbit) and must converge onto the periodic gait. It can be seen that the more traditionally formulated controllers do not converge quickly enough in the first step, causing an amplification of error in the second, while the two cases with the Lyapunov term in the cost do. When the Lyapunov term is in the cost, the existence of the hard convergence constraint (7.43) does not significantly affect the response. While the performance differs between the four controllers, the torque applied from each is similar in magnitude and form, as can be seen in Fig. 8.2. The various ID-CLF control torques are overall smoother, and the controllers with the Lyapunov derivative term in the cost have the smoothest torque profiles and best convergence performance.

3D Compliant Walking Simulation

The second simulation case is a time-based walking gait on the 3D compliant model of Cassie [128]. A single continuous domain is prescribed, and forward progress of each step is dictated by $\tau(t)$, a parameterization of time by the gait duration:

$$\tau(t) := \frac{t - t_0}{t_f - t_0} \tag{8.3}$$

where t_0 and t_f are the start and end times of the current domain.

For this formulation, the relative degree 1 output is disregarded and four new relative degree 2 outputs — both hip yaws (θ_{hy}), the swing hip roll (θ_{hr}), and the floating base roll (φ_r) - are added. A 6th-order Bézier polynomial is again chosen for the relative degree 2 desired outputs, $y_2^d(\tau, \alpha)$, and all of the relative degree 2 actual outputs are identical to those selected for the compliant motion library developed in Sec. 6.2 using the neutral leg substitution $\bar{q}_l \in \{\mathcal{Q}_l \mid q_{\text{sp}} = 0, q_{\text{hs}} = 0, q_{\text{t}} = 13^o - q_{\text{k}}\}$ to obtain a feedback controller via optimization that encodes the passive compliance.

The theory referenced in Chap. 7 assumes purely continuous control, however, in reality torques are applied at discrete intervals. Thus while in the rigid and planarized case we compared controller convergence as an approximately continuous variable time step Matlab ODE, the compliant simulation will study the effect of realistic control discretization on controller performance. Because the CLF-QP relies on an inequality constraint to enforce convergence, there is the possibility for high-frequency "chattering" in the controller as the robot repeatedly triggers and violates the bound. This discretization will also exacerbate any discontinuities in the control cost.

Each proposed feedback control method is implemented in simulation, where the dynamics is integrated using Matlab ode23s, which handles stiff ODEs such as the Cassie model considerably better than ode45. In order to simulate a discrete controller implemented on a continuous system, the walking sequence is discretized according to the specified control frequency and placed in a loop where each iteration evaluates one timestep. In each iteration, the associated controller is evaluated and then applied as a constant value for a forward simulation over the given time window.

The Lyapunov function convergence and motor torques for each of the controllers can be seen in Fig. 8.4 and Fig. 8.5, respectively. The controllers are first simulated at

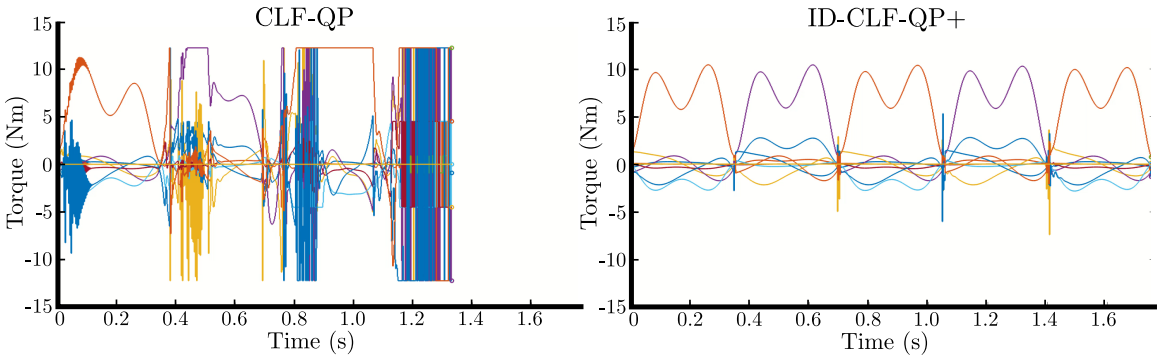

Figure 8.3: The effect of the various controllers subjected to a 500 Hz control frequency over several steps of forward walking. On the left is the traditional CLF-QP and on the right is the ID-CLF-QP+ controller.

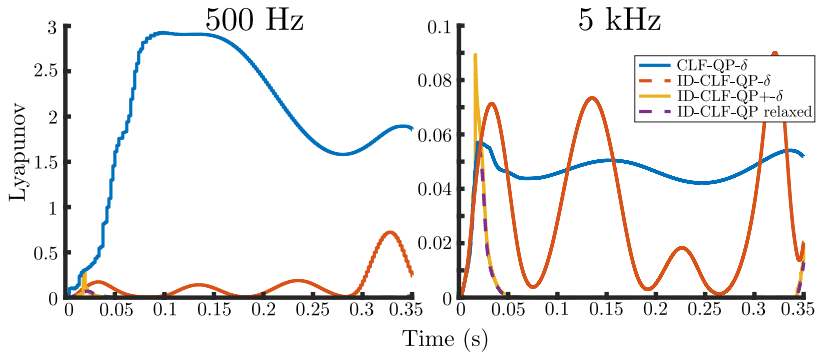

Figure 8.4: Lyapunov function convergence on the 3D compliant robot for a time based step at 500 Hz and 5 kHz control frequencies.

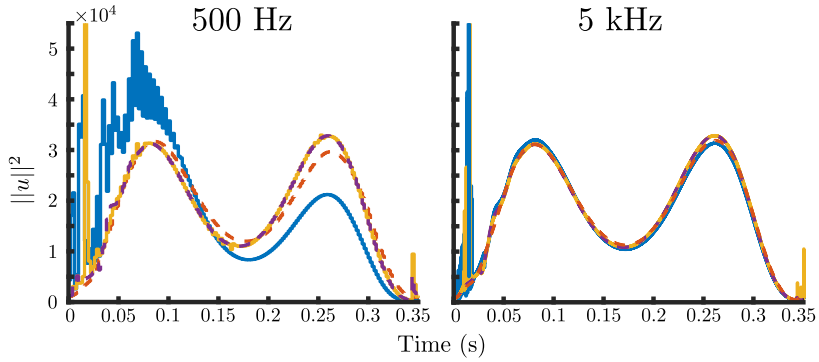

Figure 8.5: Torque of the 3D compliant robot for a time-based step at 500 Hz and 5 kHz control frequencies.

500 Hz and then increased by one order of magnitude to 5 kHz. An animation of the resulting simulation is provided[1], and shown as a sequence of gait tiles in Fig. 8.1.

While the traditional (CLF-QP-δ) and (ID-CLF-QP δ) controllers see a marked degradation as loop rates decrease, the controllers which have Lyapunov derivative terms in the cost, (ID-CLF-QP$^+$ δ) and (ID-CLF-QP$^+$), seem minimally affected. Because this controller is run on the compliant model, the ODE is much more numerically stiff than in the rigid planar case. As such, we see that (ID-CLF-QP δ), which uses the inverted form of the mass inertia matrix, is much more sensitive when applied at coarse frequencies. The degradation of the controller over time is dramatically illustrated for the 500 Hz walking in Fig. 8.3, where on the left the (CLF-QP) goes unstable after the control inputs violently chatter while the (ID-CLF-QP) remains smooth with the exception of near impact. This jump in the (ID-CLF-QP) appears to correspond to the portion of the gait where the vertical ground reaction forces become very small, and the controller triggers the friction pyramid.

[1]Video of the 3D compliant simulation and experimental results for crouching on Cassie: `https://youtu.be/CI9xv_OcfWw`

8.1.2 Dynamic Crouching with Control Lyapunov Functions

In this section, we discuss the *first successful implementation of an* (ID-CLF-QP) *on physical hardware*. To validate the methods outlined in Sec. 7.3, an output tracking problem was constructed for a crouching motion on the Cassie biped, which remains purely continuous and involves no impacts or alternating contact conditions. The motivation for beginning with a simpler and continuous application was to ensure that the controller could sufficiently track the outputs without subjecting the robot to impacts, which significantly complicates the tuning and implementation process.

A general description of the actual outputs, desired crouching polynomial, and QP controller are first introduced for the specific crouching behavior. These are then applied directly to hardware, with the results illustrating the accuracy of the tracking problem and relatively smooth torque profiles. The resulting motion can be seen in Fig. 8.6, where Cassie is mid-crouch in both a Gazebo simulation and on hardware.

Problem Description and Feedback Control Development

The motion task that we wish to stabilize for this experimental implementation is a fast vertical crouch, using a set of outputs which are defined in such a way that they can be tracked for any feasible contact configuration with both feet on the ground. The contact configuration of the robot is shown on the right in Fig. 8.7, corresponding to the geometry presented in Sec. 6.1 for double-support during walking.

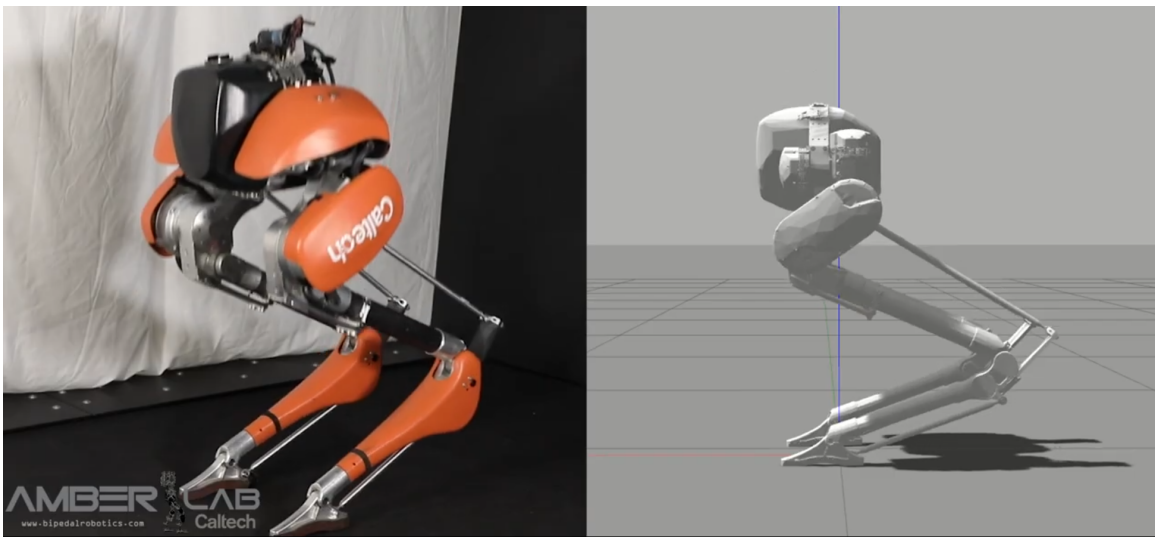

Figure 8.6: Video still of Cassie performing a dynamic crouch in Gazebo and on hardware with a CLF-QP controller.

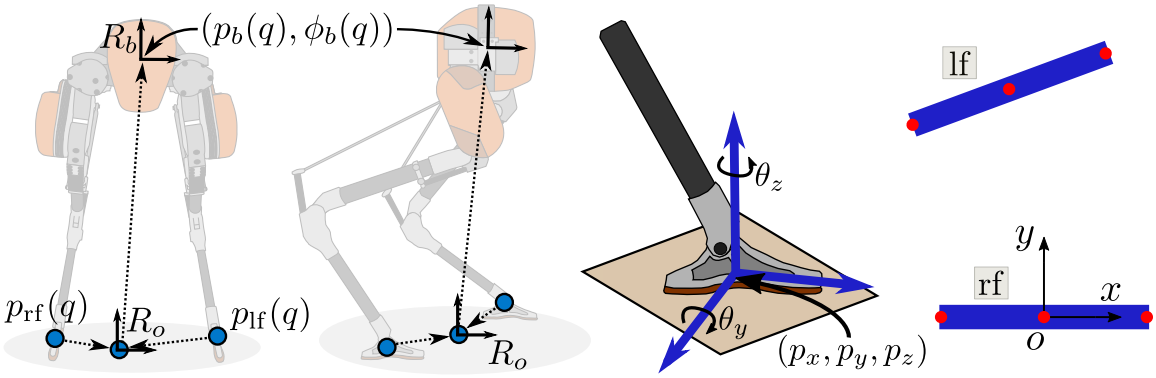

Figure 8.7: The contact geometry and output definitions for relative position and orientation of the pelvis with respect to the mean foot configuration.

Recall from the Cassie model development in Sec. 4.2 that we allow for only one mode of contact for each foot, which is consistent with a line contact along the x axis of the foot. In accordance with the fact that with both feet in contact with the ground, we define the holonomic constraints as:

$$\eta_{\text{crouch}}(q) := (p_{\text{lf}}, \varphi^{yz}_{\text{lf}}, p_{\text{rf}}, \varphi^{yz}_{\text{rf}}, \eta_{\text{4bar}}, \eta_{sp}) \in \mathbb{R}^{16} \tag{8.4}$$

where p_{sf} and p_{nsf} are the positions of the stance foot (center) and non-stance foot (center), η_{sp} demonstrates that we are making a significant assumption on the rigidity of the leg in order to simplify this behavior (we will show how to mitigate this assumption in Sec. 8.2 for compliant walking), and it can be seen that the closure of the compliant multi-bar leg is enforced through the holonomic constraint (4.18). As opposed to the "rigid" model presented in Sec. 4.3, $\eta_{sp}(q)$ is applied at the measured value of the spring deflection, and we do not make any coordinate substitutions to the tarsus. This means that the leg geometry will be correct in the kinematic terms, but the controller will assume that the springs instantaneously transfer torque to connected joints essentially acting as a series-elastic actuator.

Actual outputs: Six relative degree two outputs for standing were prescribed for the position and orientation of the pelvis relative to the mean foot configuration:

$$y^a_{2,\text{b},y}(q) = p_b - \frac{1}{2}(p_{\text{lf}}(q) + p_{\text{rf}}(q)), \qquad \text{(pelvis position)}$$
$$y^a_{2,\text{sll}}(q) = \varphi_b - \frac{1}{2}(\varphi_{\text{lf}}(q) + \varphi_{\text{rf}}(q)). \qquad \text{(pelvis orientation)}$$

Because we have subtracted the forward kinematics from the floating base, we are left with a relative expression which is only a function of internal joint angles and thus is not dependent on the measured values of p_b and φ_b. In addition, because of the rigid

spring assumption in (8.4), the behavior is over-actuated with 22 unpinned DOF, 16 constraints, 10 actuators, and 6 outputs.

Desired polynomial: The prescribed motion was a repeated crouch which moved the pelvis vertically from 0.9 m to 0.5 m and back, with each segment being two seconds in duration. This was formulated as two time-dependent sigmoids for the vertical, with the full crouch motion taking place over 4 seconds:

$$\begin{bmatrix} p_z^d(t) \\ \dot{p}_z^d(t) \\ \ddot{p}_z^d(t) \end{bmatrix} = \begin{cases} \begin{bmatrix} 0.90 - \frac{0.4}{1+e^{(10-9t)}} \\ -\frac{(18e^{(10-9t)})}{5(e^{(10-9t)}+1)^2} \\ \frac{(162e^{(10-9t)})}{5(e^{(10-9t)}+1)^2} - \frac{(324e^{(20-18t)})}{5(e^{(10-9t)}+1)^3} \end{bmatrix} & \text{if } t \leq 2, \\ \begin{bmatrix} 0.5 + \frac{0.5+0.4}{1+e^{(26-9t)}} \\ \frac{(18e^{(26-9t)})}{5(e^{(26-9t)}+1)^2} \\ \frac{(324e^{(52-18t)})}{5(e^{(26-9t)}+1)^3} - \frac{(162e^{(26-9t)})}{5(e^{(26-9t)}+1)^2} \end{bmatrix} & \text{if } 2 < t \leq 4, \end{cases} \tag{8.5}$$

where $p_z^d(t), \dot{p}_z^d(t)$, and $\ddot{p}_z^d(t)$ are the vertical position, velocity, and acceleration respectively. The remaining values are set to zero:

$$\begin{bmatrix} (p_x^d, p_y^d, \varphi_x^d, \varphi_y^d, \varphi_z^d) \\ (\dot{p}_x^d, \dot{p}_y^d, \dot{\varphi}_x^d, \dot{\varphi}_y^d, \dot{\varphi}_z^d) \\ (\ddot{p}_x^d, \ddot{p}_y^d, \ddot{\varphi}_x^d, \ddot{\varphi}_y^d, \ddot{\varphi}_z^d) \end{bmatrix} := 0. \tag{8.6}$$

In addition to the time-dependent crouching behavior, there is a trigger which can be enabled on the radio controller to place the robot into a user control mode. When in

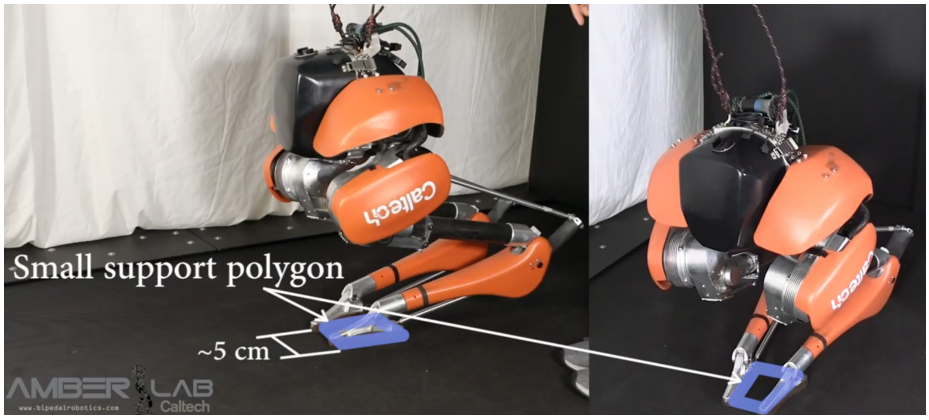

Figure 8.8: Video still of Cassie performing a crouch with a small support polygon. This illustrates how the outputs can stabilize the behavior under a variety of contact conditions, since the holonomic constraints and their arising constraint forces are directly considered by the optimization.

this mode, the various knobs on the remote control can be used to adjust the vertical position, lateral position, and pitch to constant setpoints.

ID-CLF-QP specification: The controller which was realized on hardware was in the form of (ID-CLF-QP$^+$), where all cost terms except for the CLF derivative are specified as a least-squares term:

$$\begin{aligned}\mathcal{X}^* = \underset{\mathcal{X}\in\mathbb{X}_{\text{ext}}}{\text{argmin}} \quad & ||A(x)\mathcal{X} - b(x)||^2 + \dot{V}_\varepsilon(q,\dot{q},\ddot{q}) \\ \text{s.t.} \quad & D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_s^T(q)\lambda_s \\ & \lambda_s \in \mathcal{AC}_{\text{crouch}}(\lambda_s) \\ & u_{\text{lb}} \le u \le u_{\text{ub}}\end{aligned}$$

where we have added torque bounds to the problem and $\mathcal{AC}_{\text{crouch}}(\lambda_s)$ is the admissibility conditions on the constraint forces. The constraint feasibility associated with holonomic foot constraints in Sec. 6.1 for double-support are used and are given as:

$$\mathcal{AC}_{\text{crouch}}(\lambda_s) = \begin{bmatrix} \{\lambda^z_{\text{lf}}, \lambda^z_{\text{lf}}\} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{\text{lf}} - \{|\lambda^x_{\text{lf}}|, |\lambda^y_{\text{lf}}|\} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{\text{rf}} - \{|\lambda^x_{\text{rf}}|, |\lambda^y_{\text{rf}}|\} \\ \frac{l_f}{2}\lambda^z_{\text{lf}} - \{|\lambda^{my}_{\text{lf}}|\} \\ \frac{l_f}{2}\lambda^z_{\text{rf}} - |\lambda^{my}_{\text{rf}}| \end{bmatrix} \ge 0, \tag{8.7}$$

corresponding to the friction pyramid (2.22) and foot rollover condition (2.23). The CLF derivative is given by:

$$\dot{V}_\varepsilon(q,\dot{q},\ddot{q}) := \begin{bmatrix} L_G V_\varepsilon(x) J_y(q) & 0 & 0 \end{bmatrix} \mathcal{X}, \tag{8.8}$$

with the remaining cost terms defined as:

$$A(x) = \begin{bmatrix} w_{\text{reg}} I \\ w_y J_y(q) \\ w_\lambda J_c(q) \end{bmatrix}, \qquad b = \begin{bmatrix} w_{\text{reg}} \mathcal{X}^*(t) \\ w_y \left(\ddot{y}^d(t) - \dot{J}_y(q,\dot{q})\dot{q} \right) \\ -w_\lambda \dot{J}_s(q,\dot{q})\dot{q} \end{bmatrix}. \tag{8.9}$$

Because the nominal motion for the full order system is not found offline as was the case for HZD behaviors, we construct the regularization term for the robot accelerations based on the current contacts and the desired output accelerations:

$$\ddot{q}^* = \begin{bmatrix} J_c(q) \\ J_y(q) \end{bmatrix}^\dagger \left(\begin{bmatrix} 0 \\ \ddot{y}^d(t) \end{bmatrix} - \begin{bmatrix} \dot{J}_c(q,\dot{q}) \\ \dot{J}_y(q,\dot{q}) \end{bmatrix} \dot{q} \right). \tag{8.10}$$

In addition, the torque is regularized with the previous value $u^* = u_{\text{prev}}$ and the constraint forces, λ_c^*, are all zeroes except for the vertical ground reaction force $\lambda^*_{c,z} = mg/2$, which compensates for half of the robot weight.

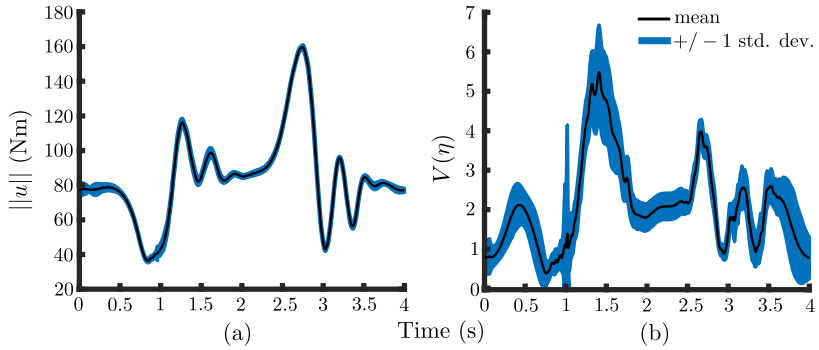

Figure 8.9: Torque and Lyapunov function values over 45 crouches on hardware with the shaded areas as +/− one std. deviation.

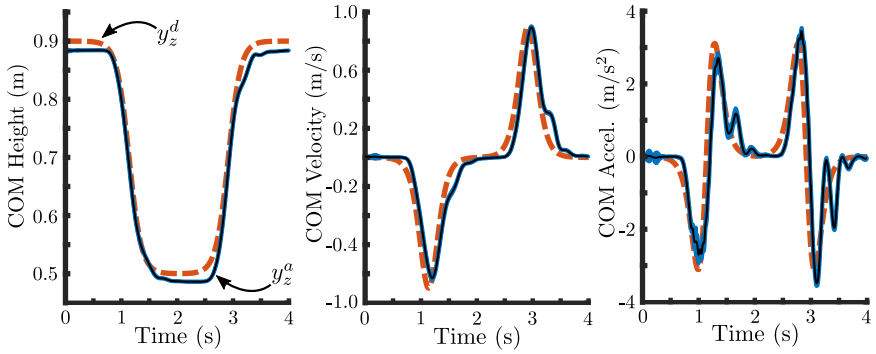

Figure 8.10: Height, velocity, and acceleration over 45 crouches. Desired outputs shown as dashed, mean as black, and shaded blue as +/− one std. deviation.

Results

Finally, the presented (ID-CLF-QP$^+$) controller was implemented on hardware. The control architecture was built upon the existing framework presented in Sec. 6.2 (see Fig. 6.17), with the primary modifications to the existing software being the addition of the (ID-CLF-QP$^+$) controller in place of (6.40) removal of the HZD gait library and the addition of the time-dependent outputs presented in this section. The source code necessary to run a Gazebo simulation or directly implement the code on hardware is also provided online[2].

The resulting QP was solved with the qpOASES package and had 49 variables and 41 constraints. As can be seen in Fig. 8.10, the height was smoothly tracked to within several centimeters for the entirety of the motion. Because we are using a task-space approach, it is not necessary to encode these objectives solely as combinations of the actuated joint angles, and no joint level stabilization (i.e. individual joint tracking or

[2]Open-sourced repository for the C++ controller code used in this work on hardware and for use in a Gazebo simulation `https://github.com/jpreher/cassie_documentation`

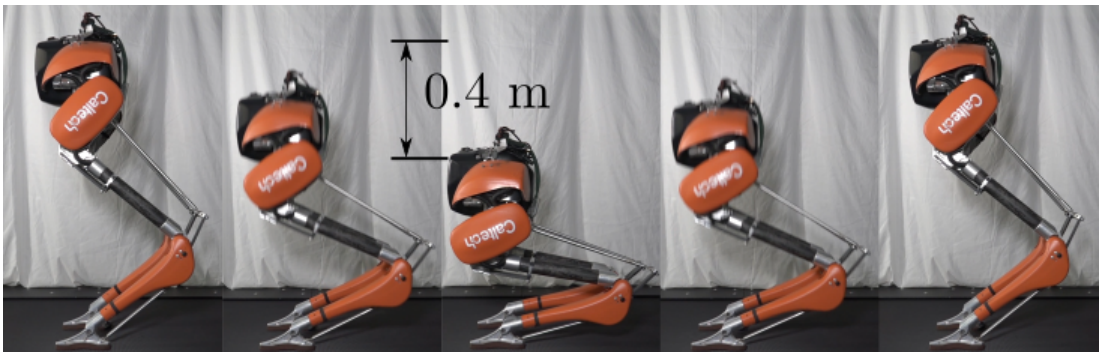

Figure 8.11: Time series motion tiles - hardware for the crouching experiment (right).

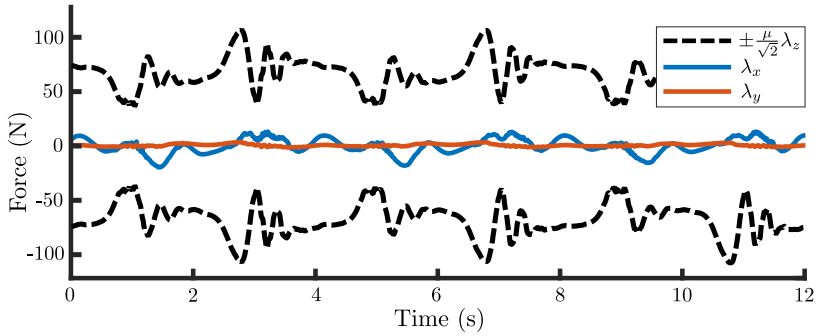

Figure 8.12: Contact forces for the left foot adhering to the pyramidal friction constraints over three consecutive crouches on hardware.

control) was used. The resulting motions thus do not require any specific placement of the feet, illustrated in Fig. 8.8, where the robot was made to crouch with the feet very close together at a yawed angle.

When this controller was created, the constraint projections considered in Sec. 7.3.4 to compensate for soft holonomic constraints were not yet developed, which can be seen by the lack of the $(\cdot)^{\perp}$ notation in the associated costs (8.9). As such, the tuning required a more careful balance between the feedback linearizing term, the CLF derivative, and the soft contact constraints. This will be addressed in Sec. 8.2, where in order to obtain for the considerably more dynamic task of walking, it was necessary to apply them in order to track the outputs well enough to ensure stability.

The norm of torque applied to all motors can be seen in Fig. 8.9, which are smooth and are satisfying all torque limitations. In addition, in Fig. 8.12 the contact forces are shown to adhere to the friction cone (2.21). A video of the experiment along with animations of the previous simulations are provided[3], while a motion tile is shown in Fig. 8.11 to illustrate the vertical height traversed for each crouch.

[3]Video of the experimental results for crouching on Cassie: `https://youtu.be/CI9xv_OcfWw`

8.2 Compliant Walking with Control Lyapunov Functions

This section discusses the *main experimental result of this thesis*, and serves to illustrate how the various aspects developed throughout the preceding chapters can be combined on hardware to obtain stable walking. Specifically, the motion library developed for hybrid zero dynamic walking in Chap. 6 on the compliant model of Cassie in Sec. 4.2 is further parameterized and then combined with the (ID-CLF-QP) control framework introduced in Chap. 7.

As indicated by the methods discussed throughout Chap. 5 and Chap. 6, there is often an "artful implementation" step that translates model-based controllers to a form that can be actually implemented on hardware. Ideally, methods can be developed that allow the exact transcription of model-based methods to hardware in a robust fashion and without heuristics. This work serves as a major step in this direction, with the walking on Cassie detailed in Sec. 8.2.4 and shown in Fig. 8.13 being the *first successful experimental implementation of a CLF for walking on a 3D biped.*

Several of the primary benefits that will be shown of the (ID-CLF-QP) version of the (CLF-QP) controller used in this implementation is that *it is computationally tractable and lends itself particularly well to intuitive tuning*, while keeping the number of heuristics as minimal as possible. With regard to the compliance in the Cassie model, the formulation of (ID-CLF-QP) shown in Sec. 8.2.2 was able to *directly optimize for the spring forces on the robot* rather than using the measured values, which are a significant source of oscillations and model inaccuracy. Posing the spring torques as a decision variable was crucial in obtaining smooth force-based control while simultaneously tracking virtual constraints and respecting the passive dynamics.

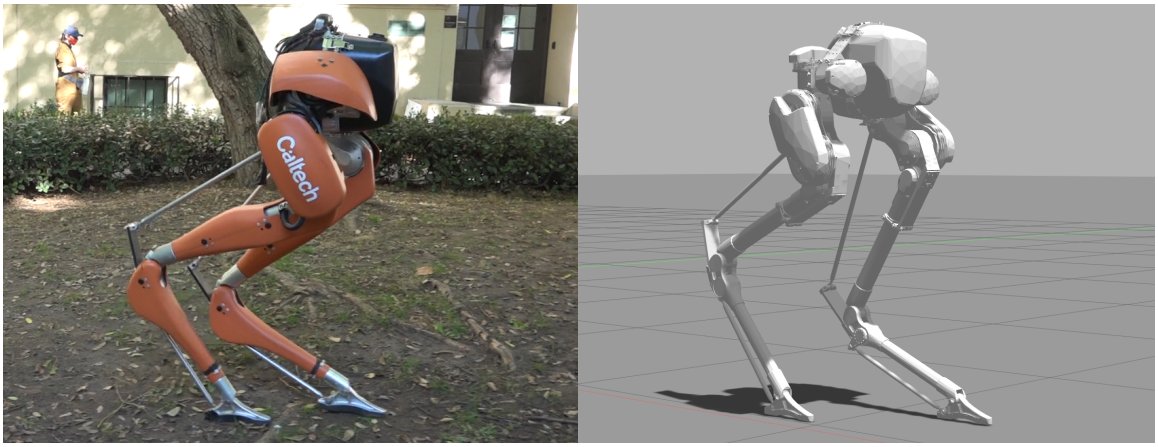

Figure 8.13: The Caltech Cassie biped walking outdoors and in a Gazebo simulation while using a version of the optimization-based (ID-CLF-QP$^+$) controller.

8.2.1 HZD Locomotion Problem

This section will provide an overview of the HZD locomotion problem that was used on Cassie for walking. The presentation will be minimal, as it builds directly from the HZD motion library which was extensively detailed in Sec. 6.2.

Locomotion Model

The hybrid locomotion problem is specified as in Sec. 6.2, which considers two continuous domains of single-support with an instantaneous double-support domain (see Fig. 6.13). The stance foot is left as a passive joint, leaving each continuous domain with 9 inputs to track the relative degree 2 outputs:

$$
\begin{aligned}
y^a_{2,\mathrm{b},x} &= \varphi^x && \text{(pelvis roll)}\\
y^a_{2,\mathrm{b},y} &= \varphi^y && \text{(pelvis pitch)}\\
y^a_{2,\mathrm{sll}} &= ||\psi_s(\bar{q}_l)||_2 && \text{(stance leg length)}\\
y^a_{2,\mathrm{nsll}} &= ||\psi_{ns}(\bar{q}_l)||_2 && \text{(swing leg length)}\\
y^a_{2,\mathrm{nsla}} &= \mathrm{atan2}\left(\psi^x_{ns}(\bar{q}_l)/\psi^z_{ns}(\bar{q}_l)\right) && \text{(swing leg pitch)}\\
y^a_{2,\mathrm{nshr}} &= q_{\mathrm{nshr}} && \text{(swing hip roll)}\\
y^a_{2,\mathrm{shy}} &= q_{\mathrm{shy}} && \text{(stance hip yaw)}\\
y^a_{2,\mathrm{nshy}} &= q_{\mathrm{nshy}} && \text{(swing hip yaw)}\\
y^a_{2,\mathrm{nsap}} &= \varphi^y(q_{\mathrm{b}}, \bar{q}_l) && \text{(swing foot pitch)}
\end{aligned}
$$

which are further described in Sec. 6.2.2 and shown in Fig. 6.14.

The stance leg is considered compliant and is modeled according to Sec. 4.2, with a rigid swing leg in order to reduce the number of passive degrees of freedom that are not significant to the walking (it is assumed that during swing the springs are sufficiently rigid to impose as a holonomic constraint). In accordance with the assumptions on the single-support domain, we define the holonomic constraints as:

$$
\eta_{ss}(q) := (p_{\mathrm{sf}}, \varphi^{yz}_{\mathrm{sf}}, \eta_{4\mathrm{bar}}, q_{nssp}, q_{nshs}) \in \mathbb{R}^9 \tag{8.11}
$$

where $p_{\square f}$ and $q_{\square sp}, q_{\square hs}$ are the positions of the stance foot (center) and the non-stance springs, pictured in Fig. 4.7. The constraint feasibility associated with holonomic foot constraints in Sec. 6.2 for single-support are used, and are given as:

$$
\mathcal{AC}_{\mathrm{ss}}(\lambda_s) = \begin{bmatrix} \{\lambda^z_{\mathrm{sf}}, \lambda^z_{\mathrm{sf}}\} \\ \frac{\mu}{\sqrt{2}}\lambda^z_{\mathrm{sf}} - \{|\lambda^x_{\mathrm{sf}}|, |\lambda^y_{\mathrm{sf}}|\} \\ \frac{l_f}{2}\lambda^z_{\mathrm{sf}} - \{|\lambda^{my}_{\mathrm{sf}}|, |\lambda^{mz}_{\mathrm{sf}}|\} \end{bmatrix} \geq 0, \tag{8.12}
$$

corresponding to the friction pyramid (2.22) and foot rollover condition (2.23).

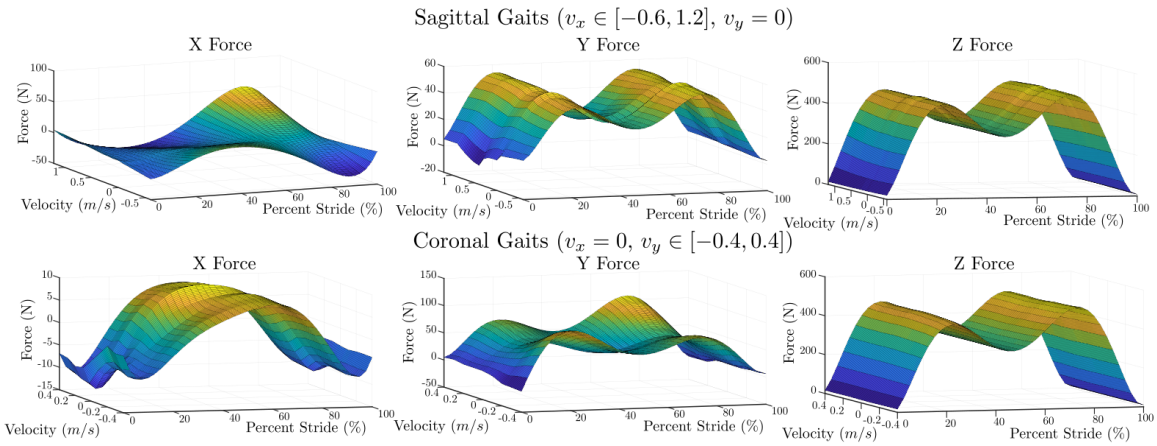

Figure 8.14: Contour plots showing the parameterized ground reaction forces for the right stance domain of walking on Cassie obtained from the HZD optimization over a range of speeds in both the sagittal and coronal directions.

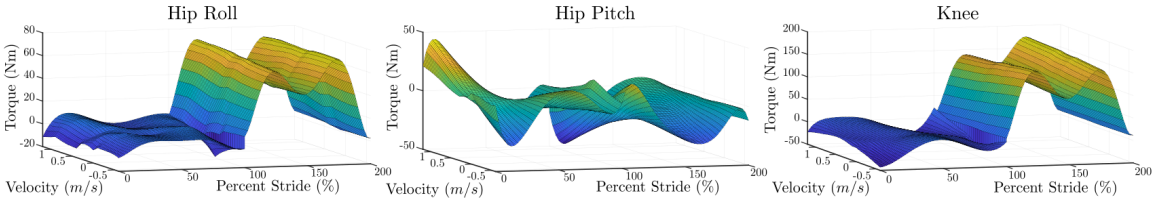

Figure 8.15: Contour plots showing the parameterized torques for walking on Cassie, pictured for a range of walking speeds in the sagittal direction.

Gait Optimization and Parameterization

The trajectory optimization which was performed in Sec. 6.2 to obtain a motion library of 171 individual walking gaits on Cassie was directly used in this section. However, the controller in (6.40) only used a parameterization of the output polynomials and generalized accelerations, shown in Fig. 6.15 and Fig. 6.16. Thus, in order to form a full parameterization of the nominal hybrid periodic orbit via:

$$\mathcal{X}^*_\alpha(t) := \begin{bmatrix} \ddot{q}^*_\alpha(t) \\ u^*_\alpha(t) \\ \lambda^*_{c,\alpha}(t) \end{bmatrix}, \tag{8.13}$$

we further parameterized the NLP result in order to obtain a 6th order Bézier polynomial for each of the ground reaction forces and rotor torques. The resulting contact force profiles over a variety of walking speeds is shown in Fig. 8.14, while the torques are shown in Fig. 8.15. These parameters were appended to the polynomial collection given in (6.32) and extracted for a given walking speed according to a bilinear interpolation routine (6.35).

8.2.2 Feedback Control Development

In this section, the feedback controller used on hardware to track the HZD locomotion problem posed in Sec. 6.2 using the concepts from Chap. 7. Several notable modifications were made in this section in order to achieve successful walking on hardware. First, several of the constraints are removed using a linear projection operator as described in Sec. 2.1.3, leaving the ground reaction forces as a decision variable so that they can be easily used with the admissibility constraints (8.12). Next, the stance shin and heel springs are added as an additional soft holonomic constraint. By adding the spring forces in this way, we can allow the QP to choose the spring torque, and can weight how closely it enforces a "quasi-static" assumption. The springs are then regularized against the measured spring force, with the weighted soft constraint smoothing the values and for allowing non-zero accelerations.

RES-CLF Specification

In order to track the virtual constraints in each of the continuous domains, a RES-CLF (7.21) is found using the continuous time algebraic Riccati equations (7.38):

$$F^T P + PF - PGR^{-1}G^T P + Q = 0,$$

for $Q = Q^T > 0$, $R = R^T > 0$, and with the solution $P = P^T > 0$. The matrices Q and R are chosen as diagonal matrices, with specific gains given in Table 8.2. The solution to (7.38), P, is thus a symmetric block matrix:

$$P = \begin{bmatrix} P_{\text{ud}} & P_{\text{od}} \\ P_{\text{od}} & P_{\text{ld}} \end{bmatrix},$$

where P_{ud}, P_{ld}, and P_{od} are found to be the diagonal matrices:

$$\begin{aligned} P_{\text{ud}} &= \text{diag}\left(794.80, 682.49, 137.30, 879.78, 1552.45, 795.89, 442.36, 189.37, 303.38\right) \\ P_{\text{ld}} &= \text{diag}\left(10.48, 11.31, 8.23, 10.88, 15.00, 12.50, 10.22, 9.83, 9.41\right) \\ P_{\text{od}} &= \text{diag}\left(60.66, 60.33, 23.37, 56.59, 82.85, 63.69, 43.27, 26.98, 32.25\right). \end{aligned}$$

From this solution, we obtain a RES-CLF according to:

$$V_\varepsilon(\eta(q,\dot{q})) = \eta^T(q,\dot{q}) \begin{bmatrix} \frac{1}{\varepsilon} I & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} \frac{1}{\varepsilon} I & 0 \\ 0 & I \end{bmatrix} \eta(q,\dot{q}), \tag{8.14}$$

with the specific value of ε given in Table 8.2.

Robot Dynamics and Partial Constraint Elimination

The full holonomic constraint vector given in (8.11) contains several constraints which are a function of the internal kinematics of the leg, and do not directly affect the physical feasibility (8.12) or lend any use in shaping as a decision variable. In addition, the measured spring forces on the stance leg are a potential source of model uncertainty, vibration, and numerical stiffness in the feedback controller. To address both of these issues, we partition the holonomic constraints into hard and soft constraints, and append the stance spring forces to the soft constraint wrench vector:

$$\lambda_c := \begin{bmatrix} \lambda_{\text{4bar}} \\ \lambda_{\text{ns,sp}} \end{bmatrix} \in \mathbb{R}^4, \tag{8.15}$$

$$\lambda_s := \begin{bmatrix} \lambda_{\text{sf}} \\ \lambda_{\text{s,sp}} \end{bmatrix} \in \mathbb{R}^7, \tag{8.16}$$

where λ_c denotes constraints which are "hard" and λ_s denotes the "soft" constraints. Instead of enforcing the hard constraints as an equality constraint in the QP, we instead remove them from the problem by forming a linear projection operator, $P_c(q) = I - J_c^\dagger(q)J_c(q)$, as described in Sec. 2.1.3 to obtain the constrained equations of motion:

$$D_c(q)\ddot{q} + H_c(q,\dot{q})\dot{q} = B_c(q)u + J_{c,s}^T(q)\lambda_s,$$

where the individual terms are:

$$\begin{aligned} D_c(q) &= D(q) + P_c(q)D(q) - (P_c(q)D(q))^T, \\ H(q,\dot{q}) &= P_c(q)H(q,\dot{q}) + D(q)J_c^\dagger(q)\dot{J}_c(q,\dot{q}), \\ B_c(q) &= P_c(q)B, \\ J_{c,s}^T(q) &= P_c(q)J_s^T(q). \end{aligned}$$

This does not significantly complicate the equations of motion, as $P_c(q)$ is simply a function of the internal leg kinematics forming the multi-bar mechanisms on each leg. Despite its addition as a soft holonomic constraint, we do not want the spring forces $\lambda_{\text{s,sp}}$ to obey $\ddot{\eta}_{\text{s,sp}} \equiv 0$. Instead, the soft constraint can be weighted to allow for acceleration violations, and the regularization vector, $\mathcal{X}_\alpha^*(t)$, is augmented to include the measured spring forces:

$$\lambda_{\text{s,sp}}^* := -K_{\text{sp}}q_{\text{sp}} = -\begin{bmatrix} 2300 & q_{\text{s,sp}} \\ 2000 & q_{\text{s,hs}} \end{bmatrix}. \tag{8.17}$$

ID-CLF-QP Specification

The resulting controller that was implemented on hardware was posed in the form of (ID-CLF-QP$^+$), where all cost terms except for the CLF derivative are specified as a least-squares term, and the decision variables are given as $\mathcal{X} = \left[\ddot{q}^T, u^T, \lambda_s^T\right]^T \in \mathbb{R}^{39}$:

$$\begin{aligned}
\mathcal{X}^* = \underset{\mathcal{X}\in\mathbb{X}_{\text{ext}}}{\text{argmin}} \quad & ||A(x)\mathcal{X} - b(x)||^2 + \dot{V}(q,\dot{q},\ddot{q}) \\
\text{s.t.} \quad & D_c(q)\ddot{q} + H_c(q,\dot{q})\dot{q} = B_c(q)u + J_{c,s}^T(q)\lambda_s \\
& \lambda_s \in \mathcal{AC}_{\text{ss}}(\lambda_s) \\
& u_{\text{lb}} \leq u \leq u_{\text{ub}} \\
& u_{\text{s,ak}} = 0
\end{aligned}$$

where $u_{\text{s,ak}}$ is the stance ankle torque, which is constrained to zero in order to ensure that the QP leaves it passive as was defined in the locomotion model. The resulting QP problem then has 39 decision variables and 31 individual constraints (excluding upper and lower bounds directly on the decision variables), and is evaluated using the qpOASES solver [208].

Because we have mixed hard and soft constraints within the QP that was posed, the cost terms which involve feedback on the generalized accelerations are projected to ensure that the terms are orthogonal to the holonomic acceleration constraint (7.48).

Table 8.1: Weights used in the (ID-CLF-QP$^+$) controller on hardware for Cassie.

Parameter	Value
ε	0.1
Q	$[4600, 3640, 390, 4575, 8580, 4056, 1872, 520, \ldots$ $\ldots 520, 16, 7.3, 1.6, 56, 115, 28.8, 18, 15, 12]$
R	$[0.8, 1, 1.4, 0.7, 0.8, 1, 1, 1.4, 1]$
$w^{\text{reg}}_{\ddot{q}_b}$	$[0.01, 0.01, 0.01, 0.01, 0.01, 0.01]$
$w^{\text{reg}}_{\ddot{q}_{l,\text{st}}}$	$[0.01, 0.01, 0.01, 0.01, 0.6, 0.01, 0.6, 0.01]$
$w^{\text{reg}}_{\ddot{q}_{l,\text{sw}}}$	$[0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]$
$w^{\text{reg}}_{u_{\text{st}}}$	$[1, 0.9, 0.5, 0.1, 1]$
$w^{\text{reg}}_{u_{\text{sw}}}$	$[1, 1, 0.9, 0.8, 1]$
$w^{\text{reg}}_{\text{grf}}$	$[0.9, 0.1, 1.9, 1.3, 1.3]$
$w^{\text{reg}}_{\text{spring force}}$	1.0
w_{grf}	$[1, 1, 1, 1.3, 1.3]$
$w_{\text{static spring}}$	1.0
w_y	1.42
$w_{\dot{V}}$	1.40

Specifically, the hard and soft constraints are stacked as:

$$J_{c,s}(q) := \left[J_c^T(q),\ J_s^T(q)\right]^T, \tag{8.18}$$

forming the orthogonal acceleration substitution (7.84):

$$\ddot{q}^{\perp} = \left(I - J_{c,s}^{\dagger}(q)J_{c,s}(q)\right)\ddot{q} + J_{c,s}^{\dagger}(q)\dot{J}_{c,s}(q,\dot{q})\dot{q}, \tag{8.19}$$

$$J_y^{\perp}(q) := J_y(q)(I - J_c^{\dagger}(q)J_c(q)), \tag{8.20}$$

$$\dot{J}_y^{\perp}(q,\dot{q}) := (\dot{J}_y(q) + J_y(q)J_c^{\dagger}(q)\dot{J}_c(q,\dot{q})). \tag{8.21}$$

The CLF derivative in the cost is then given by:

$$\dot{V}(q,\mathcal{X}) := \begin{bmatrix} L_G V_\varepsilon(x) J_y^{\perp}(q) & 0 & 0 \end{bmatrix} \mathcal{X}, \tag{8.22}$$

with the remaining cost terms defined as:

$$A(x) = \begin{bmatrix} w_{\text{reg}} I \\ w_y J_y^{\perp}(q) \\ w_\lambda J_s(q) \end{bmatrix}, \qquad b = \begin{bmatrix} w_{\text{reg}} \mathcal{X}_\alpha^*(t) \\ w_y \left(\ddot{y}^d(t) - \dot{J}_y^{\perp}(q,\dot{q})\dot{q}\right) \\ -w_\lambda \dot{J}_s(q,\dot{q})\dot{q} \end{bmatrix}. \tag{8.23}$$

The soft constraints applied to the robot allow us to have some influence via the cost with regards to the force that the robot exerts on the world. Specifically, by including the nominal ground reaction forces and accelerations into the model as a weighted trade-off with the virtual constraint tracking problem, the QP can allow for some model inaccuracy to be compensated via nonzero constraint accelerations. This is shown in Fig. 8.16, where the soft constraints for the foot constraints are shown, with violation spikes shown at impact, and with a near-constant offset in the vertical direction. The vertical acceleration offset will be discussed further in Sec. 8.2.4.

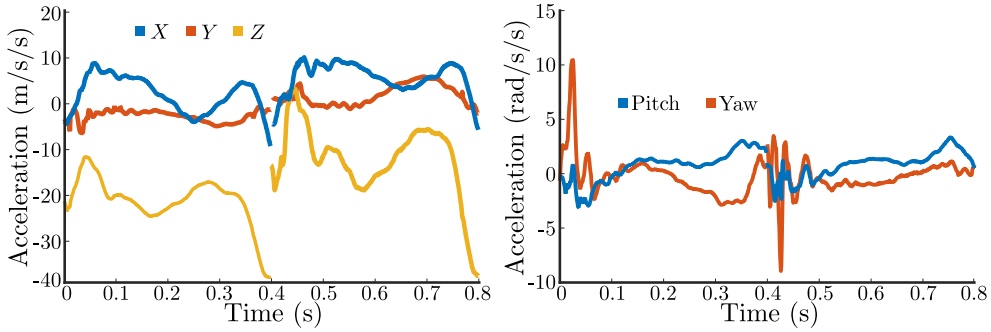

Figure 8.16: Shown are the acceleration violations of the soft constraint on the stance foot contact over two steps. The largest error is in the vertical direction, which corresponds to our observation that the total mass of the robot has some inaccuracy.

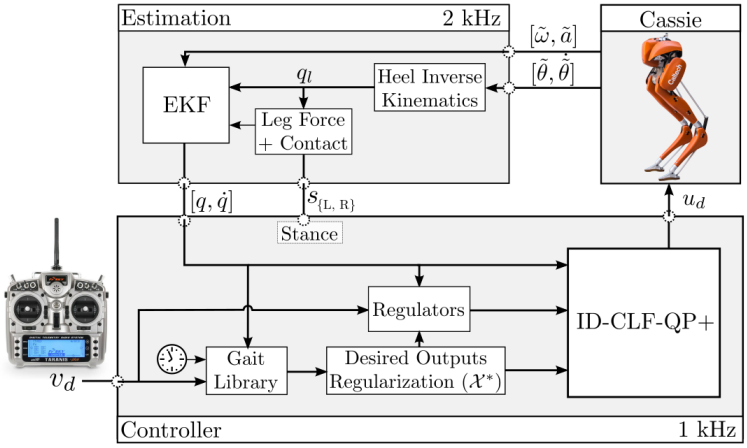

Figure 8.17: The control and estimation diagram for implementation of the (ID-CLF-QP$^+$) on Cassie. The estimation and controller ROS nodes run in separate threads on dedicated cores of the Intel NUC CPU at 2 kHz and 1 kHz, respectively.

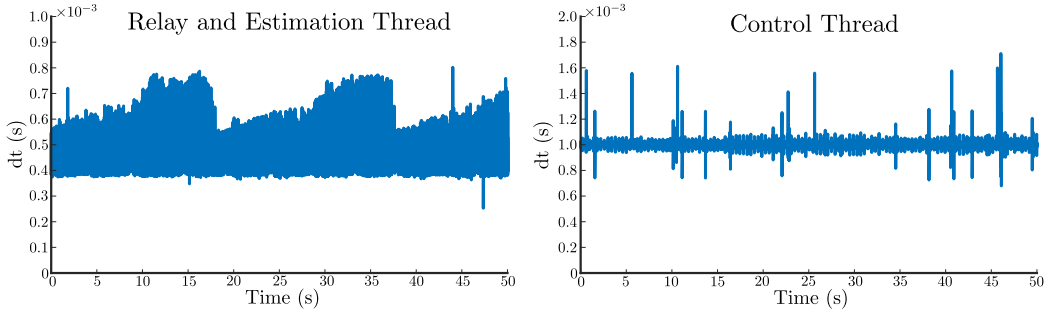

Figure 8.18: Thread timing for the two ROS nodes which are running to enable soft real-time control of Cassie. The estimation and communication relay takes place at roughly 2 kHz, while the feedback controller is evaluated at 1 kHz.

8.2.3 Implementation

The software infrastructure for feedback control and implementation was built directly on top of the codebase which was introduced in Sec. 6.2, with additions made in order to handle the QP and additional library interpolations. The code was run as two separate C++ ROS nodes elevated to real-time priority on an Intel NUC computer, with a general diagram shown in Fig. 8.17. In addition to implementation on hardware, a Gazebo simulation which directly simulates the code running on hardware was created in order to facilitate code testing and controller tuning without needing direct access to the robot. The Gazebo simulation, robot models, feedback control code, and trajectory optimization have been made available as a series of open-sourced software packages[4].

[4]Open-sourced repository for the C++ controller code used in this work on hardware and for use in a Gazebo simulation: `https://github.com/jpreher/cassie_documentation`

One of the most significant considerations which had to be made in this section as opposed to Sec. 6.2 is whether the QP controller can evaluate within the 1 kHz control frequency that is prescribed for the control thread. Loop timing was recorded over 50 seconds of walking on hardware for Cassie, and is shown in Fig. 8.18. The resulting plots show some timing jitter, with several very intermittent delays showing up in the control thread, while the estimation thread is reasonably uniform. Despite the intermittent jitter, there was no significant disruption to the controller, though there are many software improvements that could be made to improve this in future work.

The heuristic controllers given in Sec. 6.2 for foot placement and acceleration augmentation given in (6.41) and (6.42), respectively, are again used with some small adaptations. Specifically, the acceleration regulator now drops the integral term, and instead only has PD feedback:

$$\ddot{q}^d_{x,y} = \ddot{q}^*_{x,y} + k_{\ddot{q},p}\left(p^a_{x,y}(q) - p^*_{x,y}\right) + k_v(\tilde{v}^a_{x,y} - v^d_{x,y}) \tag{8.24}$$

where k_v is a gain on the step velocity tracking error and $k_{\ddot{q},p}$ is a gain affecting the x and y position errors of the pelvis relative to the stance foot. Also, the foot placement controller adds an integral term:

$$\Delta := \begin{bmatrix}\Delta_x\\ \Delta_y\\ \Delta_z\end{bmatrix} = \begin{bmatrix}\tilde{K}_{p,x}(\tilde{v}^a_x - v^d_x) + \tilde{K}_{d,x}(\tilde{v}^a_x - \bar{v}^a_{x,k-1}) + k_{i,x}\int_0^t \gamma(\tilde{v}^a_{x,y}(t') - v^d_{x,y}(t'))dt',\\ \tilde{K}_{p,y}(\tilde{v}^a_y - v^d_y) + \tilde{K}_{d,y}(\tilde{v}^a_y - \bar{v}^a_{y,k-1}) + k_{i,y}\int_0^t \gamma(\tilde{v}^a_{x,y}(t') - v^d_{x,y}(t'))dt',\\ 0\end{bmatrix},$$

$$y^d_{sw,ll} = ||p^*_{nsf}(y^d) + \Delta||_2,$$

$$y^d_{lp} = \sin^{-1}\left(\frac{p^*_x(y^d) + \Delta_x}{y^d_{sw,ll}}\right) - y^d_{b,x}(t,\alpha),$$

$$y^d_{lr} = \sin^{-1}\left(\frac{p^*_y(y^d) + \Delta_y}{y^d_{sw,ll}}\right) - y^d_{b,y}(t,\alpha),$$

where k_i is a gain on the accumulated step velocity error with a decay constant of $\gamma = 0.9995$ to avoid integral windup. The current step velocity is given by $\tilde{v}^a_{x,y} = \bar{v}^a_{k-1} + \left(v^a_{x,y} - v^*_{x,y}\right)$, $v^d_{x,y}$ is the target step velocity from the user joystick, $v^a_{x,y}$ is the instantaneous velocity of the robot relative to the stance foot, and $p^*_{nsf}(y^d) = (p^*_{nsf,x}, p^*_{nsf,y}, p^*_{nsf,z})$ are the nominal Cartesian swing foot positions..

Using the heuristics and controllers given thus far, it was found that the robot was able to reliably step in place and walk undisturbed. However, because the walking gait is now regularized around the nominal contact forces of the current walking velocity, it would often lose energy with any small disturbances and lose a significant amount

Table 8.2: Feedback control values used for the regulator heuristics on Cassie.

Parameter	Value
$\tilde{K}_p^{x,y}$	$[0.045, 0.0375]$
$\tilde{K}_d^{x,y}$	$[0.18, 0.21]$
$\tilde{K}_i^{x,y}$	$[0.06, 0]$
$k_{\ddot{q},p}^{x,y}$	$[2, 2]$
$k_v^{x,y}$	$[3, 3]$
$k_{D,L}, k_{D,L}$	$[1800, 22]$
$k_{\lambda,1}, k_{\lambda,2}, k_{\lambda,3}, k_{\lambda,4}$	$[0.025, 0.04, 0.0075, 0]$

of speed, needing several steps to recover. To overcome this energy loss, we proposed an additional regulator which finds a corrective force along the axial length of the leg, ΔF_l, adapted from the leg length heuristic introduced in [83]:

$$\Delta F_l = -k_{p,F} y_{2,\text{sll}}(t,q) - k_{d,F} \dot{y}_{2,\text{sll}}(t,q,\dot{q}) + \begin{cases} \left[k_{\lambda,1} v_x^d + k_{\lambda,2}(v_x^d - \bar{v}_{x,k-1}^a)\right] \frac{x}{x_0} & \text{if } x \leq 0 \\ \left[k_{\lambda,3} v_x^d + k_{\lambda,4}(v_x^d - \bar{v}_{x,k-1}^a)\right] \frac{x}{x_0} & \text{if } x > 0 \end{cases}$$

where x is the forward position within the stride. The QP is then regularized with the augmented value $\lambda^* = \lambda_\alpha^*(t) + R_l(q)\Delta F_l$ where $R_l(q)$ is a rotation matrix which maps the axial force into a Cartesian force at the stance foot.

All of the heuristic gains used on hardware for walking on Cassie are given in Table 8.2.

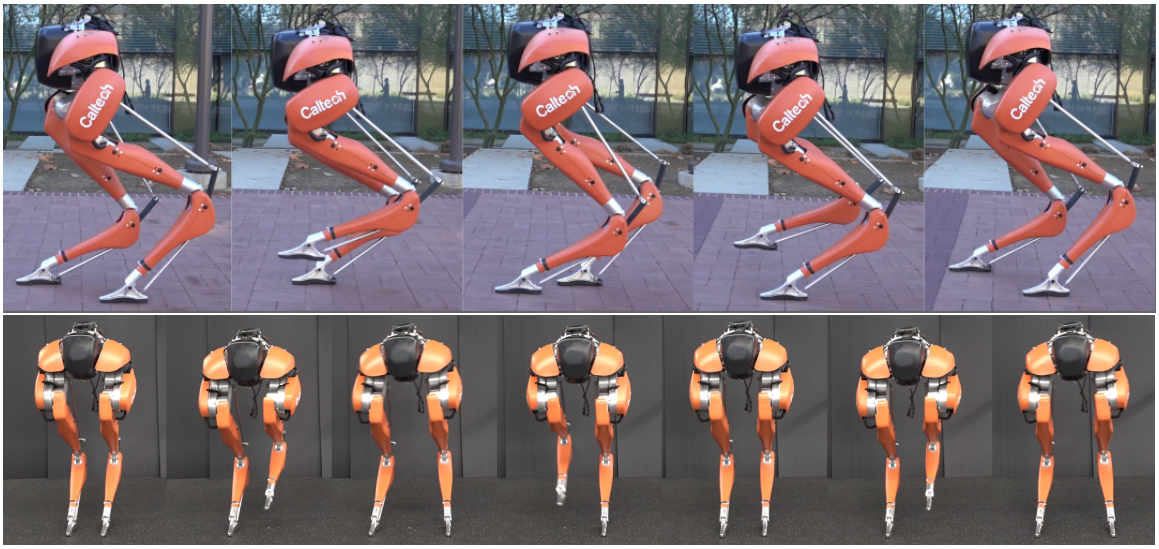

Figure 8.19: Gait tiles for Cassie walking with an ID-CLF-QP controller on a sidewalk at Caltech, and for lateral strafing indoors.

8.2.4 Results and Conclusions

The results presented in this section demonstrate stable walking on Cassie over a variety of terrains, for walking in all directions, and when subjected to a push disturbance while stepping in place. Overall, these results illustrate the *ability of the* (ID-CLF-QP) *control framework to stabilize underactuated HZD locomotion while accounting for passive compliance within a robot model.* Walking was first performed indoors for each of the behaviors to isolate segments of walking data to plot. The robot was then taken outdoors where it walked on sidewalks and over rough terrain. In addition, a video of these experiments is provided[5]. Gait tiles for forward walking and a lateral strafe are shown in Fig. 8.20, where the period-2 orbital nature of the lateral walking is clearly indicated by the contact configurations in every other frame.

The accuracy of the velocity tracking for each of the motions is shown in Fig. 8.20, where the dashed line is the nominal value from the optimization and the solid line is the estimated value obtained from an EKF running onboard. While there is some slight vibration when walking at faster speeds, the overall behavior is stable.

The virtual constraint tracking for forward walking is depicted in Fig. 8.22, where reasonable tracking performance is shown. The CLF value is also provided, which approaches its largest value near mid-step, but converges back to a low value near the end of the step and prior to impact. In addition, the commanded torques and optimal constraint forces for the same steps during a forward walking motion are given in Fig. 8.22. This is one of the more significant results in comparison to the walking in Sec. 6.2, as the QP-based controller is actively choosing both the constraint forces and the spring torques. In each of the torque and force plots, we can see that the nominal value from optimization closely matches. The torques have some minor oscillations near impact when the velocities of the system jump, but are otherwise smooth. Further, the passive dynamics of the spring forces were clearly captured by the HZD trajectory optimization, and then enforced in a compliant manner on hardware. Finally, the friction pyramid is shown with the bound becoming very small near impact due to the motion not having a planned double-support domain.

The same output and CLF plots for backwards and lateral walking are provided in Fig. 8.26 and Fig. 8.28, while the torques and forces for backward and lateral walking are given in Fig. 8.25 and Fig. 8.27. Each of these plots closely resemble the results that were discussed for the forward walking case. However, the lateral walking plots

[5]Video of experimental and simulation results for ID-CLF-QP experiments: `https://youtu.be/vACSWOF6ap0`

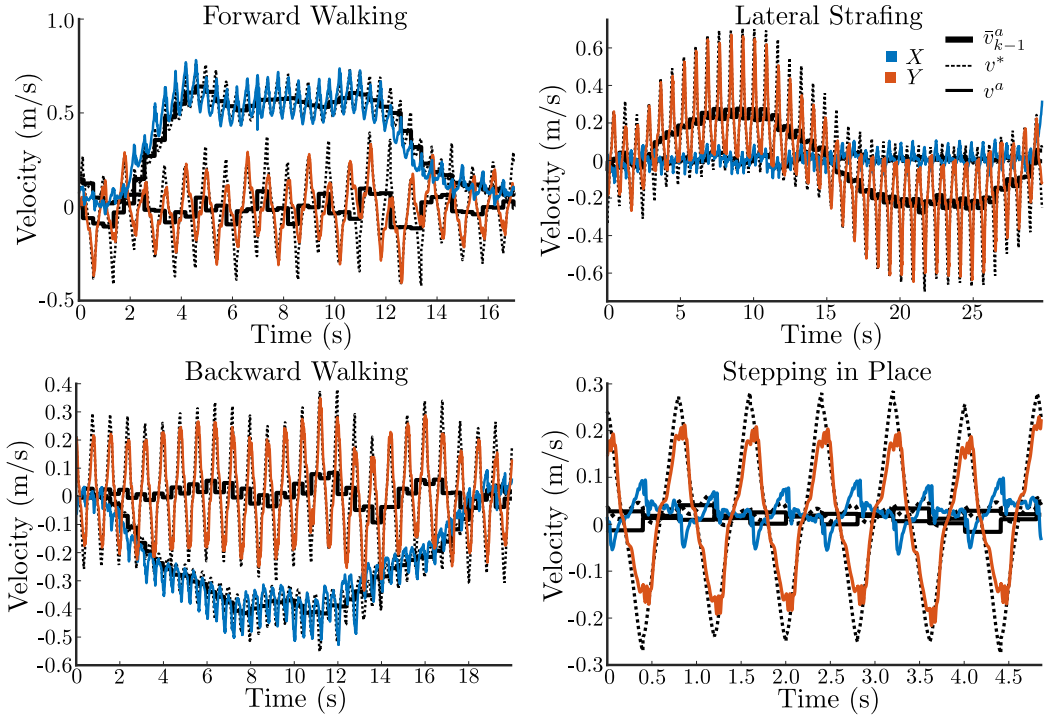

Figure 8.20: Shown is a comparison of the desired velocities from the current gait for the gait library on Cassie, $\bar{v}^a_{k-1}$, compared with the actual velocity of the robot for different behaviors while using an ID-CLF-QP controller.

show that the friction pyramid is triggered for a portion of the first step in the data that was plotted. This is particularly interesting because we can see that when the friction pyramid constraint is triggered, the CLF value rises and the forces and torques begin to deviate from the nominal values given by the optimization.

The robot also performed well under disturbances, both in the form of terrain and pushes. Motion tiles for Cassie walking outdoors over a system of roots and while being pushed aggressively from behind are shown in Fig. 8.23. The velocity, outputs, and CLF values which correspond to the push that is depicted are plotted in Fig. 8.24. It can be seen that the push drives the robot forward to approximately 1 m/s, after which the robot recovers to a near-zero speed in 5 steps, and has damped out any remaining oscillations in the velocity within 8 seconds. Also shown is the CLF value over the course of the push and recovery. The initial impact of the push coincides with a marked spike in the value, and during the push and following 5 steps, the mean value of the CLF is elevated. However, we can see that the CLF value neither grows significantly nor is driven completely to zero over time because of the constraint relaxation on the (rapidly) exponential convergence rate to the zero dynamics surface.

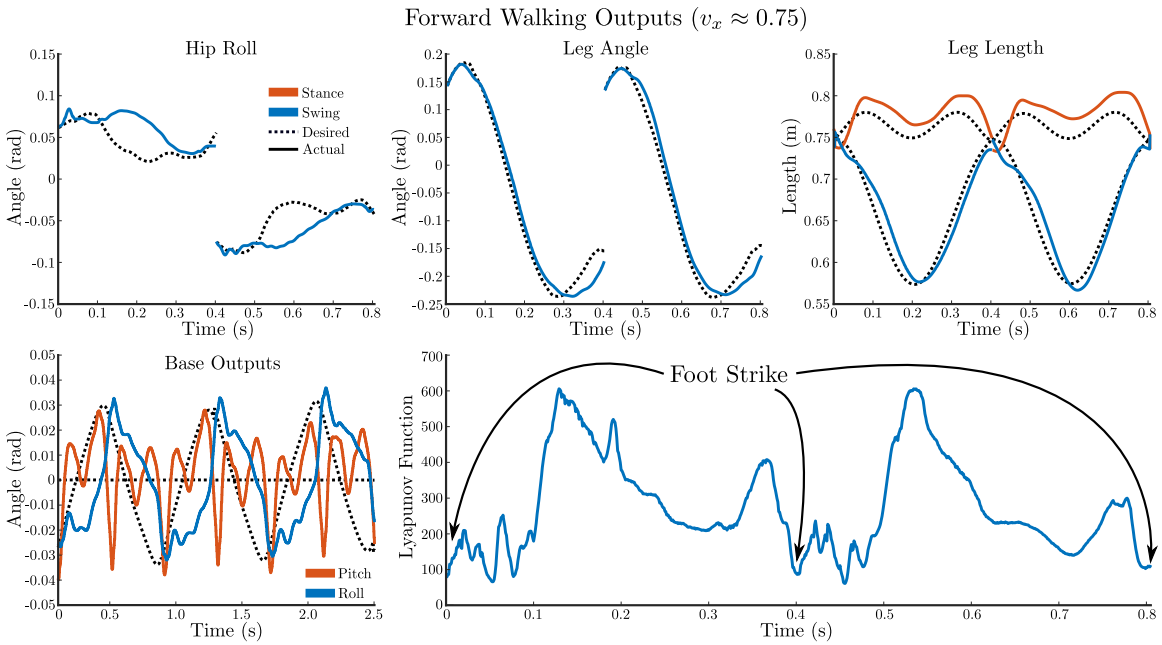

Figure 8.21: Output tracking for forward walking with an ID-CLF-QP controller on Cassie, with the Lyapunov function evolution over two steps also shown.

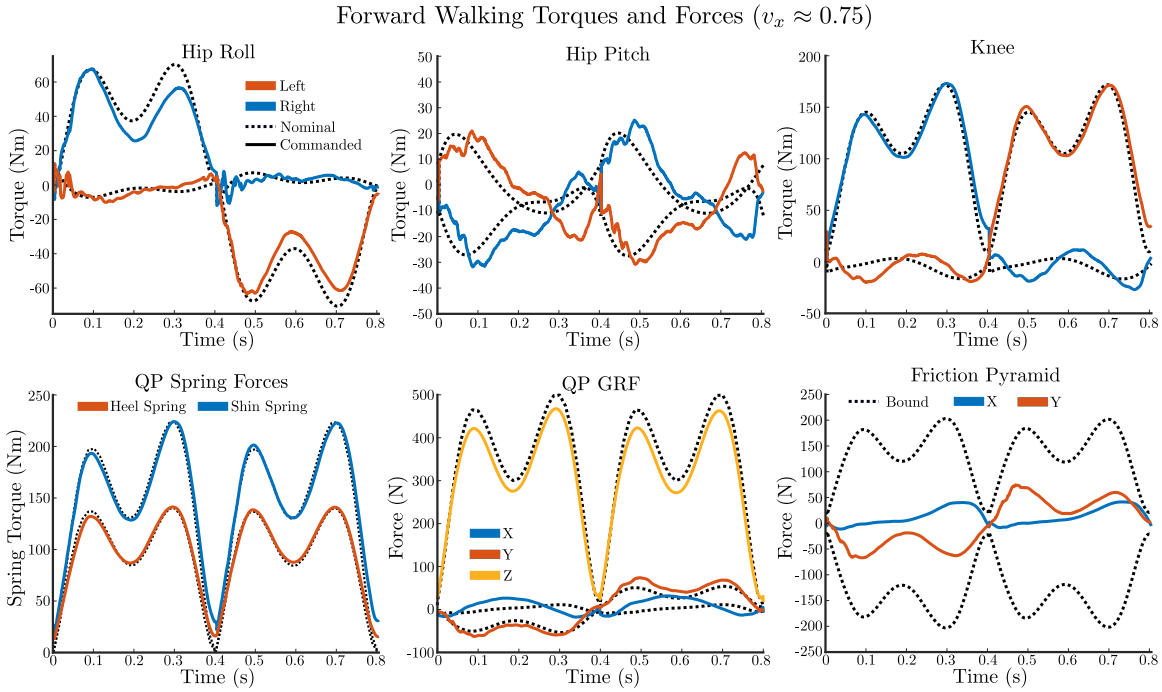

Figure 8.22: Inputs selected by the ID-CLF-QP which are applied to Cassie for forward walking. Pictured are torques and contact forces, along with the friction pyramid constraint satisfaction and the spring forces which are chosen by the QP compared to their nominal values obtained within the HZD optimization.

Figure 8.23: Gait tiles of perturbed walking on Cassie while using the ID-CLF-QP controller. Shown on the top is the robot walking over a series of roots outdoors, while on the bottom, the robot is aggressively pushed and then recovers.

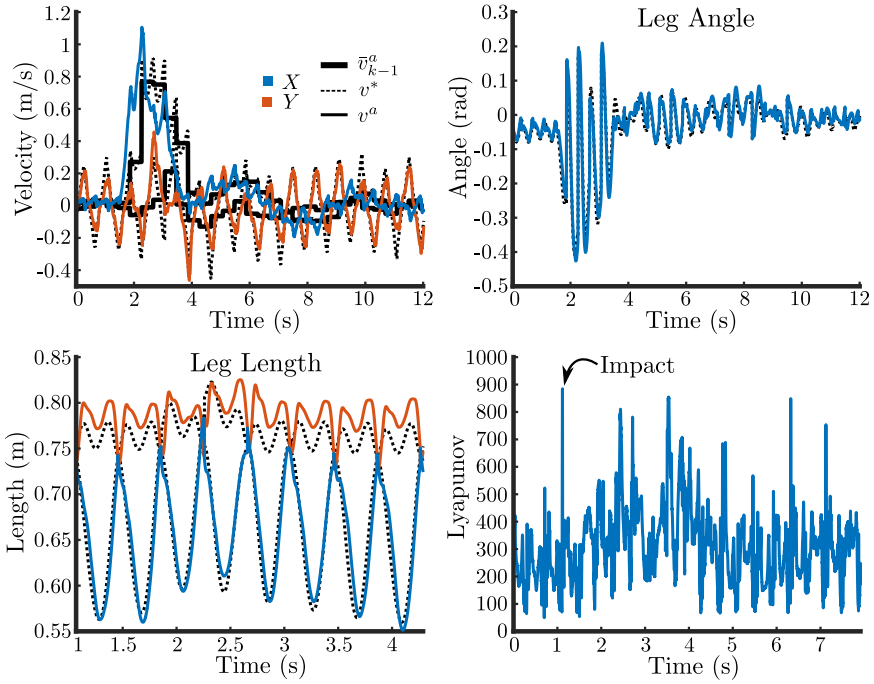

Figure 8.24: Velocity and outputs for a large push from behind while walking on Cassie. The velocity tracking plot and desired outputs show that the library accurately captures the robot motion under even perturbed walking speeds.

These results demonstrate the first successful experimental walking with CLFs on a 3D biped that the authors are aware of to date. The accuracy of the robot model and synthesized motion library allowed the control implementation to use a regularization term as described in Proposition 7.3.1.2. The use of this regularization, combined with the relaxed form of the (ID-CLF-QP$^+$) gave smooth torques and force references which would have been significantly more complex to implement on a (CLF-QP-δ).

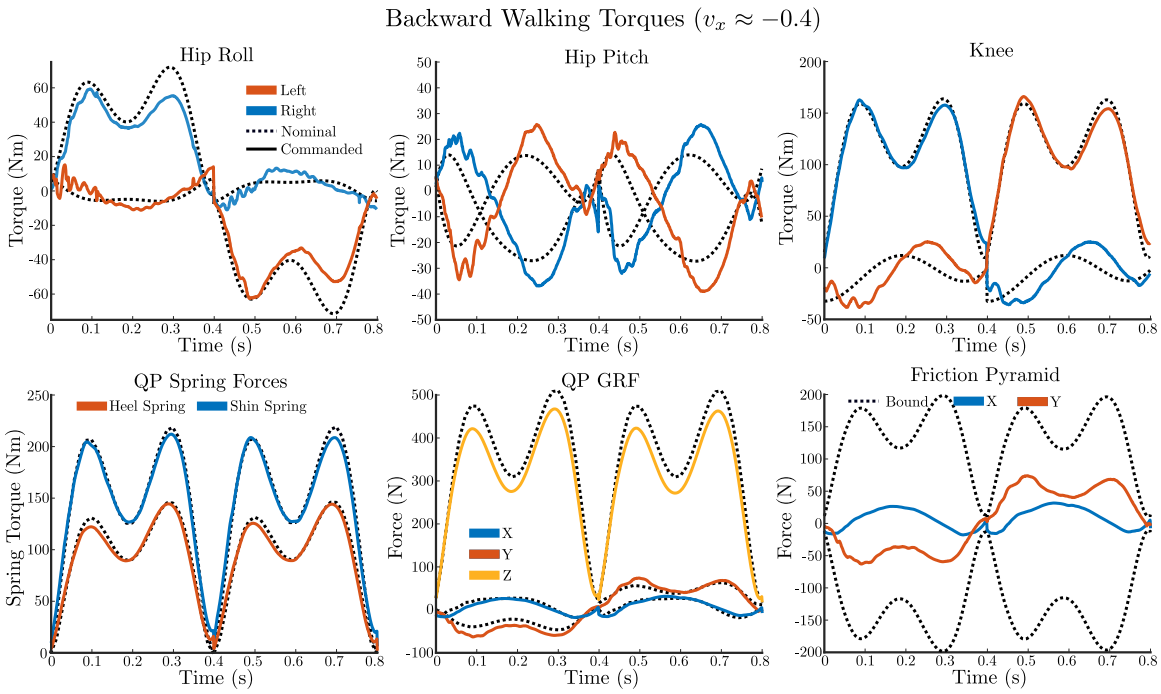

Figure 8.25: Inputs selected by the ID-CLF-QP which are applied to Cassie for backward walking. Pictured are torques and contact forces, along with the friction pyramid constraint satisfaction and the spring forces which are chosen by the QP.

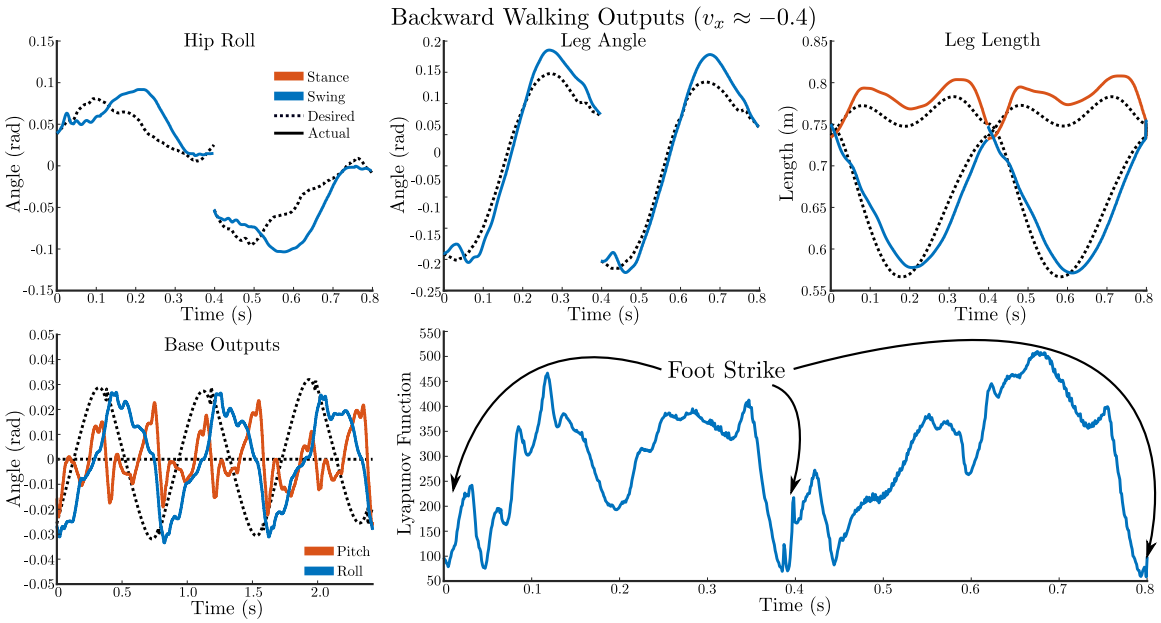

Figure 8.26: Output tracking for backwards walking with an ID-CLF-QP controller on Cassie, with the Lyapunov function evolution over two steps also shown.

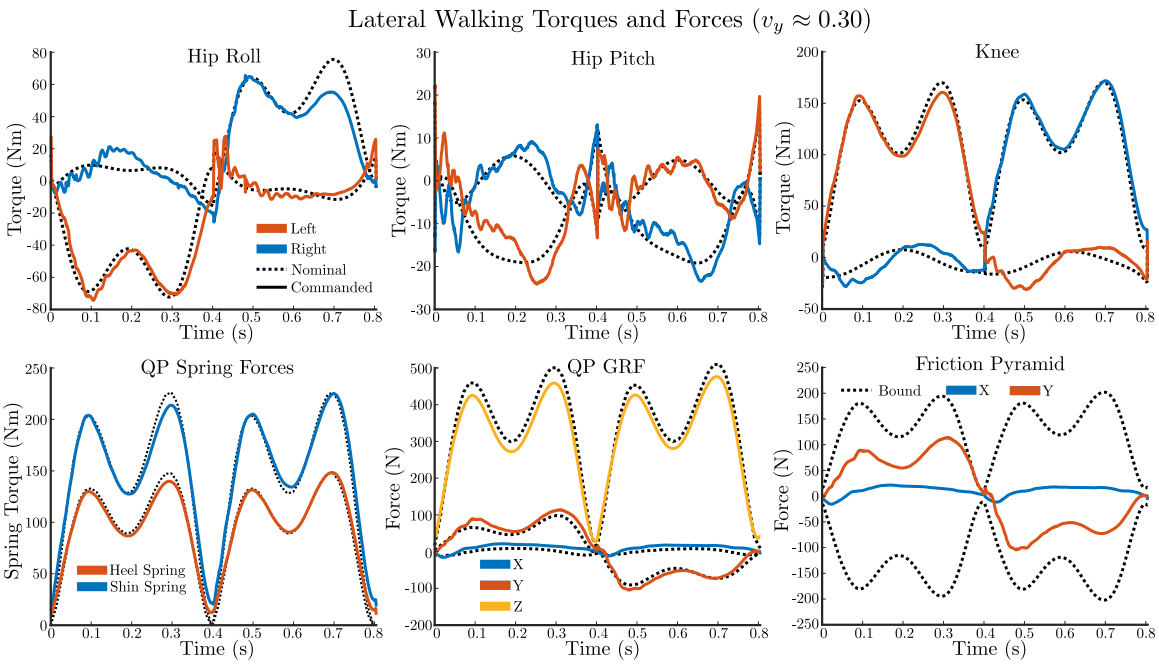

Figure 8.27: Inputs selected by the ID-CLF-QP which are applied to Cassie for lateral walking. Pictured are torques and contact forces, along with the friction pyramid constraint satisfaction. It can be seen in the bottom right figure that for the first step the friction pyramid constraint is triggered and active.

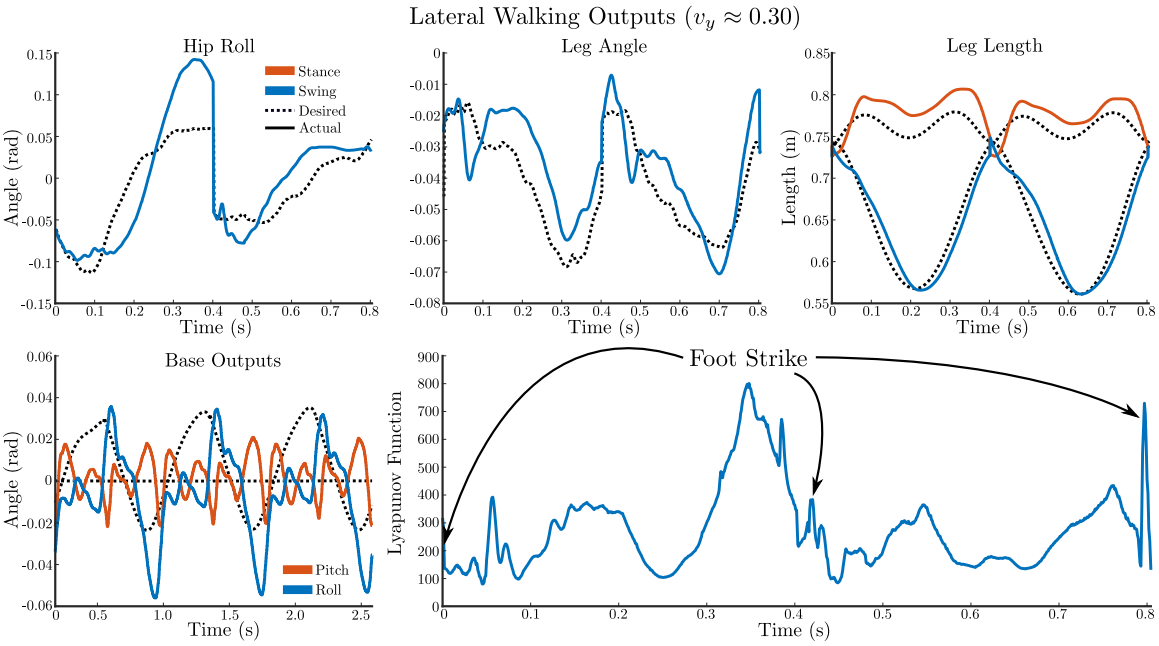

Figure 8.28: Output tracking for lateral walking with an ID-CLF-QP controller on Cassie, with the Lyapunov function evolution over two steps also shown. The period-2 nature of the walking is apparent in the reference polynomials, while the Lyapunov function is shown to have a significant relaxation at the end of the first step. This correlates with the portion of the gait when the friction pyramid was active.

Chapter 9

CONCLUSIONS

In conclusion, this thesis has presented the following discussion and contributions:

1. *Full-order nonlinear modeling of 3D underactuated and compliant bipedal robots.* Bipedal robots are inherently nonlinear with hybrid dynamical behaviors. These full order dynamics must be accounted for, either through nonlinear controllers or via optimization algorithms. This thesis focused on developing a full nonlinear and compliant model of both the DURUS and Cassie bipeds in Chap. 4, with the Cassie model being used first in an offline optimization (see Chap. 6) and then within a model-based nonlinear controller (see Chap. 8).

2. *Optimization for HZD gait generation on underactuated 3D bipeds.* An HZD-based controller was developed in Chap. 6 for Cassie to synthesize closed-loop locomotion plans while leveraging the passive compliance in the robot model to reduce impact, and generate trajectories that accurately reflect how the true compliant robotic system would evolve on hardware. In addition, this thesis presented the development of the *first and only* multi-contact humanoid walking developed with HZD to date, shown in Chap. 5, which is also the *most efficient* reported humanoid walking in the literature to date. Due to the complexity of these problems, developing and demonstrating algorithms that allow for these optimization problems to be solved efficiently was essential in instantiating successful walking gaits on hardware platforms.

3. *Parameterizing HZD optimization results for model-based control development on hardware.* This thesis develops a compliant HZD motion library for Cassie in Chap. 6 that leverages its full-body dynamics, including passive compliance. While motion libraries for sagittal motions under the assumption of sufficient rigidity have been successfully realized elsewhere on Cassie [127] and on other robots [213], all of the existing literature within the HZD field on gait libraries ignores the robot's inherent compliance. It was shown how we can use the additional information extracted from the optimization to generate regularization terms as described in Proposition 7.3.1.2 that respect the planned constraints on the robot. The end result is a set of trajectories for the virtual constraints,

forces, and torques that can be used within a model-based controller that is able to leverage the compliance of Cassie for all motion primitives in the library.

4. *Theoretical development and analysis of an inverse dynamics-based CLF controller for constrained robotic systems.* In this work, a novel formulation of an optimization-based control framework is introduced in Chap. 7 that couples convergence constraints from control Lyapunov functions with desirable formulations existing in other areas of the bipedal locomotion field that have proven successful in practice, such as inverse dynamics control and quadratic programming approaches. A theoretical analysis is shown which demonstrates several useful properties of the approach for tuning and implementation, and the stability of the controller for HZD locomotion is proven. This was extended to a relaxed version of the CLF controller, which removes a convergence inequality constraint in lieu of a conservative CLF cost within a quadratic program to achieve tracking.

5. *Control laws and methodologies for hardware realization.* Optimization-based control laws allow for the generated gaits to, ultimately, be realized on hardware in a manner which respects both the model and active constraints on the system. These control algorithms are the final step in realizing dynamic walking on bipedal robots that was shown in this thesis. In Chap. 8, a detailed formulation of the (ID-CLF-QP$^+$) controller was developed for use in real-time on the Cassie biped. The results indicate that the control method, when combined with the HZD motion library and parameterized optimization results from Chap. 6, chose smooth input torques, passive spring torques, and feasible ground reaction forces on hardware while reliably tracking the planned virtual constraints. The resulting experiments demonstrate the *first successful realization of a CLF controller on a 3D biped* in the literature.

Future Directions

There are many future directions that both the locomotion planning and real-time control development could take, including but not limited to:

1. *Generalized notions of stability and safety.* The walking considered herein, and the notions of stability, were entirely periodic in nature. To better represent a wide variety of behaviors, the idea of stability should be extended to include aperiodic walking motions [252], [253]. More generally, safety as represented

by set invariance [254] could provide a powerful tool for more generally understanding locomotion.

2. *Real-time optimal gait planning.* It was seen that nonlinear constrained optimization plays an essential role in generating dynamic walking behaviors that leverage the full-body dynamics of the robot. These methods have become very efficient, even allowing for online calculation in simple scenarios [255]. However, for the DURUS and Cassie bipeds shown in this work, finding a feasible walking gait still consumed approximately 5-10 minutes on a laptop computer. Further improving computational efficiency may enable real-time implementation of these methods yielding new paradigms for gait generation.

3. *Bridging the gap between theory and practice.* As indicated by the implementation methodologies discussed in Chapters 5, 6, and 8, there is often an "artful implementation" step that transcribes model-based controllers into a form that can actually implemented on hardware, or layers several heuristics such as foot placement in order to further stabilize the locomotion. Ideally, methods can be developed that allow the exact transcription of formal methods to hardware in a robust fashion and without heuristics.

4. *Robustness, adaptation and learning.* Dynamic walking behaviors often work in isolated instances and predefined environments. Translating these ideas to the real-world will require robustness to uncertainty – both in the internal dynamics and external environment. Adaptive and learning-based controllers can help mitigate model uncertainty and unplanned interactions with the world, from uncertain contact conditions to walking on surfaces with complex interactions, e.g., sand.

5. *Real-world deployment of bipedal robots.* While most of the experiments in this work were shown to be effective outdoors, they relied on purely proprioceptive estimation and did not actively account for changing terrains. The ultimate challenge is the ability to deploy bipedal robots in a variety of real-world scenarios. This ranges from everyday activities, to aiding humans, to venturing into dangerous environments. Examples include bipedal robotics in a healthcare setting, e.g., exoskeletons for restoring mobility [256], to humanoid robots capable of exploring remote or unsafe locations autonomously.

BIBLIOGRAPHY

[1] H.-O. Lim and A. Takanishi, "Biped walking robots created at Waseda University: WL and WABIAN family," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 49–64, 2007.

[2] M. Vukobratovic, D. Hristic, and Z. Stojiljkovic, "Development of active anthropomorphic exoskeletons," *Medical and Biological Engineering*, vol. 12, no. 1, pp. 66–80, 1974.

[3] M. Vukobratović and J. Stepanenko, "On the stability of anthropomorphic systems," *Mathematical Biosciences*, vol. 15, no. 1-2, pp. 1–37, 1972.

[4] M. Vukobratović and B. Borovac, "Zero-moment point—thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[5] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[6] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of Honda humanoid robot," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, IEEE, vol. 2, 1998, pp. 1321–1326.

[7] M. Hirose and K. Ogawa, "Honda humanoid robots development," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 11–19, 2007.

[8] K. Kaneko, F. Kanehiro, S. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, S. Ota, and T. Isozumi, "Design of prototype humanoid robotics platform for HRP," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 2431–2436.

[9] K. Akachi, K. Kaneko, N. Kanehira, S. Ota, G. Miyamori, M. Hirata, S. Kajita, and F. Kanehiro, "Development of humanoid robot HRP-3P," in *IEEE-RAS International Conference on Humanoid Robots*, Dec. 2005, pp. 50–55.

[10] Y. Ito, S. Nozawa, J. Urata, T. Nakaoka, K. Kobayashi, Y. Nakanishi, K. Okada, and M. Inaba, "Development and verification of life-size humanoid with high-output actuation system," in *IEEE International Conference on Robotics and Automation*, 2014.

[11] I.-W. Park, J.-Y. Kim, J. Lee, and J.-H. Oh, "Mechanical design of humanoid robot platform KHR-3 (KAIST humanoid robot 3: HUBO)," in *IEEE-RAS International Conference on Humanoid Robots*, 2005.

[12] M. H. Raibert, H. B. Brown Jr, and M. Chepponis, "Experiments in balance with a 3D one-legged hopping machine," *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 75–92, 1984.

[13] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[14] Y. Hürmüzlü and G. D. Moskowitz, "The role of impact in the stability of bipedal locomotion," *Dynamics and Stability of Systems*, vol. 1, no. 3, pp. 217–234, 1986.

[15] T. McGeer *et al.*, "Passive dynamic walking," *I. J. Robotic Res.*, vol. 9, no. 2, pp. 62–82, 1990.

[16] M. W. Spong and F. Bullo, "Controlled symmetries and passive walking," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 1025–1031, 2005.

[17] D. Pekarek, A. D. Ames, and J. E. Marsden, "Discrete mechanics and optimal control applied to the compass gait biped," in *2007 46th IEEE Conference on Decision and Control*, IEEE, 2007, pp. 5376–5382.

[18] A. D. Ames, R. D. Gregg, and M. W. Spong, "A geometric approach to three-dimensional hipped bipedal robotic walking," in *2007 46th IEEE Conference on Decision and Control*, IEEE, 2007, pp. 5123–5130.

[19] A. D. Ames, R. W. Sinnet, and E. D. Wendel, "Three-dimensional kneed bipedal walking: A hybrid geometric approach," in *HSCC*, Springer, 2009, pp. 16–30.

[20] 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*, vol. 46, no. 1, pp. 51–64, 2001.

[21] F. Plestan, J. W. Grizzle, E. R. Westervelt, and G. Abba, "Stable walking of a 7-DOF biped robot," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 653–668, 2003.

[22] B. Morris and J. W. Grizzle, "A restricted Poincaré map for determining exponentially stable periodic orbits in systems with impulse effects: Application to bipedal robots," in *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, 2005, pp. 4199–4206.

[23] E. Westervelt, G. Buche, and J. Grizzle, "Inducing dynamically stable walking in an underactuated prototype planar biped," in *IEEE International Conference on Robotics and Automation (ICRA), 2004.*, IEEE, vol. 4, 2004, pp. 4234–4239.

[24] B. Morris, E. Westervelt, C. Chevallereau, G. Buche, and J. Grizzle, "Fast motions symposium on biomechanics and robotics," in, K. Mombaur and M. Dheil, Eds., ser. Lecture Notes in Control and Information Sciences. Heidelberg, Germany: Springer-Verlag, 2006, ch. Achieving Bipedal Running with RABBIT: Six Steps toward Infinity, pp. 277–297.

[25] H.-W. Park, K. Sreenath, J. W. Hurst, and J. W. Grizzle, "Identification of a bipedal robot with a compliant drivetrain," *IEEE Control Systems*, vol. 31, no. 2, pp. 63–88, 2011.

[26] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[27] K. Sreenath, H.-W. Park, and J. Grizzle, "Design and experimental implementation of a compliant hybrid zero dynamics controller with active force control for running on MABEL," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 51–56.

[28] H.-W. Park, K. Sreenath, A. Ramezani, and J. W. Grizzle, "Switching control design for accommodating large step-down disturbances in bipedal robot walking," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 45–50.

[29] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[30] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.

[31] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[32] C. Shih, J. Grizzle, and C. Chevallereau, "Asymptotically stable walking of a simple underactuated 3D bipedal robot," in *The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Taipei, Taiwan, Nov. 2007, pp. 2766–2771.

[33] J. W. Grizzle, C. Chevallereau, and C.-L. Shih, "HZD-based control of a five-link underactuated 3D bipedal robot," in *2008 47th IEEE Conference on Decision and Control*, IEEE, 2008, pp. 5206–5213.

[34] C. Chevallereau, J. Grizzle, and C. Shih, "Asymptotically stable walking of a five-link underactuated 3D bipedal robot," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 37–50, Feb. 2009.

[35] C. Chevallereau, J. W. Grizzle, and C.-L. Shih, "Steering of a 3D bipedal robot with an underactuated ankle," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 1242–1247.

[36] C.-L. Shih, J. Grizzle, and C. Chevallereau, "From stable walking to steering of a 3D bipedal robot with passive point feet," *Robotica*, vol. 30, no. 7, pp. 1119–1130, 2012.

[37] I. Poulakakis and J. Grizzle, "Monopedal running control: SLIP embedding and virtual constraint controllers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, Oct. 2007, pp. 323–330.

[38] I. Poulakakis and J. W. Grizzle, "The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1779–1793, 2009.

[39] I. Poulakakis and J. Grizzle, "Formal embedding of the spring loaded inverted pendulum in an asymmetric hopper," in *2007 European Control Conference (ECC)*, IEEE, 2007, pp. 3159–3166.

[40] R. J. Full and D. E. Koditschek, "Templates and anchors: Neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[41] K. Sreenath, H.-W. Park, I. Poulakakis, and J. 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*, vol. 32, no. 3, pp. 324–345, 2013.

[42] W.-L. Ma, S. Kolathaya, E. R. Ambrose, C. M. Hubicki, and A. D. Ames, "Bipedal robotic running with durus-2d: Bridging the gap between theory and experiment," in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, 2017, pp. 265–274.

[43] 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 *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 1794–1801.

[44] J. Reher, A. Hereid, S. Kolathaya, C. M. Hubicki, and A. D. Ames, "Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot DURUS," in *Twelfth International Workshop on Algorithmic Foundations on Robotics*, 2016.

[45] M. Vukobratovic, A. A. Frank, and D. Juricic, "On the stability of biped locomotion," *IEEE Transactions on Biomedical Engineering*, no. 1, pp. 25–36, 1970.

[46] T. Arakawa and T. Fukuda, "Natural motion generation of biped locomotion robot using hierarchical trajectory generation method consisting of GA, EP layers," in *Proceedings of International Conference on Robotics and Automation*, IEEE, vol. 1, 1997, pp. 211–216.

[47] C.-L. Shih, Y. Li, S. Churng, T.-T. Lee, and W. A. Gruver, "Trajectory synthesis and physical admissibility for a biped robot during the single-support phase," in *IEEE International Conference on Robotics and Automation*, IEEE, 1990, pp. 1646–1652.

[48] J.-i. Yamaguchi, A. Takanishi, and I. Kato, "Development of a biped walking robot compensating for three-axis moment by trunk motion," *Journal of the Robotics Society of Japan*, vol. 11, no. 4, pp. 581–586, 1993.

[49] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. Center of pressure-zero moment point," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, 2004.

[50] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A universal stability criterion of the foot contact of legged robots-adios ZMP," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, IEEE, 2006, pp. 1976–1983.

[51] S. Kajita, O. Matsumoto, and M. Saigo, "Real-time 3d walking pattern generation for a biped robot with telescopic legs," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, IEEE, vol. 3, 2001, pp. 2299–2306.

[52] 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 *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, IEEE, vol. 2, 2003, pp. 1620–1626.

[53] F. Pfeiffer, K. Loffler, and M. Gienger, "The concept of jogging Johnnie," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, IEEE, vol. 3, 2002, pp. 3129–3135.

[54] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, IEEE, 2006, pp. 200–207.

[55] A. L. Hof, "The 'extrapolated center of mass' concept suggests a simple control of balance in walking," *Human Movement Science*, vol. 27, no. 1, pp. 112–125, 2008.

[56] M. B. Popovic, A. Goswami, and H. Herr, "Ground reference points in legged locomotion: Definitions, biological trajectories and control implications," *The International Journal of Robotics Research*, vol. 24, no. 12, pp. 1013–1032, 2005.

[57] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[58] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, *et al.*, "Team IHMC's lessons learned from the DARPA robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.

[59] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[60] M. DeDonato, V. Dimitrov, R. Du, R. Giovacchini, K. Knoedler, X. Long, F. Polido, M. A. Gennert, T. Padır, S. Feng, *et al.*, "Human-in-the-loop control of a humanoid robot for disaster response: A report from the DARPA robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 275–292, 2015.

[61] J. E. Pratt and S. V. Drakunov, "Derivation and application of a conserved orbital energy for the inverted pendulum bipedal walking model," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, 2007, pp. 4653–4660.

[62] J. H. Park and K. D. Kim, "Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control," in *1998 IEEE International Conference on Robotics and Automation*, IEEE, vol. 4, 1998, pp. 3528–3533.

[63] B. Stephens, "Push recovery control for force-controlled humanoid robots," PhD thesis, Carnegie Mellon University, The Robotics Institute, 2011.

[64] 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*, vol. 31, no. 10, pp. 1117–1133, 2012.

[65] A. Hof, M. Gazendam, and W. Sinke, "The condition for dynamic stability," *Journal of Biomechanics*, vol. 38, no. 1, pp. 1–8, 2005.

[66] S.-H. Hyon, J. G. Hale, and G. Cheng, "Full-body compliant human–humanoid interaction: Balancing in the presence of unknown external forces," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 884–898, 2007.

[67] B. Stephens, "Humanoid push recovery," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2007, pp. 589–595.

[68] A. Takanishi, T. Takeya, H. Karaki, and I. Kato, "A control method for dynamic biped walking under unknown external force," in *IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, IEEE, 1990, pp. 795–801.

[69] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 4420–4427.

[70] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot-1st report: Walking gait pattern generation," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 1084–1091.

[71] S. Kazuo, K. Tsuchiya, and K. Tsujita, "The intelligent ASIMO system overview and integration," in *ICRA*, New Orleans, LA, 2004, pp. 3043–3048.

[72] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, 2006.

[73] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, vol. 22, no. 11-12, pp. 1217–1227, 1989.

[74] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[75] M. Ahmadi and M. Buehler, "Controlled passive dynamic running experiments with the ARL-monopod II," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 974–986, 2006.

[76] R. Altendorfer, D. E. Koditschek, and P. Holmes, "Stability analysis of legged locomotion models by symmetry-factored return maps," *The International Journal of Robotics Research*, vol. 23, no. 10-11, pp. 979–999, 2004.

[77] J. Rummel, Y. Blum, H. M. Maus, C. Rode, and A. Seyfarth, "Stable and robust walking with compliant legs," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2010, pp. 5250–5255.

[78] G. Zeglin and B. Brown, "Control of a bow leg hopping robot," in *1998. Proceedings. 1998 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, vol. 1, 1998, pp. 793–798.

[79] I. W. Hunter, J. M. Hollerbach, and J. Ballantyne, "A comparative analysis of actuator technologies for robotics," *Robotics Review*, vol. 2, pp. 299–342, 1991.

[80] P. Kormushev, B. Ugurlu, S. Calinon, N. G. Tsagarakis, and D. G. Caldwell, "Bipedal walking energy minimization by reinforcement learning with evolving policy parameterization," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 318–324.

[81] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, *et al.*, "Valkyrie: NASA's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[82] J. A. Grimes and J. W. Hurst, "The design of ATRIAS 1.0 a unique monopod, hopping robot," in *Proceedings of the 2012 International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2012, pp. 548–554.

[83] S. Rezazadeh, C. Hubicki, M. Jones, A. Peekema, J. Van Why, A. Abate, and J. Hurst, "Spring-mass walking with ATRIAS in 3D: Robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot," in *Proceedings of the ASME 2015 Dynamic Systems and Control Conference*, 2015.

[84] A. M. Abate, "Mechanical design for robot locomotion," PhD thesis, Oregon State University, 2018.

[85] M. Ahmadi and M. Buehler, "The ARL monopod II running robot: Control and energetics," in *Proceedings 1999 IEEE International Conference on Robotics and Automation*, IEEE, vol. 3, 1999, pp. 1689–1694.

[86] B. Brown and G. Zeglin, "The bow leg hopping robot," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, IEEE, vol. 1, 1998, pp. 781–786.

[87] S.-H. Hyon and T. Mita, "Development of a biologically inspired hopping robot-"kenken"," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, IEEE, vol. 4, 2002, pp. 3984–3991.

[88] A. Stumpf, S. Kohlbrecher, D. C. Conner, and O. von Stryk, "Open source integrated 3d footstep planning framework for humanoid robots," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, IEEE, 2016, pp. 938–945.

[89] K. Nishiwaki and S. Kagami, "Online walking control system for humanoids with short cycle pattern generation," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 729–742, 2009.

[90] P.-B. 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*, IEEE, 2006, pp. 137–142.

[91] B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2010, pp. 52–59.

[92] P.-B. Wieber, "Viability and predictive control for safe locomotion," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2008, pp. 1103–1108.

[93] Q. Huang, S. Kajita, N. Koyachi, K. Kaneko, K. Yokoi, H. Arai, K. Komoriya, and K. Tanie, "A high stability, smooth walking pattern for a biped robot," in *Proceedings 1999 IEEE International Conference on Robotics and Automation*, IEEE, vol. 1, 1999, pp. 65–71.

[94] 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 *2003 IEEE International Conference on Robotics and Automation*, IEEE, vol. 2, 2003, pp. 1620–1626.

[95] K. Nagasaka, H. Inoue, and M. Inaba, "Dynamic walking pattern generation for a humanoid robot based on optimal gradient method," in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, vol. 6, 1999, pp. 908–913.

[96] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[97] P. Hebert, M. Bajracharya, J. Ma, N. Hudson, A. Aydemir, J. Reid, C. Bergh, J. Borders, M. Frost, M. Hagman, *et al.*, "Mobile manipulation and mobility as manipulation—design and algorithms of robosimian," *Journal of Field Robotics*, vol. 32, no. 2, pp. 255–274, 2015.

[98] M. Zucker, S. Joo, M. X. Grey, C. Rasmussen, E. Huang, M. Stilman, and A. Bobick, "A general-purpose system for teleoperation of the DRC-HUBO humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 336–351, 2015.

[99] T. McGeer, "Passive walking with knees," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, IEEE, 1990, pp. 1640–1645.

[100] S. H. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive-dynamic walking robot with two legs and knees," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 607–615, 2001.

[101] S. H. Collins and A. Ruina, "A bipedal walking robot with efficient and human-like gait," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 1983–1988.

[102] P. Bhounsule, A. Ruina, *et al.*, "Cornell ranger: Energy-optimal control," *Dynamic walking*, 2009.

[103] R. Tedrake, T. W. Zhang, and H. S. Seung, "Stochastic policy gradient reinforcement learning on a simple 3d biped," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, vol. 3, 2004, pp. 2849–2854.

[104] R. Tedrake, T. W. Zhang, M.-f. Fong, and H. S. Seung, "Actuating a simple 3d passive dynamic walker," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, IEEE, vol. 5, 2004, pp. 4656–4661.

[105] S. O. Anderson, M. Wisse, C. G. Atkeson, J. K. Hodgins, G. Zeglin, and B. Moyer, "Powered bipeds based on passive dynamic principles," in *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, IEEE, 2005, pp. 110–116.

[106] M. Wisse and R. Q. Van der Linde, *Delft pneumatic bipeds*. Springer Science & Business Media, 2007, vol. 34.

[107] J. Reher, N. Csomay-Shanklin, D. L. Christensen, B. Bristow, A. D. Ames, and L. Smoot, "Passive dynamic balancing and walking in actuated environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020.

[108] F. Pfeiffer and C. Glocker, *Multi-Body Dynamics with Unilateral Constraints*, ser. Wiley Series in Nonlinear Science. New York: JW, 1996.

[109] Y. Hurmuzlu and D. B. Marghitu, "Rigid body collisions of planar kinematic chains with multiple contact points," *The International Journal of Robotics Research*, vol. 13, no. 1, pp. 82–92, 1994.

[110] A. Goswami, B. Thuilot, and B. Espiau, "A study of the passive gait of a compass-like biped robot: Symmetry and chaos," *The International Journal of Robotics Research*, vol. 17, no. 12, pp. 1282–1301, 1998.

[111] 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*, vol. 54, no. 8, pp. 1751–1764, 2009.

[112] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. C. de Wit, and J. Grizzle, "Rabbit: A testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

[113] J. W. Grizzle, J. Hurst, B. Morris, H.-W. Park, and K. Sreenath, "MABEL, a new robotic bipedal walker and runner," in *American Control Conference, 2009. ACC'09.*, IEEE, 2009, pp. 2030–2036.

[114] S. N. Yadukumar, M. Pasupuleti, and A. D. Ames, "From formal methods to algorithmic implementation of human inspired control on bipedal robots," in *Algorithmic Foundations of Robotics X*, Springer, 2013, pp. 511–526.

[115] K. A. Hamed and J. W. Grizzle, "Event-based stabilization of periodic orbits for underactuated 3-D bipedal robots with left-right symmetry," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 365–381, 2013.

[116] A. Ramezani, J. W. Hurst, K. Akbari Hamed, and J. W. Grizzle, "Performance analysis and feedback control of ATRIAS, a three-dimensional bipedal robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 2, 2014.

[117] H.-H. Zhao, W.-L. Ma, M. B. Zeagler, and A. D. Ames, "Human-inspired multi-contact locomotion with AMBER2," in *ICCPS'14: ACM/IEEE 5th International Conference on Cyber-Physical Systems (with CPS Week 2014)*, IEEE Computer Society, 2014, pp. 199–210.

[118] E. Ambrose, W.-L. Ma, C. Hubicki, and A. D. Ames, "Toward benchmarking locomotion economy across design configurations on the modular robot: Amber-3m," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2017, pp. 1270–1276.

[119] A. D. Ames, E. A. Cousineau, and M. J. Powell, "Dynamically stable bipedal robotic walking with nao via human-inspired hybrid zero dynamics," in *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, 2012, pp. 135–144.

[120] B. G. Buss, A. Ramezani, K. A. Hamed, B. A. Griffin, K. S. Galloway, and J. W. Grizzle, "Preliminary walking experiments with underactuated 3D bipedal robot MARLO," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, IEEE, 2014, pp. 2529–2536.

[121] H. Zhao, J. Horn, J. Reher, V. Paredes, and A. D. Ames, "Multicontact locomotion on transfemoral prostheses via hybrid system models and optimization-based control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 502–513, 2016.

[122] ——, "First steps toward translating robotic walking to prostheses: A nonlinear optimization based control approach," *Autonomous Robots*, vol. 41, no. 3, pp. 725–742, 2017.

[123] H. Zhao, E. Ambrose, and A. D. Ames, "Preliminary results on energy efficient 3d prosthetic walking with a powered compliant transfemoral prosthesis," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1140–1147.

[124] O. Harib, A. Hereid, A. Agrawal, T. Gurriet, S. Finet, G. Boeris, A. Duburcq, M. E. Mungai, M. Masselin, A. D. Ames, *et al.*, "Feedback control of an exoskeleton for paraplegics: Toward robustly stable, hands-free dynamic walking," *IEEE Control Systems Magazine*, vol. 38, no. 6, pp. 61–87, 2018.

[125] T. Gurriet, S. Finet, G. Boeris, A. Duburcq, A. Hereid, O. Harib, M. Masselin, J. Grizzle, and A. D. Ames, "Towards restoring locomotion for paraplegics: Realizing dynamically stable walking on exoskeletons," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 2804–2811.

[126] I. Poulakakis, "Stabilizing momopedal robot running: Reduction-by-feedback and compliant hybrid zero dynamics.," PhD thesis, The University of Michigan, 2009.

[127] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a Cassie bipedal robot: Walking, standing, and riding a segway," in *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 4559–4566.

[128] J. Reher, W.-L. Ma, and A. D. Ames, "Dynamic walking with compliance on a Cassie bipedal robot," in *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 2589–2595.

[129] A. Hereid and A. D. Ames, "FROST: Fast robot optimization and simulation toolkit," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 719–726.

[130] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[131] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[132] F. Pfeiffer and C. Glocker, *Multibody dynamics with unilateral contacts*. John Wiley & Sons, 1996.

[133] M. Anitescu and F. A. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dynamics*, vol. 14, no. 3, pp. 231–247, 1997.

[134] Y. Hurmuzlu, F. Génot, and B. Brogliato, "Modeling, stability and control of biped robots-A general framework," *Automatica*, vol. 40, no. 10, pp. 1647–1664, 2004.

[135] B. Brogliato and B. Brogliato, *Nonsmooth mechanics*. Springer, 1999.

[136] J. J. Moreau, "Quadratic programming in mechanics: Dynamics of one-sided constraints," *SIAM Journal on control*, vol. 4, no. 1, pp. 153–158, 1966.

[137] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[138] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, "Models, feedback control, and open problems of 3D bipedal robotic walking," *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.

[139] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, Jun. 1981.

[140] N. H. McClamroch and D. Wang, "Feedback stabilization and tracking of constrained robots," in *1987 American Control Conference*, IEEE, 1987, pp. 464–469.

[141] J. J. Murray and G. H. Lovell, "Dynamic modeling of closed-chain robotic manipulators and implications for trajectory control," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 522–528, 1989.

[142] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 3406–3412.

[143] F. Aghili, "A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: Applications to control and simulation," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 834–849, 2005.

[144] T. Apgar, P. Clary, K. Green, A. Fern, and J. W. Hurst, "Fast online trajectory optimization for the bipedal robot Cassie," in *Robotics: Science and Systems*, 2018.

[145] M. Vukobratović, B. Borovac, D. Surla, and D. Stokić, "Dynamics of biped locomotion," in *Biped Locomotion*, Springer, 1990, pp. 1–52.

[146] A. D. Ames, R. Vasudevan, and R. Bajcsy, "Human-data based cost of bipedal robotic walking," in *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, 2011, pp. 153–162.

[147] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.

[148] C. Liu, Z. Zhao, and B. Brogliato, "Frictionless multiple impacts in multibody systems. I. Theoretical framework," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 464, no. 2100, pp. 3193–3211, 2008.

[149] Y. Or and A. D. Ames, "Stability and completion of zeno equilibria in lagrangian hybrid systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1322–1336, 2010.

[150] A. Lamperski and A. D. Ames, "Lyapunov theory for zeno stability," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 100–112, 2012.

[151] A. D. Ames, "Characterizing knee-bounce in bipedal robotic walking: A zeno behavior approach," in *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, 2011, pp. 163–172.

[152] L. Marton and B. Lantos, "Modeling, identification, and compensation of stick-slip friction," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 511–521, 2007.

[153] W.-L. Ma, Y. Or, and A. D. Ames, "Dynamic walking on slippery surfaces: Demonstrating stable bipedal gaits with planned ground slippage," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3705–3711.

[154] A. D. Ames, "First steps toward automatically generating bipedal robotic walking from human data," in *Robot Motion and Control 2011*, Springer, 2012, pp. 89–116.

[155] ——, "Human-inspired control of bipedal walking robots," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, 2014.

[156] H. Zhao, A. Hereid, W.-l. Ma, and A. D. Ames, "Multi-contact bipedal robotic locomotion," *Robotica*, vol. 35, no. 5, pp. 1072–1106, 2017.

[157] D. G. Hobbelen and M. Wisse, "Limit cycle walking," in *Humanoid Robots, Human-like Machines*, IntechOpen, 2007.

[158] L. Perko, *Differential equations and dynamical systems*. Springer Science & Business Media, 2013, vol. 7.

[159] E. D. Wendel and A. D. Ames, "Rank properties of poincaré maps for hybrid systems with applications to bipedal walking," in *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*, 2010, pp. 151–160.

[160] S. Sastry, *Nonlinear systems: Analysis, stability, and control*. Springer Science & Business Media, 2013, vol. 10.

[161] D. F. Rogers, "Mathematical elements for computer graphics," 1990.

[162] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice Hall Upper Saddle River, NJ, 2002, vol. 3.

[163] A. Isidori, *Nonlinear control systems*. Springer-Verlag, 1997.

[164] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 370–387, 2018.

[165] A. D. Ames, K. Galloway, and J. W. Grizzle, "Control Lyapunov functions and hybrid zero dynamics," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, IEEE, 2012, pp. 6837–6842.

[166] A. Hereid, S. Kolathaya, M. S. Jones, J. Van Why, J. W. Hurst, and A. 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*, ACM, 2014, pp. 263–272.

[167] C. Chow and D. Jacobson, "Studies of human locomotion via optimal programming," *Mathematical Biosciences*, vol. 10, no. 3-4, pp. 239–306, 1971.

[168] P. Channon, S. Hopkins, and D. Pham, "Derivation of optimal walking motions for a bipedal walking robot," *Robotica*, vol. 10, no. 2, pp. 165–172, 1992.

[169] K. Mombaur, "Using optimization to create self-stable human-like running," *Robotica*, vol. 27, no. 3, pp. 321–330, 2009.

[170] K. D. Mombaur, H. G. Bock, J. P. Schlöder, and R. W. Longman, "Open-loop stable solutions of periodic optimal control problems in robotics," *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics*, vol. 85, no. 7, pp. 499–515, 2005.

[171] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2014, pp. 295–302.

[172] M. Posa, S. Kuindersma, and R. Tedrake, "Optimization and stabilization of trajectories for constrained dynamical systems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 1366–1373.

[173] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2015, pp. 874–880.

[174] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[175] J. Denk and G. Schmidt, "Synthesis of a walking primitive database for a humanoid robot using optimal control techniques," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2001, pp. 319–326.

[176] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 1447–1454.

[177] A. Hereid, C. M. Hubicki, E. A. Cousineau, J. W. Hurst, and A. D. Ames, "Hybrid zero dynamics based multiple shooting optimization with applications to robotic walking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 5734–5740.

[178] E. Cousineau and A. D. Ames, "Realizing underactuated bipedal walking with torque controllers via the ideal model resolved motion method," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 5747–5753.

[179] J.-H. Shin and S.-M. Kwon, "On the lobe profile design in a cycloid reducer using instant velocity center," *Mechanism and Machine Theory*, vol. 41, no. 5, pp. 596–616, 2006.

[180] J. Blanche and D. Yang, "Cycloid drives with machining tolerances," *Journal of Mechanical Design*, vol. 111, no. 3, pp. 337–344, 1989.

[181] A. Abate, J. W. Hurst, and R. L. Hatton, "Mechanical antagonism in legged robots.," in *Robotics: Science and Systems*, Ann Arbor, MI, vol. 6, 2016.

[182] S. Rezazadeh, A. Abate, R. L. Hatton, and J. W. Hurst, "Robot leg design: A constructive framework," *IEEE Access*, vol. 6, pp. 54 369–54 387, 2018.

[183] J. Reher and A. D. Ames, "Dynamic walking: Toward agile and efficient bipedal robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, 2020.

[184] T. Yang, E. R. Westervelt, J. P. Schmiedeler, and R. A. Bockbrader, "Design and control of a planar bipedal robot ERNIE with parallel knee compliance," *Autonomous robots*, vol. 25, no. 4, p. 317, 2008.

[185] A. Hereid, O. Harib, R. Hartley, Y. Gong, and J. W. Grizzle, "Rapid trajectory optimization using C-FROST with illustration on a Cassie-series dynamic walking biped," *arXiv preprint arXiv:1807.06614*, 2018.

[186] J. Reher and A. D. Ames, "Inverse dynamics control of compliant hybrid zero dynamic walking," *arXiv preprint arXiv:2010.09047*, 2020.

[187] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based full body control for the DARPA robotics challenge," *Journal of Field Robotics*, vol. 32, no. 2, pp. 293–312, 2015.

[188] J. Reher, A. Hereid, E. Cousineau, C. Hubicki, P. Birkmeyer, S. Morfey, Z. Shivers, C. Salisbury, and A. D. Ames, "The humanoid robot DURUS: How hybrid system models, novel electromechanical design and nonlinear control realized new levels of humanoid efficiency," *IEEE Control Systems Magazine*, 2021.

[189] A. Mazumdar, S. J. Spencer, C. Hobart, J. Salton, M. Quigley, T. Wu, S. Bertrand, J. Pratt, and S. P. Buerger, "Parallel elastic elements improve energy efficiency on the steppr bipedal walking robot," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 898–908, 2016.

[190] G. Nelson, A. Saunders, N. Neville, B. Swilling, J. Bondaryk, D. Billings, C. Lee, R. Playter, and M. Raibert, "Petman: A humanoid robot for testing chemical protective clothing," *Journal of the Robotics Society of Japan*, vol. 30, no. 4, pp. 372–377, 2012.

[191] H. P. Crowell III, A. C. Boynton, and M. Mungiole, "Exoskeleton power and torque requirements based on human biomechanics," DTIC Document, Tech. Rep., 2002.

[192] N. Handharu, J. Yoon, and G. Kim, "Gait pattern generation with knee stretch motion for biped robot using toe and heel joints," in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2008, pp. 265–270.

[193] R. Sellaouti, O. Stasse, S. Kajita, K. Yokoi, and A. Kheddar, "Faster and smoother walking of humanoid HRP-2 with passive toe joints," in *2006 IEEE / RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2006, pp. 4909–4914.

[194] M. Ackermann, "Dynamics and energetics of walking with prostheses," PhD thesis, University of Stuttgart, Stuttgart, 2007.

[195] V. T. Inman, "Human locomotion," *Canadian Medical Association Journal*, vol. 94, no. 20, p. 1047, 1966.

[196] B. Griffin and J. Grizzle, "Nonholonomic virtual constraints for dynamic walking," in *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, 2015, pp. 4053–4060.

[197] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[198] R. W. Sinnet, M. J. Powell, R. P. Shah, and A. D. Ames, "A human-inspired hybrid control approach to bipedal robotic walking," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6904–6911, 2011.

[199] P. A. Bhounsule, J. Cortell, A. Grewal, B. Hendriksen, J. D. Karssen, C. Paul, and A. Ruina, "Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1305–1321, 2014.

[200] Y. Gong and J. Grizzle, "Angular momentum about the contact point for control of bipedal locomotion: Validation in a LIP-based controller," *arXiv preprint arXiv:2008.10763*, 2020.

[201] A. Shkolnik, M. Levashov, I. R. Manchester, and R. Tedrake, "Bounding on rough terrain with the LittleDog robot," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 192–215, 2011.

[202] S. Dalibard, A. E. Khoury, F. Lamiraux, A. Nakhaei, M. Taïx, and J.-P. Laumond, "Dynamic walking and whole-body motion planning for humanoid robots: An integrated approach," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1089–1103, 2013.

[203] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[204] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and IMU," *Robotics*, vol. 17, pp. 17–24, 2013.

[205] R. Hartley, M. G. Jadidi, J. W. Grizzle, and R. M. Eustice, "Contact-aided invariant extended Kalman filtering for legged robot state estimation," *arXiv preprint arXiv:1805.10410*, 2018.

[206] P. Varin and S. Kuindersma, "A constrained Kalman filter for rigid body systems with frictional contact," in *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2018.

[207] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini, "Probabilistic contact estimation and impact detection for state estimation of quadruped robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1023–1030, 2017.

[208] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "Qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[209] J. G. Ziegler, N. B. Nichols, *et al.*, "Optimum settings for automatic controllers," *trans. ASME*, vol. 64, no. 11, 1942.

[210] 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," *The International Journal of Robotics Research*, vol. 23, no. 6, pp. 559–582, 2004.

[211] X. Da, O. Harib, R. Hartley, B. Griffin, and J. W. Grizzle, "From 2D design of underactuated bipedal gaits to 3D implementation: Walking with speed tracking," *IEEE Access*, vol. 4, pp. 3469–3478, 2016.

[212] M. J. Powell, A. Hereid, and A. 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 (ICRA)*, IEEE, 2013, pp. 4803–4810.

[213] X. Da, R. Hartley, and J. W. Grizzle, "Supervised learning for stabilizing underactuated bipedal robot locomotion, with outdoor experiments on the wave field," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 3476–3483.

[214] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot Atlas," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1 650 007, 2016.

[215] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. Panne, "Learning locomotion skills for Cassie: Iterative design and sim-to-real," in *Conference on Robot Learning*, 2020, pp. 317–329.

[216] J. Reher, C. Kann, and A. D. Ames, "An inverse dynamics approach to control Lyapunov functions," in *2020 American Control Conference (ACC)*, IEEE, 2020.

[217] J. Reher and A. D. Ames, "Control Lyapunov functions for compliant hybrid zero dynamic walking," *IEEE Transactions on Robotics*, 2021, In Preparation.

[218] L. Sentis, *Synthesis and control of whole-body behaviors in humanoid systems*. Stanford University, USA, 2007.

[219] S. S. Kim and M. J. Vanderploeg, "QR decomposition for state space representation of constrained mechanical dynamic systems," *Journal of Mechanisms Transmissions and Automation in Design*, vol. 108, no. 2, p. 183, 1986, ISSN: 07380666. DOI: 10.1115/1.3260800.

[220] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[221] L. Sentis and O. Khatib, "Control of free-floating humanoid robots through task prioritization," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 1718–1723.

[222] L. Sentis, J. Park, and O. Khatib, "Compliant control of multicontact and center-of-mass behaviors in humanoid robots," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 483–501, 2010.

[223] J. Park and O. Khatib, "Contact consistent control framework for humanoid robots," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, IEEE, 2006, pp. 1963–1969.

[224] L. Sentis, J. Park, and O. Khatib, "Modeling and control of multi-contact centers of pressure and internal forces in humanoid robots," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 453–460.

[225] A. D. Ames and M. Powell, "Towards the unification of locomotion and manipulation through control Lyapunov functions and quadratic programs," in *Control of Cyber-Physical Systems*, Springer, 2013, pp. 219–240.

[226] M. Hutter, H. Sommer, C. Gehring, M. Hoepflinger, M. Bloesch, and R. Siegwart, "Quadrupedal locomotion using hierarchical operational space control," *The International Journal of Robotics Research*, vol. 33, no. 8, pp. 1047–1062, 2014.

[227] Å. Björck, *Numerical methods for least squares problems*. SIAM, 1996.

[228] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[229] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 2589–2594.

[230] 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*, vol. 3, pp. 323–332, 2015.

[231] Q. Nguyen and K. Sreenath, "Optimal robust safety-critical control for dynamic robotics," *arXiv preprint arXiv:2005.07284*, 2020.

[232] A. D. Ames and J. Holley, "Quadratic program based nonlinear embedded control of series elastic actuators," in *53rd IEEE Conference on Decision and Control*, IEEE, 2014, pp. 6291–6298.

[233] A. Hereid, M. J. Powell, and A. D. Ames, "Embedding of SLIP dynamics on underactuated bipedal robots through multi-objective quadratic program based control," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, IEEE, 2014, pp. 2950–2957.

[234] X. Xiong and A. D. Ames, "Coupling reduced order models via feedback control for 3D underactuated bipedal robotic walking," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2018, pp. 1–9.

[235] ——, "Bipedal hopping: Reduced-order model embedding via optimization-based control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3821–3828.

[236] E. D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," *SIAM Journal on Control and Optimization*, vol. 21, no. 3, pp. 462–471, 1983.

[237] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 7, no. 11, pp. 1163–1173, 1983.

[238] R. Freeman and P. V. Kokotovic, *Robust nonlinear control design: State-space and Lyapunov techniques*. Springer Science & Business Media, 2008.

[239] Y. Abe, M. Da Silva, and J. Popović, "Multiobjective control with frictional contacts," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 2007, pp. 249–258.

[240] J. Betts and I. Kolmanovsky, *Practical methods for optimal control using nonlinear programming*. American Society of Mechanical Engineers Digital Collection, 2002.

[241] Q. Nguyen and K. Sreenath, "Optimal robust control for bipedal robots through control Lyapunov function based quadratic programs," in *Robotics: Science and Systems*, Rome, Italy, 2015.

[242] S. Kolathaya, A. Hereid, and A. D. Ames, "Time dependent control Lyapunov functions and hybrid zero dynamics for stable robotic locomotion," in *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 3916–3921.

[243] E. G. Gilbert and I. J. Ha, "An approach to nonlinear feedback control with applications to robotics," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 6, pp. 879–884, 1984.

[244] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, IEEE, vol. 1, 1994, pp. 314–321.

[245] Q. Nguyen and K. Sreenath, "Optimal robust control for constrained nonlinear hybrid systems with application to bipedal locomotion," in *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 4807–4813.

[246] A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames, "Episodic learning with control Lyapunov functions for uncertain robotic systems," *arXiv preprint arXiv:1903.01577*, 2019.

[247] J. Choi, F. Castaneda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions," *arXiv preprint arXiv:2004.07584*, 2020.

[248] B. Morris, M. J. Powell, and A. D. Ames, "Sufficient conditions for the Lipschitz continuity of qp-based multi-objective control of humanoid robots," in *52nd IEEE Conference on Decision and Control*, IEEE, 2013, pp. 2920–2926.

[249] B. J. Morris, M. J. Powell, and A. D. Ames, "Continuity and smoothness properties of nonlinear optimization-based feedback controllers," in *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, 2015, pp. 151–158.

[250] A. Hereid, "Dynamic humanoid locomotion: Hybrid zero dynamics based gait optimization via direct collocation methods," PhD thesis, Georgia Institute of Technology, 2016.

[251] S.-P. Han and O. L. Mangasarian, "Exact penalty functions in nonlinear programming," *Mathematical Programming*, vol. 17, no. 1, pp. 251–269, 1979.

[252] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3d dynamic walking on stepping stones with control barrier functions," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 827–834.

[253] A. D. Ames, P. Tabuada, A. Jones, W.-L. Ma, M. Rungger, B. Schürmann, S. Kolathaya, and J. W. Grizzle, "First steps toward formal controller synthesis for bipedal robots with experimental implementation," *Nonlinear Analysis: Hybrid Systems*, vol. 25, pp. 155–173, 2017.

[254] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[255] A. Hereid, S. Kolathaya, and A. D. Ames, "Online optimal gait generation for bipedal walking robots using legendre pseudospectral optimization," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 6173–6179.

[256] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, IEEE, 2018, pp. 98–106.