Nonholonomic Virtual Constraints and Gait Optimization for Robust Robot Walking Control

by

Brent Griffin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in the University of Michigan
2016

Doctoral Committee:

Professor Jessy W. Grizzle, Chair
Professor Art Kuo
Assistant Professor C. David Remy
Assistant Professor Shai Revzen
Assistant Professor Ram Vasudevan

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Jessy Grizzle, without whom, this work would not have been possible. I am also grateful to the members of my dissertation committee, Prof. Art Kuo, Prof. C. David Remy, Prof. Shai Revzen, and Prof. Ram Vasudevan, for their support and insightful feedback. I extend my thanks to Jonathan Hurst and his incredible team at Oregon State University for all of their help keeping MARLO a working robot. I am grateful to Kaveh Akbari Hamed and Kevin Galloway for helping me with theory-related questions when I was first starting at Michigan. I also want to acknowledge Hamed Razavi; I very much enjoyed our conversations on three-dimensional walking. Gabriel Buche, Xingye Da, Ross Hartley, and Omar Harib are sincerely thanked for all of their help with experiments. I will never forget when Dennis demonstrated his determination and camaraderie by helping me with experiments from dusk until dawn the same morning that he left for China. I am greatly indebted to Brian Buss, my PhD brother. Finally, I would like to thank my family and friends, who make it all worthwhile.

Brent Griffin
Ann Arbor, July 2016

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Nonholonomic Virtual Constraints and Gait Optimization for Robust Robot Walking Control

by

Brent Griffin

Chair: Jessy W. Grizzle

Bipedal locomotion is well suited for mobile robotics because it promises to allow robots to traverse difficult terrain and work effectively in man-made environments. Despite this inherent advantage, however, no existing bipedal robot achieves human-level performance in multiple environments. A key challenge in robotic bipedal locomotion is the design of feedback controllers that function well in the presence of uncertainty, in both the robot and its environment. This dissertation addresses the design of feedback controllers and periodic gaits that function well in the presence of modest terrain variation, without reliance on perception or a priori knowledge of the environment. Model-based design methods are introduced and subsequently validated in simulation and experiment on MARLO, an underactuated three-dimensional bipedal robot that is roughly human size and has six actuators and thirteen degrees of freedom. Innovations include virtual nonholonomic constraints that enable continuous velocity-based posture regulation and an optimization method that accounts for multiple types of disturbances and more heavily penalizes deviations that persist

during critical stages of walking. Using a single continuously-defined controller taken directly from optimization, MARLO traverses sloped sidewalks and parking lots, terrain covered with randomly thrown boards, and grass fields, all while maintaining average walking speeds between 0.9-0.98 m/s and setting a new precedent for walking efficiency in realistic environments.

CHAPTER I

Introduction

1.1 Motivation

Robots have been used successfully in manufacturing for decades [26]; however, most of these applications are restricted to structured environments with fixed operation. Current robotics research aims to extend these applications to include locomotion through unstructured environments where the terrain varies unexpectedly. This extension presents a challenge because a robot that can move perfectly in a controlled environment might fall after its first encounter with a terrain disturbance. In this dissertation, our primary motivation is to achieve bipedal locomotion (i.e., locomotion in which the robot is upright and makes forward progress with support on alternating legs) through unstructured environments without over reliance on perception or a priori knowledge of the terrain. Legged robots, in general, can traverse difficult terrain inaccessible to wheeled vehicles [111]; bipedal robots, more specifically, can also effectively traverse man-made environments intended for human use [109]. To improve the performance of bipedal robots in unstructured environments, we develop control methods for sustaining bipedal locomotion despite unknown terrain variations.

1.2 Objective

The objective of this research is to develop control methods for dynamic bipedal locomotion over a variety of terrains without perception or a priori knowledge of the environment. To this end, we derive a novel optimization procedure for the design of feedback controllers that are robust to a family of unknown disturbances. We introduce a new class of virtual nonholonomic constraints that depend on velocity through generalized conjugate momenta while maintaining the property of being relative degree two. Using nonholonomic virtual constraints, we implement velocity-based posture regulation that accounts for the full dynamics of the biped, as well as a range of terrain variation. To substantiate these control methods and complete our objective, we demonstrate that MARLO, an underactuated bipedal robot that is roughly human size and is equipped with an IMU and joint encoders, can walk through a variety of environments with unknown terrain.

1.3 Summary

The remainder of this dissertation is organized into six chapters.

Chapter II provides a brief review of relevant literature and clarifies how work in the dissertation contributes to that literature. Specifically, the literature review addresses bipedal locomotion, robust walking, simplified walking models, virtual constraints, and three-dimensional walking.

Chapter III describes the general concepts used in the dissertation. We use a hybrid walking model that consists of a continuous single support phase and a discrete impact map for double support. The ATRIAS 2.1 model robot, MARLO, is the basis of our simulation models and is used for experiments. We use virtual constraints and hybrid zero dynamics to control trajectories to a lower dimensional submanifold where actuated coordinates evolve as a function of unactuated coordinates.

Chapter IV introduces the *Optimization for Accommodation of Unknown Disturbances* method. This optimization method minimizes costs that account for perturbations arising from a finite set of disturbances. Trajectory and control deviations are related to a nominal periodic orbit using a gait phasing variable. The gait phasing variable is also used to more heavily penalize deviations that persist late into the gait. This focused penalization method is compared with alternative methods through simulations and experiments using planar MARLO.

Chapter V introduces *nonholonomic virtual constraints*. These constraints are used to design controllers that modify optimized walking gaits to reject velocity perturbations. By adding this perturbation response, control solutions are shown to handle a wider range of terrain variations and exogenous forces. Additionally, including nonholonomic virtual constraints allows foot placement control to be rigorously designed on the basis of the full dynamic model of the biped, instead of on the basis of an inverted pendulum approximation of its center of mass, as is commonly done in the bipedal robotics literature.

Chapter VI focuses on three-dimensional (3D) walking. A substantial portion of our 3D approach coincides with planar control implementations; however, in order to handle additional challenges that arise in 3D walking, significant modifications are made. First, we introduce an additional set of velocity perturbations to the robust control optimization, which is shown to improve performance for repeated disturbances. Second, novel nonholonomic outputs are designed to handle coupled sagittal and frontal plane dynamics. Finally, for the robot implementation, a phase estimator and reduced-order Luenberger observer are designed for estimating the gait phasing variable and angular momentum. Using the resulting continuously-defined controller taken directly from optimization, MARLO is able to traverse a variety of unstructured environments without perception or a priori knowledge of the terrain.

Finally, Chapter VII provides concluding remarks.

1.4 Contributions

This section summarizes the primary contributions of this dissertation. These contributions are organized with respect to prior work on bipedal robot walking with terrain variations, swing foot placement methods, embedding simplified model behavior in higher dimensional control systems, virtual constraints, and feedback control of three-dimensional bipedal robots. These contributions are revisited within the context of related literature in Chapter II.

Contributions to prior work on bipedal robot walking with terrain variations include: allow a family of nonlinear controllers to be searched over with respect to disturbance attenuation; introduce a finite set of perturbations to velocity during control optimization and demonstrating efficacy; synchronize the calculation of trajectory and control deviations of a biped's gait via a gait phasing variable; more heavily penalize trajectory deviations that persist late into a step, when ground contact is likely to occur; and demonstrate in experiment the potential utility of trading off deviations early in the step for improved attenuation of the disturbance toward the end of the step.

Many bipedal robot researchers use swing foot placement, namely some policy for choosing the relative step position for the foot approaching the ground. Researchers in [110] derived an event-based swing foot placement policy based on the center of mass velocity of a linear inverted pendulum. Contributions of the dissertation to the swing foot placement policy implemented in [110] include: control foot placement based on velocity throughout the step rather than just the horizontal velocity of the center of mass at mid-step; include the dynamics of the full model (e.g., impact losses, varying center of mass height); and include a pre-specified range of terrain disturbances in the controller design process.

Contributions to prior work on embedding simplified model behavior in higher dimensional control systems include: embed a velocity-dependent simple model-based

behavior into a full-order control model using nonholonomic virtual constraints, and include a specified range of disturbances in the embedding process to ensure adherence to the desired control behavior over a range of conditions.

Contributions to prior work on virtual constraints include: introduce a new class of virtual constraints that include velocity, but maintain control outputs that are relative degree two for ease of implementation, and demonstrate superior ability to attenuate terrain and velocity perturbations.

Contributions to prior work on feedback control of three-dimensional bipedal robots include: introduce a model-based design framework that is able to achieve dynamic three-dimensional walking without hand-tuning of the optimized walking gait; demonstrate robustness by traversing sloped sidewalks and parking lots, terrain covered with randomly thrown boards, and grass fields without a priori knowledge of the environment or external sensing (the robot uses only an IMU and joint encoders); and set a new precedent by evaluating walking efficiency for a variety of realistic terrains.

CHAPTER II

Literature Review

2.1 Bipedal Locomotion

Bipedal locomotion is a type of locomotion in which an upright two-legged system makes forward progress with support on alternating legs. When applied to robots, bipedal locomotion can be characterized by the degree to which the walking gait is controlled or actuated, whether the walking gait is static or dynamic, and how the robot senses its environment while walking. Each decision offers a unique set of advantages and disadvantages.

Bipedal locomotion can be realized with varying degrees of actuation. In the 1980s, the MIT Leg Lab used actuated robots capable of running, jumping up stairs, and even doing flips [58, 106, 112]. However, they were actuated by tethered hydraulics that had a high energy cost and only performed these feats when the terrain profile was known in advance. Shortly thereafter, Tad McGeer demonstrated the alternative use of passive walking [89]. In contrast to the actuated walkers of the MIT lab, passive walkers are mechanically simpler, more energy efficient, and capable of walking down slopes without any actuation. Passive walkers, however, cannot climb stairs or handle rough terrain. As seen in [20] and citations therein, recent research has blurred the distinction between these two frameworks by demonstrating the use of partially actuated walking.

Bipedal locomotion can be static or dynamic. Static walkers can maintain equilibrium throughout a walking gait by controlling the Zero-Moment Point (ZMP) [132] of a robot to stay within the polygon of ground contact [110, 115]. More specifically, the ZMP is the point on the ground where the total horizontal inertia and gravity forces with respect to the robot equals zero. If this point is kept within the polygon of ground contact points, the robot will not "tip over the edge" and fall over[1]. While this approach works well for balancing in place, fully actuated control results in unnatural and inefficient walking gaits. Dynamic limit cycle walkers, on the other hand, do not require local controllability during their entire gait [55], and thus have more natural, more energy efficient, and potentially faster walking gaits than do static walkers [20, 89].

Bipedal locomotion control methods require varying degrees of terrain sensing. Sensing requirements have a significant impact on how generalizable a control method is to other robots. For example, Atlas, a Boston Dynamics robot, depends on stereo cameras and a laser rangefinder to traverse rough outdoor terrain [2]. In contrast, University of Michigan robot MABEL, predecessor to MARLO (see Figure 2.1), performs planar walking with terrain drops of up to 20 cm using only shin and foot contact switches [98]. Because the position of a biped's links and ground contact can indirectly provide an estimate of the terrain, blind walking, like that used by MARLO, becomes possible [19, 107]. One advantage of blind walking control methods over terrain sensing dependent methods is that blind methods can be applied to almost all bipeds. When terrain information is available, the robustness of control designed for unexpected terrain variations can instead alleviate complications arising from sensing errors.

Of particular interest to the current research is bipedal locomotion that is partially actuated, dynamic, and does not require terrain sensing. This locomotion is achieved

[1] An alternative approach is to keep the center of mass within the polygon of ground contact, which maintains static equilibrium.

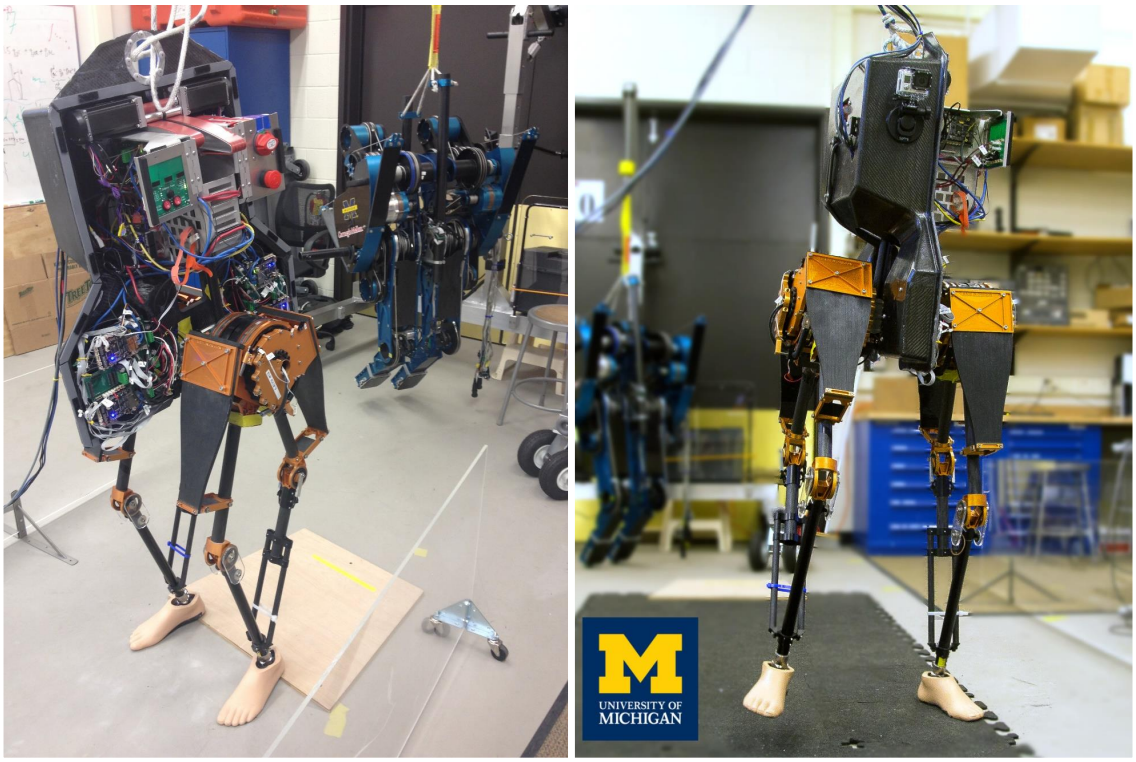

Figure 2.1: MARLO and MABEL. MABEL (back) is the predecessor of MARLO (front). MABEL requires a boom for lateral stability when planar walking, while MARLO is able to walk in three dimensions due to laterally actuated hips (right, photo: Joseph Xu). Both bipedal robots are underactuated and require no a priori knowledge of the environment to walk.

using *virtual constraints* and *hybrid zero dynamics* [46, 134, 135, 136]. Unlike *physical* constraints defined by mechanical properties of a system, *virtual* constraints are enforced through actions of actuators as controlled outputs. Virtual constraints have the advantage of being re-programmable without any physical modification. *Hybrid zero dynamics* signify the lower dimensional sub manifold of bipedal dynamics that occurs when control outputs are zeroed and evolve as a function of the remaining unactuated states. When virtual constraints and hybrid zero dynamics are applied to bipedal locomotion, they permit dynamic walking energy efficiency, are more robust to velocity perturbations compared to many time-based techniques, and can be modifiable for varied impact conditions [94]. See Chapter III for the derivation of this bipedal locomotion control approach.

2.1.1 Optimization for Bipedal Locomotion

Optimization is commonly used to identify a control policy for bipedal locomotion. An effective way to make policy search computationally feasible is to parameterize control variables [72, 78, 122]. In this dissertation, control trajectories are parameterized using Bézier polynomials [136, pp. 138]. For optimization of control variables, many robotics researchers use policy gradient methods [72, 75, 103, 116, 129]. Because the gradient can be difficult to solve analytically, researchers often use approximations of gradients for policy search, using finite policy differences [72, 129] or even just a signed derivative [74]. Other interesting policy search methods include genetic algorithms that avoid local optimality [141, 143], or evolving policy parameterizations that initiate with fewer parameters and then increase in complexity [78]. More recently, multiple shooting optimization methods have been used for fast and reliable designing of walking gaits [52]. In this dissertation, optimizations are performed using a direct shooting, policy gradient approach by means of `fmincon` in MATLAB [88].

2.2 Robust Walking

For bipedal locomotion to be useful in unstructured environments, bipeds must be able to traverse uneven terrain with imperfect knowledge of the ground profile. In such conditions, failures often result from differences between ideal models and actual conditions. Sources of error in bipedal robot locomotion include unknown terrain disturbances, parametric errors, friction in joints and motors, and sensor errors, which compound to cause unexpected ground impact times and conditions. If the resulting perturbations are severe enough, a biped's nominal gait will not be able to reject the disturbance and the biped will fall.

We consider many methodologies to quantify and improve the capacity of a bipedal robot to walk over uneven terrain. The terrain variations can be deterministic or

random, and the control policy may or may not involve switching. Various ways of quantifying stability and robustness are outlined in Section 2.2.1; robust control methods are discussed in Section 2.2.2.

2.2.1 Stability and Robustness Measures

Stability and robustness measures are useful for determining if a walking gait is feasible for bipedal locomotion. An unstable gait can cause a robot to fall, even if no disturbances are present, and the ability to reject disturbances determines a gait's robustness. Stability of periodic gaits can be determined using Poincaré maps. The domain of states from which a robot converges on a stable gait is called the region of attraction. Extended definitions of stability for aperiodic walking gaits are not used in this dissertation [140]. Robustness measures include the Gait Sensitivity Norm, the N-Step Capture Stability Margin, and the Mean Time To Failure.

2.2.1.1 Stability Measures

Poincaré maps are commonly used to measure the stability of bipedal locomotion. Poincaré maps apply to periodic walking gaits and provide a discrete map to future states on the Poincaré section. An example Poincaré section for a biped is the hypersurface for all states where the progressing swing foot initiates ground contact. A state that maps back to itself through the Poincaré map is called a fixed point and corresponds to a periodic walking gait. If points near the fixed point converge to the fixed point via the Poincaré map, the fixed point is considered attractive and corresponds to a stable periodic gait.

Poincaré maps of bipedal systems often cannot be found using a closed-form representation [25], but can be approximated by linearizing about a fixed point. The resulting state Jacobian matrix can act as a discrete map for approximating future states from points near the fixed point [136]. The eigenvalues of this Jacobian matrix

(also called Floquet multipliers) can then be used to determine if the fixed point is locally exponentially stable or not [70]. If eigenvalues are outside the unit circle, the fixed point is determined to be unstable. If eigenvalues are inside the unit circle, the fixed point is locally exponentially stable. The closer eigenvalues are to zero, the faster the convergence to the fixed point [136].

In addition to determining stability of bipedal locomotion, Poincaré maps are useful for control design. In [18], two control design approaches use Poincaré maps in two distinct ways. For the first approach, Poincaré maps are used as the basis of an event-based discrete linear quadratic regulator (DLQR) controller. Essentially, an additional Jacobian matrix that maps control inputs is used for the step to step DLQR controller for quicker convergence on a specific gait. For the second approach, Poincaré maps are used during optimization to keep Floquet multipliers within the unit circle and guarantee local exponential stability. Other work in [6] translates a Poincaré map-based sensitivity analysis into a set of bilinear matrix inequalities (BMIs). A BMI optimization is then used to find control parameters that achieve exponential stability of a periodic gait.

Poincaré maps can also demonstrate dimension reduction of hybrid bipedal systems. Work in [13] shows that when Poincaré maps of a hybrid system have constant-rank iterates (a fairly common condition) there exists an invariant subsystem that attracts all trajectories near the fixed point. This result shows that even complicated bipedal systems can be governed by simpler dynamics, as is done in this dissertation using hybrid zero dynamics.

While Poincaré maps do have many uses, they also have limitations. First, the eigenvalues do not necessarily correspond to specific failure modes. Second, if a Poincaré map is linearly approximated, it is less valid for large perturbations from the fixed point. Third, because Poincaré maps are constructed around a periodic gait, they do not directly apply to locomotion on varying terrain under non-periodic

impact conditions [84]. Finally, Poincaré maps are evaluated once per step, missing opportunities for continuous measurement throughout the walking gait. Despite these limitations, Poincaré maps are useful for guaranteeing stability of the holonomic control solution in Chapter VI. As done in [18], we impose stability conditions using eigenvalues of a linearized Poincaré map during optimization.

Another metric for measuring stability is the *Region of Attraction* (ROA). The ROA is the domain of states from which a system asymptotically converges to an equilibrium point. For bipedal walking, the ROA determines the range of initial states that converge on a periodic walking gait. Unfortunately, literature for computing the ROA of a periodic orbit (applicable to bipedal locomotion) is more limited than that of asymptotically stable equilibrium points [47].

Developing feasible methods for calculating the ROA is an open problem in research. For systems with two or three states, the ROA can be approximated by starting from initial points and recording points that converge on the stable equilibrium. Quadratic Lyapunov functions are commonly used to estimate the ROA. However, because estimates change with choice of Lyapunov function, finding a function that accurately represents the ROA can be challenging [70].

Lyapunov conditions of stability can be verified using polynomial functions that are *sum of squares* (SOS). SOS optimization methods have been used to iteratively improve an inner estimate of the ROA [97]. For computational reasons, applications of these ROA methods have been restricted to systems with a limited number of states, inputs, and polynomial vector complexity. Progress has recently been made in [83] for estimating the ROA by including the dual problem of finding points that *cannot* reach the target set. These points form an outer approximation of the *backwards reachable set* (BRS), which simplifies ROA estimation by removing the nonconvexity required of bilinear decision variables corresponding to Lyapunov functions and control inputs. Applying the algorithm from [83] to higher dimensional systems is still restricted by

the *semidefinite programs* (SDPs) used to determine whether polynomials are SOS. However, researchers in [83] are currently extending their work to include higher dimensional hybrid systems applicable to bipedal locomotion.

2.2.1.2 Robustness Measures

We consider several techniques for measuring robustness, including the Gait Sensitivity Norm, the N-Step Capture Stability Margin, and the Mean Time To Failure.

The Gait Sensitivity Norm is a robustness measure that attempts to make up for some of the weaknesses of using the eigenvalues of the linearized Poincaré map alone [55]. The Gait Sensitivity Norm uses *disturbance parameters* to measure particular disturbances (e.g., terrain slope variation) and *gait indicators* to measure perturbations (e.g., step time). Mathematically, the Gait Sensitivity Norm is the H_2 norm of the change in gait indicators per change in disturbance parameters. More specifically, the Gait Sensitivity Norm is the result of adding all of the squared changes in gait indicators for a perturbed step, rooting their sum, and then normalizing this quantity to account for the magnitude of the disturbance.

The Gait Sensitivity Norm requires prudent selection of disturbance parameters and gait indicators to be a useful robustness measure. An effective gait indicator must measure a behavior that, if perturbed with a large enough magnitude, will result in a fall. The effect of the disturbance parameters on gait indicators must be directly measurable and reveal how robust the walking gait is to disturbances. In [55, 57, 138], the Gait Sensitivity Norm measures deviations in state trajectories arising from unknown *step decreases* in ground height.

Another measure of robustness, the N-Step Capture Stability Margin [110], calculates the number of steps it takes for a biped to reach a Capture Point. A Capture Point is defined as the point at which a biped's center of mass can come to rest over its foot (i.e., come to a complete stop). If a biped is already at a Capture Point, the

system is considered very stable. Even if a biped requires one step to get to the Capture Point, the system is still fairly robust. This one step requirement is reasonable because people, who are expert bipedal walkers, often have to take another step to prevent a fall while in the middle of a stable walking gait. Because the goal of the current research is to maintain steady dynamic walking (i.e., not come to rest at a Capture Point), the N-Step Capture Stability Margin will not be used as a robustness measure.

Robustness Measures in this Dissertation

In this dissertation, we assess robustness in a number of ways. First, during control optimization, simulations of perturbed steps determine how well controllers reject terrain disturbances by their relative proximity to their nominal periodic gait. Next, to evaluate robustness during simulated walking experiments, an optimized control solution walks over various kinds of random terrain until falling. This is similar to the work of Byl and Tedrake [17], who use the Mean Time To Failure to assess walking performance in the presence of *stochastic ground height variations*. In Chapter IV, our optimizations, simulated walking experiments, and additional robot walking experiments are discussed in detail. In Chapter V, another method for evaluating robustness is introduced. Specifically, controllers are subjected to gradually increasing magnitudes of disturbances until failure. This escalating disturbance method is ideal for determining the performance limits of a control solution for repeated disturbances, such as different types of terrain profiles and exogenous forces.

2.2.2 Robustness Methods

Using robustness measures, it is possible to assess various robustness methods. We consider many methods for robust bipedal locomotion, including swing leg retraction, control that leverages advance knowledge of terrain, adaptive state machines for

switching controllers, and optimizations that by design generate robust walking gaits.

Swing leg retraction is when the swing leg has a negative velocity prior to impact. Swing leg retraction occurs naturally in bipedal and four-legged animals [124], and has been observed to be helpful for robust bipedal walking and running [15, 50, 57]. One of the primary stabilizing properties of swing leg retraction is that when a gait moves too quickly, the swing leg impacts early and results in a larger step. Large steps take more energy to complete and slow the walking gait until it converges back to the nominal walking gait. If a walking gait is moving too slowly, the swing leg has more time to retract, resulting in a smaller step. Small steps require less energy to complete and quicken the walking gait until the nominal periodic gait is reached.

It should also be noted that in time-based control systems, too much swing leg retraction results in an under-damped oscillatory behavior called the "totter" mode [57], which is an oscillatory attempt to match step length with speed that results in multiple alternating large, slow and small, fast steps. Absence of swing leg retraction can lead to an unstable speed mode in which faster steps lead to shorter steps that, in turn, become even faster. Ultimately, selecting the correct amount of swing leg retraction is important for maintaining a nominal gait that is robust to changes in velocity and self regulates in a critically damped fashion.

Many researchers leverage advance knowledge of terrain to achieve robust walking. Step length adjustments control forward velocity and achieve planned foot positions over uneven terrain in [58]. Likewise, in [85], control decisions are made with advance knowledge of terrain to generate desirable conditions when traversing each individual obstacle. A more restricted approach is developed in [119] where control selection is based on terrain that is only one step ahead. While [58, 85, 119] are all valid robustness methods, the current research aims to develop control methods that do not require a priori knowledge of the environment to broaden applicability.

Another approach for dealing with terrain variation is an adaptive control archi-

tecture that *switches among a finite-set of controllers* [85, 98, 119, 140]. Individual controllers are designed to handle specific conditions and then collectively form a set that handles many walking conditions. Using this approach, Park successfully demonstrated robust walking with MABEL for up to 20 cm terrain disturbances without a priori knowledge of the terrain [98]. The current work develops a single (non-switching) controller and nominal periodic gait that are insensitive to a predetermined and finite set of terrain variations and velocity perturbations. This choice of a non-switching controller is motivated, in part, by ease of implementation. However, even in the context of a switching controller, it would be desirable that one of the controllers be insensitive to a pre-determined range of disturbances.

Robust bipedal locomotion can also be achieved using walking gaits from optimizations specifically designed for robustness. The simultaneous design of a periodic walking gait and a linear time-varying controller that minimizes deviations induced by ground height changes is addressed by Dai and Tedrake in [24, 25]. The results are illustrated through simulation on the compass gait biped and on Rabbit, a five-link biped with knees. Their work is motivated by Differential Dynamic Programming, employed by Morimoto to iteratively apply optimization to produce a robust controller [93], and partially motivated by the work of Ernst, who showed that an infinite number of control strategies exist to maintain constant running on unknown terrain [29]. In [24, 25], Dai and Tedrake present a quantitative measure to observe robustness throughout a walking gait, and then use this measure to pose a cost function for optimizing a walking gait. They define an L_2 gain from terrain perturbations to deviations from the nominal limit cycle and then use iterative optimization to minimize this gain.

The effect of Dai and Tedrake's optimization on gait design can be understood as follows. When a biped steps onto an uneven surface, the resulting state and torque trajectory of the perturbed step is compared to its nominal gait and errors are

continuously integrated. Gaits in which terrain disturbances result in large deviations from the nominal gait will have a large associated error cost. By optimizing with this error as a cost, perturbed trajectories are squeezed toward the nominal gait during perturbed steps, thus keeping perturbed steps closer to the nominal walking gait. Because the disturbances do not cause as large of a deviation from the nominal periodic gait, the optimized gait is more robust to disturbances.

A few additional robustness methods have realized bipedal robot walking over terrain variations. A time-invariant linear controller using transverse linearization and a receding-horizon control framework is developed in [84]; experiments are performed on a compass gait walker with 2 cm step down heights. Another robustness method in [73], uses an event-based controller that updates parameters in a fixed controller to achieve a *dead-beat control response*. Control is dead-beat in the sense that, following a terrain disturbance, it steers the robot's state back to its value at the end of the nominal periodic gait. Experiments are performed using AMBER on a sloped treadmill oscillating between ± 1.5 degrees.

Robustness Methods in this Dissertation

Motivated by the approach of [24, 25], we seek a periodic walking gait that can accommodate a finite set of perturbations in ground height. Additionally, we introduce a finite set of perturbations to velocity, which is shown to improve performance for repeated disturbances. Trajectory and control deviations induced by the perturbations are defined with respect to a nominal periodic orbit via a gait phasing variable. As in [136], a parameterized family of *nonlinear controllers* is assumed to be known, and constrained parameter optimization is used to select a periodic solution of the closed-loop system that satisfies limits on torque, ground reaction forces, and other physical quantities. We also ensure swing leg retraction via optimization constraints. Similar to [24, 25], the cost function is augmented with terms that penalize devia-

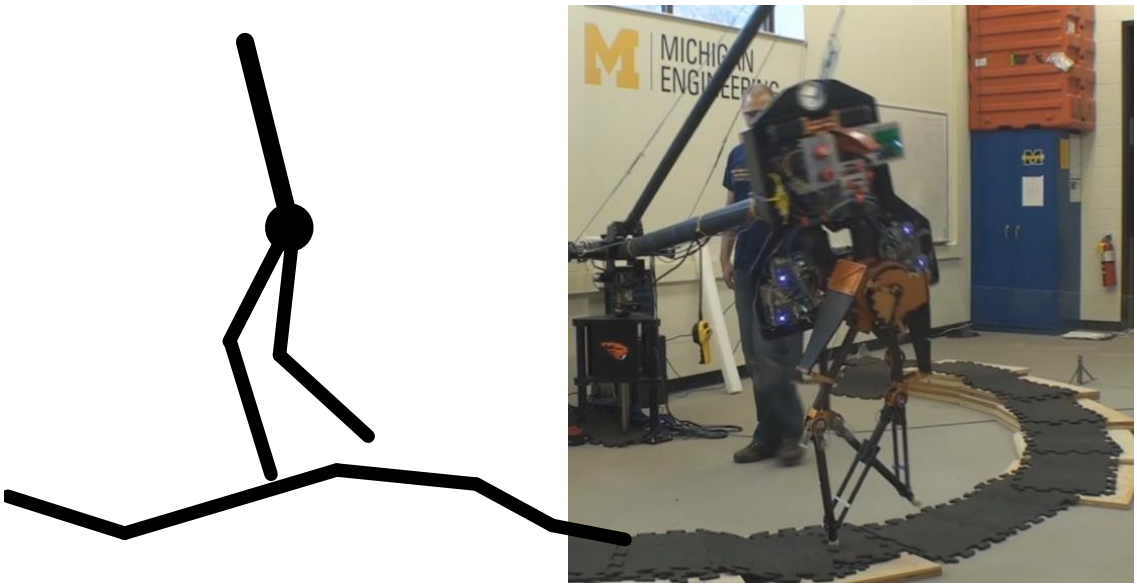

Figure 2.2: Five-link model and planar MARLO walking over uneven terrain.

tions in the state and control trajectories arising from the terrain perturbations. Two choices of cost function are studied. In the first cost function, the gait phasing variable is used to penalize more heavily those deviations that persist late into the gait. Conversely, in the second cost function, no distinction is made for where deviations occur. Focusing on deviations late in the gait is shown to improve the ability of the robot to handle terrain deviations. The method is illustrated both in simulation and in experiments using planar MARLO, as seen in Figure 2.2, and for three-dimensional walking in Chapter VI.

Contributions of this dissertation with respect to prior work on bipedal robot walking with terrain variations include: allow a family of nonlinear controllers to be searched over with respect to disturbance attenuation; introduce a finite set of perturbations to velocity during control optimization and demonstrating efficacy; synchronize the calculation of trajectory and control deviations of a biped's gait via a gait phasing variable; more heavily penalize trajectory deviations that persist late into a step, when ground contact is likely to occur; and demonstrate in experiment the potential utility of trading off deviations early in the step for improved attenuation

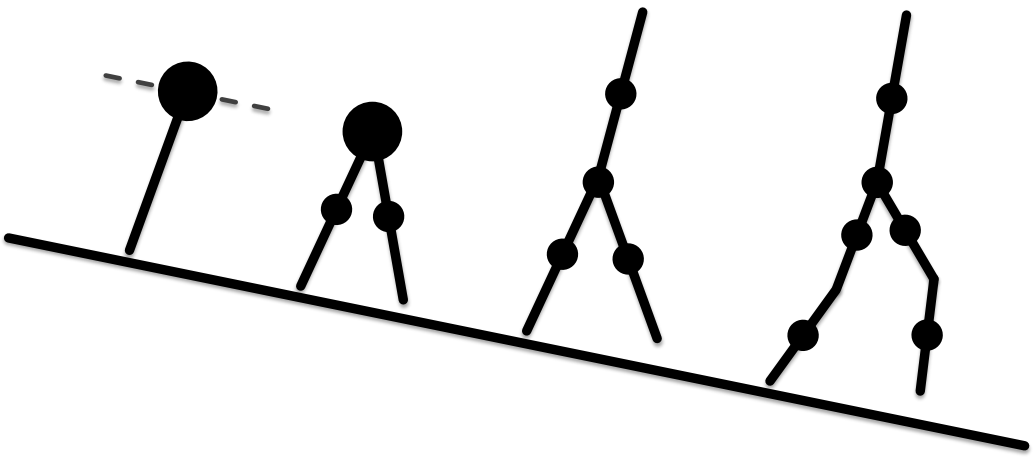

Figure 2.3: Different models of walking. From left to right: Linear Inverted Pendulum, Compass Gait Walker, Three-Link Walker, and Five-Link Walker.

of the disturbance toward the end of the step.

2.3 Simplified Walking Models

Given the complexity of many bipedal robots, simplified walking models are often a useful starting point for understanding bipedal locomotion. Simple models provide insight into the principal behaviors of walking and can motivate intuitive, and often mathematically precise, control laws. Prevailing models in biped research include the Linear Inverted Pendulum Model (LIPM) [68], the Spring Loaded Inverted Pendulum (SLIP) [11], the Compass Gait Walker [34], the Five-Link Walker [134], and variations of these [32] (Figure 2.3). These simplified models are generally computationally quicker to simulate than higher dimensional models yet reveal many of the effects of physical and control parameter variation. Understanding the physical parameters for underactuated bipeds is especially important because the dynamics are not entirely controlled.

New simplified models of bipedal systems can be created using the formal framework introduced in [31], which models high-dimensional biological systems using templates and anchors. Templates are the simplest models that exhibit a targeted behavior (e.g., an inverted pendulum), and anchors are more elaborate representations grounded in the morphology and physiology of an animal being studied (e.g., multiple legs, joints, and muscles). The general idea is that animals throw their motion into 'the hand of the mechanical template' while 'tuning up' redundant degrees of freedom around the behavior that supports the template motion.

Simplified models have been used to develop various methods for handling impact effects. In [80], idealized and anthropomorphic models of a compass gait walker demonstrate how collision losses at impact affect walking efficiency. Efficiency is improved by using impulsive energy input at toe-off instead of hip actuation, which can be four times less costly. Other work in [21] uses a one-legged hopper to demonstrate how switching conditions at impact can determine the majority of dynamics. By controlling the hopper in a feedforward manner to accommodate the impact that would occur at that particular instant, the hopper rejects terrain disturbances in a dead-beat manner. It is also shown in [21] that accurate sensing of uneven terrain is not a requirement for guaranteed stability. In [29], a SLIP model shows how spring-legged systems can maintain constant running speed despite uneven terrain and varied impact conditions.

Methods for controlling forward walking speed have also been developed using simplified models. To control forward walking speed in [110], researchers plan the desired placement of a biped's swing foot as a function of the center of mass velocity in the horizontal direction. The control law for foot placement is based on the LIPM proposed in [69], which approximates the robot's dynamics as an inverted pendulum with constant vertical height and massless legs. The pendulum's dynamic model is linear, the reset map associated with leg impact is linear and energy conserving,

and the overall hybrid model can be solved in closed form. Using the closed-form solution of this LIPM model, Pratt et al. [77, 110] propose a foot placement policy to regulate forward walking speed. They have used this policy on complex robots such as a simulation model of the M2V2 biped undergoing "shoves" of up to 15 Ns [108]. Similar adjustments are made heuristically to step length and torso pitch in [104] to improve velocity stabilization of the planar biped ERNIE.

Simplified models are also often used for control optimization and simulation prior to testing on a real system. In the early design stages of a controller, such a process can improve control while avoiding unnecessary wear or damage to hardware [73, 84, 92].

2.3.1 Embedding Simplified Model Behavior in Higher Dimensional Control Systems

Among the many challenges associated with implementing a simplified model-based controller on an actual robot, one of the most significant challenges arises from incompatible model dynamics. Even when a more complete model is used, parametric differences and sensing errors cause a gap between a simulation model and an actual robot. Researchers in [99] address this gap using robust control and parameter identification. Other researchers in [30] use supervised learning algorithms to iteratively update simulation models to match robot experiment data.

Simplified models can also be applied by making revisions that are realistic without compromising the primary desired behaviors. In [105], a formal connection is established between SLIP models and higher dimensional models using the revised Asymmetric Spring Loaded Inverted Pendulum (ASLIP). Researchers in [105] propose a framework that combines SLIP controllers with nonlinear control tools to induce exponentially stable running motions in ASLIP. In addition, by imposing a virtual constraint on torso posture, the dynamics of the closed-loop higher dimensional system is diffeomorphic to the center of mass dynamics of a SLIP model. In [62], a

SLIP model with impact compensation (SLIPec) is introduced to bring theoretical foundations closer to applications in real robots. They also increase the applicability of simplified models to their robot by explicitly designing limbs to reflect the SLIP model template.

Multiple researchers have embedded simplified model behavior into planar bipedal control. In [33], the center of mass dynamics of a SLIP model are embedded in a feedback control law for a fully actuated planar biped. For the additional degrees of freedom of the full model, torso orientation and virtual constraints for the swing foot trajectory are also controlled. In [53], researchers embed the center of mass trajectory of a SLIP model into a controller for a planar ATRIAS robot. First, they find a symmetric center of mass trajectory based on a SLIP model that is a rough approximation of ATRIAS. Next, they add the least squares fit between the center of mass trajectory of the SLIP model and the full ATRIAS model as a cost with additional constraints during full model optimization. Whereas speeds and step lengths differ, the center of mass trajectories for the SLIP model and resulting full model ATRIAS gait are similar in simulation.

Embedding Simplified Model Behavior in Higher Dimensional Control Systems in this Dissertation

In this dissertation, we develop a new framework for embedding control laws based on simple models into higher dimensional systems. Similar to [53], we embed simple model behavior using optimization costs on the full dynamic model. However, in our framework the embedded behavior is not limited to a single instance of a periodic orbit (e.g., a single center of mass trajectory). Using a parameter optimization process developed in Chapter IV, a pre-specified range of disturbances are included in the embedding process to maintain adherence to the desired control behavior over a range of conditions. We also utilize a new class of nonholonomic virtual constraints derived

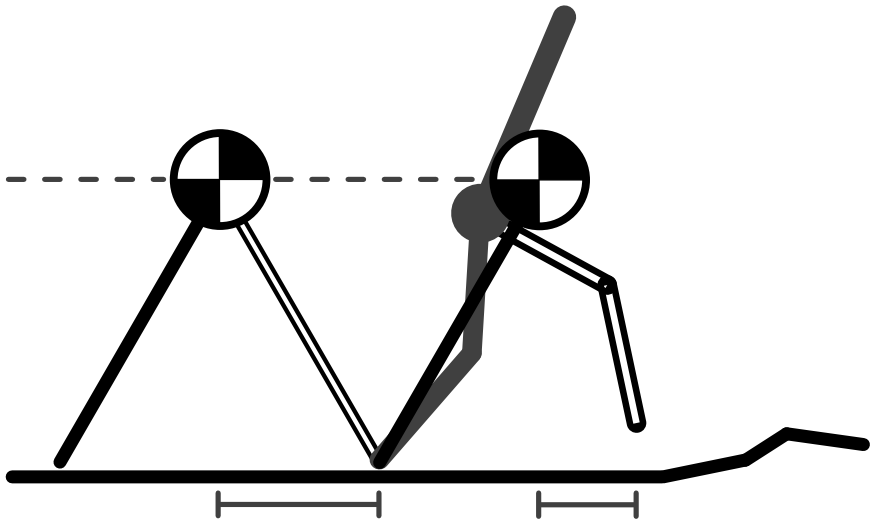

Figure 2.4: Swing foot placement on the full model. Velocity-based swing foot placement has been designed on the basis of the linear inverted pendulum model in [110]. Using velocity-dependent nonholonomic virtual constraints, it is possible to implement a swing foot placement policy that accounts for the full dynamics of the biped, as well as a range of terrain disturbances.

in Chapter V, which enable us to embed velocity-dependent behavior. Although this framework does not provide a formal diffeomorphism between models as in [105], it does not limit the complexity of the final control system.

In Chapter V, we demonstrate this framework by embedding a LIPM-based swing foot placement policy into a full dynamic model of MARLO. Using nonholonomic virtual constraints that depend on velocity through angular momentum about the stance leg end, a velocity-dependent foot-placement strategy is designed and implemented without relying on an inverted pendulum approximation of the robot. In particular, the distributed mass, multi-link nature of the robot can be fully taken into account, including energy losses upon impact. In addition, the control design process takes terrain variations into account (Figure 2.4).

Contributions of this dissertation with respect to the swing foot placement policy implemented in [110] include: control foot placement based on velocity throughout the step rather than just the horizontal velocity of the center of mass at mid-step;

include the dynamics of the full model (e.g., impact losses, varying center of mass height); and include a pre-specified range of terrain disturbances in the controller design process.

Contributions of this dissertation with respect to prior work on embedding simplified model behavior in higher dimensional control systems include: embed a velocity-dependent simple model-based behavior into a full-order control model using nonholonomic virtual constraints, and include a specified range of disturbances in the embedding process to ensure adherence to the desired control behavior over a range of conditions.

2.4 Virtual Constraints

Virtual *holonomic* constraints are functional relations among the configuration variables of a robot that are dynamically imposed through feedback control. Their purpose is to synchronize the evolution of the various links to an internal *gait phasing* or *gait timing variable*, such as the position of the robot's hip with respect to the stance leg end. The gait timing variable is selected to be monotonically increasing along a walking motion so that it can replace time as a means to parameterize command "trajectories." From a theoretical perspective, virtual constraints turn the Isidori-Byrnes theory of nonlinear zero dynamics from [64] into a formal gait and feedback design tool, while the experiments reported in [15, 35, 86, 101, 135, 142] attest to the applicability of the approach to realize dynamic locomotion that meets a range of design objectives, from speed of locomotion, to limits on actuator torque, and available friction cone, to name only a few.

Virtual Constraints in this Dissertation

In Chapter V we introduce a more general class of *nonholonomic* virtual constraints that depend on velocity through generalized conjugate momenta while main-

taining the property of being relative degree two. Nonholonomic virtual constraints allow foot placement control to be rigorously designed on the basis of the full dynamic model of the biped, instead of on the basis of an inverted pendulum approximation of its center of mass, as is commonly done in the bipedal robotics literature. The foot placement control implementation in Chapter V uses virtual constraints that depend on velocity through angular momentum about the stance leg end (in addition to the robot's configuration variables). A set of parameterized splines appearing in the virtual constraints are designed using the *Optimization for Accommodation of Unknown Disturbances* method introduced in Chapter IV. The robustness of the resulting control law to terrain and velocity perturbations is evaluated through simulation and compared to other control laws. The control law based on *nonholonomic constraints* is able to accommodate a wider range of perturbations than a control law based on *holonomic constraints*.

We use nonholonomic virtual constraints again in Chapter VI, where a second velocity-dependent posture-regulating strategy is designed and implemented on MARLO for three-dimensional walking. In this second control implementation, we find again that all control solutions based on *nonholonomic constraints* accommodate a wider range of perturbations than those based on *holonomic constraints*.

Contributions of this dissertation with respect to prior work on virtual constraints include: introduce a new class of virtual constraints that include velocity, but maintain control outputs that are relative degree two for ease of implementation, and demonstrate superior ability to attenuate terrain and velocity perturbations.

2.5 Three-Dimensional Walking

As motivated in Section 2.1, this dissertation focuses on underactuated dynamic walking control. However, three-dimensional (3D) walking control is commonly realized using *zero-moment point* (ZMP) methods for static stability [47]. Examples of

ZMP 3D walking robots include Atlas [2], KHR-3 HUBO [102], ASIMO [121], and previous Honda robots [54]. Following their inception, ZMP methods have been augmented by underactuated control methods [65]. Likewise, the control methods presented in this dissertation that are motivated by challenges of underactuated walking can be extended to fully actuated bipedal robots. Some conditions that motivate this extension include cases where fully actuated robots seek to relax restrictions on ground contact conditions, conserve a limited energy supply through dynamic walking, and continue operation despite motor failure using underactuated methods.

2.5.1 Simplified 3D Models

One of the most intuitive ways to understand the auxiliary roll and yaw behaviors of 3D walking is to study simplified 3D models. Roll and yaw instability of simple unactuated 3D models are studied in [90] and then compared with systems that are inherently stable. McGeer finds that foot design has a significant effect on trading off lateral and yaw stability. He also discusses how people eliminate rolling torques in the lateral plane by limiting lateral leg separation. These results are extended in [79], where a simple passive walker is augmented with various minimal actuator techniques to provide roll and yaw stability. Specifically, they consider three direct and two indirect stabilization methods. The three direct stabilization methods involve adding ankle torques, reaction wheels, and roll torso motions; the two indirect stabilization methods involve adjusting step width and adding a torsional spring at the hip. It is found that direct stabilization methods are most effective when applied immediately and require energy input at least equal in magnitude to the perturbation. Conversely, indirect stabilization methods can be performed over an entire step without penalty and are more energy efficient, as they essentially adjust foot contact conditions to reject a perturbation by the end of the following step. The benefits of indirect stabilization methods found in [79] motivate extensions of the swing foot

placement strategies implemented in Chapter V to 3D for roll and yaw stabilization as opposed to any direct stabilization methods (e.g., ankle torque).

Another simplified 3D model, the Three-Dimension Linear Inverted Pendulum Mode (3D-LIPM), is introduced in [66]. In this work, researchers derive analytical solutions for foot placement that achieve a desired walking speed and direction given current conditions. This model is used in [114] to formalize the notion of self-synchronization, which refers to the periods of pendular motion converging on a common period for the sagittal and frontal planes for a given level of kinetic energy. Using the 3D-LIPM in [114], researchers are able to prove the existence of a self-synchronized walking gait, and then extend a generalization of their work to a 9 degree of freedom (DOF) 3D model of ATRIAS that is asymptotically stable.

2.5.2 Yaw Restriction Models

3D walking models can have varying levels of yaw restriction in the traverse plane. For example, walking models in [18, 79] are completely yaw restricted. Bipeds are free to pivot in the roll and pitch directions, but are unable to rotate in the yaw direction. In contrast, the walking model in [125] is unrestricted and allows the biped to rotate freely about the stance foot. To deal with this additional freedom, a yaw controller is implemented to follow desired headings. Finally, the walking models in [8, 113] use moderate yaw-restriction. Yaw motion about the stance foot is limited by viscous friction, which increases yaw damping forces with rotation velocity. As an aside, in addition to the single support models, there are also different modeling assumptions for yaw restriction at impact or during double support. For a physical robot, different levels of yaw restriction can be implemented by varying the ground contact conditions through foot design.

2.5.3 Foot Design

A significant factor for 3D walking is foot design, which heavily influences roll and yaw stability [90]. Unfortunately, as shown in the survey article [47], while much work has studied the effect of foot geometry in the planar case, the explicit study of 3D foot geometry is limited. In [86], researchers formally extend the use of virtual constraints and hybrid zero dynamics from point feet to curved feet for planar control. They later study the trade-offs of foot radius and ankle offset in [87], and find that foot design based on simulation data works well for experiments. An alternative study of foot shape in [5] finds that foot length has a greater effect on gait mechanics than foot radius. Energy losses at impact are caused by changes in the center of mass velocity. Rolling onto the toe of a longer foot reduces velocity change by redirecting the center of mass velocity upward prior to impact. In this way, some push off can be achieved with passive feet with an unactuated ankle. Other researchers in [8] test 3D foot geometries by developing a compliant ground model with passive prosthetic feet to ensure the stability of a 3D controller originally designed using point feet. We have explored the pros and cons of different ground contact models, and we have used multiple types of feet for 3D walking experiments. Hardware options available for MARLO's feet are presented in Figure 6.3 and Appendix A, Figure A.11.

2.5.4 3D Hybrid Zero Dynamics and Virtual Constraints

The work most relevant to this dissertation is that of underactuated 3D walking using hybrid zero dynamics and virtual constraints. As shown in [47] and citations therein, this research is still in its early stages and much work remains to be done. Fundamental 3D-specific principals of these methods are provided in [18], where researchers achieve asymptotic stability of a 3D biped with point feet in simulation using three different and informative feedback control methods. The first method enforces stability by limiting the maximum eigenvalues of the linearized Poincaré map

during gait optimization. The second method uses a stabilizing event-based controller designed to adjust step-to-step continuous feedback control parameters. The gain matrix used for modifying the control parameters is calculated using a discrete linear quadratic regulator (DLQR) and brings the eigenvalues of the linearized Poincaré map within the unit circle. The final method involves making a judicious choice of virtual constraints for control outputs, which stabilizes a previously unstable periodic gait. This initial work is yaw constrained, but an unconstrained implementation of this work that achieves asymptotically stable walking and steering is provided in [125].

Hybrid zero dynamics and virtual constraints are used in [8] to achieve 3D walking with a 13-DOF simulation model of MARLO. Researchers in [8] use an event-based control scheme with the added objective of robustness to external disturbances and model uncertainty. They simplify control by assuming a symmetric model about the yz plane of the torso frame. External shoves of 70 N applied to the robot's center of mass for over 50% of a gait are rejected. The walking gait is also robust to changes in the yaw friction coefficient and spring parameters. Because these parameters are difficult to accurately define on the robot, these results are encouraging.

Recently, a bilinear matrix inequalities (BMI) optimization has been posed in [6] to achieve stable walking with MARLO without an event-based controller. This BMI method applies to periodic orbits that are invariant under the choice of control parameters. The problem of stabilization of periodic orbits is then converted into a set of bilinear matrix inequalities. A BMI optimization is used to tune parameters of the feedback controller to stabilize the 3D gait. One goal of the implementation is to replace the physical intuition required for selecting virtual constraints with a systematic approach to achieve stability or other control objectives.

2.5.5 3D Walking Over Rough Terrain

Bipedal robots have achieved outdoor walking but often with control techniques that result in slow locomotion. One example implementation in [67] achieves outdoor walking on uneven pavement using a 42 DOF HRP-4C robot. To reduce the complexity of walking control design, the system is approximated as a linear inverted pendulum with ZMP delay. The ZMP delay is approximated to account for the real robot ZMP lagging behind the idealized ZMP reference trajectory due to mechanical compliance and control. The HRP-4C robot traverses pavement outdoors with a 3-degree slope at approximately 0.17 m/s. Recently, the DARPA Robotics Challenge Trials featured a number of teams working to complete a set of tasks that mimic a disaster response using an Atlas robot. One of these tasks was navigating rough terrain outdoors [3]. Unfortunately, no detailed quantitative data from these trials is currently available. However, some quantitative results on Atlas can be found in [27]. Researchers in [27] achieve simulated and experimental 3D bipedal walking control using a variant of capture point walking they call the divergent component of motion (DCM). In simulation, they control bipedal robot TORO to traverse stairs with height changes varying between 12 cm and -18 cm at approximately 0.15 m/s. In experiments, they control an Atlas robot to achieve flat ground walking at approximately 0.09 m/s.

3D Walking in this Dissertation

This dissertation addresses the design of feedback controllers and periodic gaits that function well in the presence of modest terrain variation, without reliance on perception and a priori knowledge of the environment. We introduce model-based design methods that are theoretically grounded, systematic, and generalizable. These methods include a feedback control design that enables continuous velocity-based posture regulation and an optimization method that accounts for multiple types of distur-

bances. For optimization, we use a simulation model that is unrestricted in the yaw direction. Thus, the resulting walking gait is not heavily dependent on yaw damping, which is beneficial for uncertain physical ground contact conditions. These methods are validated in simulation and experiment on an underactuated three-dimensional bipedal robot that has only an IMU and joint encoders for sensors, as shown in Figure 2.5. Using a single continuously-defined controller taken directly from optimization, MARLO traverses a variety of terrains while maintaining a mechanical cost of transport (MCOT) between 0.67-0.69 and average walking speeds between 0.9-0.98 m/s.

Contributions of this dissertation with respect to prior work on feedback control of three-dimensional bipedal robots include: introduce a model-based design framework that is able to achieve dynamic three-dimensional walking without hand-tuning of the optimized walking gait; demonstrate robustness by traversing sloped sidewalks and parking lots, terrain covered with randomly thrown boards, and grass fields without a priori knowledge of the environment or external sensing (the robot uses only an IMU and joint encoders); and set a new precedent by evaluating walking efficiency for a variety of realistic terrains.

Figure 2.5: MARLO walking on various outdoor terrains. MARLO is able to traverse man-made (top) and natural (bottom) terrain using a single continuously-defined controller based on the mathematical model of the robot. The mobile gantry does not provide any stabilization or support during walking.

CHAPTER III

Background

Chapter III describes the general concepts used in this dissertation. Section 3.1 outlines fundamental assumptions and derivation of our walking model. Section 3.2 describes the ATRIAS 2.1 model robot, MARLO, which is the basis of our simulation models and is used for experiments. Section 3.3 shows the feedforward and feedback control derivation used in this work. Finally, Section 3.4 introduces hybrid zero dynamics, which uses virtual constraints to redefine a hybrid system on a lower dimensional submanifold.

3.1 Walking Model

3.1.1 Overview

Almost all models of bipedal locomotion are based on a two-phase hybrid model, as seen in Figure 3.1. This hybrid model consists of a single support or swing phase and a double support or impact phase. The single support phase involves the continuous dynamics and control that occur when the stance leg is grounded and the swing leg swings past the stance foot to contact the ground and complete a step. Once the swing leg has come into contact with the ground, an impact occurs, as shown in Figure 3.2. The impact phase uses a different model than that in the single support phase. Specifically, the impact model uses the final states of the previous step to

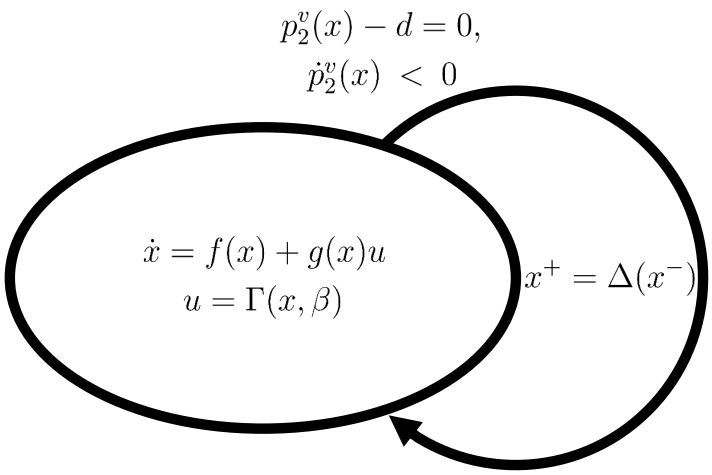

Figure 3.1: Hybrid model for walking. The single support phase exhibits continuous dynamics until the point of impact. Impact occurs when the position of the swing foot reaches the ground height, d, with a negative velocity. The start of the next walking step is initialized by the impact map, which inputs the pre-impact states and outputs the post-impact states for the next step's initial dynamics.

generate the initial states for the next step. The initial states then lead into the continuous single support phase of the next step after the control for the stance and swing legs are swapped. The model of each phase is derived using the Lagrange Method discussed in Section 3.1.3. Important factors for bipedal locomotion using this model include the physical parameters that affect the hybrid dynamics of the system and the method of control used to advance the swing leg ahead of the stance leg during each step.

3.1.2 Walking Model Assumptions

We make four assumptions for our walking model. First, we assume that the stance leg stays pinned to the ground and does not move during the swing phase. This is a fair assumption for two reasons: (1) our biped model has point feet, and (2) our optimization has constraints that ensure that the dynamics of the robot maintain necessary forces to preserve ground contact and avoid slipping for a given coefficient of friction.

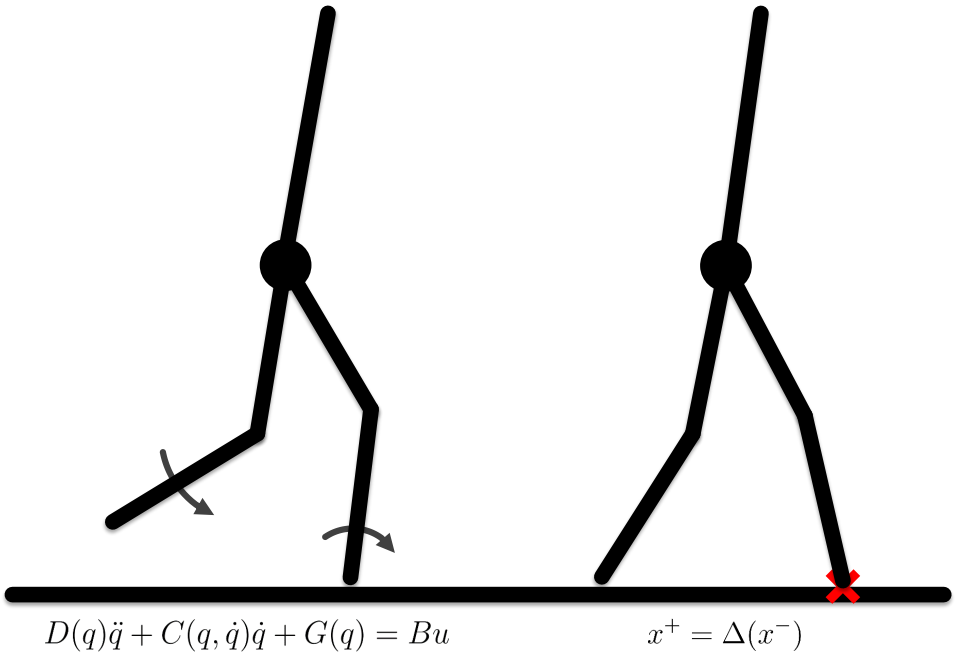

Figure 3.2: Single and double support phase. The single support phase exhibits continuous dynamics governed by (3.7) (left). Double support begins after the swing leg impacts the ground (right), and maps pre-impact conditions to post-impact conditions for the next step using (3.20).

Second, we assume that impact happens instantaneously, which implies an instantaneous change in velocities at impact. This assumption means that the biped is in double support for only an instant before impulsive contact forces cause the previous stance leg to lift off the ground and become the swing leg in the next continuous phase.

Third, we assume a no-slip condition, which guarantees no changes in position during impact. That means that the point of impact is guaranteed to be the point of the pinned stance leg during the next single support phase.

Last, we assume that the biped is able to detect impact. This assumption allows swapping stance and swing leg control after impact, which leads into a new step and re-initializes the continuous control portion of our hybrid model.

3.1.3 Mathematical Model

The basis of our simulation dynamics calculations for both the single support and impact phases is the Lagrangian ($\mathcal{L}$). The Lagrangian is a function of the total kinetic (K) and potential (V) energy of a system. The Lagrangian is defined as

$$\mathcal{L}(q, \dot{q}) := K(q, \dot{q}) - V(q), \tag{3.1}$$

where V is a function of link positions and their mass being acted upon by gravity, K is a function of velocities and the mass-inertia matrix (see (3.3)), and q is a vector of the position coordinates of the system. This energy-based approach serves as an alternative to the Newton-Euler force balance methods for calculating dynamics of a rigid body [22]. Lagrange's equation, which provides the equations of motion, is

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \Gamma, \tag{3.2}$$

where Γ is a vector of the resulting forces and torques. We can put (3.2) in a more useful form by defining D, B, C, and G as matrices satisfying

$$\frac{1}{2}\dot{q}' D(q)\dot{q} = K(q, \dot{q}) \tag{3.3}$$

$$Bu = \Gamma \tag{3.4}$$

$$C(q, \dot{q}) = \left(\frac{\partial}{\partial q}(D(q)\dot{q})\right) - \frac{1}{2}\left(\frac{\partial}{\partial q}(D(q)\dot{q})\right)' \tag{3.5}$$

$$G(q) = \frac{\partial V(q)}{\partial q}. \tag{3.6}$$

D is the mass-inertia matrix, which is a function of the link positions, B is a matrix matching motor input torques to link torques, C is the Coriolis matrix, and G is a

vector of gravity terms. Substituting (3.3)-(3.6) into (3.1) and (3.2) we find

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu, \tag{3.7}$$

which allows us to solve for acceleration of the system as

$$\ddot{q} = D^{-1}(Bu - C\dot{q} - G). \tag{3.8}$$

Equation (3.8) will be used for control derivation in Section 3.3. Equation (3.7) provides the state-space equations we use for simulating continuous phase dynamics. The state-space equations are [46, 136]

$$\begin{aligned}\dot{x} := \frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} &= \begin{bmatrix} \dot{q} \\ D^{-1}(q)(Bu - C(q,\dot{q})\dot{q} - G(q)) \end{bmatrix} \\ &= f(x) + g(x)u,\end{aligned} \tag{3.9}$$

where $x \in \mathcal{X}$ is the state of the system and $u \in \mathbb{R}^m$ are the control inputs. For later use, a parameterized family of continuous-time feedbacks is assumed to be given

$$u = \Gamma(x, \beta), \tag{3.10}$$

where $\beta \in \mathcal{B}$ are control parameters from an admissible set. The input u can be determined using various control methods; the general control methods used in this dissertation are discussed in Sections 3.3 and 3.4. The resulting closed-loop system is

$$\dot{x} = f^{cl}(x, \beta) := f(x) + g(x)\Gamma(x, \beta). \tag{3.11}$$

The closed-loop system is assumed to be continuously differentiable in x and β, thereby guaranteeing local existence and the uniqueness of solutions.

Next, we derive our impact model, which is the second half of the hybrid model we use for simulation. The impact is modeled as a collision of rigid bodies using the model of [59]. Pre-impact states are used to calculate post-impact velocities of the biped. The pre-impact states are calculated assuming that the stance leg is pinned. The post-impact states assume that the previously pinned stance leg is free to move as the new swing leg and that the previous swing leg is now pinned as the new stance leg. We calculate reaction forces and velocities of both leg ends with an extended model that has two more degrees of freedom. These two degrees of freedom arise from p_2, the position of the swing foot, defined as

$$p_2 := \begin{bmatrix} p_h \\ p_v \end{bmatrix}, \tag{3.12}$$

where p_h and p_v are the respective horizontal and vertical positions of the swing foot.

The double support phase begins when the swing foot strikes the ground. This occurs when

$$p_2^v(x) - d = 0, \tag{3.13}$$

for $d \in D$, a finite collection of ground heights used to account for varying terrain. It will be assumed at impact that the transversality condition $\dot{p}_2^v(x) < 0$ is met. Physically, it corresponds to the impact occurring at a point in the gait where the swing foot is moving down toward the ground, as opposed to the impact occurring early in the gait which would lead to tripping [98].

Once in double support, we begin calculating reaction forces and velocities changes using the extended model, q_e, defined as

$$q_e := \begin{bmatrix} q \\ p_2 \end{bmatrix}, \tag{3.14}$$

where q is the position vector used for continuous dynamics. By using q_e, we can include impulsive external forces from ground impact. Using the extended model with the method of Lagrange we find

$$D_e(q_e)\ddot{q_e} + C_e(q_e, \dot{q_e})\dot{q_e} + G_e(q_e) = B_e u_e + \delta F_{ext}. \tag{3.15}$$

F_{ext} is equivalent to the total change in momentum during impact. F_{ext} is found using the differences in velocity of the pre- and post-impact states and the mass inertia matrix with

$$D_e(q_e)(\dot{q_e}^+ - \dot{q_e}^-) = F_{ext} \tag{3.16}$$

$$F_{ext} = E_2(q_e^-)' F_2, \tag{3.17}$$

where $E_2(q_e) = \frac{\partial}{\partial q_e} p_2(q_e)$ and $F_2 = (F_2^T; F_2^N)$ is the vector of forces acting on the swing foot at impact. Note: $q_e^+ = q_e^-$ since position is assumed constant during impact, allowing $D_e(q_e)$ to be general in place of $D_e(q_e^+)$ and $D_e(q_e^-)$ in (3.16). Because we assume the swing foot does not slip (i.e., maintains contact with the ground after impact to become the next stance foot), we find

$$E_2(q_e^-)\dot{q_e}^+ = 0. \tag{3.18}$$

Using (3.16)-(3.18) we can construct [136]

$$\begin{bmatrix} D_e(q_e^-) & -E_2(q_e^-)' \\ E_2(q_e^-) & 0_{2\times 2} \end{bmatrix} \begin{bmatrix} \dot{q_e}^+ \\ F_2 \end{bmatrix} = \begin{bmatrix} D_e(q_e^-)\dot{q_e}^- \\ 0_{2\times 1} \end{bmatrix}. \tag{3.19}$$

Since the leftmost matrix is invertible and all the right-hand-side terms of the equation are known, we can now solve for $\dot{q_e}^+$, our post-impact velocities, and F_2, our ground reaction forces at impact.

Our impact model is complete with the post-impact velocities from (3.19). A simplified expression of the impact model is the continuously differentiable reset map

$$x^+ = \Delta(x^-), \tag{3.20}$$

which does not depend on the ground height since the vector of pre-impact states, $x^- := (q_e^-; \dot{q_e}^-)$, provides foot height at impact. Here, x^+ is a vector of the post-impact states. So that only one continuous-phase mechanical model is needed, the impact map is assumed to include leg swapping, as in [136, pp. 57]. Moreover, for reasons that will become clear in Chapter IV, the impact map is allowed to depend on β.

The overall hybrid model is written as

$$\Sigma : \begin{cases} \dot{x} = f^{cl}(x, \beta) & x^- \notin \mathcal{S}^d \\ x^+ = \Delta(x^-, \beta) & x^- \in \mathcal{S}^d \end{cases} \tag{3.21}$$

where

$$d \in D := \{d_0, d_1, \cdots, d_N\} \tag{3.22}$$

is the set of ground height variations and

$$\mathcal{S}^d := \{x \in \mathcal{X} \mid p_2^v(x) - d = 0,\ \dot{p}_2^v(x) < 0\} \tag{3.23}$$

is the hypersurface in the state space where the swing leg impact occurs at ground height $d \in D$. A diagram of the hybrid model cycle is provided in Figure 3.1.

Remark: The reference [136, pp. 109] shows how to augment the state variables with control parameters in order to accommodate event-based control, as used in [73]. This extension is employed later in (4.21).

3.1.4 Model Solutions

For a given value of $\beta \in \mathcal{B}$, a solution of the hybrid model (3.21) is defined by piecing together solutions of the differential equation (3.11) and the reset map (3.20); see [136, pp. 56],[60]. We are interested in periodic orbits and their perturbations and exclude Zeno and other complex behavior from our notion of a solution.

In the following, for compactness of notation, explicit dependence on β is dropped. A *step* of the robot starts at time t_0 with $x_0 \in \mathcal{S}^{\bar{d}_0}$ for a given value of $\bar{d}_0 \in D$. The reset map is applied, giving an initial condition $\Delta(x_0)$ for the ODE (3.11), with solution $\varphi(t, t_0, \Delta(x_0))$. The step is completed if the solution of the ODE can be continued until a (first) time $t_1 > t_0$ when $x_1 = \varphi(t_1, t_0, \Delta(x_0)) \in \mathcal{S}^{\bar{d}_1}$ for a given value of $\bar{d}_1 \in D$. Not all steps can be completed, but when one is completed, the next step begins by solving the ODE with initial condition $\Delta(x_1)$ at time t_1, etc. The solution (or step) is *periodic* if $\varphi(t_1, t_0, \Delta(x_0)) = x_0$, and $T = t_1 - t_0$ is the *period*. Because the model is time invariant, wherever convenient, the initial time is taken as $t_0 = 0$ and the solution denoted as $\varphi(t, \Delta(x_0))$.

3.2 Bipedal Robot MARLO

The model parameters we use in simulation are based on MARLO, the University of Michigan copy of the ATRIAS 2.1-series of robots built by Jonathan Hurst (see Figure 3.3) [45]. The robot's mass is approximately 55 kg and its legs are one meter long. MARLO is a biped with six actuators. Two motors are used for lateral hip movement and four motors control the links in the legs. MARLO is similar to MABEL in that it is intended to be able to walk over difficult terrain [49]. However, unlike MABEL, MARLO is able to walk in 3D without assistance due to lateral hip actuation.

The robot can be planarized through attachment to a boom. MARLO's planar

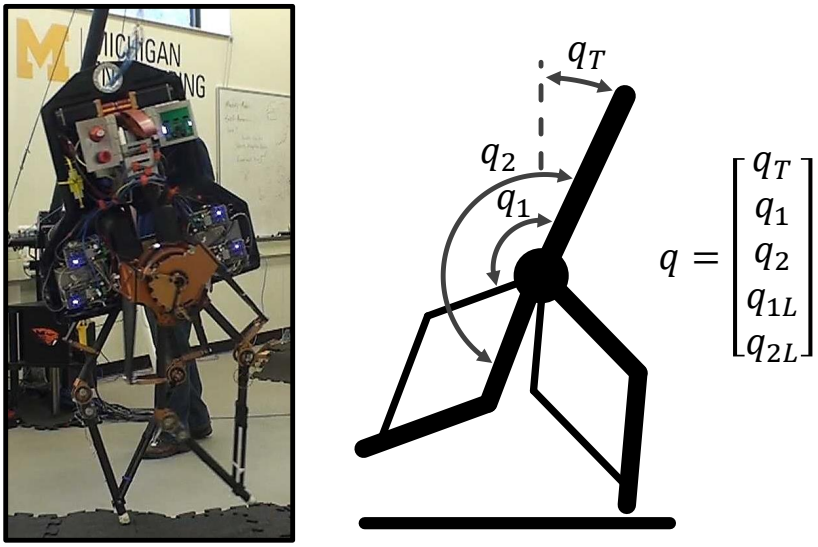

Figure 3.3: Robot MARLO and state description for model of MARLO used in planar simulations. q_T is the absolute angle of the torso with respect to the world vertical frame; the remaining coordinates define the relative link positions.

actuation is applied to four-bar linkages in each leg. Four-bar linkages allow the knee angles to be actuated from the hip, thereby keeping the legs relatively light. However, control of a four-bar linkage in its native coordinate system is not intuitive. We use coordinate transformations to redefine actuated coordinates in terms of more intuitive outputs, as described in Section 3.3.

The motors inside MARLO's hips are attached to the links in the legs through springs to absorb shock and a harmonic drive for a 50:1 gear ratio and mechanical advantage. Furthermore, while the robot has series elastic actuators, the springs are stiff and in this study are removed from the model. With this simplification, the planar robot has 5-DOF when in single support and four actuators and the 3D robot with unconstrained yaw has 9-DOF when in single support and six actuators.

MARLO's motors have torque limits and the legs cannot be fully extended. Limitations such as these are applied as constraints in optimization and torques are saturated in simulated walking. Using the original ATRIAS configuration in Chapter IV, motors are limited to 3 Nm of torque. Following amplifier upgrades, sagittal

motors are limited to 5 Nm of torque in Chapter VI.

3.3 Control Derivation

Control of a hybrid model can be discrete, continuous, or some combination of the two. Discrete control decisions can take place from step to step, whereas continuous control operates during the continuous dynamics of the single support phase. Continuous control may consist of feedback control and/or feedforward control.

Feedforward control uses the derived model to calculate necessary torques for a desired trajectory. The greater the parametric errors between the simulated model and the actual robot, the less accurate feedforward control will be for real experiments. For this reason, feedback control is practical for real experiments in the presence of significant parametric errors. As an aside, results for an alternative reinforcement learning-based feedforward (RLFF) control policy are provided in Appendix B. RLFF works independently of the control model, so it is not effected by modeling errors, and is shown to improve performance for specific terrain conditions.

Feedback control provides input torques to the system to correct for output errors. To begin feedback control design, control outputs must be chosen. Outputs are the basis of what is controlled by the actuators in a simulation model and on the robot during operation. Once a set of outputs has been chosen, desired values to follow must also be defined for outputs. Any differences between the actual and desired control trajectories generate output errors that serve as a platform for feedback control.

Our feedback controller is designed using the method of *virtual constraints*. A holonomic constraint that is expressed as an output and zeroed through the action of an actuator rather than the internal forces of a physical constraint is said to be *virtual*. Virtual constraints can be used to synchronize the links of a robot in order to achieve common objectives of walking, such as supporting the torso, advancing the swing leg in relation to the stance leg, and specifying foot clearance.

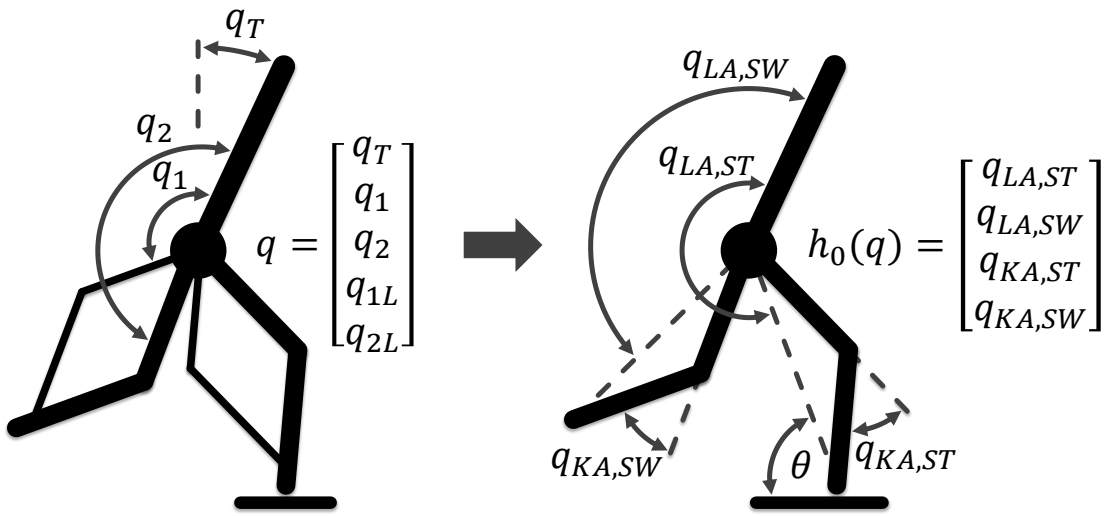

Figure 3.4: Control outputs and unactuated coordinates. For control outputs, $h_0(q)$, LA and KA are abbreviations of leg angle and knee angle, respectively, and ST and SW designate the stance and swing legs. θ defines the remaining degree of underactuation. $h_0(q)$ with θ provides a complete state description.

Our control derivation begins with the output error, $y \in \mathbb{R}^m$, defined as

$$y := h(x, \beta) = h_0(q) - h_d(x, \beta), \tag{3.24}$$

where $h_0(q)$ represents the current state of the outputs, $h_d(x, \beta)$ represents the desired output, and y represents the output error, $h(x, \beta)$. If $h(x, \beta)$ is controlled to be zero, our virtual constraints are satisfied. For compactness of notation, explicit dependence on β is dropped.

It is not intuitive to directly control MARLO's link positions, q, so we design our control outputs, $h_0(q)$, to be more meaningful. The control outputs for the five-link walking model used in Chapter IV are depicted in Figure 3.4. In this case, $h_0(q)$ can be calculated using linear transformation matrices, but other choices of outputs, such as those based on Cartesian coordinates in Chapter V, require nonlinear calculations. It is also possible to redefine our absolute world frame coordinates. In Figure 3.4, q_T is replaced by θ, the absolute angle of the stance leg clockwise from the horizontal

plane at ground contact, which will be useful for defining our gait phasing variable and desired trajectories.

The desired trajectory, h_d from (3.24), can be chosen in several ways. For example, one simple control method for swing leg angle is to keep the angle of the swing leg a mirror of the stance leg. In this case, $h_d = h_d(q)$ is a function of q alone. An even simpler h_d is to use a constant; for example, a constant stance knee angle. The research in this dissertation uses desired trajectories that evolve as a function of the a gait phasing variable τ. One common choice of τ based on θ is defined as

$$\tau(q) = \frac{\theta(q) - \theta_{min}}{\theta_{max} - \theta_{min}}, \tag{3.25}$$

where θ_{min} and θ_{max} are the beginning and ending values of $\theta(q)$ for a given periodic walking gait. For undisturbed walking, τ monotonically increases from 0 to 1. When the stance leg is at θ_{min}, $\tau = 0$, and when the stance leg is at θ_{max}, $\tau = 1$. τ can be used as a monotonically increasing gait phasing variable because θ is always increasing during the single support phase of walking. τ acts a single measure of how far along a step is in its progression. With holonomic virtual constraints and $h_d(\tau)$ a function of gait phase, desired link positions are consistent at each phases of walking, even when the walking speed changes. Such is not the case for time-based control.

In this work, the function $h_d(\tau) \in \mathbb{R}^m$ is a vector of splines that specifies the desired evolution of $h_0(q)$ in terms of the gait phasing variable $\tau(q)$. Here, the splines are Bézier polynomials, with the ith polynomial given by

$$h_{d,i}(\tau) := \sum_{k=0}^{M} \alpha_{i,k} \frac{M\,!}{k\,!(M-k)\,!} \tau^k (1-\tau)^{M-k}, \tag{3.26}$$

where the m degree-$(M+1)$ Bézier polynomials are defined by $\alpha \in \mathbb{R}^{m \times (M+1)}$ [136,

pp. 138]. In Chapter IV, we use sixth order Bézier polynomials for four outputs:

$$h_d(\tau) = \begin{bmatrix} \alpha_{1,0}(1-\tau)^5 + \alpha_{1,1}5\tau(1-\tau)^4 + \cdots + \alpha_{1,5}\tau^5 \\ \vdots \\ \alpha_{4,0}(1-\tau)^5 + \alpha_{4,1}5\tau(1-\tau)^4 + \cdots + \alpha_{4,5}\tau^5 \end{bmatrix}, \tag{3.27}$$

where $\alpha \in \mathbb{R}^{4\times 6}$ is defined by control parameters β. One important result from (3.26) is that $h_{d,i}(0) = \alpha_{i,0}$ and $h_{d,i}(1) = \alpha_{i,M}$. This result means that our first column of (3.27) should be the nominal desired output values at the beginning of our periodic walking gait, and the last column should be the desired output values at the end of our periodic walking gait. The columns in between determine how this transition occurs. Swing knee angle, for instance, may have higher Bézier parameter values in the middle columns to avoid scuffing the ground with the swing foot for $0 < \tau < 1$.

Bézier polynomials are also useful for calculating velocity by taking the derivative of $h_d(\tau)$ from (3.26) with respect to τ:

$$\frac{\partial h_{d,i}(\tau)}{\partial \tau} = \sum_{k=0}^{M-1} (\alpha_{i,k+1} - \alpha_{i,k}) \frac{M\,!}{k\,!(M-1-k)\,!} \tau^k (1-\tau)^{M-1-k} \tag{3.28}$$

Similarly, taking the derivative of (3.27) with respect to τ we find

$$\frac{\partial h_d(\tau)}{\partial \tau} = \begin{bmatrix} (\alpha_{1,1} - \alpha_{1,0})5(1-\tau)^4 + (\alpha_{1,2} - \alpha_{1,1})20\tau(1-\tau)^3 + \cdots + (\alpha_{1,5} - \alpha_{1,4})5\tau^4 \\ \vdots \\ (\alpha_{4,1} - \alpha_{4,0})5(1-\tau)^4 + (\alpha_{4,2} - \alpha_{4,1})20\tau(1-\tau)^3 + \cdots + (\alpha_{4,5} - \alpha_{4,4})5\tau^4 \end{bmatrix}. \tag{3.29}$$

One result from (3.28) is that $\frac{\partial h_{d,i}(0)}{\partial \tau} = (\alpha_{i,1} - \alpha_{i,0})5$ and $\frac{\partial h_{d,i}(1)}{\partial \tau} = (\alpha_{i,5} - \alpha_{i,4})5$. For nominal periodic walking, this result means that the desired velocity of each ith actuated state just before and after impact are a function of only two Bézier param-

eters each. We have shown that Bézier parameter values change the desired walking trajectory with respect to position and velocity. This behavior makes hand-tuning or optimizing a desired walking gait relatively straight-forward using Bézier parameters.

With control outputs and desired trajectories defined, we find our new output equation using (3.24)-(3.29) to be

$$y = h_0(q) - h_d(\tau), \tag{3.30}$$

which is differentiated as

$$\frac{dy}{dt} = \dot{y} = \frac{\partial h_0(q)}{\partial q}\dot{q} - \frac{\partial h_d(\tau)}{\partial \tau}\dot{\tau} \tag{3.31}$$

$$\ddot{y} = \frac{\partial h_0(q)}{\partial q}\ddot{q} + \frac{\partial}{\partial q}\left(\frac{\partial h_0(q)}{\partial q}\dot{q}\right)\dot{q} - \frac{\partial h_d(\tau)}{\partial \tau}\ddot{\tau} - \frac{\partial^2 h_d(\tau)}{\partial^2 \tau}\dot{\tau}^2. \tag{3.32}$$

Recognizing from (3.25) that $\ddot{\tau}$ is a function of $\ddot{q}$ via the chain rule,

$$\ddot{\tau} = \frac{\partial \tau(q)}{\partial q}\ddot{q}, \tag{3.33}$$

and then taking the partial derivative of $h(x, \beta)$ from (3.24),

$$\frac{\partial h}{\partial q} = \frac{\partial h_0(q)}{\partial q} - \frac{\partial h_d(\tau)}{\partial \tau}\frac{\partial \tau(q)}{\partial q}, \tag{3.34}$$

we simplify (3.32) to

$$\ddot{y} = \frac{\partial h}{\partial q}\ddot{q} + \frac{\partial}{\partial q}\left(\frac{\partial h_0(q)}{\partial q}\dot{q}\right)\dot{q} - \frac{\partial^2 h_d(\tau)}{\partial^2 \tau}\dot{\tau}^2, \tag{3.35}$$

where $\frac{\partial h}{\partial q}$ is called the decoupling matrix.

We now derive our feedforward and feedback control using the previous result. Substituting (3.8) in (3.35) for $\ddot{q}$, we use the relationship between the inverse dynamics

and our control outputs to find the feedforward component of our control scheme as

$$\ddot{y} = \frac{\partial h}{\partial q} D^{-1}(Bu - C\dot{q} - G) + \frac{\partial}{\partial q}\left(\frac{\partial h_0(q)}{\partial q}\dot{q}\right)\dot{q} - \frac{\partial^2 h_d(\tau)}{\partial^2 \tau}\dot{\tau}^2. \tag{3.36}$$

Our feedback control requires that y is relative degree two, which is confirmed in (3.36), since $\ddot{y}$ is the first derivative of y with any dependence on u. Our feedback control is designed to drive y to 0 and identically satisfy $\dot{y} = \ddot{y} = 0$. Using derivative and proportional gains, k_d and k_p respectively, we relate the feedback component of our control scheme to our control outputs as

$$\ddot{y} + k_d\dot{y} + k_p y = 0 \rightarrow \ddot{y} = -k_d\dot{y} - k_p y. \tag{3.37}$$

Combining (3.36) and (3.37) we find

$$\frac{\partial h}{\partial q} D^{-1}(Bu - C\dot{q} - G) + \frac{\partial}{\partial q}\left(\frac{\partial h_0(q)}{\partial q}\dot{q}\right)\dot{q} - \frac{\partial^2 h_d(\tau)}{\partial^2 \tau}\dot{\tau}^2 = -k_d\dot{y} - k_p y,$$

which leads to a precise solution for our control input u,

$$u = \left(\frac{\partial h}{\partial q} D^{-1} B\right)^{-1}\left(\frac{\partial h}{\partial q} D^{-1}(C\dot{q} + G) - \frac{\partial}{\partial q}\left(\frac{\partial h_0(q)}{\partial q}\dot{q}\right)\dot{q} + \frac{\partial^2 h_d(\tau)}{\partial^2 \tau}\dot{\tau}^2 - k_d\dot{y} - k_p y\right). \tag{3.38}$$

u in (3.38) uses a combination of feedforward control based on inverse dynamics and feedback control for driving outputs to zero. The feedback-only solution that is implemented in some full dynamic simulations and experiments is

$$u = \left(\frac{\partial h}{\partial q} D^{-1} B\right)^{-1}(-k_d\dot{y} - k_p y). \tag{3.39}$$

This form of feedback-only control is not as sensitive to parametric modeling errors because it relies less on C and D, which are approximated from the modeled dynamics.

3.4 Hybrid Zero Dynamics

Hybrid zero dynamics (HZD) enables us to redefine our hybrid model on a lower dimensional submanifold [46, 134, 135, 136]. Using a lower dimensional model with less complicated simulation calculations allows us to optimize our control design from Section 3.3 more quickly. The primary assumption behind HZD is that we are able to control all of our outputs to be identically zero, thus reducing our modeled degrees of freedom and simplifying the complexity of the system. The simplified system is said to be contained to the *zero dynamics manifold* and the simplified dynamics of the system are called the *zero dynamics*, which consists of any remaining degrees of freedom given all outputs are zeroed. If a hybrid system is invariant to the zero dynamics manifold, *hybrid* zero dynamics are achieved.

A key concept in the HZD derivation is the distinction between actuated and unactuated coordinates. We define an alternate coordinate system

$$\bar{q} := \begin{bmatrix} q_u \\ \\ q_a \end{bmatrix}, \tag{3.40}$$

with unactuated coordinates q_u and actuated coordinates q_a. Using the five-link example in Figure 3.5, $q_u = \theta$ and q_a consists of the leg and knee angles. The linear map between q and $\bar{q}$ is

$$\begin{aligned} q =& T_0\bar{q} + T_1 \\ \rightarrow \bar{q} =& T_0^{-1}(q - T_1) = \begin{bmatrix} \theta \\ \\ q_a \end{bmatrix}, \end{aligned} \tag{3.41}$$

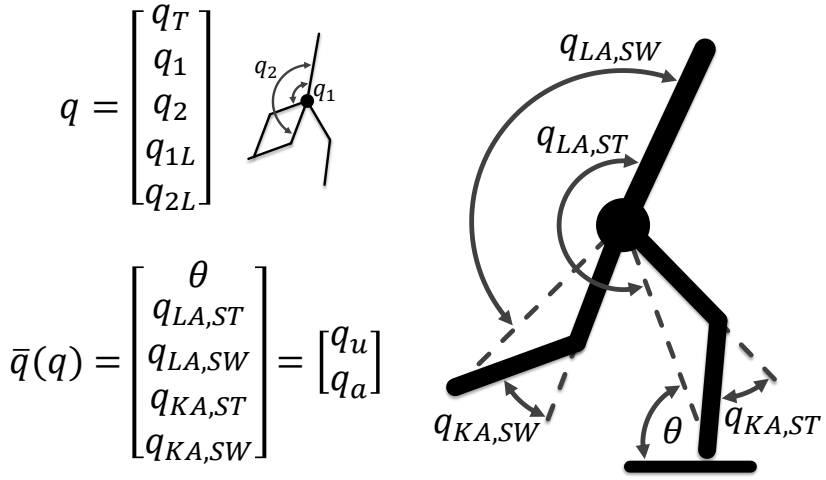

Figure 3.5: Change of coordinates from q to $\bar{q}$.

where

$$T_0 = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 & -\frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{2} \end{bmatrix}, \; T_1 = \begin{bmatrix} \frac{3\pi}{2} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The HZD model derivation starts by solving for the dynamics of the actuated coordinates, q_a. First, use (3.25) to redefine $\dot{\tau}$ in terms of $\dot{\theta}$

$$\dot{\tau} = \frac{\dot{\theta}}{\theta_{max} - \theta_{min}} = \frac{\dot{\theta}}{\Delta\theta}. \tag{3.42}$$

Next, we assume all outputs are controlled to zero,

$$\ddot{y} + K_d\dot{y} + K_p y \equiv 0, \tag{3.43}$$

and recognize q_a can be substituted for $h_0(q)$ in (3.24) to find

$$y = 0 = h_0(q) - h_d(\tau) = q_a - h_d(\tau) \rightarrow q_a = h_d(\tau). \tag{3.44}$$

Equation (3.44) shows that because outputs are zeroed, all virtual constraints are satisfied and actuated coordinates are exactly equal to their desired trajectories. To find the velocities and accelerations of the actuated states, (3.42)-(3.44) are applied to derivatives (3.31)-(3.32) to find

$$\dot{y} = 0 \rightarrow \dot{q}_a = \frac{\partial h_d(\tau)}{\partial \tau}\dot{\tau} = \frac{\partial h_d(\tau)}{\partial \tau}\frac{\dot{\theta}}{\Delta\theta} \tag{3.45}$$

$$\ddot{y} = 0 \rightarrow \ddot{q}_a = \frac{\partial h_d(\tau)}{\partial \tau}\ddot{\tau} + \frac{\partial^2 h_d(\tau)}{\partial^2 \tau}\dot{\tau}^2 = \frac{\partial h_d(\tau)}{\partial \tau}\frac{\ddot{\theta}}{\Delta\theta} + \frac{\partial^2 h_d(\tau)}{\partial^2 \tau}\Big(\frac{\dot{\theta}}{\Delta\theta}\Big)^2. \tag{3.46}$$

Equations (3.44)-(3.46) express the dynamics of the actuated coordinates when virtual constraints are satisfied and a system is invariant to the zero dynamics manifold. Since τ is a function of θ, which is the only unactuated state in $\bar{q}$, the entire system can be described by τ. The entire five-link walking model, for example, reduces to a single degree of freedom. Since τ monotonically increases with each step, the system can simply be thought of as a periodic orbit of a single variable. From step to step, all link positions of the biped are the same during each orbit for the same corresponding τ values. This synchronization also makes the dynamics of the system easier to calculate.

With the actuated coordinate dynamics solved, we turn our attention to the dynamics of the full system. First, define $H(q, \dot{q})$ as

$$H(q, \dot{q}) := C(q, \dot{q})\dot{q} + G(q),$$

and then simplify the dynamics equation (3.7) to

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu. \tag{3.47}$$

After transforming coordinates to $\bar{q}$ as shown in (3.41), we need to perform similar transformations to $D(q)$, $H(q, \dot{q})$, and B to match the transformed coordinate system

$$\bar{D}(\bar{q}) = T_0' D(q) T_0 \tag{3.48}$$

$$\bar{H}(\bar{q}, \dot{\bar{q}}) = T_0' H(q, \dot{q}) \tag{3.49}$$

$$\bar{B} = T_0' B, \tag{3.50}$$

and then transform the simplified dynamics equation (3.47) to the new coordinate system as

$$\bar{D}(\bar{q})\ddot{\bar{q}} + \bar{H}(\bar{q}, \dot{\bar{q}}) = \bar{B}u. \tag{3.51}$$

Two interesting properties of the newly derived $\bar{B}$ and $\bar{D}(\bar{q})$ are: (1) $\bar{B}$ contains all zeros in the first row, since this row corresponds to the unactuated coordinate θ in $\bar{q}$; and (2) that $\bar{D}(\bar{q})$ is a function of only the actuated coordinates, q_a. These properties are shown as

$$\bar{B} = \begin{bmatrix} 0 \\ \\ \bar{B}_a \end{bmatrix}, \ \bar{D}(\bar{q}) = \bar{D}_a(q_a), \tag{3.52}$$

where $\bar{B}_a \in \mathbb{R}^{4\times 4}$.

The unactuated dynamics are separated from the full dynamics to make their solution simpler. To separate actuated and unactuated dynamics, we rewrite the

transformed dynamics equation (3.51) as

$$\begin{bmatrix} \bar{D}_{11}(q_a) & \bar{D}_{12}(q_a) \\ \bar{D}_{21}(q_a) & \bar{D}_{22}(q_a) \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{q}_a \end{bmatrix} + \begin{bmatrix} \bar{H}_1(\bar{q}, \dot{\bar{q}}) \\ \bar{H}_2(\bar{q}, \dot{\bar{q}}) \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{B}_a \end{bmatrix} u, \tag{3.53}$$

where

$$\bar{D}_a = \begin{bmatrix} \bar{D}_{11}(q_a) & \bar{D}_{12}(q_a) \\ \bar{D}_{21}(q_a) & \bar{D}_{22}(q_a) \end{bmatrix}, \tag{3.54}$$

$$\bar{H}(\bar{q}, \dot{\bar{q}}) = \begin{bmatrix} \bar{H}_1(\bar{q}, \dot{\bar{q}}) \\ \bar{H}_2(\bar{q}, \dot{\bar{q}}) \end{bmatrix}. \tag{3.55}$$

The unactuated dynamics of the system correspond to the top row of the separated dynamics equation (3.53),

$$\bar{D}_{11}(q_a)\ddot{\theta} + \bar{D}_{12}(q_a)\ddot{q}_a + \bar{H}_1(\bar{q}, \dot{\bar{q}}) = 0, \tag{3.56}$$

which can be revised with the solution of $\ddot{q}_a$ from (3.46) to find

$$\bar{D}_{11}(q_a)\ddot{\theta} + \bar{D}_{12}(q_a)\left(\frac{\partial h_d(\tau)}{\partial \tau}\frac{\ddot{\theta}}{\Delta\theta} + \frac{\partial^2 h_d(\tau)}{\partial^2 \tau}\left(\frac{\dot{\theta}}{\Delta\theta}\right)^2\right) + \bar{H}_1(\bar{q}, \dot{\bar{q}}) = 0. \tag{3.57}$$

Finally, we use the unactuated dynamics equation (3.57) to define the lower dimensional zero dynamic model of the system,

$$D_{zero}(\theta)\ddot{\theta} + H_{zero}(\theta, \dot{\theta}) = 0, \tag{3.58}$$

where

$$D_{zero}(\theta) = \bar{D}_{11}(q_a) + \bar{D}_{12}(q_a)\frac{\partial h_d(\tau)}{\partial \tau}\frac{1}{\Delta\theta}, \tag{3.59}$$

$$H_{zero}(\theta, \dot{\theta}) = \bar{D}_{12}(q_a)\frac{\partial^2 h_d(\tau)}{\partial^2 \tau}\Big(\frac{\dot{\theta}}{\Delta\theta}\Big)^2 + \bar{H}_1(\bar{q}, \dot{\bar{q}}), \tag{3.60}$$

and q_a is considered a function of θ since $q_a = h_d(\tau(\theta))$. Using $D_{zero}(\theta)$ and $H_{zero}(\theta, \dot{\theta})$, the zero dynamics are solved from (3.58) as

$$\ddot{\theta} = -D_{zero}(\theta)^{-1} H_{zero}(\theta, \dot{\theta}). \tag{3.61}$$

Once the zero dynamics are solved, we find the necessary control torques, u, to maintain zero output error and contain the full system within the zero dynamics manifold. The dynamics of the full system corresponding to u appear on the bottom row of the separated dynamics equation (3.53),

$$\bar{D}_{21}(q_a)\ddot{\theta} + \bar{D}_{22}(q_a)\ddot{q}_a + \bar{H}_2(\bar{q}, \dot{\bar{q}}) = \bar{B}_a u. \tag{3.62}$$

Using $\ddot{\theta}$ and inverse kinematics, we solve (3.62) for u as

$$u = \bar{B}_a^{-1}\big(\bar{D}_{21}(q_a)\ddot{\theta} + \bar{D}_{22}(q_a)\ddot{q}_a + \bar{H}_2(\bar{q}, \dot{\bar{q}})\big). \tag{3.63}$$

Hybrid zero dynamics are achieved for bipeds if the zero dynamics manifold is invariant under the impact map. This condition occurs for periodic walking and is achieved with perturbed impacts in this dissertation using the control parameter reset map introduced in Section 4.2.3. Compared to a higher dimensional model, the lower dimensional hybrid zero dynamic model is computationally less expensive and results in quicker convergence for control optimizations.

CHAPTER IV

Optimization for Accommodation of Unknown Disturbances

In this chapter, the design of periodic gaits that will also function well in the presence of modestly uneven terrain is investigated. The gait design problem is formulated in terms of parameter optimization, with a cost function that accounts for periodicity under nominal walking conditions, and additional terms that specifically account for trajectory and control-effort perturbations arising from a finite set of ground height changes. Trajectory and control deviations are related to a nominal periodic orbit via a gait phasing variable, which is more natural than comparing solutions on the basis of time. The gait phasing variable is also used to penalize more heavily deviations that persist "late" into the gait. When the method is evaluated on planar MARLO, for modest terrain height variations typical of sidewalks, parking lots, and maintained grass fields, it is observed that a cost function that favors *quasi dead-beat* rejection of terrain disturbances results in the best performing gaits of the three tested approaches, both in simulation and in experiments.

Parts of this work can be found in [42]. Videos of experiments are available at [39].

4.1 Optimization for Accommodation of Unknown Disturbances

Let the nominal change in ground height *step to step* be represented by $d_0 \in D = \{d_0, \cdots, d_N\}$ from (3.22). Using the notation introduced in Section 3.1.4, we seek control parameters $\beta \in \mathcal{B}$ and initial conditions $x_0 \in \mathcal{X}$ giving rise to a periodic solution of the closed-loop system (3.21); that is, for which there exists $T_0 > 0$ such that

$$x_0 = \varphi(T_0, \Delta(x_0)). \tag{4.1}$$

Moreover, for the same value of $\beta \in \mathcal{B}$, we desire that the periodic orbit ensures the existence of the following additional solutions of the closed-loop system: $\forall\, 1 \leq j \leq N$, $d_j \in D$, $\exists\, 0 < t_j < \infty$, and $0 < T_j < \infty$ such that

$$x_j := \varphi(t_j, \Delta(x_0)) \in \mathcal{S}^{d_j}, \tag{4.2}$$

and

$$\varphi(T_j, \Delta(x_j)) \in \mathcal{S}^{d_0}. \tag{4.3}$$

In plain words, there exist steps that begin on the periodic orbit, end at ground height d_j, and continue for at least one more step at nominal ground height d_0.

In the following, we set up a parameter optimization problem in (β, x_0) for finding a periodic solution that meets these conditions. Moreover, we will pose a cost function on the steps following the change in ground height that favors solutions that "return closely" to the nominal periodic solution, that is, the closed-loop system attenuates the effects of the set of ground height variations.

Remark: This optimization framework is not exclusive to terrain disturbances. The nominal periodic solution can be perturbed using alternative disturbance types or a combination thereof. This extension is introduced in Chapter VI.

4.1.1 Gait Phase and Trajectory Deviations

Compared to time-based methods, phase-based synchronization of walking trajectories is shown to be more natural to humans in [36] and advantageous for control in [76]. For this optimization method, we have found that computing deviations of the perturbed solutions from the nominal periodic solution does not work well when the trajectories are parameterized by time. This is because terrain disturbances cause varying initial conditions, which cause perturbed trajectories to be unsynchronized with respect to time. We use instead a *gait phasing variable*, $\bar{\tau} : \mathcal{X} \to \mathbb{R}$, that is strictly increasing along walking steps. Examples include the horizontal position of the center of mass, the horizontal position of the hips, or the angle of the line connecting the hip and the ground contact point of the stance leg, which will be used in Section 4.2. The gait phase can be thought of as a measure of progress through each step. We further assume that the units are normalized on the periodic orbit so that it takes values in $[0, 1]$, namely

$$\bar{\tau}(\Delta(x_0)) = 0 \tag{4.4}$$

$$\bar{\tau}(x_0) = 1, \tag{4.5}$$

and that $L_g\bar{\tau}(x) := \frac{\partial\bar{\tau}}{\partial x}(x)g(x) = 0$.

Let $\bar{\tau}_j(t) := \bar{\tau}(\varphi(t, \Delta(x_j))$, for $0 \leq t \leq T_j$, and as in [24], denote by τ_j^+ and τ_j^- the initial and final values of $\bar{\tau}$ along the trajectory. Due to the strictly increasing assumption, the inverse map $\bar{\tau}_j^{-1} : [\tau_j^+, \tau_j^-] \to [0, T_j]$ exists. Define

$$\tilde{x}_j(\tau) := \varphi(\bar{\tau}_j^{-1}(\tau), \Delta(x_j)) \tag{4.6}$$

$$\tilde{u}_j(\tau) := \Gamma(\varphi(\bar{\tau}_j^{-1}(\tau), \Delta(x_j)), \beta). \tag{4.7}$$

For $1 \leq j \leq N$, deviations in the state and control trajectories are defined as

$$\delta x_j(\tau) := \begin{cases} \tilde{x}_j(\tau) - \tilde{x}_0(0) \text{ if } \tau < 0 \\ \tilde{x}_j(\tau) - \tilde{x}_0(\tau) \text{ if } \tau \in [0,1] \\ \tilde{x}_j(\tau) - \tilde{x}_0(1) \text{ if } \tau > 1 \end{cases} \tag{4.8}$$

$$\delta u_j(\tau) := \begin{cases} \tilde{u}_j(\tau) - \tilde{u}_0(0) \text{ if } \tau < 0 \\ \tilde{u}_j(\tau) - \tilde{u}_0(\tau) \text{ if } \tau \in [0,1] \\ \tilde{u}_j(\tau) - \tilde{u}_0(1) \text{ if } \tau > 1 \end{cases} \tag{4.9}$$

for $\tau_j^+ \leq \tau \leq \tau_j^-$ [1].

Using (4.8) and (4.9), the weighted square error is defined as

$$||\delta x_j(\tau)||^2 :=< Q\delta x_j(\tau), \delta x_j(\tau) > \tag{4.10}$$

$$||\delta u_j(\tau)||^2 :=< R\delta u_j(\tau), \delta u_j(\tau) >, \tag{4.11}$$

for Q and R positive semi-definite (constant) matrices.

4.1.2 Robust Control Cost Function

The problem of defining a cost function $\mathcal{J}_0$ and appropriate equality and inequality constraints for determining a nominal periodic solution of (3.11) has been addressed in [136, pp. 151-155], [127, 135] using parameter optimization. Here we define additional terms that penalize deviations induced by the disturbances in D.

[1] A more comprehensive approach for calculating errors of perturbed trajectories is available in [118].

For $1 \leq j \leq N$, we define

$$\mathcal{J}_j = \frac{1}{(\tau_j^- - \tau_j^+)} \int_{\tau_j^+}^{\tau_j^-} \left(||\delta x_j(\tau)||^2 + ||\delta u_j(\tau)||^2 \right) \frac{(\tau - \tau_j^+)}{(\tau_j^- - \tau_j^+)} d\tau. \tag{4.12}$$

The term $\frac{(\tau - \tau_j^+)}{(\tau_j^- - \tau_j^+)}$ under the integral scales the errors so that initial deviations from the nominal periodic trajectory are discounted with respect to errors toward the end of the step. The rationale for this is that if the closed-loop system were to rejoin the nominal periodic orbit by the end of the step, the disturbance would have been rejected and a next step would be guaranteed. The scale factor allows the optimization to focus on approximately achieving this objective. The term $\frac{1}{(\tau_j^- - \tau_j^+)}$ outside the integral is included so that perturbed step costs are normalized w.r.t. the varying ranges of τ_j resulting from higher and lower terrain disturbances. The benefit of the scalings introduced in (4.12) will be illustrated by comparing control solutions that include them against those that do not.

The overall cost function is

$$\mathcal{J} = \mathcal{J}_0 + \sum_{j=1}^{N} w_j \mathcal{J}_j, \tag{4.13}$$

where w_j determines the relative weight of each perturbation.

Parameter optimization problem: Find $(\beta; x_0)$ that (locally) minimize $\mathcal{J}$ subject to the existence of a periodic solution of (3.21) that respects ground contact conditions, torque limits, and other relevant physical properties, as illustrated in Section 4.2.4.

Remark: $\mathcal{J}_j$ in (4.13) can account for costs arising from multiple perturbed steps following a single disturbance. This is applied in Section 5.2.5.

4.2 Optimization Implementation

We now provide an implementation of the optimization approach for rejecting terrain disturbances presented in Section 4.1.

4.2.1 Configuration Variables

Control is based on the rigid five-link model of MARLO presented in Section 3.2, which has five DOF when in single support and four actuators. The configuration variables we use are depicted in Figure 3.4. Specifically,

$$q = [\theta, q_a]', \tag{4.14}$$

where θ is the absolute angle of the stance leg clockwise from the horizontal plane at ground contact and

$$q_a = \begin{bmatrix} q_{LA,ST} \\ q_{LA,SW} \\ q_{KA,ST} \\ q_{KA,SW} \end{bmatrix} \tag{4.15}$$

is a vector of the actuated configuration variables. In (4.15), LA and KA are abbreviations of leg angle and knee angle, respectively, and ST and SW designate the stance and swing legs. The Lagrangian model for single support and the impact model, (3.9) and (3.20), use $x = (q; \dot{q}) \in \mathcal{X}$ an open subset of $\mathbb{R}^{10}$ and $u \in \mathbb{R}^4$ for one degree of underactuation during single support.

4.2.2 Family of Feedback Controllers

The feedback controller is designed using the method of *virtual constraints* and *hybrid zero dynamics* (see Sections 3.3 and 3.4, [46, 134]). For planar MARLO, four virtual constraints are defined, one for each available actuator. The output vector y

is defined in terms of the configuration variables and a set of parameters κ and β,

$$y = h(q, \kappa, \beta), \tag{4.16}$$

in such a way that the output has vector relative degree 2 [64, pp. 220] on a subset of interest, $\mathcal{X} \times \mathcal{K} \times \mathcal{B}$. Specifically, κ is used to maintain *hybrid zero dynamics* following impacts with terrain disturbances. The feedback controller is based on input-output linearization, namely

$$u_{ff}(q, \dot{q}, \kappa, \beta) := -\big[L_g L_f h(q, \kappa, \beta)\big]^{-1} L_f^2 h(q, \dot{q}, \kappa, \beta), \tag{4.17}$$

$$u_{fb}(q, \dot{q}, \kappa, \beta) := -\big[L_g L_f h(q, \kappa, \beta)\big]^{-1} \big(K_p y + K_d \dot{y}\big), \tag{4.18}$$

with

$$u = \Gamma(q, \dot{q}, \kappa, \beta) := u_{ff}(q, \dot{q}, \kappa, \beta) + u_{fb}(q, \dot{q}, \kappa, \beta), \tag{4.19}$$

where (4.19) is the simplified Lie derivative form of (3.38). Along solutions of the closed-loop system, $\ddot{y} + K_d \dot{y} + K_p y \equiv 0$ (3.43).

Section 4.2.3 gives an explicit construction of $h(q, \kappa, \beta)$ in terms of the actuated variables q_a and a set of degree-$(M + 1)$ Bézier polynomials. Moreover, with this output choice, it is straightforward to construct a function $\Psi : \mathcal{S}^d \times \mathcal{B} \to \mathcal{K}$ such that for all

$$\beta \in \mathcal{B} \text{ and } \begin{bmatrix} q^+ \\ \dot{q}^+ \end{bmatrix} = \Delta(q^-, \dot{q}^-)$$

the initial values of the outputs are zeroed, that is,

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} y^+ \\ \dot{y}^+ \end{bmatrix} = \begin{bmatrix} h(q^+, \kappa^+, \beta) \\ \frac{\partial}{\partial q} h(q^+, \kappa^+, \beta) \dot{q}^+ \end{bmatrix} \tag{4.20}$$

for $\kappa^+ = \Psi(q^-, \dot{q}^-, \beta)$.

The parameters κ are constant within each step and are reset at the end of each step. They are thus formally states and are included in the dynamics with

$$x_e := [q, \dot{q}, \kappa]' \tag{4.21}$$

and $\dot{\kappa} = 0$. The closed-loop model used in the optimization is then

$$\Sigma : \begin{cases} \dot{x}_e = f^{cl}(x_e, \beta) & x_e^- \notin \mathcal{S}_e^d \\ x_e^+ = \Delta_e(x_e^-) & x_e^- \in \mathcal{S}_e^d, \end{cases} \tag{4.22}$$

where

$$f^{cl}(x_e, \beta) = f^{cl}(x, \kappa, \beta) := \begin{bmatrix} f(x) + g(x)\Gamma(x, \kappa, \beta) \\ 0 \end{bmatrix}, \tag{4.23}$$

$$\Delta_e(x_e^-, \beta) := \begin{bmatrix} \Delta(q^-, \dot{q}^-) \\ \Psi(q^-, \dot{q}^-, \beta) \end{bmatrix}, \tag{4.24}$$

and

$$\mathcal{S}_e^d := \mathcal{S}^d \times \mathcal{K}. \tag{4.25}$$

Remarks: (a) The reset map is independent of the current value of κ. (b) Because of the second-order system (3.43) and the reset map in (4.20), solutions of (4.23) that are initialized in $\mathcal{S}_e^d$ satisfy $y(t) \equiv 0$. This has two consequences: (i) The solutions evolve on the zero dynamics manifold, a 2-dimensional invariant surface and can thus be computed from a 2-dimensional vector field [134], [94]. This fact is used to accelerate the parameter optimization process. (ii) The feedback term u_{fb} in (4.18) is identically zero, and thus Γ in (4.19) is independent of the gains K_p and K_d.

4.2.3 Bézier Parameter Reset Derivation

In Section 4.2.2, we discuss how control parameters κ must be reset such that we satisfy (4.20). First, we define our output

$$y = h(q, \kappa, \beta) = q_a(q) - h_d(q, \kappa, \beta), \tag{4.26}$$

where $h_d \in \mathbb{R}^4$ are desired trajectories defined by Bézier polynomials. Each ith polynomial is defined as

$$h_{d,i}(q, \kappa, \beta) := \sum_{k=0}^{M} \alpha_{i,k} \frac{M\,!}{k\,!(M-k)\,!} \tau^k (1-\tau)^{M-k}. \tag{4.27}$$

A set of four degree-$(M+1)$ Bézier polynomials can be defined by $\alpha \in \mathbb{R}^{4\times(M+1)}$ [136, pp. 138]. We designate the first two columns of parameters, α_0 and α_1, as κ,

$$\kappa = [\alpha_0, \alpha_1]. \tag{4.28}$$

α_0 and α_1 have the most effect on trajectories immediately after impact during low τ values. The remaining fixed columns, β, determine trajectories toward the end of the gait. Hence, perturbed trajectories return to the nominal gait as τ increases.

Let $y^+ = \dot{y}^+ = 0$ as in (4.20). Using (4.26), this implies that

$$h_d(q^+, \kappa^+, \beta) = q_a^+. \tag{4.29}$$

Note, to match h_d to q_a^+, we must reset at least one column of Bézier parameters. To guarantee desired trajectories match post-impact velocities, we reset a second column to satisfy

$$\frac{\partial h_d(q^+, \kappa^+, \beta)}{\partial \tau} \dot{\tau}^+ = \dot{q}_a^+. \tag{4.30}$$

Solving (4.29) and (4.30) using α_0 and α_1 we find

$$\alpha_0 = \frac{q_a^+ - \sum_{k=1}^{M} \alpha_k \frac{M\,!(\tau^+)^k(1-\tau^+)^{M-k}}{k\,!(M-k)\,!}}{(1-\tau^+)^M} \tag{4.31}$$

$$\alpha_1 = \frac{\frac{\dot{q}_a^+}{\dot{\tau}^+} - \alpha_2 M(M-1)\tau^+(1-\tau^+)^{M-2} - a + b}{M((1-\tau^+)^{M-1} + \tau^+(1-\tau^+)^{M-2})}, \tag{4.32}$$

where

$$a = \sum_{k=2}^{M-1} (\alpha_{k+1} - \alpha_k) \frac{M\,!(\tau^+)^k(1-\tau^+)^{M-1-k}}{k\,!(M-1-k)\,!}, \tag{4.33}$$

$$b = \frac{M}{1-\tau^+} \left(q_a^+ - \sum_{k=2}^{M} \alpha_k \frac{M\,!(\tau^+)^k(1-\tau^+)^{M-k}}{k\,!(M-k)\,!} \right). \tag{4.34}$$

Reseting control parameters α_0 to (4.31) and α_1 to (4.32) forms a solution for κ^+ that always satisfies (4.20).

4.2.4 Gait Phase and Three Periodic Orbits

Along periodic walking gaits, the coordinate θ shown in Figure 3.4 is monotonic and cycles between a minimum value θ_{min} and a maximum value θ_{max}. The gait phasing variable is $\tau(q) = \frac{\theta(q)-\theta_{min}}{\theta_{max}-\theta_{min}}$ as defined in (3.25).

The cost function for the nominal periodic orbit is taken as

$$\mathcal{J}_0 = \frac{1}{\text{step length}} \int_0^{T_0} \sum_{i=1}^{4} |u_i \dot{q}_{motor,i}| dt, \tag{4.35}$$

where T_0 is the period, u is the 4-vector of motor torques, and $\dot{q}_{motor}$ is the corresponding 4-vector of motor angular velocities, obtained from the link velocities and gear ratios [113]. The product of u_i and $\dot{q}_{motor,i}$ is the instantaneous mechanical power from each motor.

The nominal periodic orbit was computed for walking on level ground, that is $d_0 =$

0, by optimizing (4.35) subject to (4.22), and the following additional constraints: peak motor torque less than 2.5 Nm; vertical ground reaction force positive and friction coefficient less than 0.6; minimum foot clearance at mid-stance of 0.05 m; minimum knee bend of 22^o to avoid hyperextension; average walking speed of at least 0.75 m/s; minimum swing-leg retraction of 7^o; dimensionless swing-leg retraction less than -0.5 [57]. The computations were performed offline with `fmincon` in MATLAB.

The set of terrain variations was taken as $D = \{0, \pm 2 \text{ cm}, \pm 4 \text{ cm}\}$. A second periodic gait was found that minimized the cost function (4.13), with $w_j = 100$ for $1 \leq j \leq 4$. Taking the weights all equal is analogous to assuming a uniform distribution of terrain variations [25].

To investigate the utility of discounting trajectory deviations that occur early in the perturbed steps, a third periodic orbit was found with the term $\frac{(\tau - \tau_j^+)}{(\tau_j^- - \tau_j^+)}$ removed from (4.12), resulting in

$$\mathcal{J}_j = \frac{1}{(\tau_j^- - \tau_j^+)} \int_{\tau_j^+}^{\tau_j^-} \left(||\delta x_j(\tau)||^2 + ||\delta u_j(\tau)||^2 \right) d\tau. \tag{4.36}$$

In total, three gaits have been computed: a periodic gait that does not account for terrain variation and two that do. These will be denoted as Nominal, $NS_{4\text{cm}}$ and $S_{4\text{cm}}$, where the NS (not scaled) refers to the cost function (4.36) and S (scaled) refers to the cost function (4.12). In the next section, these gaits are evaluated both in simulation and experimentally.

4.3 Results

The "raw" simulation and experimental results are given here, with discussion given in Section 4.4. Videos of the experiments are available at [39].

4.3.1 Simulation Results

4.3.1.1 Control Law

The simulations are conducted with the same controller that will be used in the experiments. Because the model of the robot is imperfect, even with the initialization (4.20), the outputs (4.16) (see also (4.26)) will not remain zero. Hence, the feedback term (4.18) is used with $K_p = \left(\frac{1}{0.03}\right)^2$ and $K_d = 60$. Due to the 50:1 gear ratio of the harmonic drives, the feedforward term (4.17) is not essential and is dropped, as in [135].

The parameter update portion of the reset map (4.20) is pre-computed and interpolated using $\tau(x^+)$ of each step. The motivation for this is that $\tau(x^+)$ can be calculated using MARLO's link positions alone, which is helpful for actual experiments with non-instantaneous impacts and potentially noisy velocity measurements. When generating the Bézier table for a given controller, disturbance height has a one-to-one correspondence with τ_j^+. This bijection exists because the Bézier table is generated using a hybrid zero dynamic model that assumes no output errors. An example Bézier table is provided in Table 4.1. Bézier parameter columns $\alpha_{i,2}$-$\alpha_{i,5} \in \beta$ remain constant for all disturbances.

4.3.1.2 Terrain and Results

Two types of terrain profiles were generated, stepped and sloped, as shown in Figure 4.1. Step-terrain profiles consist of one vertical displacement per step, modeled as an i.i.d. uniform random variables with -4 cm $\leq d \leq 4$ cm. Fifty such terrains were generated, each with a length of 10,000 steps. The sloped terrain is meant to more closely approximate real ground variation. It uses an additional i.i.d. uniform random variables to determine the horizontal intervals between vertical displacements. Because the average step length of the three periodic gaits was approximately 0.5 m,

Table 4.1: Bézier table for $S_{4\text{cm}}$ with 4 cm disturbance maximum.

Bézier Table								
Disturbance d_j (cm)	τ_j^+	i	$\alpha_{i,0}$	$\alpha_{i,1}$	$\alpha_{i,2}$	$\alpha_{i,3}$	$\alpha_{i,4}$	$\alpha_{i,5}$
4	-0.1127	1	3.5755	3.4642	3.3935	3.3794	3.2324	3.0915
		2	3.0467	2.8714	2.8780	4.0221	3.7405	3.5699
		3	0.6345	0.6990	0.5961	0.4828	0.3911	0.5510
		4	0.5853	0.7724	1.0444	0.8554	0.6969	0.4877
2	-0.0565	1	3.5697	3.4478	3.3935	3.3794	3.2324	3.0915
		2	3.0730	2.9106	2.8780	4.0221	3.7405	3.5699
		3	0.5707	0.6882	0.5961	0.4828	0.3911	0.5510
		4	0.5691	0.7528	1.0444	0.8554	0.6969	0.4877
0	0	1	3.5699	3.4319	3.3935	3.3794	3.2324	3.0915
		2	3.0915	2.9141	2.8780	4.0221	3.7405	3.5699
		3	0.4877	0.6756	0.5961	0.4828	0.3911	0.5510
		4	0.5510	0.7568	1.0444	0.8554	0.6969	0.4877
-2	0.0485	1	3.5701	3.4253	3.3935	3.3794	3.2324	3.0915
		2	3.1175	2.8604	2.8780	4.0221	3.7405	3.5699
		3	0.3819	0.6681	0.5961	0.4828	0.3911	0.5510
		4	0.5133	0.8366	1.0444	0.8554	0.6969	0.4877
-4	0.0755	1	3.5455	3.4658	3.3935	3.3794	3.2324	3.0915
		2	3.1542	2.7483	2.8780	4.0221	3.7405	3.5699
		3	0.2547	0.7027	0.5961	0.4828	0.3911	0.5510
		4	0.4149	1.1169	1.0444	0.8554	0.6969	0.4877

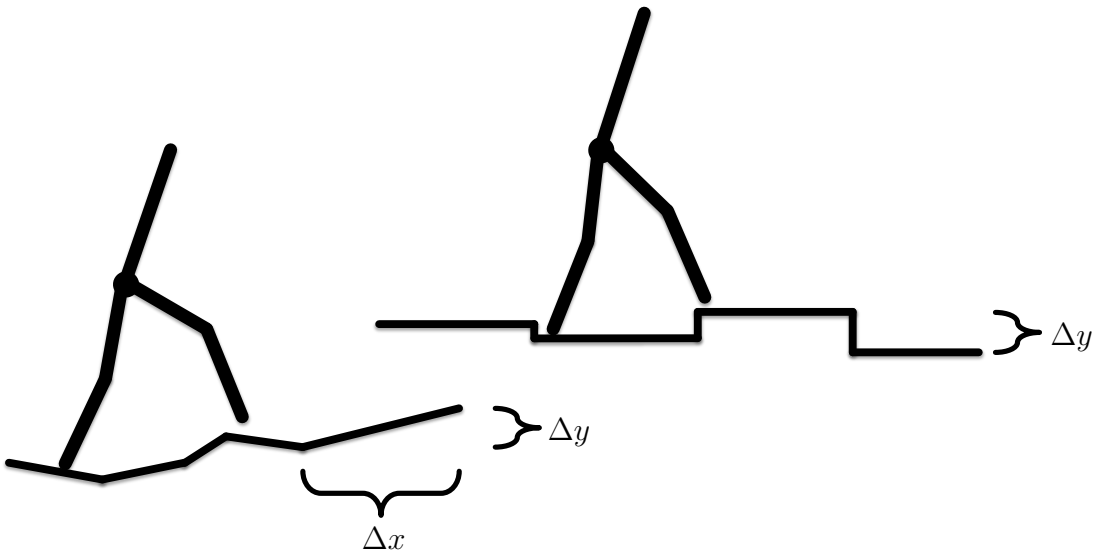

Figure 4.1: Sloped (bottom left) and step (top right) terrain.

Table 4.2: Simulation results for all control solutions.

Control	Number of Steps					Finished Trials
	Mean[a]	Med.	Min.	Max.	Variation Coef.	
4 cm Step Terrain						
Nominal	192	112	6	1157	1.22	0
$NS_{4\text{cm}}$	85	67	5	404	0.96	0
$S_{4\text{cm}}$	4616	4499	165	10^4	0.73	4
4 cm Sloped Terrain						
Nominal	481	331	17	2224	0.93	0
$NS_{4\text{cm}}$	218	178	15	740	0.79	0
$S_{4\text{cm}}$	4543	3335	40	10^4	0.71	8

[a] $S_{4\text{cm}}$ mean could be higher, but trials were limited to 10,000 steps.

the horizontal intervals are chosen uniformly between 0.25 m and 0.75 m. Because the intervals between height changes are random, it is possible to have more than one vertical displacement in the span of a single walking step. As a result, the sloped-terrain profile admits disturbances that exceed 4 cm over a single step. The 4 cm sloped terrain is representative of potential disturbances found on sidewalks, driveways, and parking lots. Fifty sloped terrains were generated, each long enough that at least 10,000 steps would be possible.

Each of the three gaits was evaluated over each of the 100 terrain profiles, 50 stepped and 50 sloped. A simulation over a given terrain profile was initiated at the gait's fixed-point and terminated when the robot reached 10,000 steps or fell. A fall could occur from losing momentum and falling backward or slipping after violating ground contact constraints. The results of these simulations are summarized in Table 4.2. An additional set of simulations over terrain with periodic, constant stepped height changes was performed and the cost function $\mathcal{J}_0$ in (4.35) was evaluated. The results are in Table 4.3.

Table 4.3: Cost function $\mathcal{J}_0$ evaluated on periodic terrain with constant step height changes.

Constant Step Disturbance	Periodic Efficiency (J/m) [a]		
	Nominal	$NS_{4\text{cm}}$	$S_{4\text{cm}}$
3 cm	Unstable	Unstable	58.5
2 cm	41.2	42.4	55.7
0 cm	35.6	39.6	52.5
-2 cm	35.2	37.8	48.4
-4 cm	42.3	36.0	39.7

[a]Periodic efficiency is calculated using (4.35).

4.3.2 Robot Experiments

4.3.2.1 Experiment Setup

The robot MARLO with point feet is attached to a 2.4 m boom to impose a planar gait. The center of the boom is mounted near a wall of the laboratory, and hence the maximum distance of an experiment is 7.5 m. Because the robot is walking in a circle, the outside leg travels a longer distance than the inside leg. To partially compensate for this, in the last 25% of a gait, the lateral hip angles are commanded to move the feet toward the center of the robot; to avoid leg collisions, the legs are moved outward toward the middle of the gait.

A terrain of variable height is constructed by stacking sections of plywood that are approximately 61 cm long [99, Fig. 25]. The plywood terrain is then overlaid with rubber mats to increase friction. Each experiment is initiated from a static pose with the robot's center of mass a few millimeters in front of the stance leg. Each terrain begins with a few steps downward so that the transfer of potential energy to kinetic energy will cause the robot to quickly transition from zero velocity to approximately its velocity on the periodic orbit.

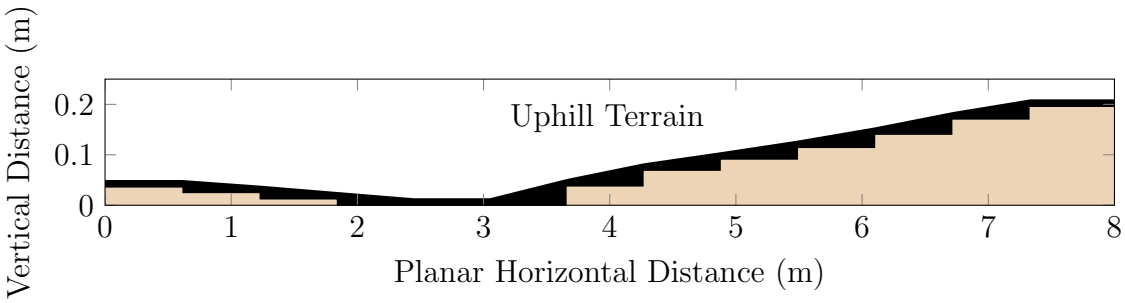

Figure 4.2: Planar view of experimental terrain used for comparing control. Stair heights are accurate (brown), while the rubber mat heights are approximate (black).

Table 4.4: Experimental results for all control solutions. Three consecutive trials are performed for each controller on the Uphill Terrain depicted in Figure 4.2. $S_{4\text{cm}}$ is the only controller to complete the terrain on all three trials.

Control	Number of Steps			Stalls	Falls	Finished Trials
	Trial 1	Trial 2	Trial 3			
Uphill Terrain						
$S_{4\text{cm}}$	12	12	12	0	0	3
Nominal	12	12	7	1	0	2
$NS_{4\text{cm}}$	7	6	7	2	1	0

4.3.2.2 Experiment Results

When a fall occurred in the simulations, it was only after consecutive uphill steps. We thus set up an uphill terrain, shown in Figure 4.2, to compare the Nominal, $NS_{4\text{cm}}$, and $S_{4\text{cm}}$ gaits. Each of the three gaits was executed over the uphill terrain three times for nine total trials, as shown in Table 4.4. The $S_{4\text{cm}}$ controller was able to complete all three trials with a consistent walking speed, as shown in Figure 4.3. The Nominal controller was able to complete the terrain course twice, but stalled[2] during one trial. The $NS_{4\text{cm}}$ controller was not able to complete the terrain on any trials due to stalling on two trials and falling on one.

In the next set of experiments, the $S_{4\text{cm}}$ gait was further evaluated over the terrains illustrated in Figure 4.4. These trials demonstrate successful walking over a variety of modest terrain disturbances. Each terrain was completed for two consecutive

[2]A stall occurs when the robot lacks adequate momentum to complete a step, and thus settles backward onto the previous stance leg rather than transition to the next step.

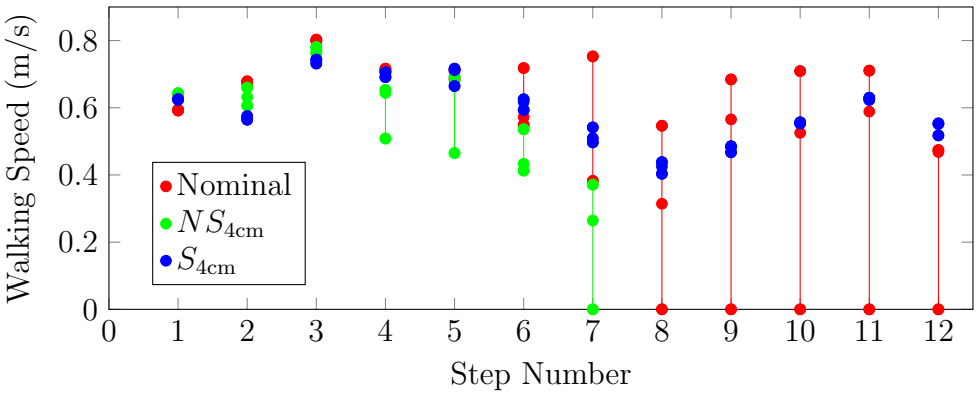

Figure 4.3: Walking speed vs. step number for Uphill Terrain experiments. Lines connect data for each controller from maximum to minimum speeds. Data points for incomplete steps are set at zero.

Table 4.5: Height changes between wood steps for experimental terrains. Distance between steps is approximately 61 cm.

Terrain	Height Changes (cm)
Uphill	-1.1, -1.25, -1.25, 0, 0, 3.8, 3.1, 2.2, 2.3, 2.6, 3, 2.5
Hill	-2.4, -2.9, -4.1, 4.1, 3.6, 4.2, -4, -2.1, -2.8, -1.5, -2.5, -2.5
Valley	-4.3, -3.4, -4.3, -2.5, -2.5, 0, 0, 3.7, 4.1, 3.6, 2.5, 3.8
Mogul	-2.5, -2.9, -4.1, 4, 3.6, -2.4, -3.9, -2.3, 4, 3.9, 2.5

trials. The results are documented in the video [39]. Height changes for experimental terrains are given in Table 4.5.

4.4 Discussion

Table 4.6 presents the minimum angular momentum about the stance leg over the step following a terrain disturbance of height $d_i \in D$ as well as the corresponding impact losses. Each controller stabilizes forward walking speed in part with lower impact losses for increasing terrain height and much greater impact losses when stepping to lower terrain. This helps to offset the velocity changes attendant with increasing or decreasing potential energy when walking uphill or downhill. The S_{4cm} gait maintains on average greater angular momentum at peak potential energy than the other gaits. Furthermore, with a single 4 cm disturbance the minimum angular momentum

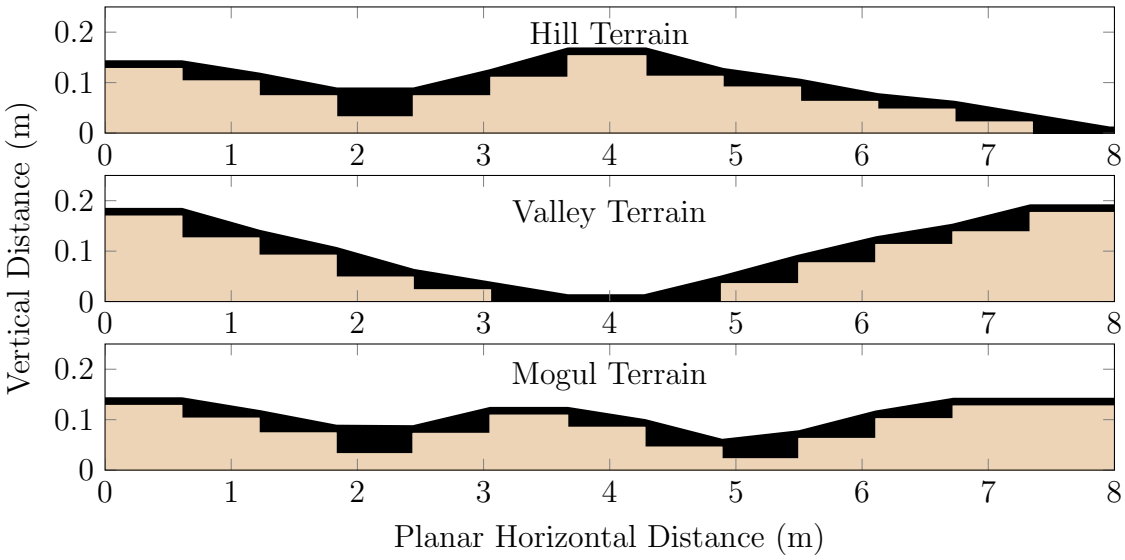

Figure 4.4: Additional experimental terrain for $S_{4\text{cm}}$.

Table 4.6: Minimum angular momentum about the stance foot and impact losses for perturbed steps in optimization.

	Minimum Angular Momentum (Nms)				Impact Losses (J)			
Control	-4 cm	0 cm	4 cm	Rng.	-4 cm	0 cm	4 cm	Rng.
Nominal	53.0	54.4	40.5	13.9	50.4	17.8	8.3	42.1
$NS_{4\text{cm}}$	52.3	54.7	38.3	16.4	47.4	18.8	10.1	37.3
$S_{4\text{cm}}$	54.6	53.5	43.8	10.8	39.2	24.6	13.9	25.3

of the Nominal and $NS_{4\text{cm}}$ gaits decreases 26% and 30% respectively, while the $S_{4\text{cm}}$ gait decreases 18%. In simulation, we found falling backward after losing momentum from repeated uphill disturbances to be the only failure mode. Having a more reliable reserve of angular momentum explains in part why the $S_{4\text{cm}}$ gait was able to outperform the other gaits in simulation and experiments (see Tables 4.2 and 4.4).

Because consecutive uphill steps led to every fall in the simulations, the initial experiments focused on comparing all controllers on the uphill terrain of Figure 4.2. The Nominal and $NS_{4\text{cm}}$ gaits were unreliable. The lower and less consistent swing foot trajectories in Figure 4.5 are prone to scuffing and premature impacts, both of which inhibit a consistent forward velocity (see Figure 4.3). The $S_{4\text{cm}}$ gait exhibited

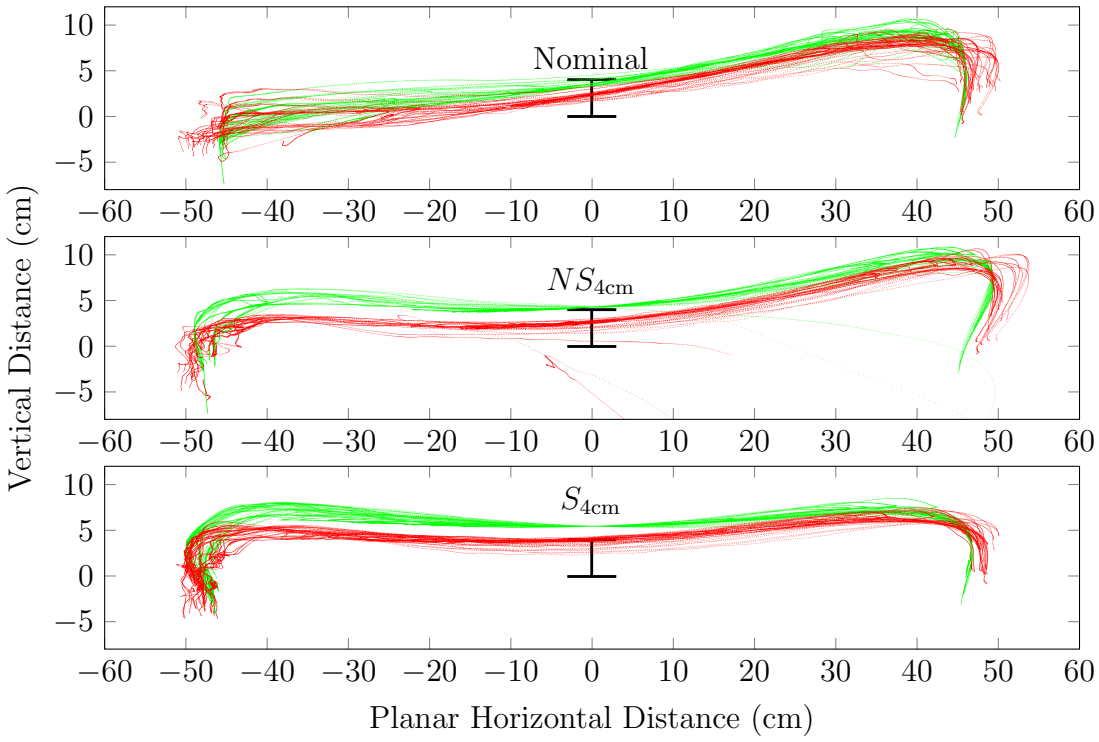

Figure 4.5: Actual (red) and desired (green) swing foot trajectories relative to stance foot. Top black bar on each plot indicates 4 cm above the stance foot position for each trajectory. Plots generated using all data from Uphill Terrain experiments, hence, the swing foot can start below the stance foot for some trajectories.

much better disturbance attenuation in terms of swing foot trajectory and speed. However, a consequence of the S_{4cm} gait's higher swing foot clearance is a higher impact loss on flat ground (see Table 4.6).

Table 4.3 shows that the Nominal gait is the most energy efficient for flat terrain. However, with some disturbances the other two gaits are more energy efficient than the Nominal gait. Hence, the energy advantages of the Nominal gait are dependent on avoiding terrain disturbances, which may be inconsistent with outdoor operation. As emphasized in [120], efficiency may be out-weighed by robustness.

Given that the NS_{4cm} gait is optimized explicitly for terrain disturbances, it may seem surprising that the Nominal gait is more reliable on uneven terrain than the NS_{4cm} gait. We postulate that this is not a general phenomenon.

Overall, the $S_{4\text{cm}}$ gait outperformed the $NS_{4\text{cm}}$ gait. We believe that allowing the optimizer to accept actions in the beginning of the step that result in smaller errors later in the step, near the moment of impact, is the main reason for this. The difference between the $S_{4\text{cm}}$ and $NS_{4\text{cm}}$ optimizations was the use of scaling variables in the $S_{4\text{cm}}$ to emphasize end-of-step errors. This gait was shown to work well in a variety of environments.

CHAPTER V

Nonholonomic Virtual Constraints

Virtual constraints are functional relations (*i.e., constraints*) on the state variables of a robot's model that are achieved through the action of actuators and feedback control instead of physical contact forces. They are called *virtual* because they can be re-programmed on the fly without modifying any physical connections among the links of the robot or its environment. Previous analytical and experimental work has established that vector relative degree two virtual *holonomic* (i.e., only configuration dependent) constraints are a powerful means to synchronize the links of a bipedal robot so as to achieve walking and running motions over a variety of terrain profiles. This chapter introduces a class of virtual *nonholonomic* constraints that depend on velocity through (generalized) angular momentum, while maintaining the property of being relative degree two. This additional freedom is shown to yield control solutions that handle a wider range of gait perturbations arising from terrain variations and exogenous forces. Moreover, including angular momentum in the virtual constraints allows foot placement control to be rigorously designed on the basis of the full dynamic model of the biped, instead of on the basis of an inverted pendulum approximation of its center of mass, as is commonly done in the bipedal robotics literature.

This new class of control laws is shown in simulation to be robust to a variety of common gait disturbances. Two feedback controllers are studied that use non-

holonomic virtual constraints. The first is an application of the *Optimization for Accommodation of Unknown Terrain Disturbances* method presented in Chapter IV applied to nonholonomic constraints. The second is an implementation of swing foot placement [110] that accounts for the full dynamics of the biped, as well as a range of terrain disturbances.

Parts of this work can be found in [40]. A video demonstration of this work is available at [41].

5.1 Relative Degree Two Nonholonomic Outputs

Assume an n-degree of freedom mechanical model

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu, \tag{5.1}$$

with m actuators and Lagrangian

$$\mathcal{L}(q,\dot{q}) := \frac{1}{2}\dot{q}^\top D(q)\dot{q} - V(q). \tag{5.2}$$

Assume moreover that the configuration variables $q = (q_u, q_a)'$ have been selected such that $q_u = (q_1, \cdots, q_{(n-m)})'$ are unactuated and $q_a = (q_{(n-m+1)}, \cdots, q_n)'$ are actuated, so that, by Lagrange's equation,

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_u} - \frac{\partial \mathcal{L}}{\partial q_u} = 0. \tag{5.3}$$

The quantity

$$\sigma := \frac{\partial \mathcal{L}}{\partial \dot{q}_u}(q,\dot{q}) \tag{5.4}$$

is the *momenta conjugate to* q_u, and for $1 \leq i \leq (n-m)$, is equal to

$$\sigma_i = D_i(q)\dot{q}, \tag{5.5}$$

where $D_i(q)$ is the i-th row of the mass-inertia matrix. From (5.3) and (5.4),

$$\frac{d}{dt}\sigma = \frac{\partial \mathcal{L}}{\partial q_u}(q, \dot{q}), \tag{5.6}$$

and thus if σ has a relative degree, it is two or greater. Indeed, differentiating σ a second time gives terms that depend on acceleration, which, via (5.1), may in turn depend on the input torque.

Functional relations involving *momenta* are classic examples of *nonholonomic constraints* [12]. Consider now a nonholonomic output function of the form

$$y = h(q, \sigma) \tag{5.7}$$

$$=: \tilde{h}(q, \dot{q}). \tag{5.8}$$

Then from the chain rule, its derivative along trajectories of the model is

$$\begin{aligned} \dot{y} = & \frac{\partial h(q,\sigma)}{\partial q}\dot{q} + \frac{\partial h(q,\sigma)}{\partial \sigma}\dot{\sigma} \\ = & \frac{\partial h(q,\sigma)}{\partial q}\dot{q} + \frac{\partial h(q,\sigma)}{\partial \sigma}\frac{\partial \mathcal{L}}{\partial q_u}(q, \dot{q}) \end{aligned} \tag{5.9}$$

and thus the relative degree cannot be less than two.

Remark: Equation (5.9) holds for one or more degrees of underactuation. Thus, it can be applied to both planar and 3D biped models, as well as models with or without compliant elements.

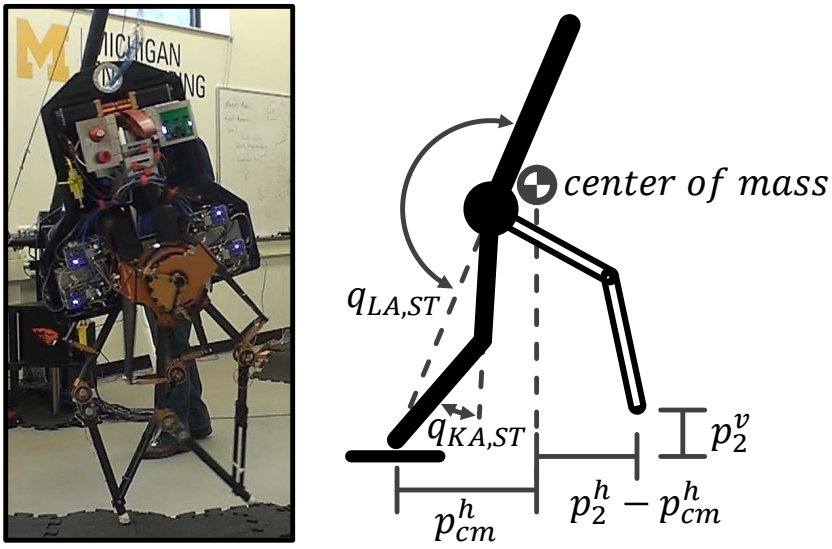

Figure 5.1: Robot MARLO and state description for planar model of MARLO used for simulation and control design.

5.2 Control Design

This section details the controller's design using the nonholonomic virtual constraints presented in Section 5.1.

5.2.1 Configuration Variables

Control is based on the rigid five-link model of MARLO presented in Section 3.2, which has 5-DOF when in single support and four actuators. The configuration variables we use are depicted in Figure 5.1. Specifically,

$$q_u = p_{cm}^h, \tag{5.10}$$

where p^h_{cm} is the horizontal position of the center of mass relative to the stance foot and

$$q_a = \begin{bmatrix} q_{LA,ST} \\ q_{KA,ST} \\ p^h_2 - p^h_{cm} \\ p^v_2 \end{bmatrix}. \tag{5.11}$$

LA and KA are abbreviations of leg angle and knee angle, ST designates the stance leg, and p^h_2 and p^v_2 are the horizontal and vertical positions of the swing foot relative to the stance foot. *With this choice of configuration variables, σ is the angular momentum about the stance foot end.* The Lagrangian model for single support and the impact model, (3.9) and (3.20), use $q = (q_u; q_a)$, $x = (q; \dot{q}) \in \mathcal{X}$ an open subset of $\mathbb{R}^{10}$, and $u \in \mathbb{R}^4$ for one degree of underactuation during single support.

5.2.2 Family of Feedback Controllers

The feedback controller is designed using the method of *virtual constraints* and *hybrid zero dynamics* (see Sections 3.3 and 3.4, [46, 134]). For planar MARLO, four virtual constraints are defined, one for each available actuator. The output vector y is defined in terms of the configuration variables, q, angular momentum, σ, and a set of parameters κ and β,

$$y = h(q, \sigma, \kappa, \beta), \tag{5.12}$$

in such a way that the output has vector relative degree 2 [64, pp. 220] on a subset of interest, $\mathcal{X} \times \mathcal{K} \times \mathcal{B}$. The parameters κ are used to achieve invariance of the *zero dynamics* manifold induced by (5.12), while the parameters β will be tuned through optimization to achieve a desirable periodic orbit. Section 5.2.2.1 provides details for calculating κ.

The feedback controller is based on input-output linearization, namely

$$u_{ff}(q,\dot{q},\kappa,\beta) := -\big[L_gL_fh(q,\dot{q},\kappa,\beta)\big]^{-1}L_f^2h(q,\dot{q},\kappa,\beta), \tag{5.13}$$

$$u_{fb}(q,\dot{q},\kappa,\beta) := -\big[L_gL_fh(q,\dot{q},\kappa,\beta)\big]^{-1}\big(K_py + K_d\dot{y}\big), \tag{5.14}$$

and[1] $u = \Gamma(q,\dot{q},\kappa,\beta) := u_{ff}(q,\dot{q},\kappa,\beta) + u_{fb}(q,\dot{q},\kappa,\beta)$ (4.19). Along solutions of the closed-loop system $\ddot{y} + K_d\dot{y} + K_py \equiv 0$ (3.43).

An explicit choice of $h(q,\sigma,\kappa,\beta)$ is now made,

$$h(q,\sigma,\kappa,\beta) = h_0(q) - h_d(\tau(q),\sigma,\kappa,\beta), \tag{5.15}$$

$$= h_0(q) - \left[h_{d,\sigma}(\sigma,\beta) + h_{d,\tau}(\tau(q),\kappa,\beta)\right], \tag{5.16}$$

where $h_d(\tau(q),\sigma,\kappa,\beta)$ specifies the desired evolution of $h_0(q)$ and

$$h_0(q) = q_a, \tag{5.17}$$

$$h_{d,\sigma}(\sigma,\beta) = [0,0,k_1(\beta)\sigma + k_2(\beta)\sigma^2,0]'. \tag{5.18}$$

The inclusion of angular momentum in the third component of $h_{d,\sigma}$ allows step length to vary with velocity. The function $h_{d,\tau}(\tau(q),\kappa,\beta) \in \mathbb{R}^4$ is a vector of splines that specifies the desired evolution of defined $h_0(q) - h_{d,\sigma}(\sigma,\beta)$ in terms of the gait phasing variable $\tau(q)$. Here, the splines are Bézier polynomials, with the ith polynomial given by

$$h_{d,\tau,i}(\tau,\kappa,\beta) := \sum_{k=0}^{M}\alpha_{i,k}\frac{M\,!}{k\,!(M-k)\,!}\tau^k(1-\tau)^{M-k}, \tag{5.19}$$

where the four degree-$(M+1)$ Bézier polynomials are defined by $\alpha(\kappa,\beta) \in \mathbb{R}^{4\times(M+1)}$ [136, pp. 138]. The gait phasing variable $\tau(q)$ is selected to be an affine function of

[1] A procedure for transforming a Lagrangian system with feedback control into a control-free Lagrangian system with a new class of trajectories is available in [126].

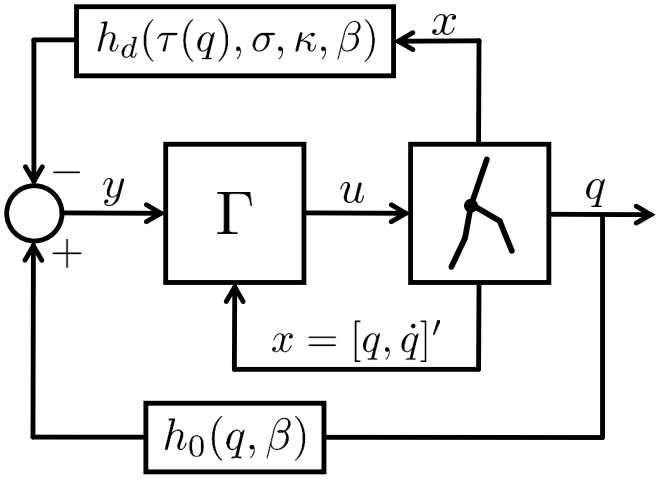

Figure 5.2: Nonholonomic virtual constraint control schematic.

p^h_{cm} and is normalized on the periodic orbit to take values in $[0, 1]$.

The complete output equation using (5.12) and (5.16)-(5.19) is

$$y = \begin{bmatrix} q_{LA,ST} \\ q_{KA,ST} \\ p^h_2 - p^h_{cm} \\ p^v_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ k_1\sigma + k_2\sigma^2 \\ 0 \end{bmatrix} - h_{d,\tau}(\tau(q), \kappa, \beta). \tag{5.20}$$

In the optimization phase, values for k_1 and k_2 will be chosen such that a perturbation in velocity, and attendant deviation of σ, results in a corrective change in swing foot placement. Specifically, this will adjust the amount of time the center of mass spends behind the stance foot, versus in front of the stance foot, and will enable quicker convergence to the periodic orbit [69, 77, 110]. The complete output control schematic is shown in Figure 5.2.

5.2.2.1 Extended Model for Invariant Hybrid Zero Dynamics

Parameters κ are used to maintain *hybrid zero dynamics* following impact deviations. With output (5.20), it is straightforward to construct an equivalent function

$\Psi : \mathcal{S}^d \times \mathcal{B} \to \mathcal{K}$ such that for all

$$\beta \in \mathcal{B} \text{ and } \begin{bmatrix} q^+ \\ \dot{q}^+ \end{bmatrix} = \Delta(q^-, \dot{q}^-)$$

the initial values of the outputs are zeroed, that is,

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} y^+ \\ \dot{y}^+ \end{bmatrix} = \begin{bmatrix} h(q^+, \sigma^+, \kappa^+, \beta) \\ \frac{\partial}{\partial q} h(q^+, \sigma^+, \kappa^+, \beta)\dot{q}^+ + \frac{\partial}{\partial \sigma} h(q^+, \sigma^+, \kappa^+, \beta)\dot{\sigma}^+ \end{bmatrix} \tag{5.21}$$

for $\kappa^+ = \Psi(q^-, \dot{q}^-, \beta)$. The current implementation of κ^+ is derived in Section 5.2.3.

Parameters κ are constant within each step and are reset at the end of each step, hence, they are included as states in the dynamics as derived in Section 4.2.2.

5.2.3 Bézier Parameter Reset Derivation for Nonholonomic Virtual Constraints

Control parameters κ must be reset such that post-impact outputs are zeroed in (5.21). Using (5.12), (5.15), (5.16), and (5.19), output terms dependent on κ are defined as

$$h_\kappa(\tau(q), \alpha(\kappa, \beta)) := h_{d,\tau}(\tau(q), \kappa, \beta) \tag{5.22}$$

and output terms independent of κ are defined as

$$h_\beta(q, \sigma, \beta) := h_0(q) - h_{d,\sigma}(\sigma, \beta). \tag{5.23}$$

Next, h_κ and h_β substitute into (5.12) as

$$y = h_\beta(q, \sigma, \beta) - h_\kappa(\tau(q), \alpha(\kappa, \beta)). \tag{5.24}$$

From (5.24) and (5.19), we find that the desired trajectory of h_β along τ is specified by Bézier parameters
$\alpha(\kappa, \beta) \in \mathbb{R}^{6\times(M+1)}$, which are defined as

$$\alpha(\kappa, \beta) := \Big[\alpha_0(\kappa), \alpha_1(\kappa), \alpha_2(\beta), \ldots, \alpha_M(\beta)\Big]. \tag{5.25}$$

It is evident from (5.19) that $\alpha_0(\kappa)$ and $\alpha_1(\kappa)$ have the most effect on trajectories during low τ values immediately after impact. The remaining columns of $\alpha(\kappa, \beta)$, defined by fixed parameters β, determine trajectories toward the end of the gait. Hence, perturbed trajectories return to the nominal gait as τ increases.

Let $y^+ = \dot{y}^+ = 0$ as in (5.21). Using (5.24), this implies that

$$h_\kappa(\tau(q^+), \alpha(\kappa^+, \beta)) = h_\beta(q^+, \sigma^+, \beta), \tag{5.26}$$

or simply $h_\kappa(\tau^+, \kappa^+, \beta) = h_\beta^+$. To satisfy (5.26), we must reset at least one column of Bézier parameters, $\alpha_0(\kappa^+)$. To guarantee desired trajectories match post-impact velocities, we reset a second column, $\alpha_1(\kappa^+)$, to satisfy

$$\frac{\partial h_\kappa(\tau^+, \kappa^+, \beta)}{\partial \tau}\dot{\tau}^+ = \dot{h}_\beta^+. \tag{5.27}$$

Using (5.22), (5.19), and (5.25), we solve for $\alpha_0(\kappa^+)$ and $\alpha_1(\kappa^+)$ in (5.26) and (5.27) as

$$\alpha_0(\kappa^+) = \frac{h_\beta^+ - \sum_{k=1}^{M} \alpha_k \frac{M\,!(\tau^+)^k(1-\tau^+)^{M-k}}{k\,!(M-k)\,!}}{(1-\tau^+)^M} \tag{5.28}$$

$$\alpha_1(\kappa^+) = \frac{\frac{\dot{h}_\beta^+}{\dot{\tau}^+} - \alpha_2 M(M-1)\tau^+(1-\tau^+)^{M-2} - a + b}{M((1-\tau^+)^{M-1} + \tau^+(1-\tau^+)^{M-2})}, \tag{5.29}$$

where $a = \sum_{k=2}^{M-1}(\alpha_{k+1} - \alpha_k)\frac{M\,!(\tau^+)^k(1-\tau^+)^{M-1-k}}{k\,!(M-1-k)\,!}$ and

$b = \frac{M}{1-\tau^+}\left(h_\beta^+ - \sum_{k=2}^{M} \alpha_k \frac{M\,!(\tau^+)^k(1-\tau^+)^{M-k}}{k\,!(M-k)\,!}\right)$.
(5.28) and (5.29) are a solution for $\kappa^+ = \Psi(q^-, \dot{q}^-, \beta)$ that always satisfies (5.21).

5.2.4 Extended Gait Phasing Variable and Bézier Polynomials

A *gait phasing* variable $\tau : \mathcal{X} \to \mathbb{R}$ that is strictly increasing along walking steps is used. The gait phase can be thought of as a measure of progress through each step. Along periodic walking gaits, the coordinate p^h_{cm} shown in Figure 5.1 is monotonic and cycles between a minimum value $p^h_{cm,min}$ and a maximum value $p^h_{cm,max}$. The nominal gait phasing variable is defined as

$$\tau_{nom}(q) = \frac{p^h_{cm}(q) - p^h_{cm,min}}{p^h_{cm,max} - p^h_{cm,min}}, \tag{5.30}$$

which is a variant of (3.25). Let $x_0 = (q_0; \dot{q}_0)$ be the initial condition for a periodic orbit with ground height d_0. It follows that $\tau_{nom}(\Delta(x_0)) = 0$, $\tau_{nom}(x_0) = 1$, and $x_0 \in \mathcal{S}^{d_0}$ using (3.23).

The periodic orbit is departed, however, when $x_0 \notin \mathcal{S}^d$ for some $d < d_0$. When this occurs, $\tau_{nom}(x) > 1$ and the desired trajectory as defined by the nominal gait phasing variable and Bézier polynomials can become counterproductive. To avoid this, an alternative trajectory is defined using an extended gait phasing variable,

$$\tau_{ext}(q) = \frac{p^h_{cm}(q) - p^h_{cm,max}}{p^h_{cm,max} - p^h_{cm,min}}, \tag{5.31}$$

and a second set of Bézier polynomials, α_{ext}.

The complete τ and α used in (5.19) are defined using their nominal values and

equivalent extensions as

$$\tau(q) := \begin{cases} \tau_{nom}(q) \text{ if } p^h_{cm}(q) \leq p^h_{cm,max} \\ \tau_{ext}(q) \text{ if } p^h_{cm}(q) > p^h_{cm,max} \end{cases} \tag{5.32}$$

$$\alpha(\kappa, \beta) := \begin{cases} \alpha_{nom}(\kappa, \beta) \text{ if } p^h_{cm}(q) \leq p^h_{cm,max} \\ \alpha_{ext}(\beta) \text{ if } p^h_{cm}(q) > p^h_{cm,max} \end{cases} . \tag{5.33}$$

$\tau_{nom}(q)$, $\tau_{ext}(q)$, $\alpha_{nom}(\kappa, \beta)$, and $\alpha_{ext}(\beta)$ should be defined such that (5.19) is continuous. One way of achieving this is by defining τ_{ext} such that $\{(q, \dot{q})' \in \mathcal{X} \mid \tau(q) = 1\}$

$$\tau_{\text{ext}}(q) = 0 \tag{5.34}$$

$$\dot{\tau}_{\text{ext}}(q, \dot{q}) = \dot{\tau}(q, \dot{q}), \tag{5.35}$$

and α_{ext} such that

$$\alpha_{ext,0} = \alpha_{nom,M} \tag{5.36}$$

$$\alpha_{ext,1} = \alpha_{ext,0} + \big(\alpha_{nom,M} - \alpha_{nom,(M-1)}\big)\frac{M}{M_{ext}}, \tag{5.37}$$

where $\alpha_{ext,i}$ is the $(i+1)$ column of α_{ext} and $(M+1)$ and $(M_{ext}+1)$ are the degree of Bézier polynomials associated with α_{ext} and α_{nom} respectively. If $M \neq M_{ext}$, the order of the polynomials used in (5.19) must be updated when transitioning to the extended controller.

5.2.5 Optimization for Three Control Solutions

Three controllers are designed and subsequently tuned via parameter optimization: a controller that does not include nonholonomic virtual constraints and two controllers that do. These will be denoted as HVC, NHVC, and NHVC-SFP, where

HVC and NHVC refer to the use of holonomic and nonholonomic virtual constraints, respectively, and SFP refers to an additional objective of swing foot placement suggested by the LIPM model. The controller based on HVC serves as a comparison to work in Chapter IV. The swing foot placement policy used for the NHVC-SFP optimization is derived in Section 5.3.

To account for uneven terrain, the *Optimization for Accommodation of Unknown Disturbances* method presented in Chapter IV is used. This is a parameter optimization problem in (β, x_0), the parameters in h_d and the initial state of the robot, such that the resulting closed-loop system has a periodic solution and can also accommodate (i.e., take valid steps following) a given set of terrain disturbances. The cost function is chosen so that it favors perturbed solutions that "return closely" to the nominal periodic solution; in other words, the cost function is designed so that the closed-loop system attenuates the potentially deleterious effects of the given set of ground height variations. A key feature is that the gait phasing variable is used to penalize more heavily deviations that persist "late" into the gait.

Two primary changes have been made with respect to the implementation in Chapter IV. First, nonholonomic virtual constraints are incorporated into the outputs using (5.18). Second, two steps following a terrain disturbance are included in the cost so that the effect of swing foot placement at the end of the first step after the perturbation is captured during the second step.

Our optimization cost to penalize deviations is induced by 4 terrain height disturbances in $D = \{\pm 4 \text{ cm}, \pm 8 \text{ cm}\}$. The nominal periodic solution corresponds to a terrain height of 0 cm. For perturbed steps $1 \leq j \leq 8$, the deviation costs are defined as $\mathcal{J}_j = \frac{1}{(\tau_j^- - \tau_j^+)} \int_{\tau_j^+}^{\tau_j^-} \left(||\delta x_j(\tau)||^2 + ||\delta u_j(\tau)||^2 \right) \frac{(\tau - \tau_j^+)}{(\tau_j^- - \tau_j^+)} d\tau$ (4.12). $\delta x_j(\tau)$ and $\delta u_j(\tau)$ are the differences in perturbed state and control trajectories from the closest existing periodic trajectories characterized by τ (see (4.8) and (4.9)). The term $\frac{(\tau - \tau_j^+)}{(\tau_j^- - \tau_j^+)}$ under the integral scales the errors so that initial deviations from the nominal periodic

trajectory are discounted with respect to errors toward the end of the step. The term $\frac{1}{(\tau_j^- - \tau_j^+)}$ outside the integral is included so that perturbed step costs are normalized w.r.t. the varying ranges of τ_j resulting from higher and lower terrain disturbances.

The overall cost function is

$$\mathcal{J} = \mathcal{J}_0 + \sum_{j=1}^{8} w_j \mathcal{J}_j, \tag{5.38}$$

where w_j determines the relative weight of each perturbation and the energy efficiency $\mathcal{J}_0$. $\mathcal{J}_0$ is calculated using step distance and mechanical actuator work of the nominal solution as in (4.35).

Parameter optimization problem: Find $(\beta; x_0)$ that (locally) minimize $\mathcal{J}$ subject to the existence of a periodic solution that respects the following constraints: motor torque is saturated at 6 Nm; vertical ground reaction force greater than 100 N and friction coefficient less than 0.6; minimum swing foot clearance of 0.1 m over stance foot; minimum knee bend of 10^o to avoid hyperextension; average walking speed between 0.6-0.8 m/s. The computations were performed offline with `fmincon` in MATLAB.

5.3 Swing Foot Placement using Nonholonomic Virtual Constraints

Figure 5.3 shows the Linear Inverted Pendulum model used to derive the foot placement strategy of [110]

$$v_{k+1}^2 = v_k^2 - \frac{g}{\ell}(r_{accel,k}^2 - r_{step,k}^2), \tag{5.39}$$

where k is the step number, v_k is the center of mass velocity when the pendulum is vertical, and $r_{step,k}$ and $r_{accel,k}$ are the horizontal distances the center of mass travels

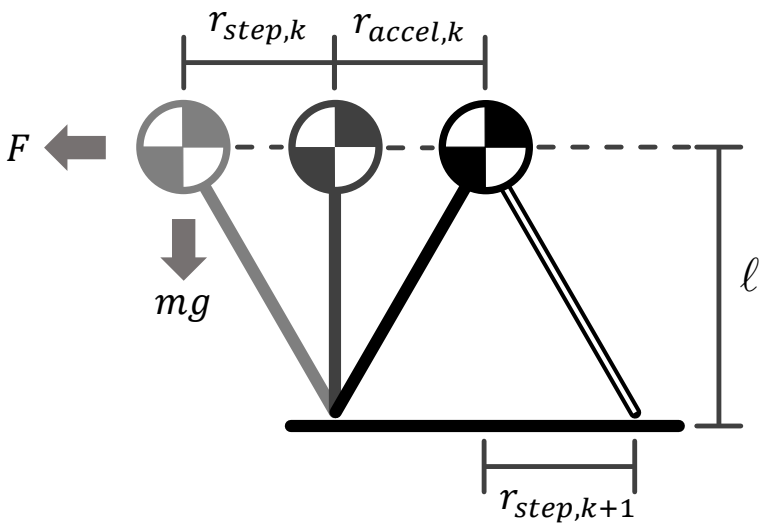

Figure 5.3: Linear Inverted Pendulum model. Horizontal force F results from the ground reaction forces caused by gravity.

from behind the stance foot to in front of the stance foot during step k, resulting in the velocity v_{k+1} in the next step.

For implementation on the full dynamic model of MARLO, adjustments are made to (5.39). First, the height of the center of mass, ℓ, is calculated in this work as the average center of mass height during the periodic orbit of the full model. Second, (5.39) does not take impact losses into account and will generally require $r_{accel} > r_{step}$ to compensate for this. Finally, because swing foot height relative to the stance foot is included in (5.20), r_{accel} does not change when walking on flat ground.

In (5.39), let v^*, r^*_{step}, and r^*_{accel} denote nominal values on a periodic orbit, so that

$$v^{*2} = v^{*2} - \frac{g}{\ell}({r^*_{accel}}^2 - {r^*_{step}}^2). \tag{5.40}$$

In (5.39), setting $r_{accel,k} = r^*_{accel}$ and $v_{k+1} = v^*$ from (5.40) gives step length to return to the nominal velocity, namely

$$r_{step,k} = \sqrt{\frac{\ell}{g}(v_k^2 - v^{*2}) + {r^*_{step}}^2}. \tag{5.41}$$

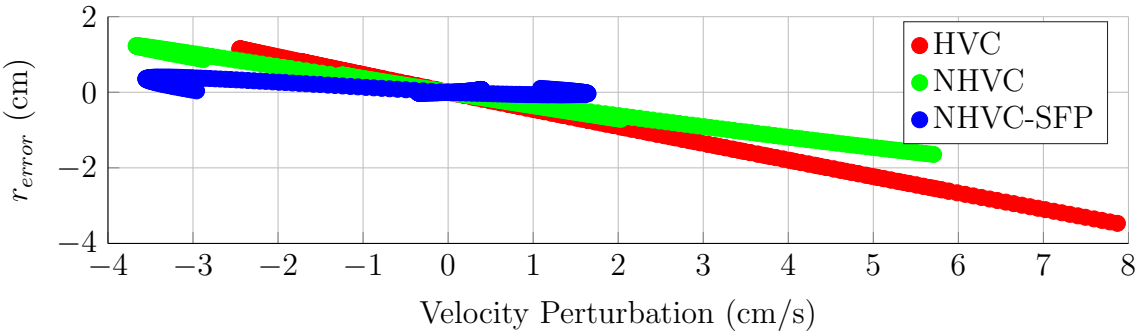

Figure 5.4: r_{error} vs. velocity perturbation. Perturbations occur during the two steps following a single terrain height disturbance $d \in [-8, 8]$ (cm).

The step-length policy (5.41) is implemented using the full model and nonholonomic virtual constraints. The variable r_{step} is equivalent to the configuration variable $p_2^h - p_{cm}^h$ at the end of a step. Since $p_2^h - p_{cm}^h$ is paired with an angular momentum-based virtual constraint in (5.20), the horizontal position of the swing foot can vary with velocity.

Define $r_{error,k}$ as the difference between the actual swing foot placement and $r_{step,k}$ from (5.41). The NHVC-SFP control solution is optimized with an additional cost, $\mathcal{J}_{SFP}$, based on $r_{error,k}$

$$\mathcal{J} = \mathcal{J}_{SFP} + \mathcal{J}_0 + \sum_{j=1}^{8} w_j \mathcal{J}_j. \tag{5.42}$$

A comparison of the error in r_{step} for the three control solutions is shown in Figure 5.4. The NHVC-SFP controller stays within 4 mm of the theoretical swing foot placement policy for the terrain disturbances used during optimization.

5.4 Results

Simulation results are presented here with discussion given in Section 5.5. The nominal periodic orbits resulting from the three control solutions are very similar as shown in Table 5.1.

Table 5.1: Periodic step velocity, impact losses, and energy efficiency on flat ground. Efficiency is calculated using $\mathcal{J}_0$ from (5.38) and (5.42).

Metric	Control		
	HVC	NHVC	NHVC-SFP
Step Velocity (m/s)	0.63	0.65	0.66
Impact Losses (J)	19.9	20.1	19.6
Energy Efficiency (J/m)	347	346	365

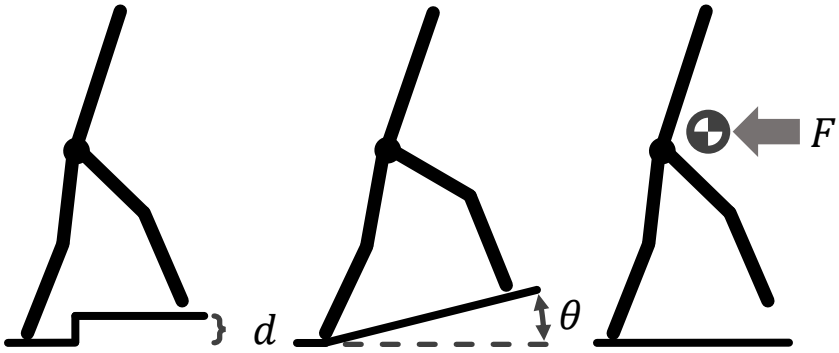

Figure 5.5: Step terrain disturbance (left), change in terrain slope (middle), and horizontal force to center of mass (right).

5.4.1 Disturbance Types

Figure 5.5 shows the three types of disturbances used to evaluate each of the control solutions. Step changes in terrain height consist of a vertical displacement of d (m) per step, as was done during optimization. A change in terrain slope of θ (deg) causes a similar disturbance as a step change in terrain height, but accounts for variations in terrain elevation with longer and shorter steps. A sloped terrain may be more representative of natural outdoor terrain. A third type of disturbance is a horizontal force F (N) applied to the center of mass over the entire duration of a step. This induces a velocity perturbation to the robot without the complication of early or late impacts that may occur with terrain disturbances.

Table 5.2: Simulation results for all control solutions.

Control	Step Disturbance (cm)			Terrain Slope (degrees)			Horizontal Force (N)		
	Min.	Max.	Rng.	Min.	Max.	Rng.	Min.	Max.	Rng.
HVC	-8.9	12.5	21.4	-12.0	28.1	40.1	-14.2	11.2	25.4
NHVC	-15.4	12.8	28.2	-17.8	35.0	52.8	-37.5	19.9	57.4
NHVC-SFP	-14.8	13.1	27.9	-16.9	37.0	53.9	-89.6	26.5	116.1

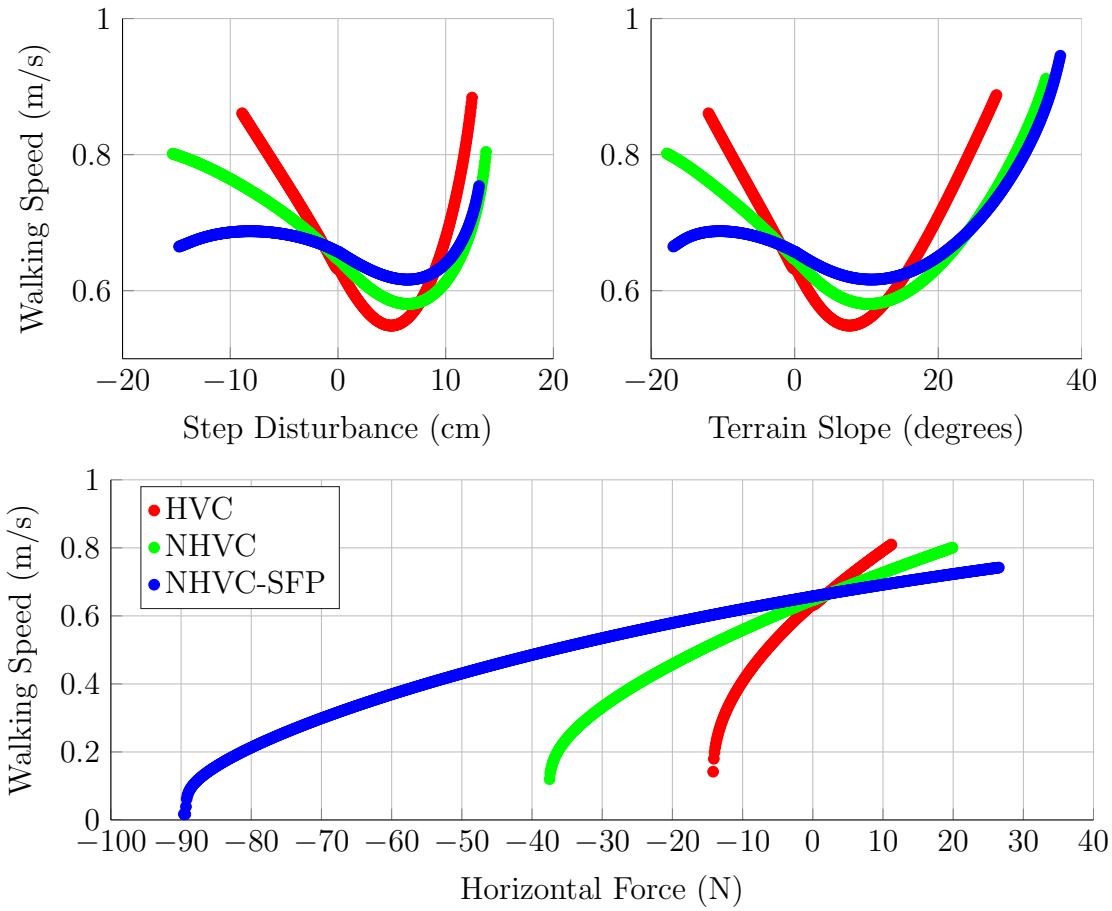

Figure 5.6: Velocity vs. step terrain disturbances (top left), changes in terrain slope (top right), and external forces on flat ground (bottom).

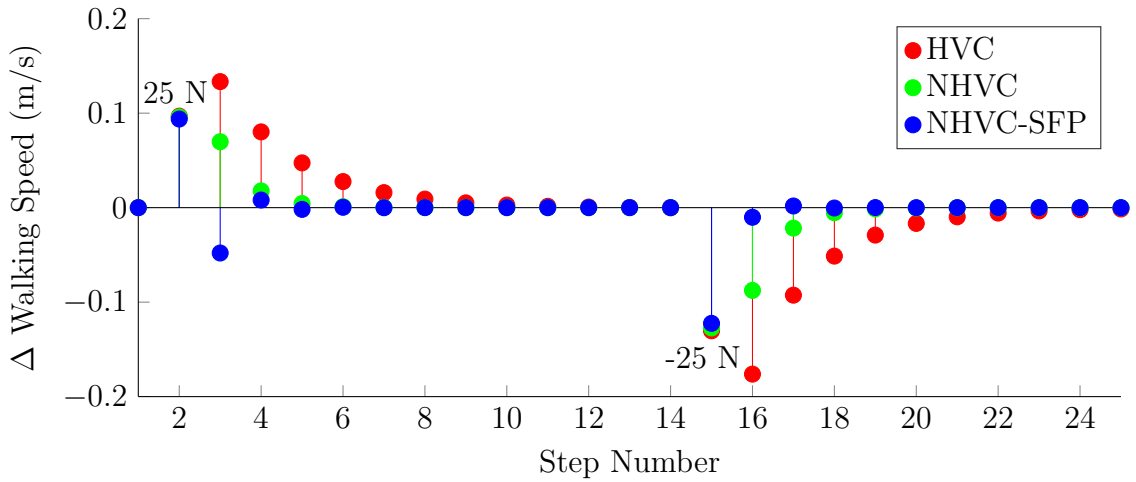

Figure 5.7: Velocity stabilization after a 25 N horizontal force applied over entire second step and a -25 N force applied over the entire fifteenth step.

5.4.2 Repeated Disturbance Limits

The three control solutions are first compared under the action of a persistent disturbance whose magnitude is gradually increased each step until the robot falls[2]. For example, with step changes in terrain, each controller is initialized on the periodic orbit, and then the terrain height change is varied as $d_{k+1} = d_k + 0.5\ mm$, where k is step number. Once a fall occurs, the simulation is reset from the periodic orbit, and a decrease of 0.5 mm is applied to d_k until failure. The same procedure is applied to terrain slopes with 0.1^o increments and to horizontal force with 0.1 N increments. The results of these simulations are summarized in Table 5.2 and the resulting perturbed velocities for each step are plotted in Figure 5.6.

5.4.3 Transient Response to Velocity Perturbations

An additional simulation is performed to evaluate the response of each controller to a velocity perturbation. The velocity perturbation is applied through a ±25 N horizontal force acting throughout an entire step, starting from the periodic orbit. The

[2]A fall can occur from losing momentum and tumbling backward, or violating ground contact constraints.

Table 5.3: Minimum angular momentum about the stance foot and impact losses for perturbed steps in optimization.

	Minimum Angular Momentum (Nms)				Impact Losses (J)			
Control	-8 cm	0 cm	8 cm	Rng.	-8 cm	0 cm	8 cm	Rng.
HVC	47.4	40.5	40.1	7.3	42.2	19.9	16.9	25.3
NHVC	46.8	41.7	39.1	7.7	46.7	20.1	15.1	31.6
NHVC-SFP	43.8	42.5	39.8	4.0	54.7	19.6	13.8	40.9

response is monitored through the resulting average velocity over the steps following the perturbation. Figure 5.7 shows the results.

5.5 Discussion

Table 5.3 presents the minimum angular momentum about the stance leg during the step following a terrain disturbance, as well as the impact losses associated with each terrain disturbance. Each controller is able to maintain a stable walking gait in part with lower impact losses for increasing terrain height and much greater impact losses when stepping to lower terrain. This helps to offset velocity changes that normally occur when increasing or decreasing potential energy walking uphill or downhill. However, NHVC-SFP is able to maintain the most consistent minimum angular momentum. This is explained in part by the larger range of impact losses, but also by a more principled swing foot placement to remove velocity perturbations.

While Table 5.1 showed that the nominal periodic gait for each controller was similar, differences emerge when testing the limits of performance as seen in Table 5.2 and Figure 5.6. The two controllers using nonholonomic constraints have a similar range on uneven terrain, but both outperform the controller based on holonomic constraints. For all three controllers, velocity initially decreases as expected on uphill disturbances, but then increases as the height change exceeds the limits used in the optimization. The speed up occurs because limited swing foot clearance leads to an early impact, which in turn results in the center of mass initializing the step in

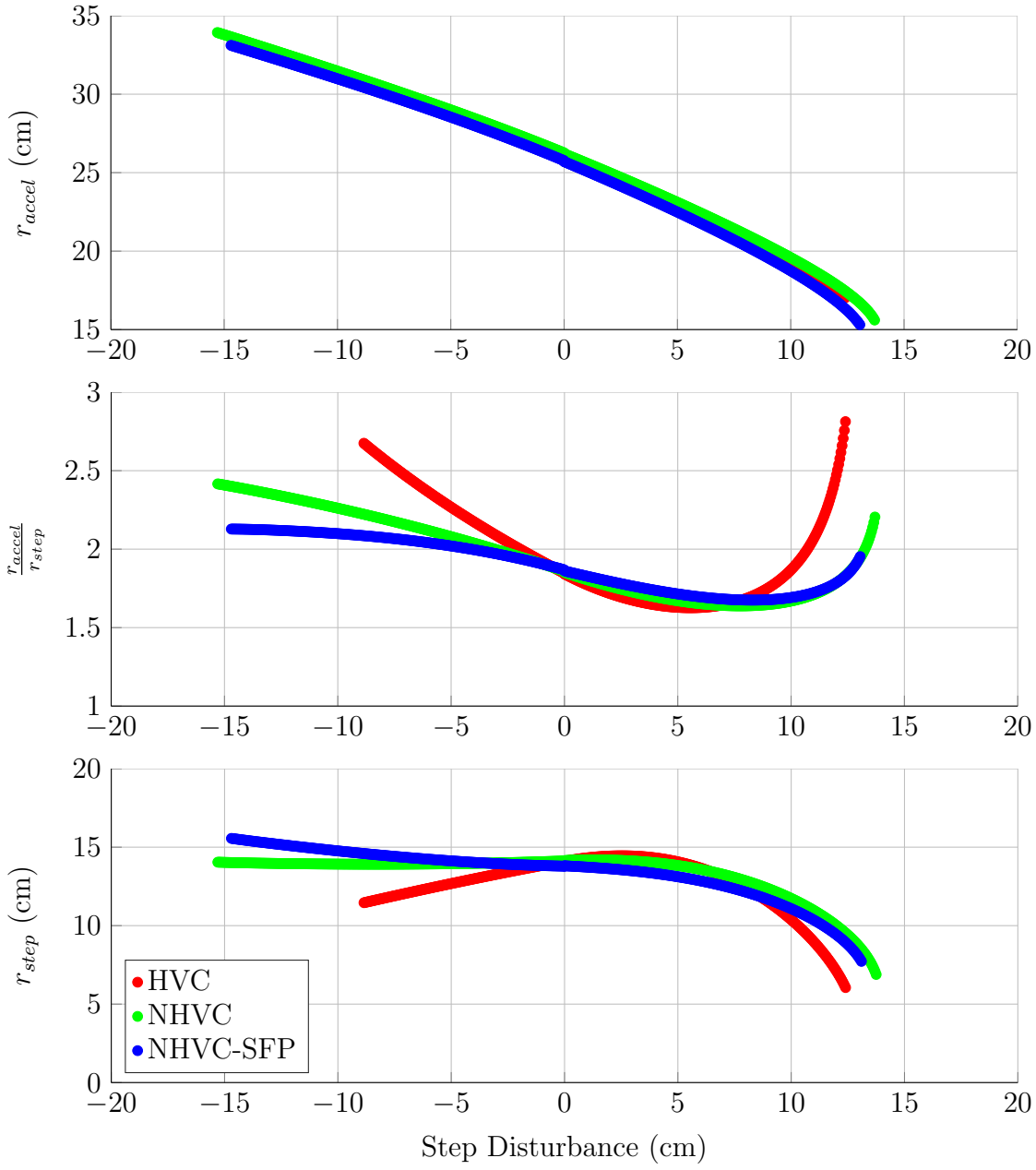

Figure 5.8: r_{accel} (top), $\frac{r_{accel}}{r_{step}}$ ratio (middle), and r_{step} (bottom) vs. step terrain disturbances.

a more forward position; this limits the deceleration period and causes an increase in velocity[3]. Figure 5.8 shows this effect to be most prominent for HVC as r_{step} values, the distance the center of mass must travel to be over the stance foot, are cut nearly in half when moving from 8 cm to 12 cm disturbances. The two controllers using nonholonomic constraints exhibit less variation because r_{step} naturally increases as velocity increases. r_{accel} is consistent between control solutions, increases with downhill walking due to late impacts, and is nominally greater than r_{step} to offset impact losses. NHVC-SFP has the greatest increase in r_{step} for negative disturbances, which leads to tighter velocity regulation in Figure 5.6 when walking downhill.

When testing the three control solutions for external forces, the NHVC-SFP controller was able to outperform the two other controllers by a fair margin, especially for negative horizontal forces. In fact, the NHVC-SFP controller can maintain a periodic gait despite negative impulses as high as 180 Ns per step. Additionally, in Figure 5.7 the NHVC-SFP controller exhibits nearly dead-beat behavior for velocity regulation following external forces.

The swing foot placement policy (5.39) is designed to reject velocity changes when walking on flat ground, and by "embedding this event-based controller" into the NHVC-SFP control solution, the resulting continuous-time swing foot placement policy is able to accommodate a wide range of disturbances. This is especially evident when the external disturbance corresponds to an extreme loss in velocity and changing r_{step} can reduce deceleration from gravity.

In this chapter, a speed control law suggested by the linear inverted pendulum [110] was embedded into the controller for a planar biped, while accounting for the full ten-dimensional hybrid model of the robot, and also accounting for unexpected terrain height changes. A key factor in the controller design was the use of nonholonomic virtual constraints. Leveraging knowledge from low-dimensional models, as

[3]Eventually the speed increases to a point where ground reaction forces are violated.

illustrated with swing foot placement, the optimization process was guided toward more successful outcomes.

CHAPTER VI

Three-Dimensional Walking in Realistic Environments

This chapter addresses the problem of designing feedback controllers that allow a three-dimensional (3D) bipedal robot to walk outdoors over sloped sidewalks, parking lots, and lawns and indoors over randomly placed planks, all without a priori knowledge of the environment or external sensing. Model-based design methods are introduced and subsequently validated in simulation and experiment on MARLO. A substantial portion of our 3D approach coincides with planar control implementations introduced in Chapters IV and V, which were, in fact, originally motivated by challenges observed during 3D walking experiments. Using a single continuously-defined controller taken directly from optimization, MARLO traverses a variety of outdoor environments while maintaining average walking speeds between 0.9-0.98 m/s and setting a new precedent for walking efficiency in realistic environments.

The remainder of this chapter is organized as follows. Section 6.1 presents an extended definition of the *Optimization for Accommodation of Unknown Disturbances* method that explicitly accounts for multiple types of disturbances. Section 6.2 details a specific 3D control implementation of the general concepts introduced throughout this dissertation. To demonstrate the efficacy of new concepts and establish best practices, Section 6.3 compares simulation results for many different control solutions.

Each control solution results from a unique design configuration selected explicitly for this purpose. Section 6.4 gives the results of robot experiments using control solutions designed for outdoor environments, with corresponding discussion given at the end of the section.

Parts of this work have been submitted for publication in [43]. Videos of outdoor experiments are available at [44]. Code used for 3D control design and simulation is available at [37]. For the convenience of the reader, an introductory guide to 3D walking concepts is given in Appendix A.

6.1 Optimization for the Accommodation of Unknown Disturbances

This section provides an extended definition of the *Optimization for Accommodation of Unknown Disturbances* method from Chapter IV that explicitly accounts for multiple types of disturbances.

6.1.1 Terrain Disturbances

Let $d_0 \in D$ represent the nominal change in ground height *step to step*. We seek $\beta \in \mathcal{B}$ and $x_0 \in \mathcal{X}$ giving rise to a periodic solution of the closed-loop system (3.21); that is, for which there exists $T_0 > 0$ such that

$$x_0 = \varphi(T_0, \Delta(x_0)). \tag{6.1}$$

Moreover, for the same value of $\beta \in \mathcal{B}$, we desire that the periodic orbit ensures the existence of the following additional solutions of the closed-loop system: $\forall\, 1 \leq i \leq N_d$,

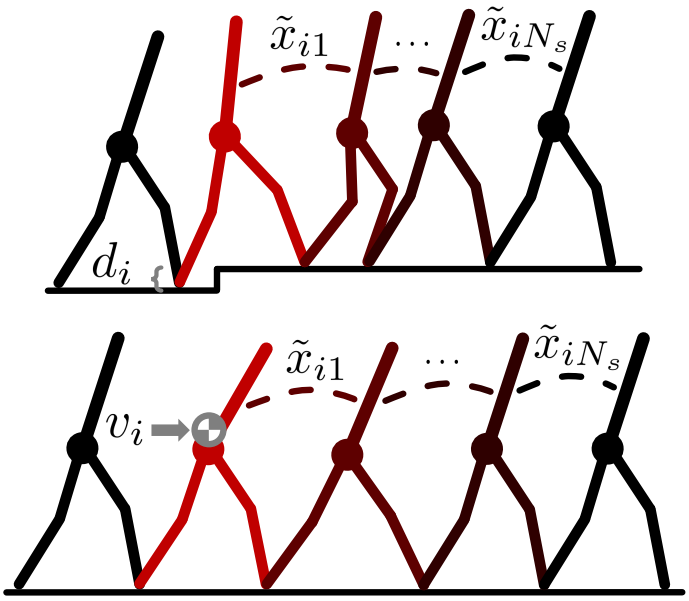

Figure 6.1: Terrain (top) and velocity (bottom) disturbances for optimization. Data are collected for N_s steps following a perturbation caused by d_i or v_i.

$d_i \in D$, $1 \leq j \leq N_s$, $\exists\ 0 < t_i < \infty$ and $0 < T_{ij} < \infty$ such that

$$x_{i1} = \varphi(t_i, \Delta(x_0)) \in \mathcal{S}^{d_i} \tag{6.2}$$

$$x_{i(j+1)} = \varphi(T_{ij}, \Delta(x_{ij})) \in \mathcal{S}^{d_0}. \tag{6.3}$$

In plain words, there exist steps that begin on the periodic orbit, end at ground height d_i, and continue for at least N_s more steps at nominal ground height d_0, as shown in Figure 6.1.

In the following, we set up a parameter optimization problem in (β, x_0) for finding a periodic solution that meets these conditions. Moreover, we will pose a cost function on the steps following the change in ground height that favors solutions that "return closely" to the nominal periodic solution, that is, the closed-loop system attenuates the effects of the set of ground height variations.

6.1.2 Velocity Disturbances

The method of Section 6.1.1 can accommodate a variety of disturbances. Here, velocity disturbances are addressed. Let $x_{v_0} \in \mathcal{X}$ represent the values of the state in the periodic orbit when the position of the center of mass is directly above the stance foot in the sagittal plane. Given a set of Cartesian velocity variations for the center of mass,

$$v \in V := \{v_1, v_2, \cdots, v_{N_v}\}, \tag{6.4}$$

where $v \in \mathbb{R}^3$, define the ith velocity perturbation as

$$x_{v_i} := x_{v_0} + \delta x_{v_i}, \tag{6.5}$$

such that $p_{cm}(x_{v_i}) = p_{cm}(x_{v_0})$ and

$$v_i = \frac{\partial p_{cm}(x_{v_0})}{\partial x} \delta x_{v_i}, \tag{6.6}$$

where $p_{cm}(x)$ gives the Cartesian position of the center of mass corresponding to x.

For the purpose of attenuating the effects of velocity variations, we desire that the periodic orbit ensures the existence of the following additional solutions of the closed loop system: $\forall\ 1 \leq i \leq N_v$, $v_i \in V$, $1 \leq j \leq N_s$, $\exists\ 0 < t_i < \infty$ and $0 < T_{ij} < \infty$ such that

$$x_{i1} = \varphi(t_i, x_{v_i}) \in \mathcal{S}^{d_0} \tag{6.7}$$

and $x_{i(j+1)} = \varphi(T_{ij}, \Delta(x_{ij})) \in \mathcal{S}^{d_0}$, as in (6.3). In plain words, there exist steps that begin on the periodic orbit, end at nominal ground height d_0 after a velocity disturbance v_i is applied mid-step, and continue for at least N_s more steps at nominal

ground height d_0, as shown in Figure 6.1.

Remarks: (a) When applying multiple disturbance types, the index i in (6.3) must be offset for each type of disturbance for calculations in Section 6.1.3. (b) We found that applying a velocity perturbation in the middle of a step is beneficial for finding solutions that satisfy the conditions in (6.7), while allowing time for the controller to make adjustments before the end of the step. It is possible, however, to apply a velocity disturbance at any point along the periodic orbit.

6.1.3 Gait Phase and Trajectory Deviations

The gait phase can be thought of as a measure of progress through each step. We further assume that the units are normalized on the periodic orbit so that the *gait phasing variable* takes values in $[0, 1]$, namely

$$\bar{\tau}(\Delta(x_0)) = 0 \tag{6.8}$$

$$\bar{\tau}(x_0) = 1, \tag{6.9}$$

and that $L_g\bar{\tau}(x) := \frac{\partial\bar{\tau}}{\partial x}(x)g(x) = 0$.

Let $\bar{\tau}_{ij}(t) := \bar{\tau}(\varphi(t, \Delta(x_{ij}))$, for $0 \leq t \leq T_{ij}$, and as in [24], denote by τ_{ij}^+ and τ_{ij}^- the initial and final values of $\bar{\tau}$ along the trajectory. Due to the assumption that $\bar{\tau}_{ij}$ is strictly increasing, the inverse map $\bar{\tau}_{ij}^{-1} : [\tau_{ij}^+, \tau_{ij}^-] \to [0, T_{ij}]$ exists. Define

$$\tilde{x}_{ij}(\tau) := \varphi(\bar{\tau}_{ij}^{-1}(\tau), \Delta(x_{ij})) \tag{6.10}$$

$$\tilde{u}_{ij}(\tau) := \Gamma(\varphi(\bar{\tau}_{ij}^{-1}(\tau), \Delta(x_{ij})), \beta). \tag{6.11}$$

For $1 \leq i \leq (N_d + N_v)$ and $1 \leq j \leq N_s$, deviations in the state and control

trajectories are defined as

$$\delta x_{ij}(\tau) := \begin{cases} \tilde{x}_{ij}(\tau) - \tilde{x}_0(0) & \text{if } \tau < 0 \\ \tilde{x}_{ij}(\tau) - \tilde{x}_0(\tau) & \text{if } \tau \in [0,1] \\ \tilde{x}_{ij}(\tau) - \tilde{x}_{0,\text{ext}}(\tau) & \text{if } \tau > 1 \end{cases} \tag{6.12}$$

$$\delta u_{ij}(\tau) := \begin{cases} \tilde{u}_{ij}(\tau) - \tilde{u}_0(0) & \text{if } \tau < 0 \\ \tilde{u}_{ij}(\tau) - \tilde{u}_0(\tau) & \text{if } \tau \in [0,1] \\ \tilde{u}_{ij}(\tau) - \tilde{u}_{0,\text{ext}}(\tau) & \text{if } \tau > 1 \end{cases} \tag{6.13}$$

for $\tau_{ij}^{+} \leq \tau \leq \tau_{ij}^{-}$, where $\tilde{x}_{0,\text{ext}}(\tau)$ and $\tilde{u}_{0,\text{ext}}(\tau)$ are forward extensions of the nominal periodic trajectories[1].

Using (6.12) and (6.13), the weighted square error is defined as

$$||\delta x_{ij}(\tau)||^2 := < Q\delta x_{ij}(\tau), \delta x_{ij}(\tau) > \tag{6.14}$$

$$||\delta u_{ij}(\tau)||^2 := < R\delta u_{ij}(\tau), \delta u_{ij}(\tau) > \tag{6.15}$$

for Q and R positive semi-definite (constant) matrices.

6.1.4 Robust Control Cost Function

The problem of defining a cost function $\mathcal{J}_0$ and appropriate equality and inequality constraints for determining a nominal periodic solution of (3.11) has been addressed in [136, pp. 151-155]; [135, 127] using parameter optimization. Here, we define additional terms that penalize deviations induced by the terrain-height disturbances in D and velocity disturbances in V.

[1] A more comprehensive approach for calculating errors of perturbed trajectories that includes backward extensions of nominal trajectories is available in [118].

For $1 \leq i \leq (N_d + N_v)$ and $1 \leq j \leq N_s$, we define

$$\mathcal{J}_{ij} := \frac{1}{(\tau_{ij}^- - \tau_{ij}^+)^2} \int_{\tau_{ij}^+}^{\tau_{ij}^-} (\tau - \tau_{ij}^+)\big(||\delta x_{ij}(\tau)||^2 + ||\delta u_{ij}(\tau)||^2\big) d\tau. \tag{6.16}$$

The term $\frac{(\tau - \tau_{ij}^+)}{(\tau_{ij}^- - \tau_{ij}^+)}$ scales the errors so that initial deviations from the nominal periodic trajectory are discounted with respect to errors toward the end of the step. The rationale for this is that if the closed-loop system were to rejoin the nominal periodic orbit by the end of the step, the disturbance would have been rejected and a next step would be guaranteed. The scale factor allows the optimization to focus on approximately achieving this objective. The benefit of the scale factor in (6.16) is demonstrated in Chapter IV by comparing optimization solutions that include the scale factor against those that do not. The additional term outside the integral, $\frac{1}{(\tau_{ij}^- - \tau_{ij}^+)}$, is included so that perturbed step costs are normalized w.r.t. the varying ranges of τ_{ij} that result from disturbances (e.g., higher and lower terrain).

The overall cost function is

$$\mathcal{J} = \mathcal{J}_0 + \sum_{i=1}^{N_d+N_v} \sum_{j=1}^{N_s} w_{ij} \mathcal{J}_{ij}, \tag{6.17}$$

where w_{ij} determines the relative weight of each step.

Parameter optimization problem: Find $(\beta; x_0)$ that (locally) minimize $\mathcal{J}$ subject to the existence of a periodic solution of (3.21) that respects ground contact conditions, torque limits, and other relevant physical properties, as illustrated in Section 6.2.4.

6.2 Control Design

This section provides an example implementation of the gait optimization method from Section 6.1 and the nonholonomic outputs from Section 5.1. Section 6.2.1 describes the bipedal robot and corresponding model. Section 6.2.2 defines the feedback control used for walking. Section 6.2.4 describes the optimization configuration for finding walking control solutions.

6.2.1 Bipedal Robot Model

The robot MARLO is the Michigan copy of the ATRIAS-series of robots built by Jonathan Hurst and is described in detail in [45, 113]. The robot's mass is approximately 55 kg and its legs are one meter long. Furthermore, while the robot has series elastic actuators, the springs used in this study are sufficiently stiff that they are ignored. Excluding the global Cartesian position, the resulting rigid model has nine DOF in single support and six actuators. Four sagittal-plane leg motors use harmonic drives with a 50:1 gear ratio, and two hip-abduction motors use a belt transmission with a 26.7:1 gear ratio. The power amplifiers for the leg and hip motors generate up to 5 Nm and 3 Nm of torque respectively.

The configuration variables $q = (q_u, q_a)'$ are shown in Figure 6.2. Specifically, the unactuated components are

$$q_u = [q_{zT}, q_{yT}, q_{xT}]', \tag{6.18}$$

and the actuated components are

$$q_a = [q_{1R}, q_{2R}, q_{3R}, q_{1L}, q_{2L}, q_{3L}]'. \tag{6.19}$$

With this choice of configuration variables, σ has three components corresponding to

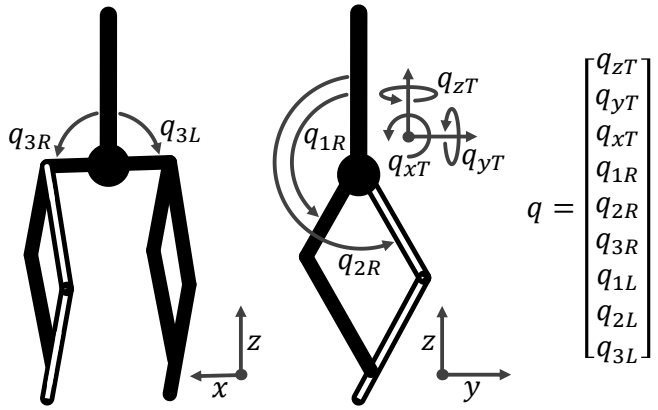

Figure 6.2: Rigid model of MARLO for control design and simulation. L and R designate left and right legs. q_{zT}, q_{yT}, and q_{xT} are the respective torso yaw, roll, and pitch Euler angles w.r.t. the world frame.

the angular momenta about the stance foot end in the yz-, xz-, and xy-planes (i.e., the sagittal, frontal, and transverse planes respectively). Because the model is 3D, the σ components can also be defined using x-, y-, and z-axes.

The complete hybrid model of the robot is derived as in [113], including the dynamic model for the single support phase and the reset map at leg impact. Using the natural state variables $x = (q, \dot{q})'$, the Lagrange model (3.7) is expressed in state variable form as in (3.9), with $x \in \mathcal{X}$ an open subset of $\mathbb{R}^{18}$ and $u \in \mathbb{R}^6$ for three degrees of underactuation during single support. Full details of the impact surface (3.23) and the reset map (3.20) are in Section 3.1.

The 3D robot model is assumed to be symmetric as in [8]. Hence, the control definition assumes right stance. During left stance, a coordinate transform on x maps the state of the robot to "right stance," and the resulting "right-stance" control inputs are then mapped back to the actual left-stance control inputs.

For control calculation, the y-axis is attached to the forward direction of the torso and control is yaw independent. Given sufficient vertical ground reaction forces, yaw motion is limited on MARLO by the two-contact-point feet shown in Figure 6.3. With these feet, MARLO pivots freely in the roll and pitch directions, which is consistent with the control model. Some outdoor experiments use the prosthetic feet shown in

Figure 6.3: Two-contact-point feet (left) and prosthetic feet (right).

Figure 6.3. Although prosthetic feet do not pivot as easily for roll and pitch, they provide a larger surface area for walking on compliant terrain, such as un-mowed grass.

6.2.2 Family of Feedback Controllers

The feedback controller is designed using the method of *virtual constraints* and *hybrid zero dynamics* as in [46, 134]. For MARLO, six virtual constraints are defined, one for each available actuator.

The output vector y is defined in terms of the configuration variables, q, angular momentum, σ, and a set of parameters κ and β,

$$y = h(q, \sigma, \kappa, \beta), \tag{6.20}$$

in such a way that the output has vector relative degree 2 [64, pp. 220] on a subset of interest, $\mathcal{X} \times \mathcal{K} \times \mathcal{B}$. The parameters κ, as shown in Section 6.2.2.1, are used to achieve invariance of the *zero dynamics* manifold induced by (6.20), while the parameters β will be tuned through optimization to achieve a desirable periodic orbit.

The feedback controller is based on input-output linearization, namely

$$u_{ff}(q,\dot{q},\kappa,\beta) := -\big[L_g L_f h(q,\dot{q},\kappa,\beta)\big]^{-1} L_f^2 h(q,\dot{q},\kappa,\beta), \tag{6.21}$$

$$u_{fb}(q,\dot{q},\kappa,\beta) := -\big[L_g L_f h(q,\dot{q},\kappa,\beta)\big]^{-1} \big(K_p y + K_d \dot{y}\big), \tag{6.22}$$

with

$$u = \Gamma(q,\dot{q},\kappa,\beta) := u_{ff}(q,\dot{q},\kappa,\beta) + u_{fb}(q,\dot{q},\kappa,\beta). \tag{6.23}$$

Along solutions of the closed-loop system, $\ddot{y} + K_d\dot{y} + K_p y \equiv 0$ (3.43).

An explicit choice of $h(q,\sigma,\kappa,\beta)$ is now made,

$$h(q,\sigma,\kappa,\beta) = h_0(q,\beta) - h_d(\tau(q),\sigma,\kappa,\beta), \tag{6.24}$$

where $h_d(\tau(q),\sigma,\kappa,\beta)$ specifies the desired evolution of the control variables

$$h_0(q,\beta) = \begin{bmatrix} q_{LA,ST} \\ q_{LA,SW} \\ q_{KA,ST} \\ q_{KA,SW} \\ q_{yT} - \xi(\beta) q_{HA,ST} \\ q_{HA,SW,ABS} \end{bmatrix}, \tag{6.25}$$

where LA, KA, and HA are abbreviations of leg angle, knee angle, and hip angle respectively, and ST and SW designate the stance and swing legs, as shown in Figure 6.4. For the lateral controller, a combination of torso roll, q_{yT}, and stance hip, $q_{HA,ST}$, are used[2]. As in [7], $\xi(\beta)$ is a free optimization parameter that changes the

[2]Because of limited actuation, selecting only the torso roll or the stance hip as a control variable causes the uncontrolled joint to drift during perturbations. However, a control variable defined by a combination of torso roll and stance hip causes the controller to respond to either component

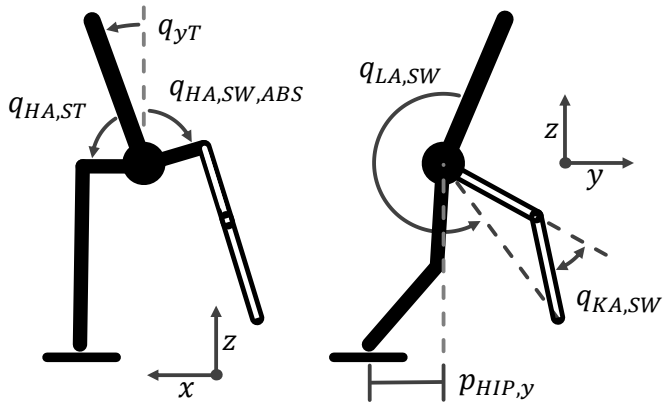

Figure 6.4: Control variables and gait phasing variable. Control trajectories are synchronized with the motion of $p_{HIP,y}$.

exact output configuration. Finally, $q_{HA,SW,ABS}$ represents the absolute swing-hip angle w.r.t. the global vertical axis.

The desired evolution of the control variables, $h_0(q, \beta)$, is chosen as

$$h_d(\tau(q), \sigma, \kappa, \beta) = h_{d,\tau}(\tau(q), \kappa, \beta) + h_{d,\sigma}(\sigma, \beta), \tag{6.26}$$

where $h_{d,\tau}(\tau(q), \kappa, \beta)$ and $h_{d,\sigma}(\sigma, \beta)$ specify holonomic and nonholonomic virtual constraints respectively.

The function $h_{d,\tau}(\tau(q), \kappa, \beta) \in \mathbb{R}^6$ is a vector of splines that specifies the desired evolution of defined $h_0(q, \beta) - h_{d,\sigma}(\sigma, \beta)$ in terms of the gait phasing variable $\tau(q)$. Here, the splines are Bézier polynomials, with the ith polynomial given by

$$h_{d,\tau,i}(\tau, \kappa, \beta) := \sum_{k=0}^{M} \alpha_{i,k} \frac{M\,!}{k\,!(M-k)\,!} \tau^k (1-\tau)^{M-k}, \tag{6.27}$$

where, as in [136, pp. 138], the six degree-$(M+1)$ Bézier polynomials are defined by $\alpha(\kappa, \beta) \in \mathbb{R}^{6\times(M+1)}$, which is derived in Appendix B. The gait phasing variable, $\tau(q)$,

drifting, even if the exact behavior of each individual joint is no longer guaranteed in perturbed conditions. For the combined control variable in (6.25), the sign convention of the stance hip is selected such that an input from the hip actuator causes consistent directional output changes for both components.

is selected to be an affine function of the y position of the center of the hips, $p_{HIP,y}$, and is normalized on the periodic orbit to take values in $[0, 1]$. If $\tau(q) > 1$ outside of the periodic orbit, extended Bézier polynomials defined in Section 6.2.3 are used in (6.27).

The nonholonomic virtual constraints are chosen as

$$h_{d,\sigma}(\sigma, \beta) = \begin{bmatrix} 0 \\ k_\sigma(k_1(\beta), \sigma_{yz}) \\ 0 \\ 0 \\ 0 \\ k_\sigma(k_2(\beta), \bar{\sigma}_{xz}) \end{bmatrix}, \tag{6.28}$$

where σ_{yz} and $\bar{\sigma}_{xz}$ are angular momentum in the sagittal and frontal planes and the nonholonomic function is defined as

$$k_\sigma(k_i, \sigma_j) := k_{i,1}\sigma_j + k_{i,2}\sigma_j^2 + k_{i,3}\sigma_j^3. \tag{6.29}$$

The complete output equation using (6.20) and (6.24)-(6.28) is

$$y = \begin{bmatrix} q_{LA,ST} \\ q_{LA,SW} \\ q_{KA,ST} \\ q_{KA,SW} \\ q_{yT} - \xi q_{HA,ST} \\ q_{HA,SW,ABS} \end{bmatrix} - \begin{bmatrix} 0 \\ k_\sigma(k_1, \sigma_{yz}) \\ 0 \\ 0 \\ 0 \\ k_\sigma(k_2, \bar{\sigma}_{xz}) \end{bmatrix} - h_{d,\tau}(\tau). \tag{6.30}$$

The inclusion of angular momentum in the third and sixth components of $h_{d,\sigma}$ allows step length and width to vary with velocity. In the optimization phase, values for

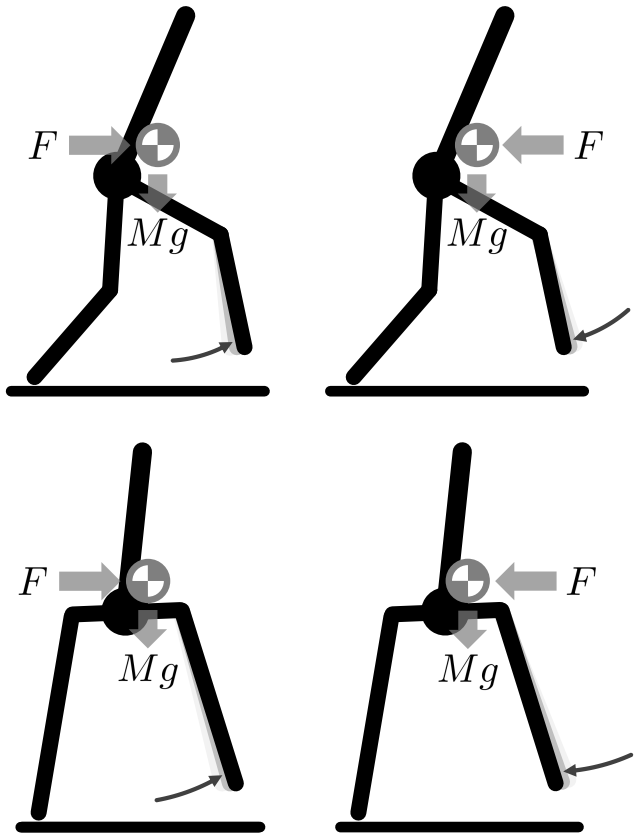

Figure 6.5: Posture changes in response to velocity perturbations from pushes in the sagittal (top) and frontal (bottom) planes. Changes in swing foot placement adapt the gravity moment between the stance foot and the center of mass during the following step. A detailed description of gravity effects on 3D walking dynamics is provided in Appendix A.

$k_1(\beta)$ and $k_2(\beta)$ will be chosen such that a perturbation in velocity, and attendant deviation of σ, results in a corrective change in swing foot placement. For the sagittal plane, this will adjust the amount of time the center of mass spends behind the stance foot, versus in front of the stance foot. For the frontal plane, this will adjust the magnitude of the lateral gravity moment proportional to the width between the stance foot and the center of mass. Both changes, shown in Figure 6.5, enable quicker convergence to the periodic orbit. Additionally, lateral stabilization through step width adjustments is shown to be more efficient than direct actuation in [79]. For more details, see [69, 77, 110].

Remarks on $\bar{\sigma}_{xz}$: During a nominal step, the robot rotates laterally both toward

and away from the stance foot, and hence σ_{xz} is negative and positive within the same step. In practice, we found it beneficial to use only the portion of σ_{xz} associated with rolling away from the stance leg during the later part of each step. To keep (6.23) continuous and smooth, we define $\bar{\sigma}_{xz}$ as

$$\bar{\sigma}_{xz} := \begin{cases} 0 & \text{if } \sigma_{xz} < 0 \\ \sigma_{xz} e^{\left(\frac{-30}{\sigma_{xz}^2}\right)} & \text{if } \sigma_{xz} \geq 0 \end{cases}. \tag{6.31}$$

6.2.2.1 Extended Model for Invariant Hybrid Zero Dynamics

Parameters κ are used to maintain *hybrid zero dynamics* following impact deviations. With output (6.30), it is straightforward to construct an equivalent function $\Psi : \mathcal{S}^d \times \mathcal{B} \to \mathcal{K}$ such that for all

$$\beta \in \mathcal{B} \text{ and } \begin{bmatrix} q^+ \\ \dot{q}^+ \end{bmatrix} = \Delta(q^-, \dot{q}^-)$$

the initial values of the outputs are zeroed, that is,

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} y^+ \\ \dot{y}^+ \end{bmatrix} = \begin{bmatrix} h(q^+, \sigma^+, \kappa^+, \beta) \\ \frac{\partial}{\partial q} h(q^+, \sigma^+, \kappa^+, \beta)\dot{q}^+ + \frac{\partial}{\partial \sigma} h(q^+, \sigma^+, \kappa^+, \beta)\dot{\sigma}^+ \end{bmatrix} \tag{6.32}$$

for $\kappa^+ = \Psi(q^-, \dot{q}^-, \beta)$. The current implementation of κ^+ is derived in Section 5.2.3.

Parameters κ are constant within each step and are reset at the end of each step, hence, they are included as states in the dynamics as derived in Section 4.2.2.

6.2.3 Extended Gait Phasing Variable and Bézier Polynomials

Along periodic walking gaits, the y position of the center of the hips, $p_{HIP,y}$ shown in Figure 6.4, is monotonic and cycles between a minimum value, $p^{min}_{HIP,y}$, and

a maximum value, $p^{max}_{HIP,y}$. The nominal gait phasing variable is defined as

$$\tau(q) := \frac{p_{HIP,y} - p^{min}_{HIP,y}}{p^{max}_{HIP,y} - p^{min}_{HIP,y}}, \tag{6.33}$$

where $p^{min}_{HIP,y}$ is the initial value of $p_{HIP,y}$ each step, and $p^{max}_{HIP,y}$ is the final value of $p_{HIP,y}$ on the periodic orbit.

If the periodic orbit is exited and $\tau(q) > 1$, the desired trajectory defined by the nominal gait phasing variable and Bézier polynomials can become counterproductive. To avoid this, an alternative trajectory is defined using an extended gait phasing variable,

$$\tau_{\text{ext}}(q) := \frac{p_{HIP,y} - p^{max}_{HIP,y}}{p^{max}_{HIP,y} - p^{min}_{HIP,y}}, \tag{6.34}$$

and a second set of Bézier polynomials, $\alpha_{\text{ext}}(\beta)$. Thus, the complete $\bar{\tau}(q)$ and $\bar{\alpha}(\kappa, \beta)$ used in (6.27) are defined using their nominal definitions and equivalent extensions as

$$\bar{\tau}(q) := \begin{cases} \tau(q) & \text{if } p_{HIP,y}(q) \leq p^{max}_{HIP,y} \\ \tau_{\text{ext}}(q) & \text{if } p_{HIP,y}(q) > p^{max}_{HIP,y} \end{cases} \tag{6.35}$$

$$\bar{\alpha}(\kappa, \beta) := \begin{cases} \alpha(\kappa, \beta) & \text{if } p_{HIP,y}(q) \leq p^{max}_{HIP,y} \\ \alpha_{\text{ext}}(\beta) & \text{if } p_{HIP,y}(q) > p^{max}_{HIP,y} \end{cases}. \tag{6.36}$$

$\tau(q)$, $\tau_{\text{ext}}(q)$, $\alpha(\kappa, \beta)$, and $\alpha_{\text{ext}}(\beta)$ should be defined such that (6.27) is continuous. One way of achieving continuity is by defining τ_{ext} such that $\{(q, \dot{q})' \in \mathcal{X} \mid \tau(q) = 1\}$

$$\tau_{\text{ext}}(q) = 0 \tag{6.37}$$

$$\dot{\tau}_{\text{ext}}(q, \dot{q}) = \dot{\tau}(q, \dot{q}), \tag{6.38}$$

and defining α_{ext} such that

$$\alpha_{\text{ext},0} = \alpha_M \tag{6.39}$$

$$\alpha_{\text{ext},1} = \alpha_{\text{ext},0} + \big(\alpha_M - \alpha_{(M-1)}\big)\frac{M}{M_{\text{ext}}}, \tag{6.40}$$

where α_i and $\alpha_{\text{ext},i}$ are the $(i+1)$ columns of α and α_{ext}, and $(M+1)$ and $(M_{\text{ext}}+1)$ are the degree of Bézier polynomials associated with α and α_{ext}. If $M \neq M_{\text{ext}}$, M_{ext} replaces M in (6.27) when using the extended parameters.

Remark: Defining $\tau(q)$, $\tau_{\text{ext}}(q)$, $\alpha(\kappa, \beta)$, and $\alpha_{\text{ext}}(\beta)$ such that control trajectories defined by (6.27) are continuous does not guarantee continuity of control inputs u in (6.23). This is evident later in Figure 6.15 where, when $\tau > 1$ (i.e., during downhill walking), the extended controller causes a jump in u that immediately requires a greater friction coefficient.

6.2.4 Robust Control Optimization Configuration and Control Solutions

The cost function for the nominal periodic orbit is based on energetic efficiency and is defined as

$$\mathcal{J}_0 := \frac{1}{\text{step length}} \int_0^{T_0} \sum_{i=1}^{6} |u_i \dot{q}_{m,i}| \, dt, \tag{6.41}$$

where step length is the distance between the stance and swing feet at impact[3], T_0 is the period, u is the 6-vector of motor torques, and $\dot{q}_m$ is the corresponding 6-vector of motor angular velocities, which is obtained from the link velocities and gear ratios as in [113]. The product of u_i and $\dot{q}_{m,i}$ is the instantaneous mechanical power from each motor.

The nominal periodic orbit was computed for walking on level ground (i.e., $d_0 = 0$) by optimizing (6.17), with nominal cost (6.41), subject to the hybrid dynamic model

[3]In (6.41), we use the absolute Cartesian distance between the stance and swing feet at impact as the step length. Alternatively, step length can be defined as the y distance between the feet in the sagittal plane.

(4.22) given in Section 4.2.2 and the following constraints: leg and hip motor torques saturate at 4 Nm and 2 Nm respectively, minimum vertical ground reaction forces of 250 N and maximum required friction coefficient of 0.5, minimum knee bend of 20^o to avoid hyperextension, maximum combined hip angles of 190^o to avoid leg collision, maximum link velocities of 200 deg/s for (q_1, q_2) and 60 deg/s for q_3, average walking speed between 0.5-1 m/s, minimum swing foot clearance of 0.1 m over stance foot, and backward swing-foot velocity at impact. Constraints based on ground reaction forces and physical limitations of MARLO also apply to perturbed steps.

The weight matrix Q in (6.14) is selected such that torso roll and pitch squared errors are multiplied by 4, hip squared errors are multiplied by 2, and velocity squared errors are divided by four. The weight matrix R in (6.15) is selected such that it has one fifth the base weighting of Q. The variables N_s and w_{ij} from (6.17) are selected such that costs are generated for two steps following a disturbance (i.e., $N_s = 2$), and the second perturbed step is multiplied by 3 (i.e., $w_{i2} = 3$). The rationale for this is to enable the optimizer to choose actions that may deviate more from the nominal trajectory directly following a disturbance, but result in quicker convergence to nominal conditions in subsequent steps.

The control solutions are found offline with `fmincon` in MATLAB, using the nonholonomic-virtual-constraints (NHVC) given in Section 6.2.2. For comparison purposes, a nominal control solution, NHVC_0, is defined for terrain-height disturbances $D = \{\pm 2 \text{ cm}, \pm 4 \text{ cm}\}$ and center of mass velocity disturbances in the x and y directions $V_x = \{\pm 7.5 \text{ cm/s}, \pm 15 \text{ cm/s}\}$ and $V_y = \{\pm 15 \text{ cm/s}, \pm 30 \text{ cm/s}\}$ for a total of twelve disturbances.

Additional control solutions are found using various configurations to test the efficacy of new concepts and establish best practices for selecting optimization disturbances. The control solutions are computed with `fmincon` initialized at the values obtained with NHVC_0, using the disturbance profiles indicated in Table 6.1. First, to

Table 6.1: Periodic-walking behavior of control solutions for different optimization configurations.

Control	Optimization Disturbance Profile D (cm)	V_x (cm/s)	V_y (cm/s)	Ratio of Disturbance/Efficiency Cost (6.17)	MCOT (6.42)	Impact Losses (J)	Forward Walking Speed (m/s)	Max. Poincaré Map λ	$\xi(\beta)$ (6.25)
Moderate Disturbances									
NHVC_0	$\pm2, \pm4$	$\pm7.5, \pm15$	$\pm15, \pm30$	2.1	0.240	6.7	0.736	0.61	0.276
NHVC_D	$\pm2, \pm4$	Ø	Ø	0.7	0.237	7.1	0.744	0.67	0.266
NHVC_V	Ø	$\pm7.5, \pm15$	$\pm15, \pm30$	1.3	0.235	7.2	0.741	0.75	0.291
Decreased Disturbances									
NHVC_{DV-}	$\pm1, \pm2$	$\pm3.75, \pm7.5$	$\pm7.5, \pm15$	1.0	0.242	7.0	0.764	0.69	0.246
NHVC_{D-}	$\pm1, \pm2$	Ø	Ø	0.5	0.241	6.9	0.765	0.72	0.287
NHVC_{V-}	Ø	$\pm3.75, \pm7.5$	$\pm7.5, \pm15$	0.4	0.237	6.9	0.749	0.72	0.314
Increased Disturbances									
NHVC_{DV+}	$\pm4, \pm8$	$\pm15, \pm30$	$\pm25, \pm50$	7.2	0.254	7.3	0.778	0.70	0.341
NHVC_{D+}	$\pm4, \pm8$	Ø	Ø	1.1	0.246	7.0	0.769	0.79	0.345
NHVC_{V+}	Ø	$\pm15, \pm30$	$\pm25, \pm50$	4.9	0.248	7.2	0.788	0.70	0.291
Varied Nonholonomic Function (6.29)									
HVC	$\pm2, \pm4$	$\pm7.5, \pm15$	$\pm15, \pm30$	7.9	0.231	7.5	0.780	0.92	0.284
$\text{NHVC}_{\text{Deg.1}}$	$\pm2, \pm4$	$\pm7.5, \pm15$	$\pm15, \pm30$	2.8	0.239	7.5	0.770	0.74	0.299
$\text{NHVC}_{\text{Deg.2}}$	$\pm2, \pm4$	$\pm7.5, \pm15$	$\pm15, \pm30$	1.8	0.239	7.1	0.729	0.69	0.239

investigate the utility of using nonholonomic outputs, a holonomic control solution, HVC, is optimized with $k_1, k_2 = 0$ in (6.28), using the same disturbance profile as NHVC_0. In order to find a stable solution for HVC, it is necessary to include optimization costs associated with the highest eigenvalue of the linearized Poincaré map, as done in [18]. Additionally, different nonholonoic functions are tested by using a linear (Deg.1) or quadratic (Deg.2) polynomial in place of (6.29). Finally, the remaining control solutions evaluate the effect of different disturbance configurations, such as incorporating only terrain (D) or velocity (V) disturbances or a decreased ($-$) or increased ($+$) range of disturbances. In the next section, these control solutions are evaluated in simulation. Additional control solutions for experiments are introduced in Section 6.4.1.

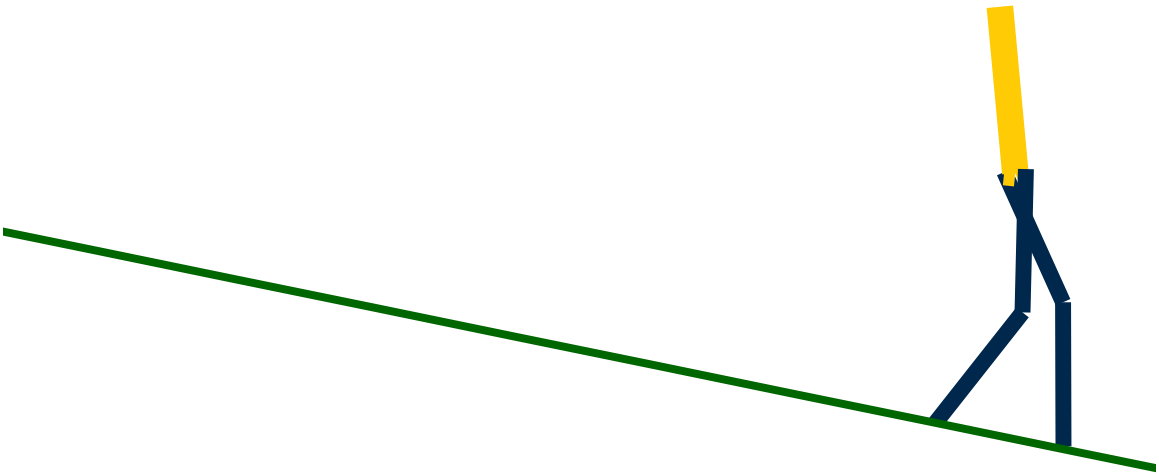

Figure 6.6: Sagittal view of $NHVC_0$ control solution walking downhill with repeated -10 cm changes in terrain height.

6.3 Simulation Results

Control solutions are compared in simulation to evaluate concepts introduced in this paper and to test the relative benefit of various disturbances for the robust control optimization. The gait designs of Table 6.1 are simulated under the influence of external forces and over terrain with varying height. Section 6.3.1 provides an initial evaluation of periodic flat-ground walking behavior. Section 6.3.2 evaluates the performance under persistent, repeated disturbances, which is a means to assess "steady-state" behavior under disturbances, whereas Section 6.3.3 focuses on transient aspects by giving results for recovery after a single disturbances. Discussion and interpretation of the simulation results are given in Section 6.3.4. A video illustrating the results is available at [44].

6.3.1 Walking on Flat Ground without External Perturbations

Each of the controllers in Table 6.1 is initially simulated over flat ground with no external perturbations. To evaluate the energetic efficiency of a control solution, the

mechanical cost of transport (MCOT) is calculated as

$$\text{MCOT} := \frac{1}{Mgd_y} \int_0^{T_0} \sum_{i=1}^{6} \max(u_i \dot{q}_{m,i}, 0) dt, \tag{6.42}$$

where M is the total mass of the biped, g is the acceleration due to gravity, d_y is the forward travel distance, and only the positive work of each actuator is considered. The reader is referred to [20] for a review of MCOT for various walking robots.

To evaluate the stability of a control solution's fixed point (i.e., periodic orbit), the eigenvalues of the linearized Poincaré map are computed, with the maximum magnitude of the eigenvalues, λ, given in Table 6.1. For the current control implementation, yaw is not regulated. Consequently, the eigenvalue associated with yaw is 1, as proved in [125, Propositions 3 and 4], and is not included in the comparison.

6.3.2 Repeated Disturbance Limits

Terrain and push disturbances are used to evaluate each control solution. For terrain disturbances, changes in terrain height consist of a vertical displacement of d (cm) per step. Figure 6.1 shows an example of a single vertical displacement, d_i, and Figure 6.6 shows walking with repeated -10 cm displacements in simulation. For push disturbances, horizontal forces of F (N) are applied to the center of mass over the duration of an entire step. This induces a velocity perturbation to the robot without the complication of terrain variation. Assuming left-right symmetry of a robot, push direction (forward or backward) in the sagittal plane determines the control response. In the frontal plane, however, whether a push is away from or toward the stance leg affects the appropriate control response, as illustrated in Figure 6.5.

Here, control solutions are compared under the action of a persistent disturbance whose magnitude is gradually increased each step until the robot falls. A fall occurs when requiring a friction coefficient greater than 0.6 or losing momentum and tum-

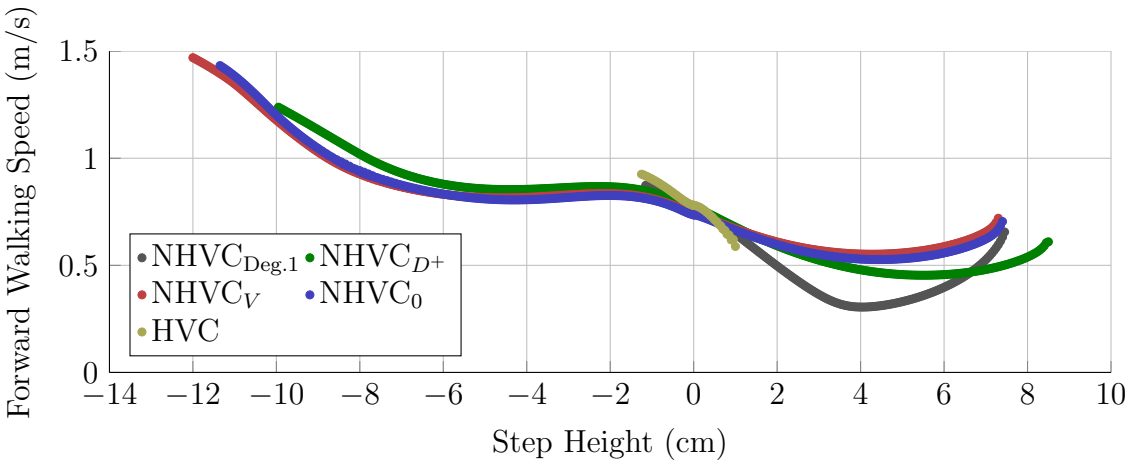

Figure 6.7: Walking speed vs. sustained terrain disturbances.

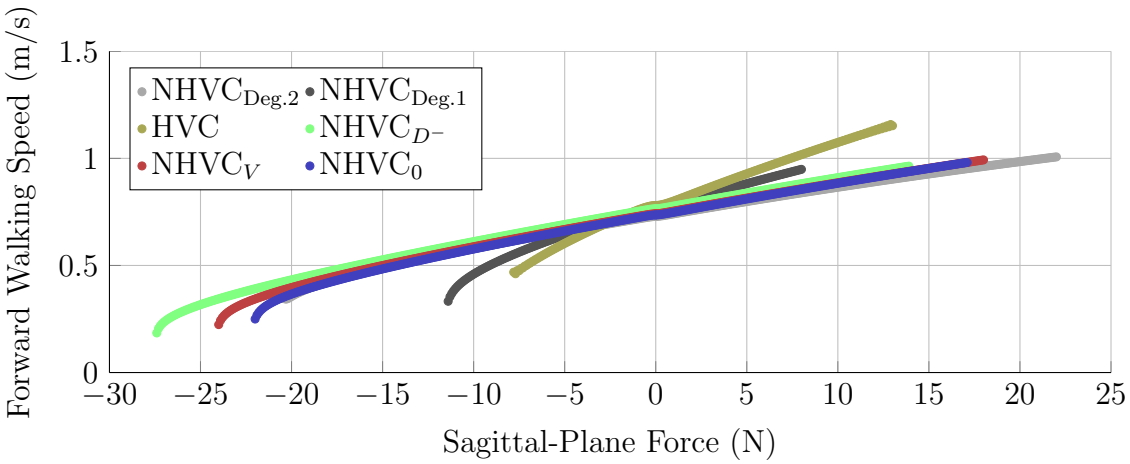

Figure 6.8: Walking speed vs. sustained sagittal-plane force.

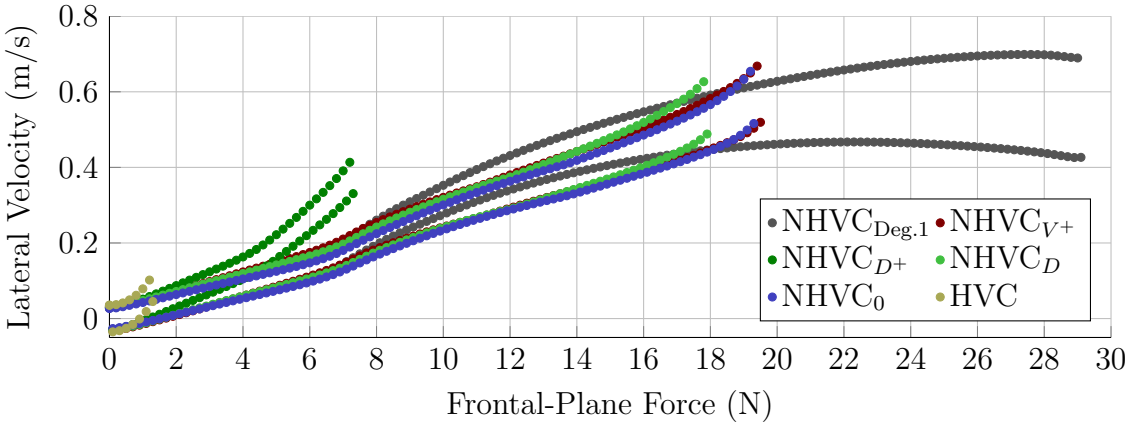

Figure 6.9: Lateral velocity vs. sustained frontal-plane force. Aperiodic lateral velocities result from alternating stance legs. Perturbed velocities are mirrored for negative forces.

Table 6.2: Disturbance limits of control solutions. Bold text indicates best and worst result for each column.

	Step Disturbance (cm)			Sagittal-Plane Force (N)			Frontal-Plane Force (N)		
Control	Min.	Max.	Range	Min.	Max.	Range	Min.	Max.	Range
$NHVC_0$	-11.35	7.40	18.75	-22.2	17.1	39.3	-20.2	20.2	40.4
$NHVC_D$	-10.90	7.35	18.25	-20.2	15.4	35.6	-18.8	18.8	37.6
$NHVC_V$	**-12.00**	7.30	**19.30**	-24.2	18.0	42.2	-18.5	18.5	37.0
$NHVC_{DV^-}$	-7.00	7.25	14.25	-22.3	12.6	34.9	-16.1	16.1	32.2
$NHVC_{D^-}$	-6.10	7.25	13.35	**-27.5**	13.9	41.4	-12.8	12.8	25.6
$NHVC_{V^-}$	-10.65	7.45	18.10	-21.5	16.2	37.7	-15.8	15.8	31.6
$NHVC_{DV^+}$	-10.75	7.60	18.35	-21.8	17.4	39.2	-18.5	18.5	37.0
$NHVC_{D^+}$	-9.95	**8.50**	18.45	-22.6	13.4	36.0	-8.5	8.5	17.0
$NHVC_{V^+}$	-10.10	6.80	16.90	-22.4	18.4	40.8	-20.4	20.4	40.8
HVC	**-1.25**	**1.35**	**2.60**	**-8.4**	13.0	21.4	**-2.4**	**2.4**	**4.8**
$NHVC_{Deg.1}$	-1.15	7.45	8.60	-11.6	**8.0**	**19.6**	**-29.5**	**29.5**	**59.0**
$NHVC_{Deg.2}$	-11.70	7.20	18.90	-21.1	**22.0**	**43.1**	-19.7	19.6	39.3

bling sideways or backward. For disturbance limits with changes in terrain height, each control solution is initialized on the periodic orbit, and then terrain height increases each step as $d_{k+1} = d_k + 0.5$ mm, where k is the step number. Once a fall occurs, the simulation is reset from the periodic orbit, and a decrease of 0.5 mm is applied to d_k until failure. The same procedure is applied through sagittal and frontal plane forces with 0.1 N increments. The results of repeated disturbance simulations are summarized in Tables 6.2 and 6.3, and, for illustrative control solutions, the perturbed velocities for each step are plotted in Figures 6.7-6.9. The lateral velocity in plots is the average velocity each step in the frontal plane.

6.3.3 Transient Response to Perturbations

Additional simulations are performed to evaluate the transient response of each control solution to individual terrain and push disturbances. Velocity deviations after terrain disturbances of ±2 cm, ±4 cm, and ±8 cm are shown in Figure 6.10. For push disturbances, 50 N forces are applied over the length of an entire step in either the

Table 6.3: Disturbance-limit averages based on optimization configuration. Bold text indicates greatest range for each category.

Optimization Disturbance Configuration	Step Disturbance (cm)			Sagittal-Plane Force (N)			Frontal-Plane Force (N)		
	Min.	Max.	Range	Min.	Max.	Range	Min.	Max.	Range
Disturbance Magnitude									
Moderate	-11.4	7.4	**18.8**	-22.2	16.8	**39.0**	-19.2	19.2	**38.3**
Decreased	-7.9	7.3	15.2	-23.8	14.2	38.0	-14.9	14.9	29.8
Increased	-10.3	7.6	17.9	-22.3	16.4	38.7	-15.8	15.8	31.6
Disturbance Type									
Terrain and Velocity	-9.7	7.4	17.1	-22.1	15.7	37.8	-18.3	18.3	**36.5**
Terrain Only	-9.0	7.7	16.7	-23.4	14.2	37.7	-13.4	13.4	26.7
Velocity Only	-10.9	7.2	**18.1**	-22.7	17.5	**40.2**	-18.2	18.2	**36.5**

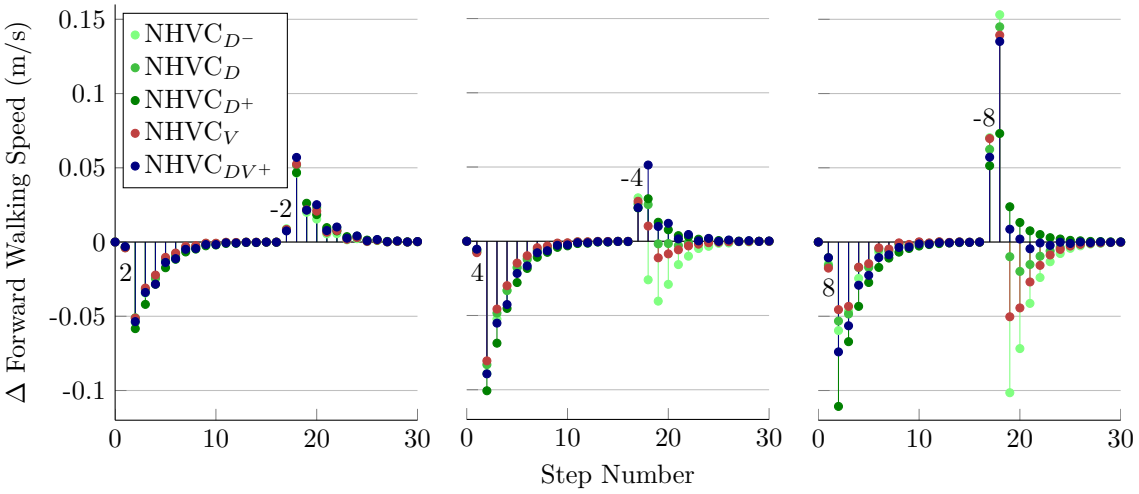

Figure 6.10: Sagittal-plane velocity deviations after ±2 cm (left), ±4 cm (center), and ±8 cm (right) terrain disturbances. Step-up and step-down disturbances occur on the first and seventeenth steps respectively. When converging back to the periodic orbit, sagittal-plane velocity is not necessarily monotonic due to the coupled dynamics of the sagittal and frontal planes. Following step-up disturbances outside of the ±2 cm range used for optimization, NHVC$_{D^-}$ is more destabilized than the other control solutions.

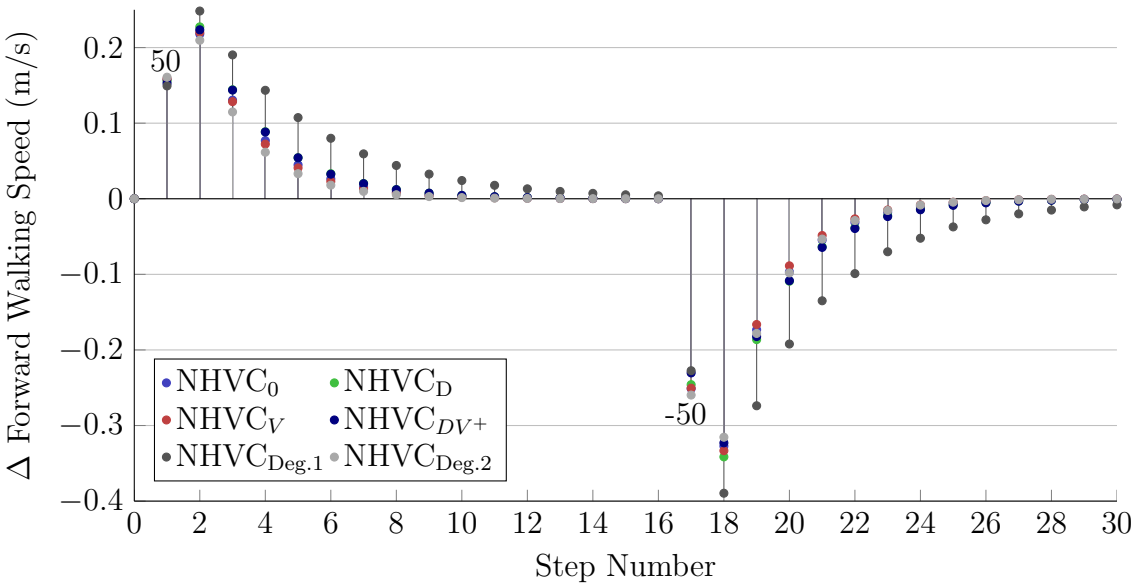

Figure 6.11: Sagittal-plane velocity deviations after pushes in sagittal plane. Forward and backward 50 N pushes occur over the entire first and seventeenth steps respectively. Figure 6.12 shows the simultaneous frontal-plane velocity deviations.

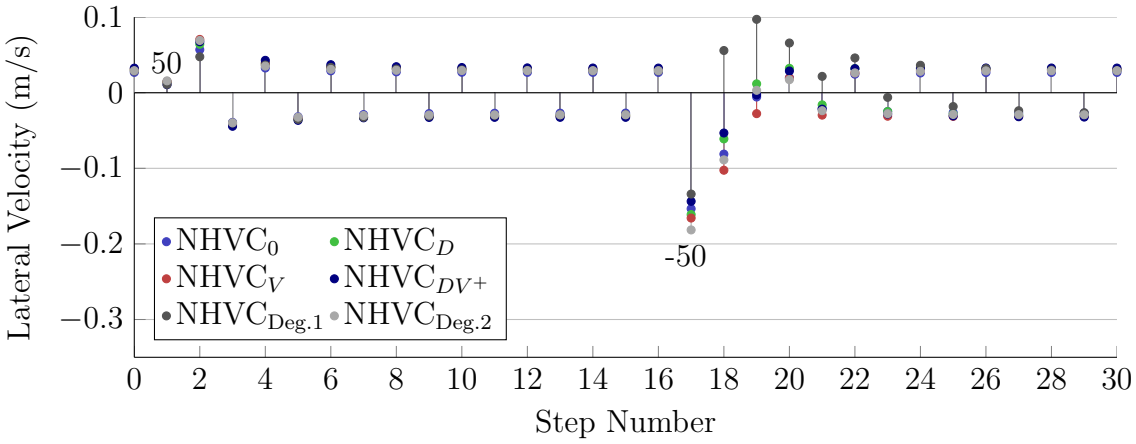

Figure 6.12: Frontal-plane velocity deviations after pushes in sagittal plane.

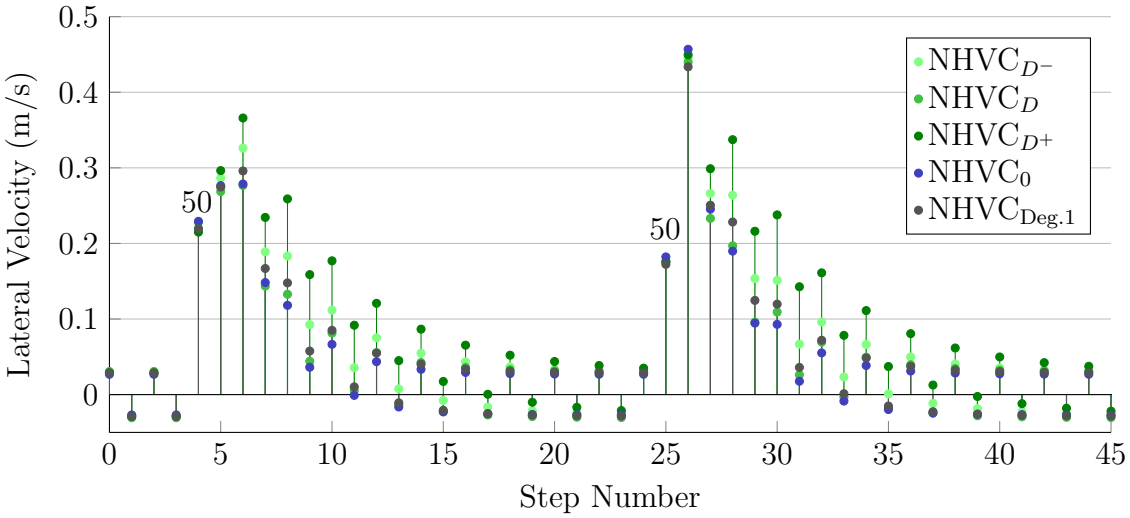

Figure 6.13: Frontal-plane velocity deviations after pushes in frontal plane. Lateral 50 N pushes away from and toward the stance leg occur over the entire fourth and twenty-fifth steps respectively.

sagittal or frontal planes. The HVC control solution is unable to recover from the disturbances used here and is not included in the analysis.

Sagittal-plane pushes are applied in the forward and backward directions, as shown in Figure 6.11. For 3D walking, the sagittal and frontal plane dynamics are coupled, as demonstrated by the simultaneous frontal-plane velocity deviations occurring with sagittal-plane pushes shown in Figure 6.12. Lateral perturbations caused by changes in forward walking speed are just one example of coupled dynamics. A loss in forward walking speed results in more time spent on a single stance leg, which subsequently causes a longer lateral gravity moment and increased lateral velocity by the end of the step. Likewise, a gain in forward walking speed results in less time spent on a single stance leg and a decreased lateral velocity. These coupled behaviors are evident in Figures 6.11 and 6.12. The role of synchronization of pendular motion in the sagittal and frontal planes to gait stability is studied in [114].

Frontal-plane pushes are applied in a single direction, as shown in Figure 6.13, but are timed such that the first lateral push is away from the stance leg and the second push is toward the stance leg. Both lateral-push behaviors are depicted in Figure 6.5

Table 6.4: Single-step pushes and corresponding impulses for $NHVC_0$.

Push Disturbance	Force (N)		Step Time (s)	Impulse (Ns)	
	x	y		x	y
None, Periodic Orbit	0	0	0.412	0	0
Forward	0	50	0.347	0	17.4
Backward	0	-50	0.609	0	-30.4
Away from Stance	50	0	0.410	20.5	0
Toward Stance	50	0	0.422	21.1	0

(bottom) for clarification.

Impulses corresponding to single-step pushes for the $NHVC_0$ control solution are provided in Table 6.4. A backward push results in the longest step time and greatest corresponding impulse.

6.3.4 Discussion of Simulation Results

Each of the control solutions in Table 6.1 have similar nominal periodic orbits with respect to forward walking speed, step length, and foot clearance at mid-step; nevertheless, as documented above, their responses to disturbances vary greatly.

Notably, the control solutions using nonholonomic outputs (NHVC) outperform the holonomic control solution (HVC). First, as shown in Table 6.2, HVC has the smallest range of admissible repeated disturbances. Although HVC handles greater forward forces than $NHVC_{Deg.1}$, it performs the worst for all other tested disturbances. Second, HVC exhibits the greatest deviations in velocity within its operating range, as shown in Figures 6.7-6.9. Finally, as shown in Table 6.1, the NHVC solutions have a smaller spectral radius (i.e., maximum magnitude of the Poincaré map eigenvalues) than HVC, suggesting quicker convergence to the periodic orbit after a (small) disturbance. Differences in the performance of the HVC and NHVC solutions are attributed to the ability of NHVC solutions to regulate walking posture with velocity (e.g., adjusting sagittal step distance with forward walking speed, as shown in

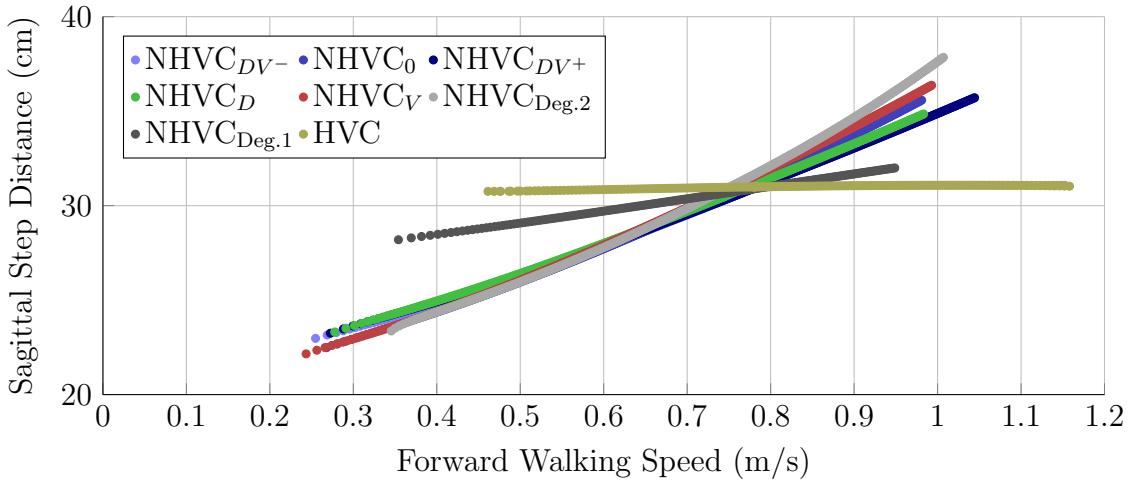

Figure 6.14: Sagittal step distance vs. forward walking speed. Details on how sagittal step distance regulates sagittal velocity are available in Chapter V.

Figure 6.14).

A comparison of NHVC solutions reveals that there are clear benefits to including velocity disturbances in the robust control optimization. First, NHVC$_V$, which incorporates only velocity disturbances during optimization, handles a wider range of repeated terrain disturbances than the other NHVC solutions (see Table 6.2). In contrast, NHVC$_{D+}$ and NHVC$_{D-}$, which incorporate only terrain disturbances during optimization, handle the smallest ranges of frontal-plane forces and have the slowest recoveries following lateral pushes (see Figure 6.13). Finally, with respect to repeated disturbances, solutions incorporating only terrain disturbances perform worse than solutions incorporating velocity disturbances (see Table 6.3). This difference in performance likely occurs because applying individual terrain disturbances during the robust control optimization does not perturb velocity to the same extent as repeated terrain disturbances. We propose that by including velocity disturbances in the robust control optimization, nonholonomic outputs are obliged to make constructive posture adjustments over a wider range of walking speeds, including speed changes that occur when walking uphill or downhill. Our analysis has considered only two types of disturbances. Investigating additional classes of disturbances to be included

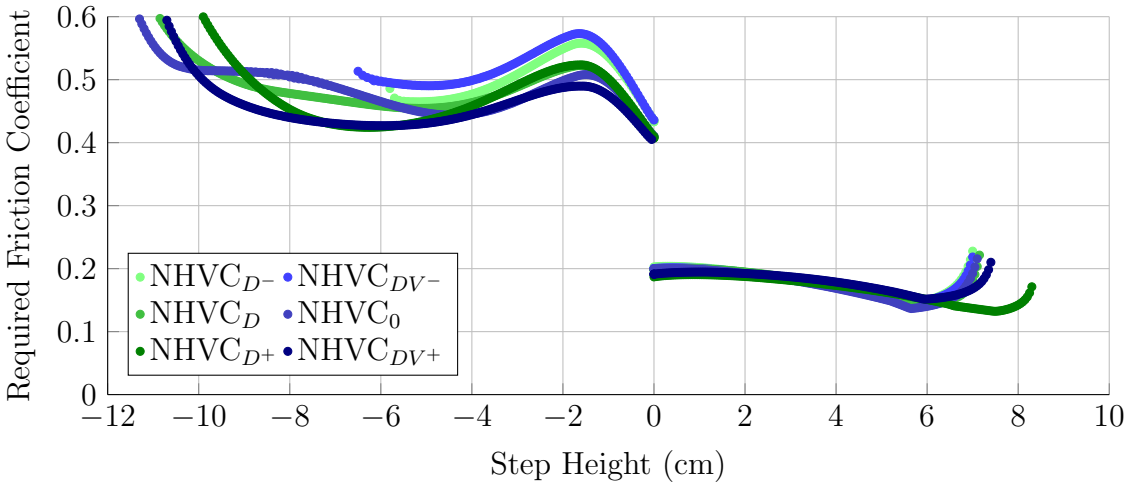

Figure 6.15: Required friction coefficient vs. sustained terrain disturbances. During downhill walking, the extended controller defined in Section 6.2.3 requires a greater friction coefficient. Essentially, the stance knee quickly bends to lower the biped.

in the control design process should be a fruitful endeavor.

The size of disturbances used for the robust control optimization is also significant. NHVC$_{D^-}$, which incorporates smaller disturbances during optimization than NHVC$_D$ or NHVC$_{D^+}$, exhibits greater deviations in velocity following the terrain disturbances illustrated in Figure 6.10. As the size of terrain disturbances incorporated during optimization increases, control solutions handle steeper uphill terrain (see Table 6.2) and require a lower friction coefficient for the majority of the repeated terrain disturbances illustrated in Figure 6.15. Incorporating larger disturbances for the robust control optimization does not, however, indiscriminately improve performance. As shown in Table 6.3, solutions incorporating only moderate disturbances handle the widest range of repeated terrain and force disturbances. For the current control implementation, we propose that incorporating larger disturbances during optimization results in the adherence to performance criteria (e.g., required friction coefficient) for a broader range of disturbances; however, this generalization comes at a cost in other aspects. This tradeoff could be avoided with a control implementation that enables tailoring for specific conditions (e.g., switching among a library of control solutions).

Many of the NHVC solutions have a similar recovery from velocity perturbations, as shown in Figures 6.11 and 6.12. This is, in part, due to using local optimization with repeated initial values. Consistent solutions for $k_1(\beta)$ and $k_2(\beta)$ in the nonholonomic function, (6.28), determine how posture adapts with velocity. There is more variability when changing the underlying nonholonomic function (i.e., HVC, $\text{NHVC}_{\text{Deg.1}}$, and $\text{NHVC}_{\text{Deg.2}}$) than when changing optimization disturbances, as shown in Figure 6.14. Implementing nonholonomic functions other than simple polynomials would likely enable additional variability. As an aside, additional nonholonomic outputs for posture regulation could enhance recovery from velocity perturbations (e.g., changing stance and swing knee angles to regulate lateral velocity through modified ground reaction forces and step duration).

Walking efficiency should be evaluated for a variety of terrain conditions and, as emphasized by [120], within the context of robustness. Although HVC exhibits the lowest periodic MCOT (see Table 6.1), when considering the limited range of traversable terrain for this control solution (see Table 6.2), the *flat-ground* walking efficiency is less relevant. Additionally, just as [87, 139] consider gait efficiency for a range of velocities, for practical walking applications, we propose that efficiency should be evaluated for a variety of terrain conditions. MCOT is plotted for a range of repeated terrain disturbances in Figure 6.16. For the NHVC solutions, MCOT increases with uphill terrain because of the additional work required to raise the center of mass. For downhill terrain, MCOT decreases with moderate declines, but increases with more severe declines. This eventual increase arises from the larger impact losses associated with downhill walking. The effects of impact losses on MCOT are well illustrated by HVC, because it makes no velocity-dependent posture adjustments. Uphill walking decreases HVC's impact losses and MCOT, whereas downhill walking increases HVC's impact losses and MCOT.

Overall, the control solutions using nonholonomic outputs are able to handle a

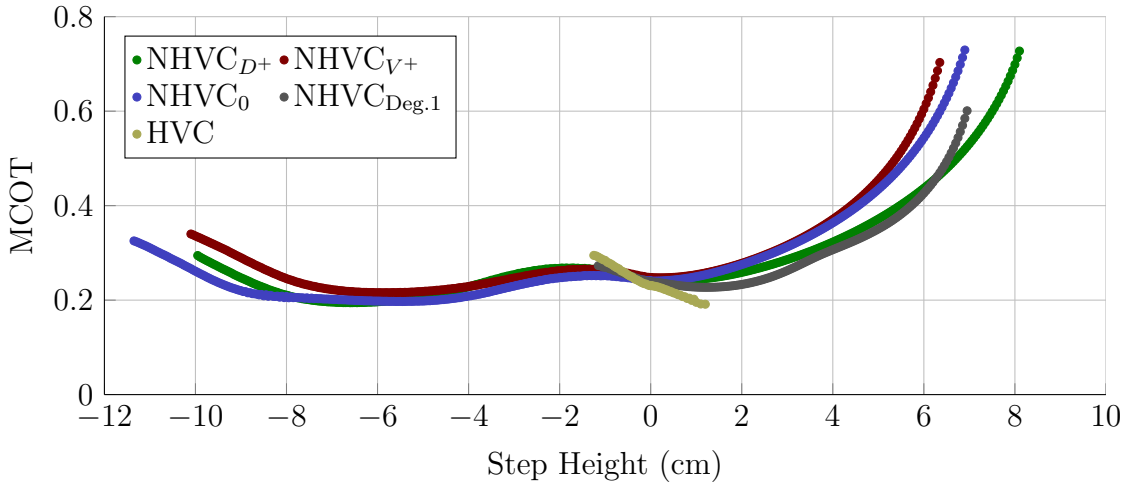

Figure 6.16: MCOT vs. sustained terrain disturbances.

wide range of disturbances and terrain conditions. NHVC$_0$ recovers from backward shoves of -30.4 Ns, handles about a 40 N range of sustained forces in the sagittal and frontal planes, and handles an 18.8 cm range of repeated terrain disturbances. Such robustness is desirable because it allows the robot to handle disturbances and difficulties associated with the robot hardware.

6.4 Experimental Results

Experiments are conducted on MARLO both indoors and outdoors. Section 6.4.1 introduces the control solutions implemented on the robot. Section 6.4.2 describes the setup for the experiments. Sections 6.4.5 and 6.4.6 present the results of the indoor and outdoor experiments respectively, with discussion given in Section 6.4.7. Videos of indoor and outdoor experiments are available at [44].

6.4.1 Control Solutions

The control solutions used on the robot are designed for outdoor terrain. Optimization terrain disturbances (D) are based on outdoor measurements and previous planar experiments with uneven terrain in [42]. Optimization velocity disturbances

Table 6.5: Periodic-walking behavior of control solutions used on the robot.

Control	Optimization Disturbance Profile D (cm)	V_x (cm/s)	V_y (cm/s)	Ratio of Disturbance/Efficiency Cost (6.17)	MCOT (6.42)	Impact Losses (J)	Forward Walking Speed (m/s)	Max. Poincaré Map λ	$\xi(\beta)$ (6.25)
Optimized Prior to Robot Experiments									
NHVC_0	$\pm 2, \pm 4$	$\pm 7.5, \pm 15$	$\pm 15, \pm 30$	2.1	0.240	6.7	0.736	0.61	0.276
$\text{NHVC}_0^{\text{Poincaré}}$	$\pm 2, \pm 4$	$\pm 7.5, \pm 15$	$\pm 15, \pm 30$	1.8	0.234	7.3	0.732	0.58	0.273
NHVC_1	$\pm 3, \pm 6$	$\pm 15, \pm 30$	$\pm 20, \pm 40$	4.7	0.219	8.0	0.751	0.75	0.258
NHVC_2	$\pm 3, \pm 6$	± 20	± 30	2.4	0.217	7.5	0.732	0.66	0.232
Optimized After Initiating Robot Experiments									
NHVC_3	$\pm 3, \pm 6$	± 30	± 30	3.0	0.267	7.3	0.811	0.74	0.246

(V) are based on forward walking speed and velocity changes attendant with repeated terrain disturbances. The nominal control solution based on nonholonomic virtual constraints, NHVC_0, is carried forward to the experiments. Prior to beginning the experimental phase of the work, additional controller designs similar to NHVC_0 are performed, as indicated in Table 6.5. The $\text{NHVC}_0^{\text{Poincaré}}$ control solution has the same disturbance profile as NHVC_0 with an additional penalty included on the spectral radius of the linearized Poincaré map (i.e., on the peak magnitude of the eigenvalues).

One additional control solution is performed after initiating the experiments. After the first day of outdoor walking, laterally-sloped terrain is identified as a significant perturbation to the gait of the robot. To account for this, an additional controller, NHVC_3, is designed with equal emphasis on velocity disturbances in the frontal and sagittal planes.

6.4.2 Experimental Setup

Virtual constraints resulting from the optimization process are implemented on the robot without modification. The feedback controller (6.23) is simplified as follows. In place of u_{ff}, constant 0.5 Nm torques are added to the stance leg and hips to

provide some friction and gravity compensation. In place of the decoupling matrix $L_g L_f h(q, \dot{q}, \kappa, \beta)$, a constant matrix is used to relate y to u_{fb}. Constant decoupling matrices are also used in [15, 14]. Additionally, commanded motor torque, u, is bounded at 5 Nm for the legs and 3 Nm for the hips. These bounds are greater than those used in optimization to compensate for unmodeled friction and other drivetrain inefficiencies on the actual robot.

Impact is detected by a rapid deflection in the springs when the swing foot contacts the ground. After swapping stance legs, α_0 from Appendix B updates such that $y = 0$. On the robot, there are no instantaneous jumps in the post-impact velocities, so, in place of updating α_1 such that $\dot{y} = 0$, α_1 updates to maintain its nominal difference with respect to α_0 on the periodic orbit. After control updates, torque bounds are initialized at 0 Nm and linearly scaled back to nominal values while $0 < \tau < 0.1$, which limits any counterproductive control inputs during the brief period of double support.

Joint angular velocities are estimated from encoder readings through numerical differentiation. It is a standard problem that such estimates appear "quite noisy" in comparison to the clean signals available in simulation. On MARLO, a low-pass Butterworth filter based on [16] attenuates only high-frequency "noise", because the cutoff frequency is necessarily high to limit phase delay in the feedback controller. Angular velocity estimates are particularly "noisy" following impacts and on the hip joint angles, which are measured on the motor side of a belt transmission. Consequently, the derivative term of the controller at the hip-angle only considers the h_0 component of (6.24) when calculating $\dot{y}$.

The gait phasing variable, τ, determines the progression of control trajectories, and angular momentum, σ, determines the velocity-based changes of control trajectories. Both τ and σ are critical for implementing nonholonomic virtual constraints. First-order filters for their estimation from measured quantities are specifically de-

signed. Section 6.4.3 defines the phase estimator for τ, and Section 6.4.4 defines a reduced-order Luenberger observer for σ. Comparisons of original signals and their estimated counterparts are provided in Section 6.4.4.1.

Finally, in the walking experiments, the robot is initialized from a standing position. The use of nonholonomic virtual constraints makes initialization straightforward, because the controller automatically adjusts step length with forward velocity. Under the evaluated controllers, initializing the robot from a static pose and hand-guiding it forward through a few steps is sufficient to enter the basin of attraction. The initialization process is illustrated in [44].

6.4.3 Phase Estimator

An estimator is used in place of direct measurement of the gait phasing variable. This is done because when $\dot{\tau}$ is determined through numerical differentiation, it presents unacceptable oscillations after impacts, which transfer to the torque signals determined by the controller.

The phase estimator is defined as

$$\dot{\hat{\tau}} := \frac{1}{T} + L(\hat{\tau})(\tau - \hat{\tau}), \tag{6.43}$$

where $\hat{\tau}$ is the estimated gait phasing variable, T is the duration of the previous step, and $L(\hat{\tau})$ is the observer gain. The term $\frac{1}{T}$ is interpreted as a model for the evolution of the normalized phase variable τ, and $L(\hat{\tau})(\tau - \hat{\tau})$ is the correction term based on observation of τ. Hence, $L(\hat{\tau})$ determines the relative dependence of the estimated phase on the time-based model and the measured gait-phasing variable. Because the numerical estimates of joint velocities appear to be most inaccurate immediately following an impact, $L(\hat{\tau})$ is chosen such that (6.43) emphasizes the time-based model immediately following impact and then smoothly returns to accurately tracking τ by

the end of the step. Specifically, $L(\hat{\tau})$ is defined as

$$L(\hat{\tau}) := \begin{cases} 20\hat{\tau} & \text{if } \hat{\tau} < 1 \\ \\ 20 & \text{if } \hat{\tau} \geq 1 \end{cases}. \tag{6.44}$$

Remarks: (a) During the first step of an experiment, previous step duration, T, is undefined. Therefore, (6.43) is modified such that $\frac{1}{T} = 0$ and $L(\hat{\tau}) = 20$. (b) In simulation, the estimated phase variable, $\hat{\tau}$, tracks well with τ and provides a reliable estimate of $\dot{\tau}$, as shown in Figure 6.18.

6.4.4 Reduced-Order Luenberger Observer for Estimating Angular Momentum

When σ_i is estimated on the robot through

$$\hat{\sigma}_i = D_i(q)\widehat{\dot{q}},$$

the resulting signal presents non-physical behavior as detailed in Appendix D. Consequently, reduced-order Luenberger observers based on [82] are developed to estimate angular momentum in the frontal and sagittal planes.

Here, the reduced-order observer is derived for angular momentum in the frontal plane. We use a process similar to [48], which was inspired by [91]. A novel aspect here is that the reduced-order design is not carried out on the complete model of the robot, but instead on a simplified inverted pendulum model. The simplified model is based on the center of mass position of the full model, as shown in Figure 6.17, but does not include "flywheel-like" dynamics from individual-link velocities and momenta.

To start our reduced-order observer derivation, the dummy state η_{xz} and its deriva-

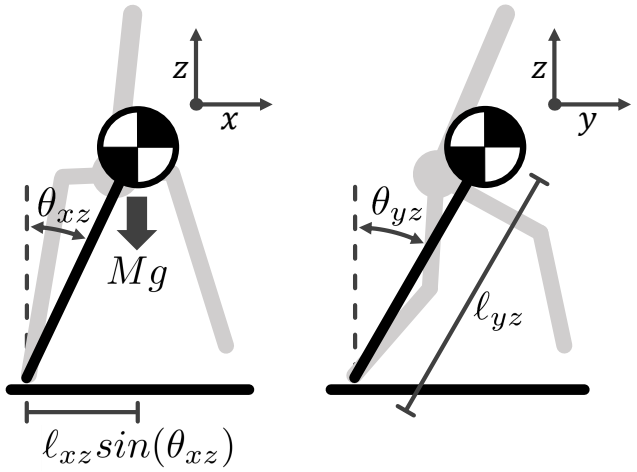

Figure 6.17: Simplified model for the reduced-order Luenberger observer. Two separate observers estimate σ for the frontal (left) and sagittal (right) planes.

tive are defined as

$$\eta_{xz} := \dot{\theta}_{xz} - L_{xz}\theta_{xz} \tag{6.45}$$

$$\dot{\eta}_{xz} = \ddot{\theta}_{xz} - L_{xz}\dot{\theta}_{xz}, \tag{6.46}$$

where $L_{xz} > 0$ is a scalar to be chosen. From the inverted pendulum model, $\ddot{\theta}_{xz}$ in (6.46) is calculated as

$$\ddot{\theta}_{xz} = \frac{g}{\ell_{xz}} sin(\theta_{xz}), \tag{6.47}$$

while (6.45) provides a substitution for $\dot{\theta}_{xz}$ in (6.46). Thus,

$$\dot{\eta}_{xz} = \frac{g}{\ell_{xz}} sin(\theta_{xz}) - L_{xz}(\eta_{xz} + L_{xz}\theta_{xz}). \tag{6.48}$$

Using (6.48), the reduced-order observer for η_{xz} is defined as

$$\dot{\hat{\eta}}_{xz} := \frac{g}{\ell_{xz}} sin(\theta_{xz}) - L_{xz}(\hat{\eta}_{xz} + L_{xz}\theta_{xz}). \tag{6.49}$$

At each impact, $\hat{\eta}_{xz}$ is updated to account for impact losses, which are calculated using the simplified model and [136, Eqn. (3.35)]. Using (6.45) with (6.49), a subsequent observer for $\dot{\theta}_{xz}$ is defined as

$$\hat{\dot{\theta}}_{xz} := \hat{\eta}_{xz} + L_{xz}\theta_{xz}. \tag{6.50}$$

Using (6.50) and the simplified model, σ_{xz} and its derivatives are estimated as

$$\sigma_{xz,L} = M\ell_{xz}\hat{\dot{\theta}}_{xz} \tag{6.51}$$

$$\dot{\sigma}_{xz,L} = Mg\ell_{xz}sin(\theta_{xz}) = \dot{\sigma}_{xz} \tag{6.52}$$

$$\ddot{\sigma}_{xz,L} = Mg\ell_{xz}\hat{\dot{\theta}}_{xz}cos(\theta_{xz}), \tag{6.53}$$

where $\sigma_{xz,L}$ is the Luenberger-observer estimate of σ_{xz}, and an equivalent process yields $\sigma_{yz,L}$ to estimate σ_{yz}.

Remarks: (a) Because $\dot{\sigma}_{xz}$ is only dependent on the center of mass position and gravity, $\dot{\sigma}_{xz,L} = \dot{\sigma}_{xz}$ in (6.52). (b) As with $\bar{\sigma}_{xz}$, $\bar{\sigma}_{xz,L}$ for the robot implementation is found using (6.31). (c) In simulation, there is little difference between the estimated angular momentum, σ_L, and the actual angular momentum, σ, as shown in Figure 6.19.

6.4.4.1 Comparison of Original and Processed Signals

The phase estimator, $\hat{\tau}$ in Figure 6.18, and Luenberger-observer angular momentum, σ_L in Figure 6.19, are compared with their corresponding original signals. Signal data are collected from simulation and the robot implementation. Simulation data corresponds to two steps from the periodic orbit of NHVC_0. Robot experiment data are taken from two steps using the NHVC_0 control solution. Angular momentum and other velocity-based quantities generally decrease at step transition due to impact

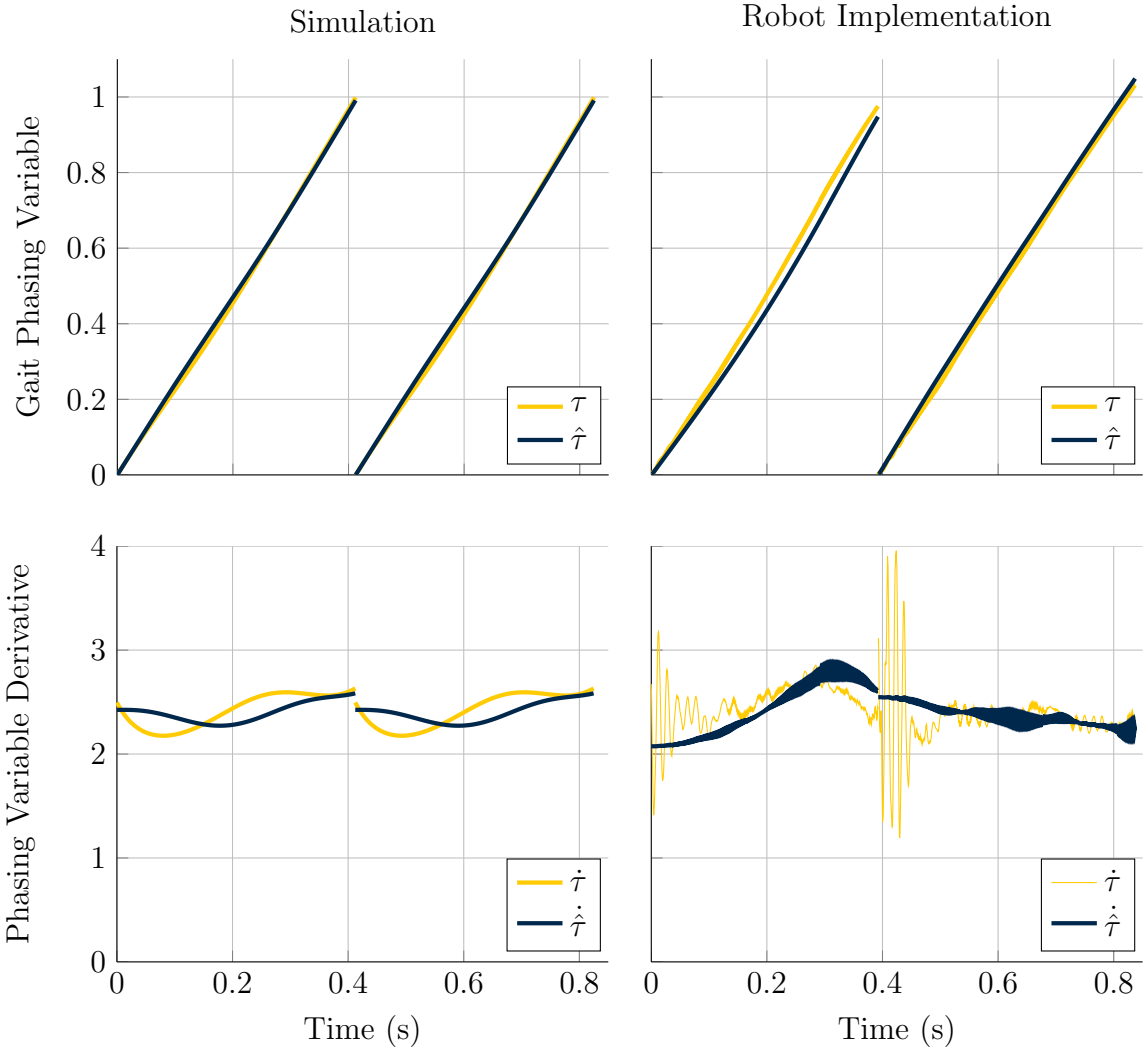

Figure 6.18: Comparison of τ and $\hat{\tau}$ using data from simulation (left) and the robot implementation (right).

losses.

6.4.5 Indoor Experiments

The first set of experiments with MARLO are performed indoors. As an initial robustness test of each control solution, terrain disturbances are created by either stacking sections of plywood boards in an organized fashion, as shown in Figure 6.20, or by throwing the boards randomly on the floor of the laboratory, as shown in Figure 6.21. Organized stacks of boards are immobile, quantifiable, and easily repro-

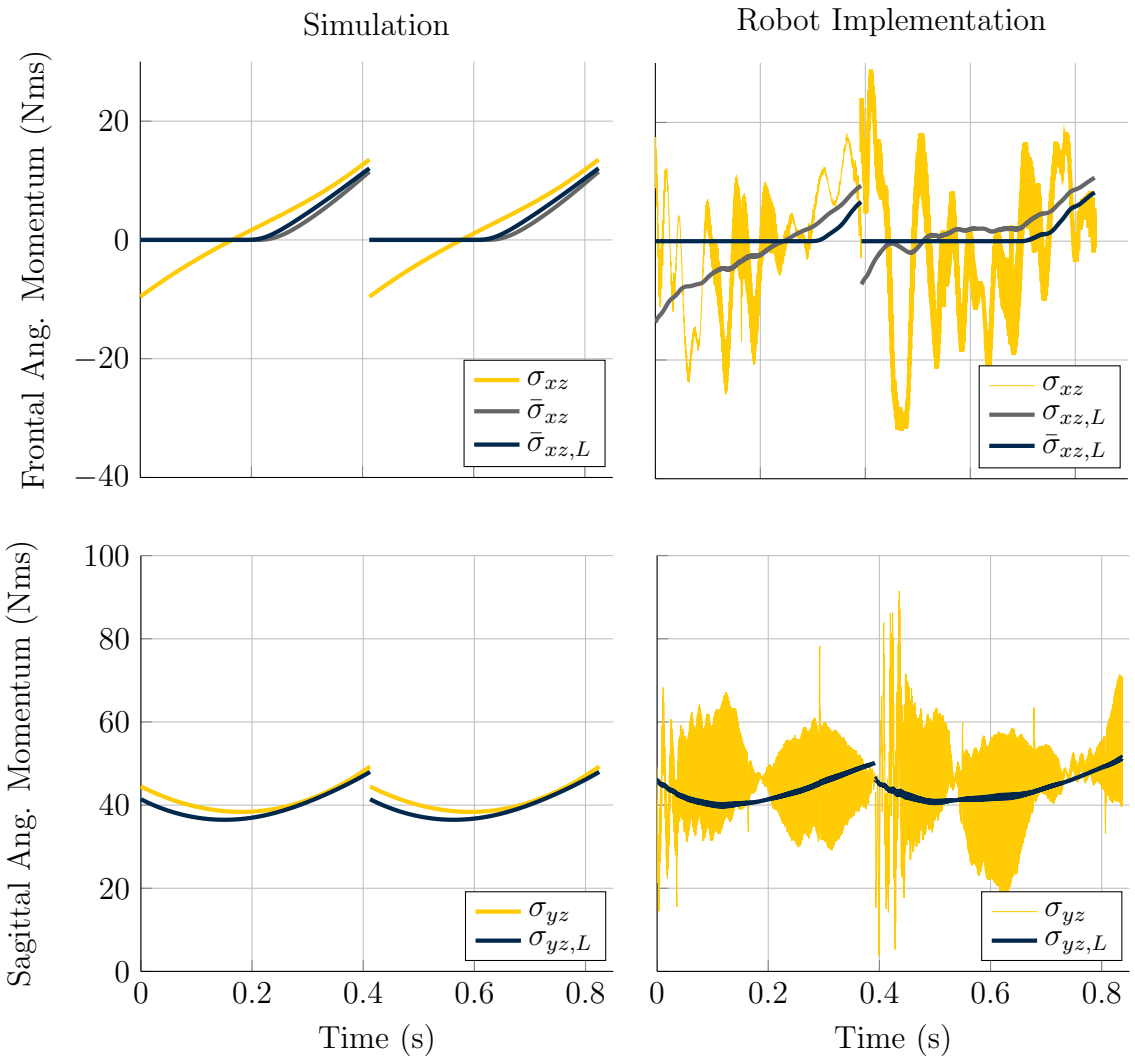

Figure 6.19: Comparison of σ and σ_L in the frontal plane (top) and sagittal plane (bottom). Data are taken from simulation (left) and the robot implementation (right). The sign convention of σ_{xz} alternates between right and left stance for symmetric control.

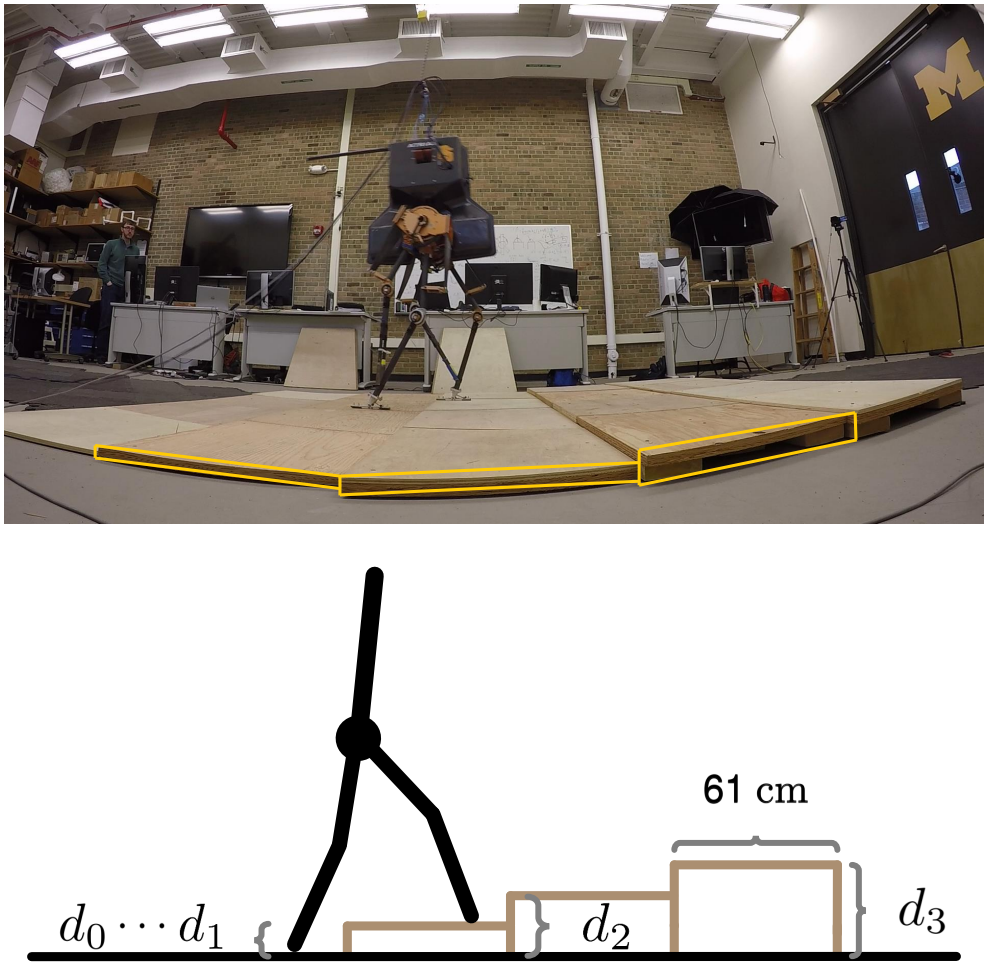

Figure 6.20: MARLO walks over stacks of boards. Sections of 61 cm-wide plywood boards are stacked to construct variable-height terrain disturbances.

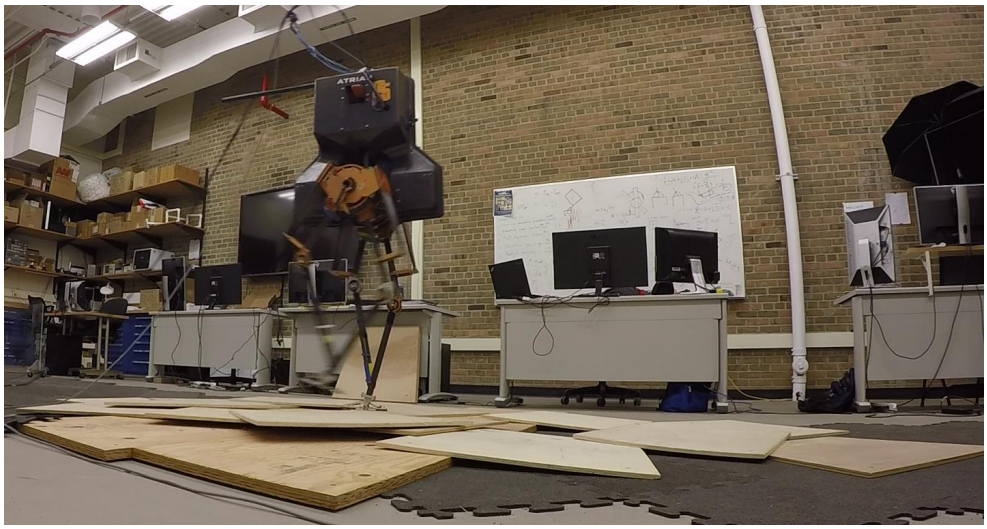

Figure 6.21: MARLO walks over randomly thrown boards.

Table 6.6: Indoor walking results from the first day of robot experiments. Terrain-disturbance profiles are created using the plywood boards shown in Figure 6.20. A "Success" indicates a successful crossing of the terrain on the first attempt, and a blank space indicates that the control solution was not tested with that terrain profile.

Terrain Disturbance Profile	Control Solution			
	$NHVC_0$	$NHVC_0^{Poincaré}$	$NHVC_1$	$NHVC_2$
Flat Ground	Success	Lateral Fall	Success	Success
0-1.2-0	Success		Success	
0-1.2-2.6-1.2-0	Success		Success	
0-2.6-2.6-0	Success		Success	Success
0-3.8-3.8-0	Success		Success	
0-5.0-5.0-0	Success		Foot Slip	
0-1.2-2.6-3.8-6.7-7.9	Success			

ducible for each experiment. Randomly thrown boards, on the other hand, present the additional challenge of shifting under applied weight.

On the first attempt, MARLO traverses the length of the lab using the $NHVC_0$ control solution, and, subsequently, walks across various terrain obstacle courses. From the point where MARLO is started to the opposite wall is approximately 11 m. Each of the control solutions listed in Table 6.6 is tested in turn on the same day. With the exception of $NHVC_0^{Poincaré}$, each of them results in MARLO traversing the lab. Videos of experiments listed in Table 6.6 and random board experiments are available at [44].

6.4.6 Outdoor Experiments

For experiments outdoors, a mobile gantry is used to transport MARLO to locations within a 1 km radius of the laboratory and to catch MARLO in case of a fall or when experiments are terminated. As shown in Figure 6.22, the gantry does not provide external support of the robot during walking experiments. Power is supplied by a set of batteries carried by the gantry, which enables MARLO to execute multiple experiments without returning to the lab for recharging. An Ethernet cable is

Figure 6.22: MARLO walks gradually uphill with a MCOT of 0.69 and an average walking speed of 0.91 m/s.

sometimes used to download data after experiments; it is partially visible in the same figure.

MARLO under the control laws developed in this paper is able to traverse sloped sidewalks, parking lots, and grass fields. Outdoors, experiments are no longer limited by the 11 m indoor lab space. Table 6.7 examines results for a few of the outdoor experiments. Data for walking over a grass area are collected from 50 consecutive steps, whereas data for all other experiments are collected from 100 consecutive steps. Grass-field experiments last until the gantry is obstructed by outdoor terrain. Other experiments are manually shut down prior to MARLO encountering an obstacle, with the exception of the downhill experiment, which ends due to an electrical-hardware malfunction. Figure 6.23 shows the the step-to-step behavior of the robot induced by walking over naturally varying outdoor terrain. Videos of the outdoor experiments listed in Table 6.7 are available at [44].

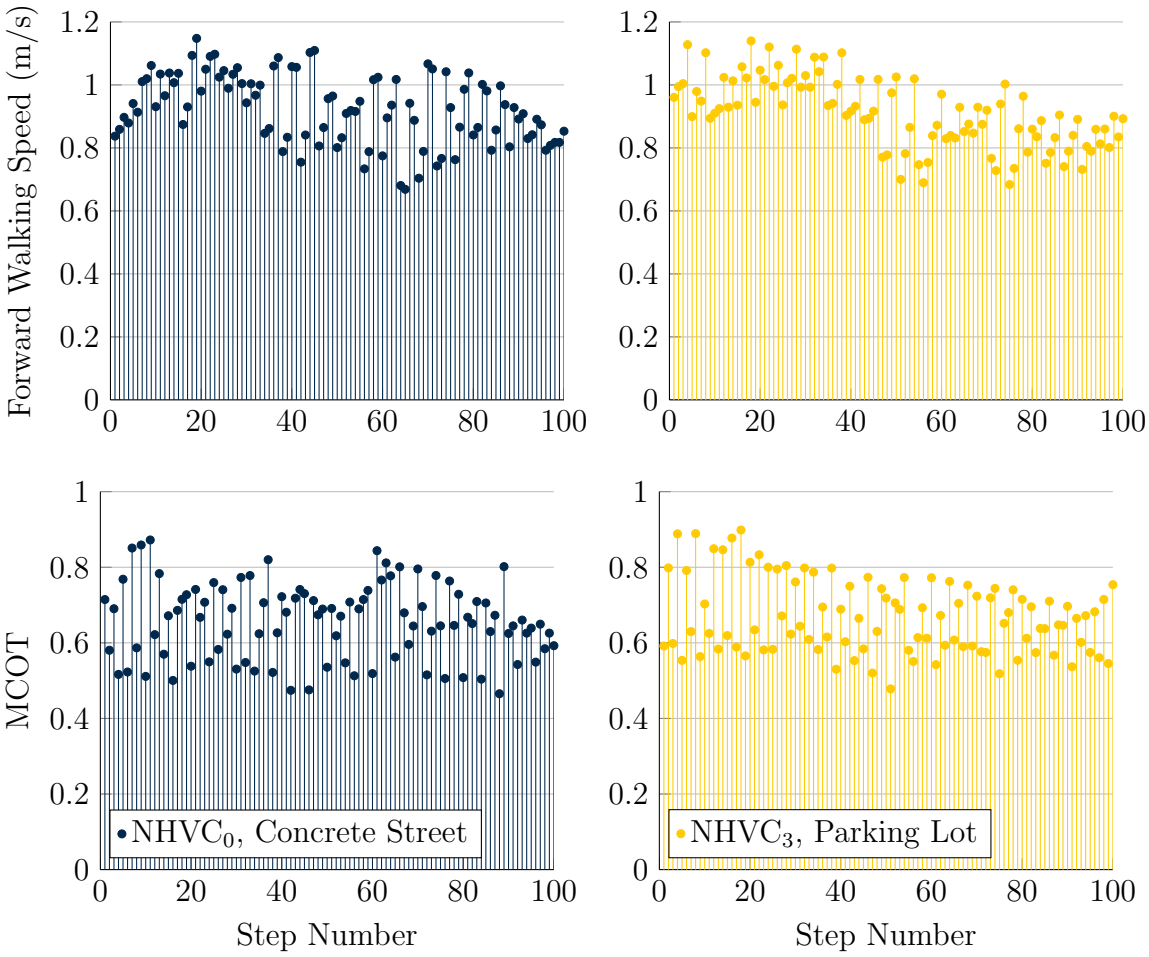

Figure 6.23: Step-to-step forward walking speed (top) and MCOT (bottom) during 100 consecutive steps of NHVC$_0$ on a concrete street (left) and NHVC$_3$ in a flat parking lot (right).

Table 6.7: Outdoor walking results for various terrains.

Control	Terrain Description	Forward Walking Speed (m/s)	MCOT (6.42)	COT		
				Liberal Estimate $\text{COT}_{\text{regen}}$ (C.5)	Conservative Estimate $\text{COT}_{\text{abs.}}$ (C.4)	P_{base} Component of COT
NHVC_0	Concrete street, fairly flat with a slight lateral slope and some potholes.	0.92	0.65	0.64	1.13	0.23
NHVC_3	Parking lot, fairly flat with a slight lateral slope.	0.90	0.67	0.62	1.19	0.24
NHVC_3	Grass field, fairly flat.	0.95	0.68	0.65	1.16	0.22
NHVC_3	Grass field using prosthetic feet, varying slope.	0.78	0.67	0.73	1.15	0.27
NHVC_3	Gradual downhill with some lateral slope in parking lot.	0.98	0.68	0.57	1.22	0.22
NHVC_3	Gradual uphill with consistent lateral slope on sidewalk.	0.91	0.69	0.69	1.16	0.23

The cost of transport (COT) is an alternative metric to MCOT when evaluating locomotion efficiency. The methods used in the literature to estimate COT vary with hardware configuration of the robot being studied. In the strictest sense, COT should be assessed on the basis of the energy required to recharge batteries after traveling a known distance. When performing outdoor experiments with MARLO, the battery pack on the mobile-gantry is not configured to measure the supplied power. However, based on experiments in the lab with supply-power measurements, conservative and liberal estimates for COT, $\text{COT}_{\text{abs.}}$ and $\text{COT}_{\text{regen}}$ respectively, can be calculated using data that are also available during outdoor experiments. Both of these quantities are derived in Appendix C and included in Table 6.7. The P_{base} components of $\text{COT}_{\text{abs.}}$ and $\text{COT}_{\text{regen}}$ account for power used for MARLO's on-board sensing and computation.

6.4.7 Discussion of Experimental Results

The robot successfully traverses the lab, both with and without obstacles, using the same control algorithms tested in simulation and applied directly out of the optimization procedure described in Section 6.2.4. With the exception of $\text{NHVC}_0^{\text{Poincaré}}$, each of the control solutions yield successful robot walking without any hand tuning. It is important to note that the actual robot differs significantly from the idealized control model. For example, the robot has estimated velocity signals for feedback control; varying levels of stiction and friction in each of the harmonic drives; series elastic actuators (springs in series with the motors in the sagittal plane and a timing belt in series with the motors in the frontal plane); a combination of manufactured and fatigued differences in individual physical components; and, due to on-going changes in hardware, an asymmetric mass distribution of 63 kg compared with the symmetric 55 kg simulation model. Despite these differences, the control solutions are sufficiently robust to handle the disturbances listed in Table 6.6 and randomly thrown piles of boards, as shown in Figure 6.21. On its first attempt, NHVC_0 traverses up and down 5 cm terrain disturbances–disturbances greater than those used during optimization.

After concluding indoor experiments, the NHVC_0 control solution is evaluated on the robot outdoors. MARLO walked for more than 100 steps across a slightly sloped paved area with potholes; the experiment was manually stopped to prevent the robot from colliding with a building. The MCOT was 0.65 and the average walking speed was 0.92 m/s, as reported in Table 6.7. These values differ from the simulation values reported in Table 6.5 for at least two reasons: (1) because of differences between the simulation model and the physical robot mentioned previously and (2) because outdoor terrain injects additional step-to-step variability, as shown in Figure 6.23.

After observing how the NHVC_0 control solution performed outdoors, it became apparent that laterally-sloped terrain caused the most significant perturbation to the robot. A new control solution, NHVC_3, was optimized to address this type

of disturbance and subsequently evaluated over a variety of terrains outdoors (see Table 6.7 for results).

Implementing NHVC_3 in multiple environments revealed many informative behaviors. First, the experiments with NHVC_3 show how $\text{COT}_{\text{abs.}}$ and $\text{COT}_{\text{regen}}$ vary with terrain. For example, $\text{COT}_{\text{regen}}$ is lower for downhill walking than for uphill walking. This is expected because walking downhill reduces the height of the robot's center of mass–a decrease in potential energy that may be recovered. Next, as shown in Table 6.7, the cost associated with P_{base} decreases with decreasing walking speed, consistent with the simulation work of [139]. Finally, the walking behavior of MARLO varies more with changes in hardware than with changes in terrain. Switching to prosthetic feet in the grass field causes a greater change in walking speed than when traversing any other terrain with the normal hardware configuration.

Outdoor experiments with MARLO set a new precedent for walking efficiency in realistic environments. Table 6.8 provides context for the outdoor walking experiments within the broader legged robotics literature. To the best of the authors' knowledge, MARLO under the NHVC_0 and NHVC_3 control solutions has achieved the lowest MCOT of any unsupported bipedal robot tested over rough terrain. Based on the conservative and liberal estimates of the COT in Table 6.7, it seems likely that this is also the case for the actual COT. Furthermore, whereas previous benchmarks have only been reported for treadmills and flat terrain, the NHVC_3 control solution provides a MCOT benchmark for walking over various realistic terrains.

Table 6.8: Mechanical cost of transport and cost of transport for various legged robots. Blank spaces indicate that information is not currently available.

Robot	MCOT	COT	Configuration for Cost of Transport Calculation							Rough Terrain Tested
			Speed (m/s)	Mass (kg)	Legs	Lateral Support	Includes Abduction Motors	On-board Power	Test Terrain	
Ranger [10] (COT record)	0.04	0.19		9.9	4	No	N/A	Yes	Linoleum Floor	No
Human [20] (estimated)	0.05	0.2			2	No				Yes
Ranger [10] (distance record)		0.28	0.59	9.9	4	No	N/A	Yes	Indoor Track	No
Denise [20, 137]	0.08		0.47	8	2	No	N/A	Yes		No
Meta	0.09[a]		0.4	12.3	4	No	N/A	Yes		Yes
MIT Cheetah [63, 123]		0.5	6	33	4	Yes	No	No	Treadmill	No[b]
MABEL [101, 127]	0.14		0.8	58	2	Yes	N/A	No	Flat	Yes
ERNIE	0.31[c]		0.60	19.6	2	Yes	N/A	No	Flat	No
MARLO ($NHVC_0$)	0.65		0.92	63	2	No	Yes	No[d]	Concrete Street	Yes
MARLO ($NHVC_3$)	0.68		0.95	63	2	No	Yes	No	Grass Field	Yes
MARLO ($NHVC_3$)	0.69		0.91	63	2	No	Yes	No	Gradual Uphill	Yes
ATRIAS [61] (OSU)		1.0			2	No	Yes	Yes		Yes
DURUS [4, 51] (DRC)		1.33	0.23	80	2	No	Yes	Yes	Treadmill	No
Asimo [20] (estimated)	1.6	3.2	0.44	52	2	No	Yes	Yes		Yes
ATLAS [4] (estimated)		20		157	2	No	Yes	Yes		Yes

[a]Calculated using the absolute mechanical power (see [56, Eqn. (4)]).

[b]The MIT Cheetah II has performed jumps over obstacles and outdoor running, but no COT information is available (see [100]).

[c]Calculated using the absolute mechanical power (see [104, Eqn. (12)]).

[d]MARLO can be configured to use an on-board 3 kg battery for power (see [96]).

CHAPTER VII

Concluding Remarks

One of the most significant technological advances our society can achieve during my lifetime is the closure of the gap between the potential of robotics and its actualization. To achieve this goal, however, we must overcome many technical challenges. In the specific context of robotic bipedal locomotion, one key challenge is the design of feedback controllers that function well in the presence of uncertainty, in both the robot and its environment. To that end, this dissertation addresses the design of feedback controllers and periodic gaits that function well in the presence of modest terrain variation, without reliance on perception and a priori knowledge of the environment.

A model-based control design methodology was developed for a class of underactuated 3D bipedal robots and evaluated both in simulations and in experiments. The first key aspect of the control design methodology was the computation of periodic orbits for walking that were robust to a finite set of perturbations. The second key aspect was the extension of the method of virtual constraints to include terms that depend on the robot's generalized velocity coordinates. Some distinguishing characteristics of the methods introduced are that they are theoretically grounded, systematic, and generalizable.

The viability of the design methodology was illustrated on MARLO, a 3D bipedal

robot with thirteen DOF in single support and six actuators. During indoor experiments, the robot walked on flat ground and over obstacles across a relatively short (i.e., 11 m) section of a laboratory. The controllers were designed on the basis of the full-order model of the robot and were implemented on the robot without hand tuning. Using the same design method during outdoor experiments, the robot traversed sloped sidewalks and parking lots and grassy areas, while maintaining average walking speeds between 0.9-0.98 m/s without perception or a priori knowledge of the terrain.

The mechanical cost of transport was evaluated for a variety of terrain conditions. To the authors' knowledge, there is no precedent for this in the robotics literature, either in simulation or in actual experiments. This is important because, for practical applications, robots must be able to traverse a variety of environments in a reliable and efficient manner. It is hoped that other robotics researchers will consider environments other than flat ground when evaluating walking efficiency of their robots.

7.1 Perspectives on Future Work

Direct extensions of this work include applying nonholonomic outputs to different applications, speeding up the optimization process, and incorporating a controller for starting and stopping. First, it would be interesting to implement nonholonomic outputs for different applications, such as prosthetics or bipedal robots with ankle actuation. Ankle torque would allow the controller to shape angular momentum about the stance foot while simultaneously making other velocity-based posture adjustments. Second, the *Optimization for Accommodation of Unknown Disturbances* method could be implemented using a faster optimization process, such as direct collocation [51]. This would make the process of designing an entire library of control solutions, each tailored for specific walking conditions, much more practical. Finally, the walking controller developed in this dissertation would ideally operate in con-

junction with a controller for walking in place [23] or standing. Ultimately, legged locomotion is not a means in itself and incorporating a controller for starting and stopping is important for practical applications.

More general extensions of this work include yaw control and perception to enable navigation and obstacle avoidance. Results for an underactuated yaw control method are described in Appendix A.6; however, a more practical solution is using additional actuation designed for turning the robot [1]. Using perception and computer vision to reason about the environment is an area of active research in its own right, and incorporating this work in real time with a walking controller is by no means an easy process. Still, even rudimentary information about the environment could dramatically improve performance by selecting terrain paths conducive for walking, avoiding large obstacles, and recognizing consistent uphill and downhill terrain prior to changes in velocity.

Finally, in the broadest sense, a significant barrier for the practical application of robotics is determining how to account for dynamic environments and varying tasks required of robots over their lifetimes. Once robots reach a certain level of competency, they will be expected to perform new and increasingly complicated tasks. Control methods that adapt to and learn from new environments, conditions, and tasks must be developed. This can be accomplished when a robot leverages the knowledge it has gathered over its lifetime while recognizing which experiences are no longer relevant to current circumstances [131]. Although this is a challenging research problem, lifelong robot learning and similar methods bring us closer to realizing more advanced robot autonomy at a faster rate than can be achieved when research addresses a single task that a robot performs (e.g., an entire dissertation on bipedal walking).

APPENDICES

APPENDIX A

Three-Dimensional Walking Concepts

This appendix provides a brief introduction to some of the fundamental concepts and challenges of three-dimensional (3D) walking. Challenges that are unique to 3D walking over planar walking include: the additional freedom and failure modes of the frontal and transverse planes; the asymmetric step-to-step roll behavior of the frontal plane; and the coupling of dynamics in the frontal, saggital, and transverse planes. The discussion assumes an underactuated model, however, when designing any 3D walking controller, the subject matter of the following sections is likely to be helpful.

A.1 Gravity as an Actuator

The importance of gravity during walking cannot be overstated. Every fall occurs because of gravity, but walking (as we know it) is not possible without it. Furthermore, gravity can act on behalf of a walking controller based on indirect stabilization methods [79, 110] to regulate velocity in the frontal and sagittal planes and reject disturbances with minimal energy costs.

The gravity-based change in sagittal-plane angular momentum, σ_{yz}, each step is

$$\Delta\sigma_{yz} = \int_0^T \dot{\sigma}_{yz}(q)dt = Mg \int_0^T p_{cm_y}(q)dt, \tag{A.1}$$

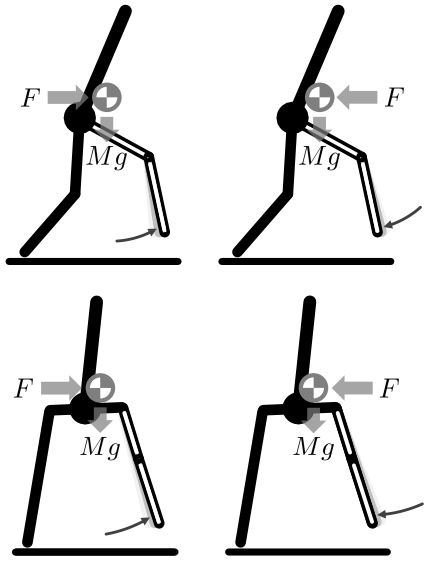

Figure A.1: Typical posture changes in response to velocity perturbations from pushes in the sagittal (top) and frontal (bottom) planes. Changes in swing foot placement adapt the gravity moment between the stance foot and the center of mass during the following step.

where T is the step time, $p_{cm_y}(q)$ is the y position of the center of mass w.r.t. the stance foot, and we assume sufficient ground reaction forces such that the biped does not yaw. For periodic walking on flat-ground, $\Delta\sigma_{yz} > 0$ to account for kinetic energy lost at impact. The equivalent change in frontal-plane angular momentum is

$$\Delta\sigma_{xz} = \int_0^T \dot{\sigma}_{xz}(q)dt = Mg \int_0^T p_{cm_x}(q)dt. \tag{A.2}$$

Adjustments in step length and width change the trajectory of p_{cm_y} and p_{cm_x} and thus change $\Delta\sigma_{yz}$ and $\Delta\sigma_{xz}$.

For velocity perturbations, and attendant deviations of σ, corrective changes in step length and width can enable quicker convergence to the periodic orbit. Example corrective changes in swing foot placement are depicted in Figure A.1. For the sagittal plane, this will adjust the amount of time the center of mass spends behind the stance foot, versus in front of the stance foot (i.e., adjust $\int p_{cm_y}$ in (A.1)). For the frontal plane, this will adjust the magnitude of the lateral gravity moment pro-

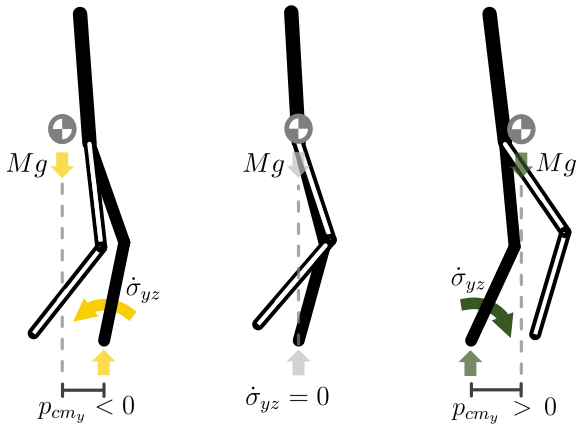

Figure A.2: Single-step gravity-moment progression for sagittal-plane angular momentum. Gravity causes σ_{yz} to decrease (maize) or increase (green) depending on if the center of mass is behind or in front of the stance foot (i.e., $p_{cm_y} < 0$ or $p_{cm_y} > 0$).

portional to the width between the stance foot and the center of mass (i.e., adjust $\int p_{cm_x}$ in (A.2)). Even when direct actuation of σ is available (e.g., ankle actuation), disturbance attenuation through step width adjustments is shown to be more efficient than direct actuation in [79].

For bipedal walking, there is a fundamental difference in how gravity changes angular momentum in the sagittal and frontal planes. In the sagittal plane, the sign of p_{cm_y} changes within each step, as shown in Figure A.2, whereas in the frontal plane, the sign of p_{cm_x} only changes between right and left stance, as shown in Figure A.3. From a walking control standpoint, this means the opportunity to use step length adjustments to increase and decrease σ_{yz} occurs every step, whereas the primary opportunity to use step width adjustments to increase or decrease σ_{xz} occurs every other step. This makes sustaining frontal-plane stability quite a bit more challenging than sagittal-plane stability. One advantage of a walking gait with quicker steps is that there is less of a time delay between right and left stance for adjusting σ_{xz}.

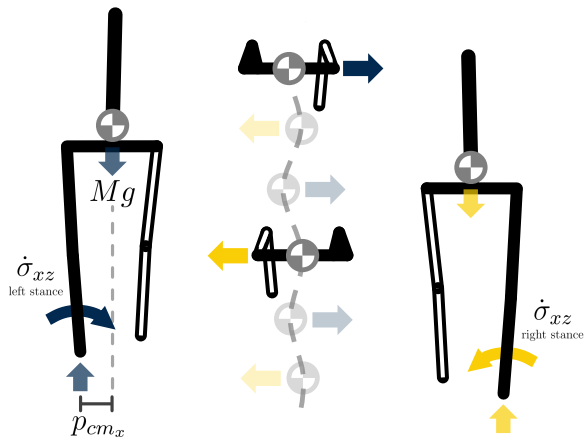

Figure A.3: Alternating step-to-step gravity moment for frontal-plane angular momentum. The direction of $\dot{\sigma}_{xz}$ alternates between left (blue) and right (maize) stance, which causes lateral oscillations during walking (middle).

A.2 Directing the Center of Mass via Ground Reaction Forces

Ground reaction forces (GRF) are another fundamental component of walking. First, the center of mass is ultimately guided by GRF. Any periodic orbit expresses a balanced sequence of GRF, even if GRF are not explicitly considered when designing the walking gait. Second, any practical robot implementation must maintain sufficient GRF such that it does not slip from requiring an unrealistic friction coefficient. Finally, with a clear understanding of GRF, it is possible to achieve improved control responses to perturbations.

The GRF-based change each step in the y-direction velocity of the center of mass, v_y, is

$$\Delta v_y = \int_0^T \dot{v}_y(q, \dot{q}, \ddot{q}) dt = \frac{1}{M} \int_0^T F_y(q, \dot{q}, \ddot{q}) dt, \tag{A.3}$$

where $F_y(q, \dot{q}, \ddot{q})$ is the y-direction GRF, which determines the y-direction center of mass acceleration, $\dot{v}_y$, and we assume sufficient GRF such that the biped does not

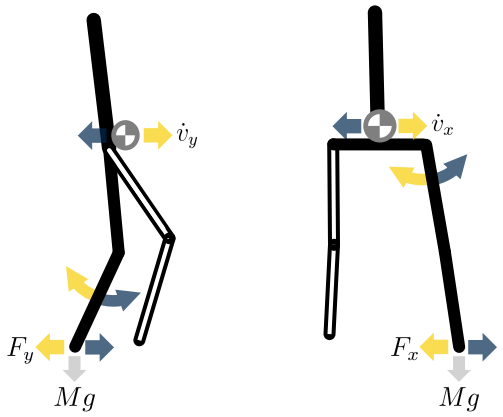

Figure A.4: Actuated horizontal forces in the sagittal (left) and frontal (right) planes. Actuation direction (maize or blue) determines the changes to $\dot{v}$.

yaw. The equivalent change in x-direction center of mass velocity is

$$\Delta v_x = \int_0^T \dot{v}_x(q, \dot{q}, \ddot{q})dt = \frac{1}{M}\int_0^T F_x(q, \dot{q}, \ddot{q})dt. \tag{A.4}$$

Horizontal forces, F_y and F_x, can result from any actuation that would cause horizontal motion of a frictionless stance foot, as shown in Figure A.4, or from reaction forces that support the center of mass, as shown in Figure A.5 and Figure A.6. Actuated horizontal forces require additional energy and must be regulated to avoid slipping given the friction coefficient and vertical GRF. However, actuated horizontal forces have a focused effect on $\dot{v}_y$ in (A.3) and $\dot{v}_x$ in (A.4), whereas reaction forces primarily effect vertical acceleration, $\dot{v}_z$.

Horizontal reaction forces are shaped by the position of the center of mass relative to the stance foot. However, these forces change when the distance between the center of mass and the stance foot are in transition. Let ℓ_{yz} represent this distance in the sagittal plane. If the center of mass is in front of the stance foot and the biped is accelerating, increasing or decreasing ℓ_{yz} causes a respective increase or decrease in acceleration, as shown in Figure A.5. However, if the center of mass is behind the stance foot and the biped is decelerating, increasing or decreasing ℓ_{yz} instead causes

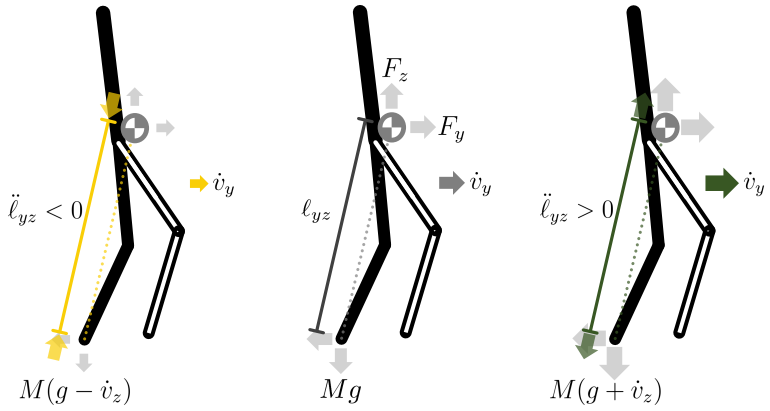

Figure A.5: Sagittal-plane reaction forces. When the center of mass is forward of the stance leg, nominal reaction forces (center) cause forward acceleration of the biped, $\dot{v}_y$. Decreasing (maize) or increasing (green) the nominal sagittal distance between the center of mass and the stance foot, ℓ_{yz}, causes a relative decrease or increase in GRF and $\dot{v}_y$.

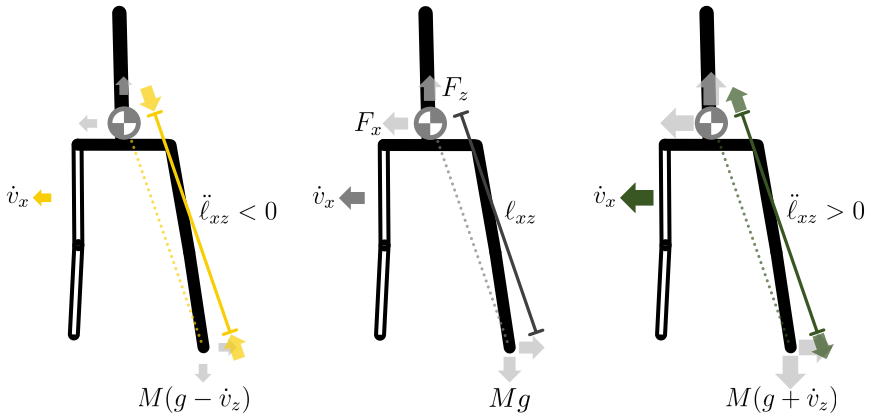

Figure A.6: Frontal-plane rection forces. Nominal reaction forces (center) cause lateral acceleration of the biped away from the stance leg, $\dot{v}_x$. Decreasing (maize) or increasing (green) the nominal ℓ_{xz} causes a relative decrease or increase in $\dot{v}_x$.

a respective increase or decrease in the deceleration. Changing ℓ_{yz} to adjust the acceleration profile is one method of regulating velocity. Similar adjustments can be made to ℓ_{xz} in the frontal-plane to adjust lateral acceleration, however, the direction of the nominal lateral acceleration is determined by right or left stance.

Varying ℓ in response to a velocity perturbation can have at least three benefits. To illustrate this, we use the example of increasing ℓ_{xz} in response to a frontal-plane push toward the stance leg. First, the increase in GRF attendant with $\ddot{\ell}_{xz} > 0$ redirects the center of mass back toward its original trajectory. Second, the increased leg length causes a greater potential energy requirement to roll the center of mass over the stance leg, which decreases the likelihood of falling laterally past the stance leg. Finally, extending the stance leg length will generally increase the step time, T, since this lifts the swing foot farther from the ground. Increasing T gives the lateral gravity moment in (A.2) and nominal GRF in (A.4) a longer time to redirect the perturbed momentum back to nominal values. One way to implement this control response in ℓ_{xz} is to use nonholonomic virtual constraints to change stance knee angle with front-plane angular momentum.

A.3 Coupled Sagittal-Plane and Frontal-Plane Dynamics

For 3D walking, the sagittal-plane and frontal-plane dynamics are coupled. One reason for this coupling is because step times and dynamics are directly linked, as demonstrated in (A.1)-(A.4). Perturbations to sagittal-plane dynamics change the step time, and perturbations to the step time change the frontal-plane dynamics. More specifically, a loss in forward walking speed results in more time spent on a single stance leg, which subsequently causes a longer lateral gravity moment and increased lateral velocity by the end of the step (i.e., T and $\Delta\sigma_{xz}$ increase in (A.2)). Likewise, a gain in forward walking speed results in less time spent on a single stance leg and a decreased lateral velocity. Such perturbations in velocity are common when

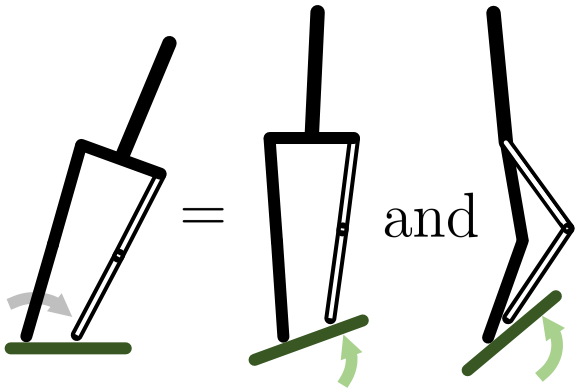

Figure A.7: Comparison of roll offset and geometrically-equivalent perturbations from terrain in the frontal and sagittal planes.

transitioning to a higher or lower level terrain in a single step. Additionally, when T varies step-to-step, there is an imbalance because more time is spent on a single stance leg, which causes greater lateral acceleration in a single direction. However, it is also possible to control lateral acceleration by purposefully varying T between right and left stance.

Frontal-plane behvaior also has a significant effect on the sagittal plane. An illustrative example of this occurs from a roll offset in the frontal plane, which is common for underactuated walking. It is clear that rolling the biped in Figure A.7 about the stance foot toward the terrain and adding "terrain roll" toward the biped are geometrically equivalent. However, the "terrain pitch" required to cause an equivalent impact to the planar version of the biped is less obvious. Thus, when a 3D biped experiences roll offsets in the frontal plane, the planar version of the biped experiences "uneven terrain," even if the 3D biped is walking on flat ground. This behavior illustrates why it is important to consider multiple types of disturbances to the biped during the control design process, as done in Chapter VI.

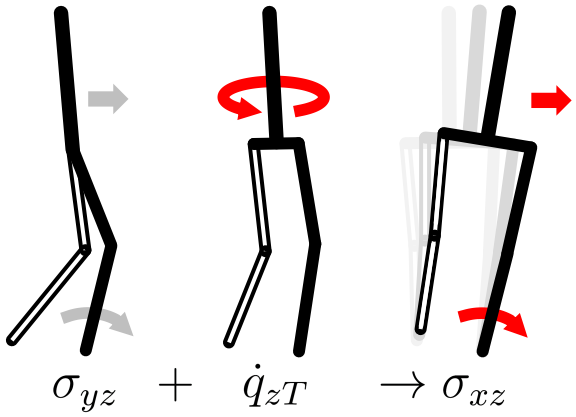

Figure A.8: Yaw-based transfer of momentum. Yaw motion, $\dot{q}_{zT}$, transfers forward-walking angular momentum, σ_{yz}, into problematic lateral angular momentum, σ_{xz}.

A.4 Consequences of Yaw Motion in the Transverse Plane

One obvious consequence of unintended yaw is turning away from a desired heading. A not so obvious consequence is that yaw motion in the transverse plane can cause a devastating perturbation to the nominal sagittal-plane and frontal-plane dynamics. Assume that the sagittal and frontal planes rotate with the biped, as in Section 6.2.1. If the biped yaws during a step, kinetic energy that was associated with forward motion in the sagittal plane transfers into lateral motion in the frontal plane of the biped. This is problematic because the magnitude of the angular momentum in the sagittal plane is normally much greater than in the frontal plane. Thus, yaw motion can greatly increase the kinetic energy in the frontal plane, which is a significant deviation from the nominal walking dynamics. This deviation is intensified by the loss of momentum in the forward walking direction, which enables more time for roll motion to occur before the step ends. In some cases, the additional angular momentum transferred to the frontal plane is sufficient to tip the biped over the stance leg, as shown in Figure A.8.

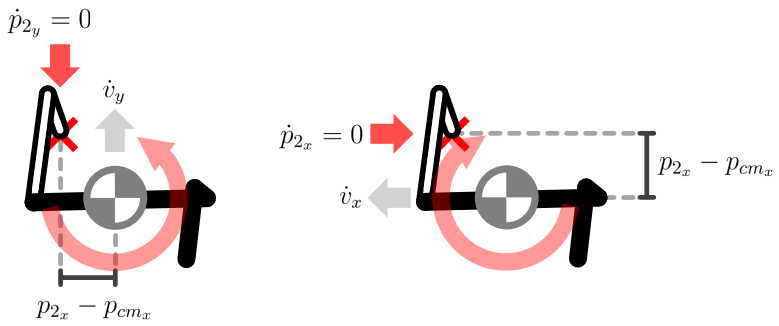

Figure A.9: Impulse-based yaw moments at impact. Impulses occur in the sagittal (left) and frontal (right) planes at impact. This causes yaw moments that are determined by the kinetic energy dissipated when the previous swing foot suddenly stops, $\dot{p}_2 = 0$, and the offset distance of the dissipated energy (e.g. $p_{2_x} - p_{cm_x}$).

A.5 Causes of Yaw Motion

Yaw motion occurs when a biped experiences a transverse-plane moment with ground contact conditions that enable rotation. Bipeds can easily yaw if using point feet, if the stance foot is high centered on an uneven surface, or if walking on unstable terrain. During these conditions, a motion as simple as accelerating the swing leg forward can trigger flywheel-like dynamics that rotate the biped. However, if a bipeds legs are relatively light, its walking gait is designed with sufficient vertical GRF, and its stance foot has multiple contact points with the ground, it is unlikely that yaw will occur throughout the majority of the walking gait. However, even well-designed bipeds can be susceptible to yaw at or around impact, when the weight of the biped is transitioning between feet and sudden opposing forces can result in a significant yaw moment.

Impulses at impact can cause yaw motion, as depicted in Figure A.9. Rapid changes in velocity occur when the swing foot strikes the ground, which can cause impulsive forces that overwhelm the frictional forces that normally keep the biped from rotating. This effect can be intensified because the stance foot is in the process of transferring weight to the previous swing foot, which means sufficient normal forces to prevent yaw motion are not necessarily guaranteed.

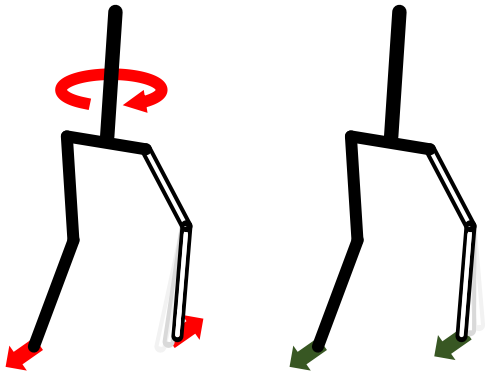

Figure A.10: Double-support GRF for 3D walking. Without swing leg retraction, the x-offset y-direction GRF of each foot are opposed at impact, which causes a yaw moment (left). Swing leg retraction can limit this effect (right).

Opposing GRF during double support can also cause yaw motion, as shown in Figure A.10 left. Opposing GRF can be difficult to avoid during double support because the control inputs for each leg change as they transition between steps. One way to minimize opposing y-direction GRF in double support is to use swing leg retraction, which occurs naturally in bipedal and four-legged animals [124]. When using swing leg retraction, the y-direction control inputs for the swing leg are consistent before and after impact. The swing leg will also generate GRF that are consistent with the previous stance leg during double support, as shown in Figure A.10 right. Additional strategies for managing yaw are discussed in Section A.6.

A.6 Unactuated Yaw Control Strategies

Yaw control enables a biped to follow specific headings and reject asymmetric disturbances that cause the biped to turn. Asymmetric disturbances include: consistent lateral ground height differences between right and left stance; constant directional forces on the center of mass; and, for a physical robot, asymmetric physical components caused by manufacturing errors and fatigue and an asymmetric mass distribution. The ideal method of turning a biped is using actuators intended for yaw control

Figure A.11: Foot designs for various levels of yaw restriction. Toroidal feet approximate point feet and provide negligible yaw friction (top right), prosthetic feet add sporadic yaw friction and toe roll (left), and the flexible two-contact-point feet restrict yaw rotation but enable pitch and roll (bottom right).

(e.g., hip or foot rotation). However, even if explicit yaw actuation is unavailable, it is still possible to manage or even control yaw through other means. Unactuated yaw control strategies include: using a foot design that restricts yaw motion through friction, as shown in Figure A.11; adding intentional yaw moments by adjusting GRF during double support; and making adjustments to the swing foot position to change impulses and yaw moments at impact.

One example underactuated yaw control method varies impact conditions and step length to follow a desired heading. When there is a change in desired heading or external disturbances turn the biped, errors between the average heading over two steps and the desired heading are used as an output error for a discrete PID controller. The yaw control input consists of step-to-step feedforward torques on both hips, which vary the impact yaw moments and step length of each leg. Because yaw is unconstrained, the yaw controller is able to turn the walking trajectory and converge on the desired heading. This yaw control method has been applied in simulation to a rigid point-foot model and a compliant ground model with toroidal feet. The turning radius of the yaw controller and an example navigation path are shown in Figure A.12.

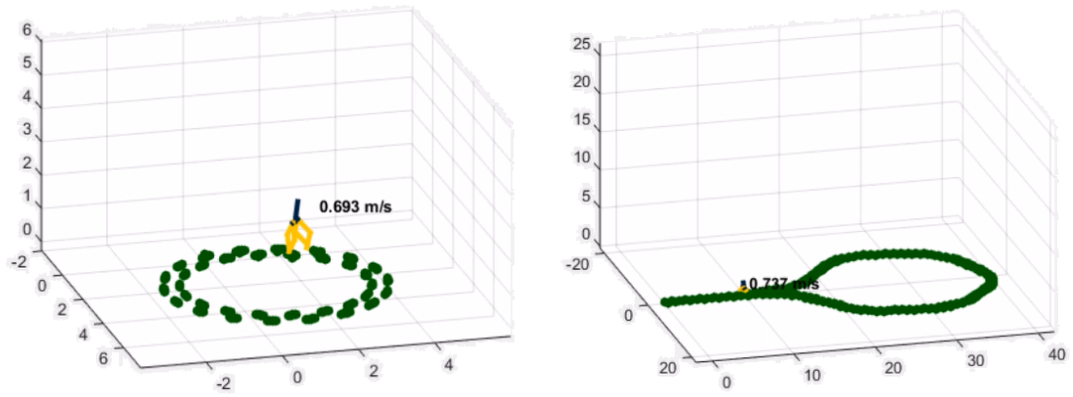

Figure A.12: Yaw control walking trajectories. The biped walks in a controlled turn radius (left) and returns on its outgoing path after following a set trajectory (right).

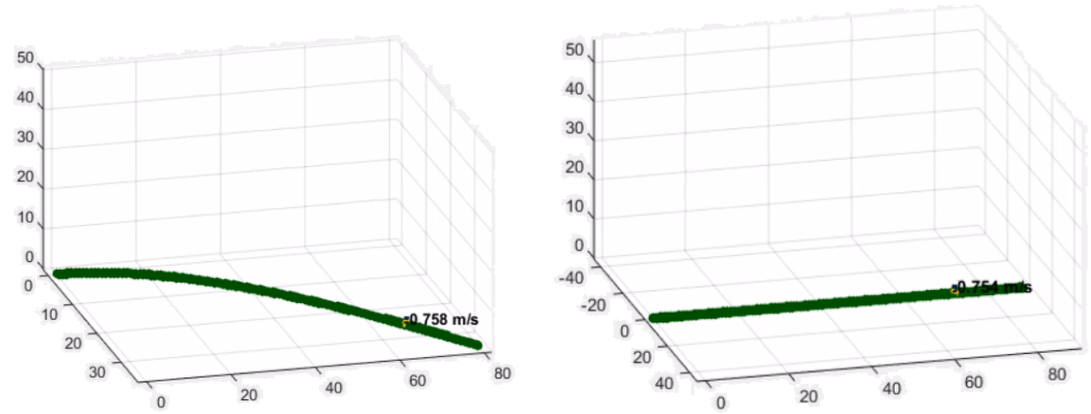

Figure A.13: Yaw control for attenuating disturbances. A constant lateral force is applied at 45 degrees relative to the initial trajectory. This disturbance causes the nominal controller to converge to the force direction (left), but adding yaw control allows the biped to stay on the initial trajectory despite the disturbance (right).

The added yaw controller enables a consistent heading despite asymmetric terrain disturbances and directional forces applied to the biped, as shown in Figure A.13.

APPENDIX B

Reinforcement Learning-Based Feedforward Control

After developing a bipedal walking control policy that rejects a range of terrain disturbances, we seek to augment our control approach using reinforcement learning to navigate terrains previously unnavigable. Control can consist of a combination of feedback and feedforward control. Feedback control is based on generating actuation to correct for errors between desired and actual walking trajectories. Conversely, one method of feedforward control results from the inverse dynamics of the desired walking trajectory for a robot. Although feedforward control based on inverse dynamics is quite accurate in simulation, it is less reliable in implementation due to the difficulty of exactly modeling an experimental robot, such as MARLO. Because of this limitation, actual experiments are often performed using only feedback control. In this work, we implement a reinforcement learning approach to find a successful feedforward control policy that works independently of any physical robot model, and works with the initial control policy to successfully traverse previously unreachable terrains. A video demonstration of this work is available at [38].

B.1 Machine Learning and Bipedal Locomotion

Reinforcement learning has been used by many researchers for the development and improvement of robot control policies, as shown in [71] and citations therein. Many of these studies specifically address bipedal walking [9, 133]. In this work we use reinforcement learning to find a feedforward control policy that improves upon an initial feedback control policy to navigate uphill terrain that was previously unreachable.

Factors that separate this work from other reinforcement learning-based walking research include our dynamic form of locomotion, the magnitude of terrain disturbances we consider, and the combination of feedback and feedforward control we use during policy search. Many researchers maintain static stability while walking with either the use of more than two legs (e.g., [72, 74, 75, 81]) or the use of Zero Moment Point calculations to maintain the center of mass within the polygon of foot contact [28, 78]. Our robot uses dynamic walking on point feet. Dynamic walking allows a more human-like, energy efficient walking gait that is less dependent on flat terrain surfaces; however, having point feet also make static walking impractical, if not impossible. Many reinforcement learning researchers that have used the dynamic walking approach have done so for either flat terrain or terrain with lower levels of terrain disturbances [92, 122]. In this work, our biped traverses through terrain with up to 7 cm disturbances (i.e., more than 9 degrees of slope) between adjacent steps.

One of the challenges of applying machine learning to robotics is the curse of dimensionality, as discussed in [71]. Parameterization of control variables is an effective way to make policy search computationally feasible [72, 78, 122]. Cublic splines, in particular, can be useful for completing trajectory planning in real time [75, 116]. In this work, we use Bézier parameters in a polynomial function that evolves with the absolute stance leg angle of the robot and maps a discrete set of parameters to a continuous trajectory of feedforward actuator profiles, as shown in Section B.2.

We perform reinforcement learning through a policy search algorithm using Bézier parameters. Many policy gradient approaches have been used successfully in robotics [72, 75, 103, 116, 129]. Because the gradient can be difficult to solve analytically, robotics researchers often use approximations of gradients for policy search using finite policy differences, as in [72, 129], or even just a signed derivative, as in [74]. Other interesting policy search methods include genetic algorithms that avoid local optimality in [141, 143] and methods in [78] that use an evolving policy parameterization that starts simple and then increases in complexity. Although all of these approaches are feasible options, for ease of implementation in the current work, policy search is performed using `fmincon` in MATLAB [88].

The idea of initializing a new component of control at zero and then performing a policy search on top of a functioning controller was originally motivated by [130]. Using reinforcement learning, Tedrake augments a control policy developed for a simulated one legged hopper. Starting from zero augmented control, he performs gradient descent using the Downhill Simplex method to reach his final control policy. Although his final policy behaved in an unexpected way, it was more robust than the initial control policy alone.

We extend this work to a higher dimensional bipedal walker and implement the new control policy strictly as feedforward control that operates independently of the physical robot model. This approach has many advantages. First, initializing at the nominal feedback control policy means learning starts from a meaningful solution and can only improve with policy changes to the feedforward control from the starting point. Second, because the feedback controller commands to desired trajectories proportionally with deviations, initially, the feedforward control will be unable to have the effect of removing the robot so far from a desired trajectory that there is no longer meaningful walking. However, as the feedforward control learns policies that operate better than the initial control policy alone, it will grow in magnitude and develop new

gaits that work better for the terrains it traverses. This is somewhat how the supervisory control in [117] works, although in our case, the feedback controller initially acts like a supervisor by design and the feedforward controller receives validation from the overall reward function alone.

The reward function we use is distance the robot is able to traverse through randomly built terrains. We use the average reward setting from step to step to avoid greedy behavior for a good transient and encourage overall stable behavior. The simplicity of this reward function is similar to [72], where overall trial velocity was used as a single measure of reward. Because reinforcement learning does not need an exact model [128], it can guide policy search through complicated control spaces without requirements of how the controller improves; it only requires that the policy does improve. Although our current improvements derive from a single measure of walking distance, the mechanism we derive in Section B.2 can incorporate more rewards from step to step in the future.

B.2 Reinforcement Learning-Based Feedforward Control

While the initial feedback control policy, $S_{4\text{cm}}$ from Chapter IV, is robust for a bounded range of terrain disturbances, there are still terrains that this controller is unable to traverse. We augment the feedback controller for planar walking with a reinforcement learning-based feedforward control policy. The complete control policy is

$$\pi(s) = \pi_{nominal}(s) + \pi_{RLFF}(s) = u, \tag{B.1}$$

where $\pi_{nominal}(s)$ is the initial feedback control policy shown in Figure B.1, $\pi_{RLFF}(s)$ is the reinforcement learning-based feedforward (RLFF) control policy, and $u \in \mathbb{R}^4$ is the resulting actions or command torques. The complete state description is defined

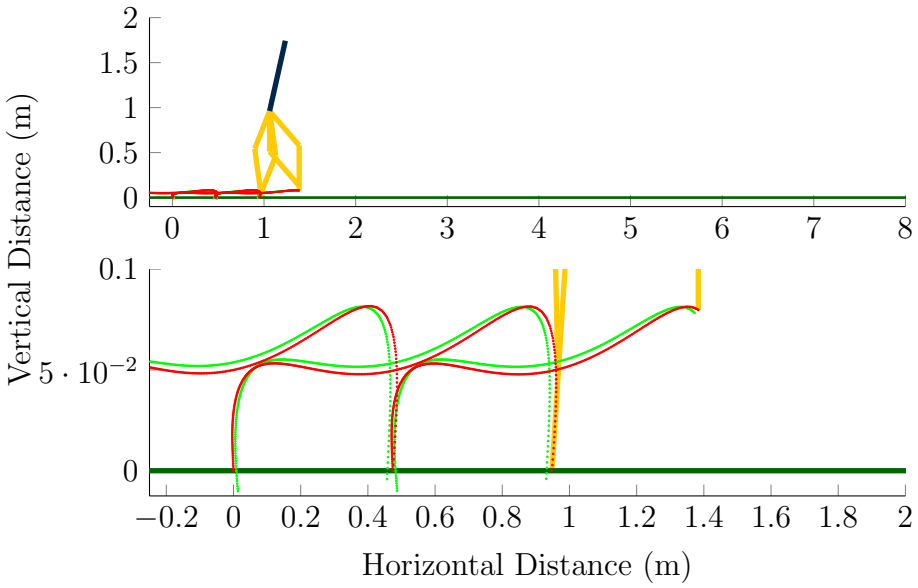

Figure B.1: Desired (green) and actual (red) trajectories of swing foot for nominal feedback control. The initial feedback control policy, $\pi_{nominal}$, is dependent on errors from the desired trajectory to generate actions.

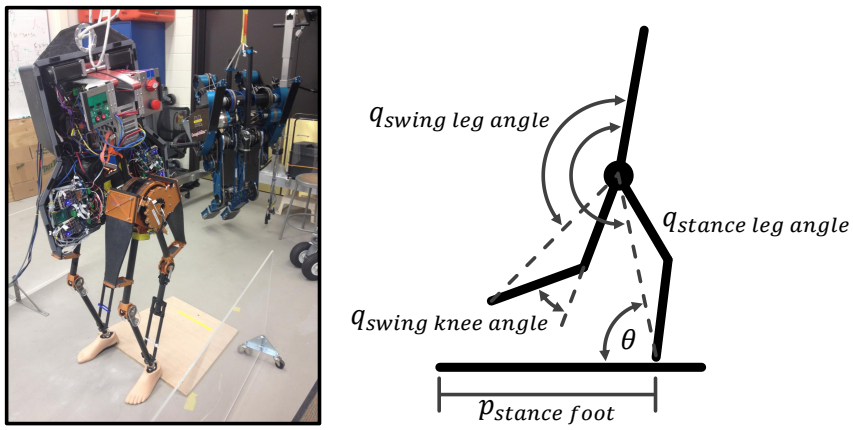

Figure B.2: Robot MARLO and model MARLO. Left, University of Michigan bipedal robots MARLO (front) and MABEL (back). Right, state description for planar model of MARLO used in this work.

as

$$s := \left[\theta,\ q_{stance\ leg\ angle},\ q_{swing\ leg\ angle},\ q_{stance\ knee\ angle},\ q_{swing\ knee\ angle},\ p_{stance\ foot}\right]', \tag{B.2}$$

where θ is the angle between the global horizontal axis and the stance leg, $p_{stance\ foot}$ is the global horizontal position of the stance foot, and the remaining q coordinates describe the internal controlled states of the walking robot shown in Figure B.2.

We parameterize the feedforward control policy with sixth order Bézier polynomials as

$$\pi_{RLFF,i}(s_{phase}) := \sum_{k=0}^{5} \alpha_{i,k} \frac{5\,!}{k\,!(5-k)\,!} s_{phase}^{k} (1 - s_{phase})^{5-k} \tag{B.3}$$

$$\pi_{RLFF}(s_{phase}) = \begin{bmatrix} \alpha_{1,0}(1 - s_{phase})^5 + \alpha_{1,1} 5 s_{phase}(1 - s_{phase})^4 + \cdots + \alpha_{1,5} s_{phase}^5 \\ \vdots \\ \alpha_{4,0}(1 - s_{phase})^5 + \alpha_{4,1} 5 s_{phase}(1 - s_{phase})^4 + \cdots + \alpha_{4,5} s_{phase}^5 \end{bmatrix}, \tag{B.4}$$

where $s_{phase}(s) = \frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}}$, hence $\pi_{RLFF}(s_{phase}) =: \tilde{\pi}_{RLFF}(s)$. s_{phase} and $\alpha \in \mathbb{R}^{4\times 6}$ completely define the feedforward torque values used to augment the initial controller at each stage of walking. One advantage of using Bézier polynomials, in addition to those discussed in Section B.1, are that the resulting feedforward torques are smooth during each step. We apply the bound $\alpha_{i,j} \in [-6, 6]\ \forall i, j$ to account for physical limitations of MARLO's motors.

The reward function is defined as

$$R(s) := p_{swing\ foot} - p_{stance\ foot}, \tag{B.5}$$

and can best be understood as the forward progress made from the previous step toward the next step. To avoid any local minimum around a policy of simply extending

the swing foot out as far as possible while not actually walking, $R(s)$ is set to zero if the biped falls. A fall results from violating frictional constraints in the simulator dependent on ground reaction forces and the coefficient of friction. A fall can also result from losing momentum and falling backward. The value function we seek to maximize, $V(s)$, is defined as

$$V(s) := \sum_{n=0}^{N} R(s_n), \tag{B.6}$$

where N is the total number of completed steps over a terrain and $R(s_n)$ is the reward from each successful step n. Note that V will change with the control policy being implemented.

We initialize the Bézier parameters as zero and apply a gradient descent method to maximize the distance the control policy is able to walk across a given terrain. We do this using the interior-point algorithm built into the MATLAB function, `fmincon` [88]. This algorithm computes a numerical approximation of the Hessian, or matrix of second-order derivatives, in order to find the minimum of the function provided. Other algorithms such as `fminsearch` and `fminunc` were tried with less success.

B.3 Results

The initial results of the reinforcement learning-based feedforward control policy are promising. The control policy that uses RLFF, π, is able to outperform the initial feedback only controller, $\pi_{nominal}$, on 4/5 of the training terrains. Using the RLFF augmented policy, the previously unnavigable portions of the terrain are much easier for the biped to walk through, as shown in Figure B.3. This is reflected by the almost order of magnitude difference in policy values between the initial and final control policies in Tables B.1 and B.2. However, the final control policy does have some shortcomings. In the remainder of this section, we explain how we generate

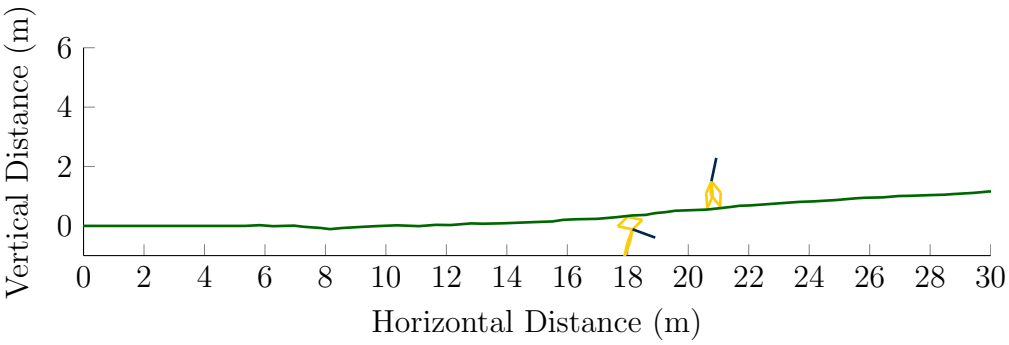

Figure B.3: Comparison of initial control and final control on Training Terrain 1. The initial control solution falls at 18.0 m while the control policy using RLFF keeps walking to 153.6 m on the same terrain.

the different terrains and then provide analysis on how the final RLFF control policy changes the initial policy.

For the training and test terrain, we built terrain that is at first navigable by the initial feedback controller, but has gradually increasing disturbance levels, thus causing the robot to fail. By using this progressive task framework, the inability to navigate the same terrain as the initial controller results in a smaller reward and outperforming the initial controller results in a greater reward. With multiple random terrain profiles, we have both training data for finding the policy and test data for evaluation. Using multiple terrain profiles for the value function during policy search also helps to avoid over-fitting.

The terrain built for policy search and testing are constructed in the following way. Terrain is a series of i connected slopes with horizontal intervals of h_i between vertical displacements of v_i. The random training and test terrains are built to be flat for the first horizontal 5 m, have uniformly random ± 4 cm range displacements the next 5 m (i.e., $v_i \in [-4, 4]$ cm), random ± 6 cm range displacements the next 5 m, and finally have random 6 cm range displacements for the remainder of the 1000 step trial, but all uphill (i.e., $v_i \in [0, 6]$ cm). The horizontal intervals between vertical displacements are also uniformly random, but between 0.25 and 0.75 m. With $h_i \in [0.25, 0.75]$ m and

Table B.1: Initial and final control policy values on training and test terrains.

Policy Value Comparison							
	Disturbance Height (cm) [a]			$\pi_{nominal}$		π	
Terrain	Mean	Std. Dev.	Max.	Failure	Value	Failure	Value
Training Terrain 1	2.32	1.68	6.80	Fall	18.0	Slip	153.6
Training Terrain 2	2.06	1.97	7.41	Fall	19.4	Slip	73.7
Training Terrain 3	0.18	2.44	4.64	Fall	18.4	Slip	8.7
Training Terrain 4	2.29	1.69	7.25	Fall	19.4	Slip	172.0
Training Terrain 5	2.38	1.47	7.02	Fall	22.3	None	425.0 [b]
Test Terrain 1	2.11	1.88	8.12	Fall	17.5	Slip	111.1
Test Terrain 2	0.18	2.14	4.59	Fall	18.0	Slip	11.6
Test Terrain 3	2.27	1.66	6.95	Fall	14.7	Slip	192.1
Test Terrain 4	0.41	2.23	5.77	Fall	19.9	Slip	13.7
Test Terrain 5	2.34	1.58	7.52	Fall	22.7	Slip	299.4

[a]Terrain Disturbance Height statistics based on longest policy travel distance.
[b]Value could be higher, but trial reached 1000 step limit.

Table B.2: Initial and final control policy values from reinforcement learning. Values are calculated using the average control policy value of all terrains for a particular set. Alternative terrains are generated to analyze conditions other than those used during policy search.

Reinforcement Learning Policy Values		
Terrain	$\pi_{nominal}$	π
Training	19.5	166.6
Test	18.6	125.6
Alternative Terrain Policy Values		
Non-Uphill	49.1	17.6

an average step distance of about 0.43 m, the resulting single-step terrain disturbance heights experienced by the robot during simulation can be greater than v_i, as shown in Table B.1.

When evaluating the margins between the two policies in Table B.1, it is apparent that when $\pi_{nominal}$ does outperform π, it is because π failed on the initial non-uphill terrain with disturbances. To verify this behavior, we built another set of 100 terrains that are random ± 6 cm range displacements without the uphill constraint. The results in Table B.2 indicate that $\pi_{nominal}$ does indeed outperform π on this type of terrain.

After evaluating the changes in walking trajectory, work input, and velocity data of the two policies, the cause of their different behavior becomes apparent. When looking at the differences in trajectories, it is evident that π learned to lift the swing leg higher and take shorter steps compared with the nominal trajectory, as shown in Figure B.4.

When looking at the differences in work input, it is evident that π injects more energy into the robot while walking than $\pi_{nominal}$. As shown in Figure B.5, this also results in a higher velocity walking gait. This explains why π was able to continue on the random terrain that had an uphill bias whereas $\pi_{nominal}$ would lose momentum after a period of time and fall backwards.

The downside of this higher velocity gait is that it more often violates the frictional constraints of the simulator, causing a slip failure. If the robot makes it to the uphill portion of the terrain, this is not as much of a concern. However, when walking on the non-uphill terrain, it is evident that $\pi_{nominal}$ is a better controller, because it maintains a lower velocity that is less likely to violate the frictional constraints of the simulator when walking on the random downhill portions of the terrain.

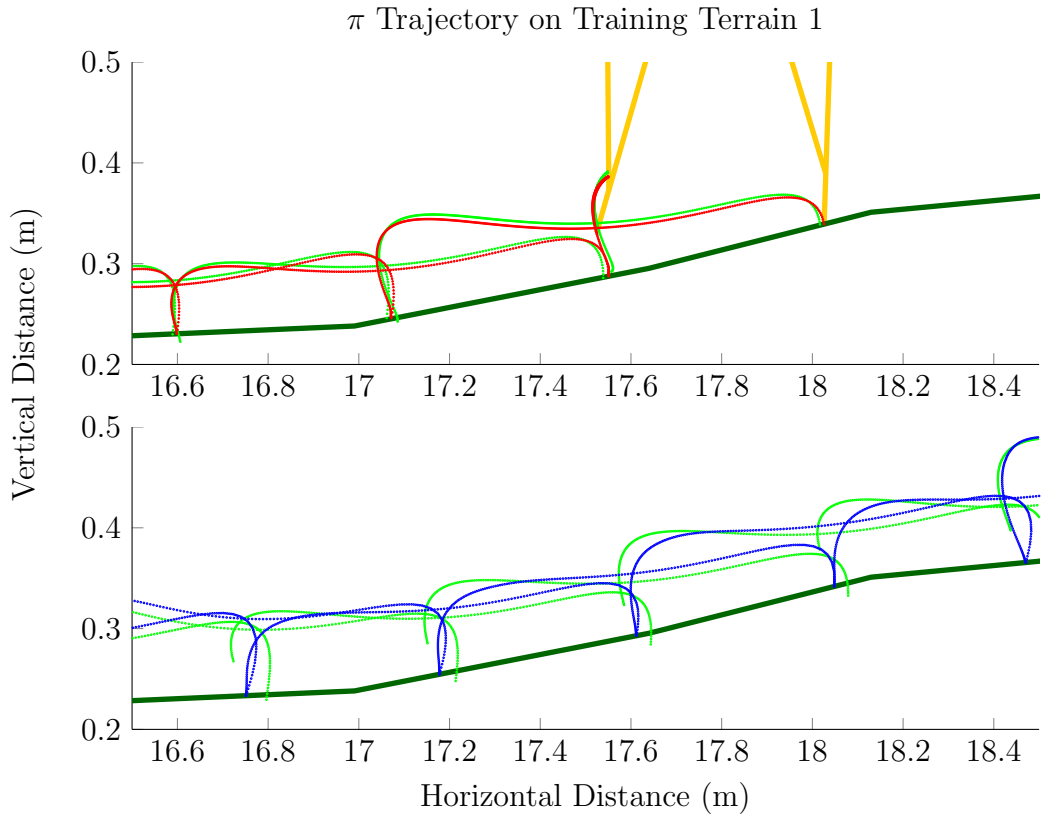

Figure B.4: Desired (green) and actual swing foot trajectories for the nominal feedback control (red) and the final control policy (blue). The final control policy learned to take higher and shorter steps than the nominal desired trajectory. This type of trajectory is better suited for walking uphill.

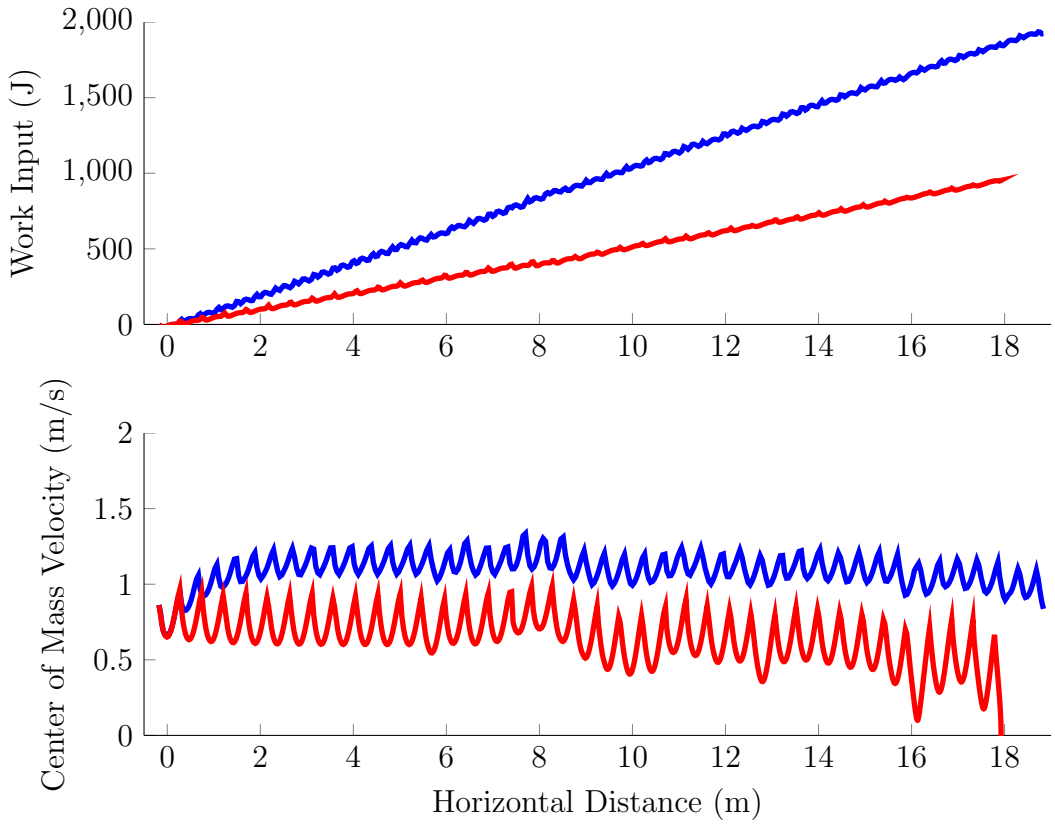

Figure B.5: Initial control policy (red) and final control policy (blue) actuator work input (top) and resulting center of mass velocity (bottom) on Training Terrain 1. The result of the learned feedforward control policy is to inject more energy into the system, resulting in a higher velocity walking gait. This gait is more robust to navigating uphill terrain, but is more prone to slipping when not traveling uphill.

B.4 Discussion

Our implementation of reinforcement learning was successful at improving the task it was assigned to learn. From a framework design standpoint, some improvements can be made to ensure that the learned policy will perform better than the nominal policy in more settings. One way to avoid a learned policy that violates frictional constraints is to change the training terrain. The new terrain could have a longer stretch of non-uphill terrain before the uphill portion of the task or even include stretches of downhill terrain in some of the terrain profiles to emphasize more general walking stability. Another approach would be to incorporate a step to step reward proportional to the margin from the frictional constraints of the simulation to include an incentive during policy search for a low slipping probability. A final proposed solution would be to use the final control policy only when the robot is slowing down, and then switch back to the nominal policy afterward. This hybrid control would be easy to develop and test in simulation and then could be evaluated against the two control policies compared in this current work.

We are also interested in seeing how the learned control policy performs during actual experiments. With experiments, it would be possible to remove some simulation inaccuracies with supervised learning for more meaningful simulations thereafter [30]. It may also be possible to perform learning with an alternative framework during experiments directly, as has been done by other researchers [72, 75, 95, 116, 129], thereby negating the necessity for an accurate simulation model.

This initial work has been a success. The policy search was given the task of improving walking on a given terrain type, and the controller using reinforcement learning-based feedforward performed with an 854% and 675% increase in value over the initial controller on the training and test terrains, respectively. We are excited to validate these results in experiments, and, using this developed framework, extend this work to incorporate a new policy search that works for more general sets of

unknown terrain.

APPENDIX C

Cost of Transport Derivation

The cost of transport (COT) is an alternative metric to the mechanical cost of transport (MCOT) for evaluating locomotion efficiency. Here, we make the distinction between COT calculated instantaneously as

$$\mathrm{COT}_P := \frac{P}{Mgv_y}, \tag{C.1}$$

where P is power consumption at forward velocity v_y, and COT calculated over a period of time as

$$\mathrm{COT} := \frac{E}{Mgd_y}, \tag{C.2}$$

where E is the energy used to travel distance d_y. (C.2) is more useful to the current work than (C.1), because it accounts for local changes that occur for non-periodic conditions, such as when traversing almost any outdoor environment.

Based on experiments in the lab with supply-power measurements, a conservative estimate for COT can be calculated using data that is also available during outdoor

experiments. The conservative estimate uses the absolute MCOT, calculated as

$$\mathrm{MCOT}_{\mathrm{abs.}} := \frac{1}{Mgd_y} \int_0^{T_0} \sum_{i=1}^{6} |u_i \dot{q}_{m,i}| dt, \tag{C.3}$$

which includes negative actuator work, as in [56, Eqn. (4)]. As in [139, Eqn. (23)], a fixed power cost, P_{base}, is added to account for ancillary electronics. Based on the highest measurement for power consumption of on-board sensing and computation on MARLO, $P_{\mathrm{base}} = 131.7$ W. The resulting conservative estimate for COT is defined as

$$\mathrm{COT}_{\mathrm{abs.}} := \frac{1}{Mgd_y} \int_0^{T_0} P_{\mathrm{base}} + \sum_{i=1}^{6} |u_i \dot{q}_{m,i}| dt, \tag{C.4}$$

which is consistently higher than the actual measured power consumption, because it does not consider any negative-work regenerative capabilities of the amplifiers and batteries.

For comparison, a liberal estimate based on the regenerative COT is defined as

$$\mathrm{COT}_{\mathrm{regen}} := \frac{1}{Mgd_y} \int_0^{T_0} P_{\mathrm{base}} + \sum_{i=1}^{6} u_i \dot{q}_{m,i} dt, \tag{C.5}$$

which is consistently lower than the actual measured power consumption due to regenerative losses in hardware. Based on power experiments with nominal periodic motion, the average power consumption based on (C.4) is about 14% higher than the actual measured values, while the average power consumption based on (C.5) is about 30% lower.

APPENDIX D

Virtual Spring

Following outdoor experiments that had severe impacts on concrete, a virtual spring was implemented on MARLO to maintain more uniform ground reaction forces. The virtual spring functions in place of the virtual constraint for stance knee angle, and is derived as

$$q_{vs} = 2\ell cos(\frac{q_{KA,ST}}{2}) \tag{D.1}$$

$$\dot{q}_{vs} = -\dot{q}_{KA,ST}\ell sin(\frac{q_{KA,ST}}{2}) \tag{D.2}$$

$$F_{vs}(q,\dot{q},\beta) = k_{vs}(\beta)(q_{vs,Rest}(\tau,\beta) - q_{vs}) + b_{vs}(\beta)\dot{q}_{vs} \tag{D.3}$$

$$u_{vs}(q,\dot{q},\beta) = F_{vs}(q,\dot{q},\beta)\ell sin(\frac{q_{KA,ST}}{2}), \tag{D.4}$$

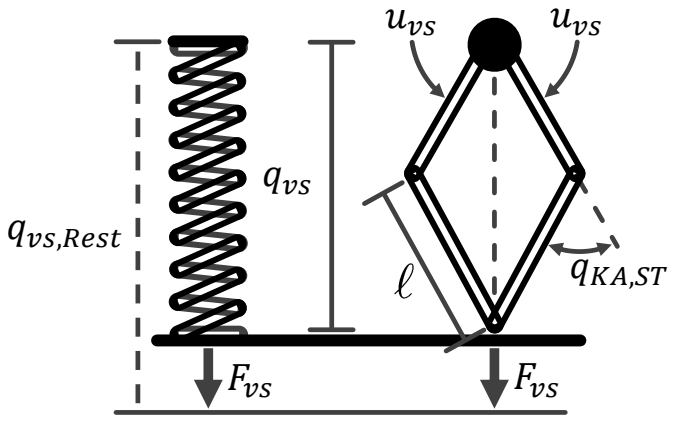

Figure D.1: Kinematic model for virtual spring.

Figure D.2: Virtual spring implementation for planar MARLO on uneven terrain. When using the virtual spring, MARLO is able to traverse up (top left) and down (right) 9 cm changes in terrain height, as well as random terrain (bottom left).

where q_{vs} is the virtual spring length, ℓ is the average link length, $q_{KA,ST}$ is the stance knee angle, F_{vs} is the effective force of the virtual spring, k_{vs} is the virtual spring constant, $q_{vs,Rest}$ is the "rest length" of the virtual spring, b_{vs} is the virtual-spring damping term, and u_{vs} is the torque required to generate F_{vs}, as depicted in Figure D.1. In practice, if $q_{vs,Rest} - q_{vs} < 0$, k_{vs} is set to 0 in (D.3) to eliminate any tensile virtual-spring forces. This virtual-spring implementation is similar to [98], but with the following improvements: (1) all parameters dependent on β are determined using the robust control optimization introduced in Chapter IV and (2) the rest length of the virtual spring, $q_{vs,Rest}$, evolves with the gait phasing variable, τ.

The planar implementation of the virtual spring enables excellent shock absorption

following large downhill changes in terrain height, as found in [98]. In addition, we found the virtual spring to be effective for walking uphill and traversing the uneven, unstable terrain shown in Figure D.2.

Although the virtual spring worked well for three-dimensional (3D) walking in simulation, this success did not translate to the physical 3D robot. Hardware differences between the two legs (e.g., differences in friction and motor performance) caused uneven forces between right and left stance, which destabilized the frontal-plane dynamics. Essentially, when using the virtual spring, 3D MARLO experiences unbalanced ground reaction forces and eventually deviates to one side, whereas planar MABEL are MARLO were not effected by this lateral instability.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Agility robotics. http://www.agilityrobotics.com, 2016.

[2] Atlas - the agile anthropomorphic robot. http://www.bostondynamics.com, 2016.

[3] Darpa robotics challenge trials. http://archive.darpa.mil/roboticschallengetrialsarchive, 2016.

[4] E. Ackerman. Durus: Sri's ultra-efficient walking humanoid robot. http://spectrum.ieee.org/automaton/robotics/humanoids/durus-sri-ultra-efficient-humanoid-robot, June 2015.

[5] P. G. Adamczyk and A. D. Kuo. Mechanical and energetic consequences of rolling foot shape in human walking. *The Journal of experimental biology*, 216(14):2722–2731, 2013.

[6] K. Akbari Hamed, B. G. Buss, and J. W. Grizzle. Continuous-time controllers for stabilizing periodic orbits of hybrid systems: Application to an underactuated 3d bipedal robot. In *53rd IEEE Conference of Decision and Control*, 2014.

[7] K. Akbari Hamed, B. G. Buss, and J. W. Grizzle. Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations. 2015.

[8] K. Akbari Hamed and J. W. Grizzle. Event-based stabilization of periodic orbits for underactuated 3-d bipedal robots with left-right symmetry. *Robotics, IEEE Transactions on*, 30(2):365–381, 2014.

[9] H. Benbrahim and J. A. Franklin. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, 22(3):283–302, 1997.

[10] 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*, 33(10):1305–1321, 2014.

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

[12] A. M. Bloch. *Nonholonomic Mechanics and Control*. Springer-Verlag, 2003.

[13] S. Burden, S. Revzen, and S. S. Sastry. Dimension reduction near periodic orbits of hybrid systems. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 6116–6121. IEEE, 2011.

[14] B. G. Buss, K. Akbari Hamed, B. A. Griffin, and J. W. Grizzle. Experimental results for 3d bipedal robot walking based on systematic optimization of virtual constraints. In *American Control Conference*, Boston, MA, USA, July 2016.

[15] B. G. Buss, A. Ramezani, K. Akbari 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*, pages 2529–2536. IEEE, 2014.

[16] S. Butterworth. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.

[17] K. Byl and R. Tedrake. Metastable walking machines. *Int. J. Rob. Res.*, 28:1040–1064, August 2009.

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

[19] C. Chew, J. Pratt, and G. Pratt. Blind walking of a planar bipedal robot on sloped terrain. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 381–386 vol.1, 1999.

[20] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

[21] G. Council, S. Yang, and S. Revzen. Deadbeat control with (almost) no sensing in a hybrid model of legged locomotion. In *Advanced Mechatronic Systems (ICAMechS), 2014 International Conference on*, pages 475–480. IEEE, 2014.

[22] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, Upper Saddle River, NJ, third edition, 2005.

[23] X. Da, O. Harib, R. Hartley, B. Griffin, and J. Grizzle. From 2d design of underactuated bipedal gaits to 3d implementation: Walking with speed tracking. *IEEE Access*, PP(99):1–1, 2016.

[24] H. Dai and R. Tedrake. Optimizing robust limit cycles for legged locomotion on unknown terrain. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1207–1213, 2012.

[25] H. Dai and R. Tedrake. L2-gain optimization for robust bipedal walking on unknown terrain. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.

[26] J. Engelberger. Economics should be main basis for using robots. *Production Engineer*, 62(12):25–, 1983.

[27] J. Englsberger, C. Ott, and A. Albu-Schaffer. Three-dimensional bipedal walking control based on divergent component of motion. *Robotics, IEEE Transactions on*, 31(2):355–368, 2015.

[28] M. J. Er and Y. Zhou. Intelligent fuzzy q-learning control of humanoid robots. In *Advances in Neural Networks–ISNN 2005*, pages 216–221. Springer, 2005.

[29] M. Ernst, H. Geyer, and R. Blickhan. Spring-Legged Locomotion on Uneven Ground: a Control Approach To Keep the Running Speed Constant. *Proceedings of the Twelfth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pages 639–644, 2009.

[30] A. Farchy, S. Barrett, P. MacAlpine, and P. Stone. Humanoid robots learning to walk faster: From the real world to simulation and back. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 39–46. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[31] R. Full and D.E. Koditschek. Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202:3325–3332, December 1999.

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

[33] G. Garofalo, C. Ott, and A. Albu-Schaffer. Walking control of fully actuated robots based on the bipedal slip model. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1456–1463. IEEE, 2012.

[34] A. Goswami, B. Espiau, and A. Keramane. Limit cycles in a passive compass gait biped and passivity-mimicking control laws. *Autonomous Robots*, 4(3):273–86, 1997.

[35] R. D. Gregg, T. Lenzi, L. J. Hargrove, and J. W Sensinger. Virtual constraint control of a powered prosthetic leg: From simulation to experiments with transfemoral amputees. *IEEE Transactions on Robotics*, 2014.

[36] R. D. Gregg, E. J. Rouse, L. J. Hargrove, and J. W. Sensinger. Evidence for a time-invariant phase variable in human ankle control. *PLoS ONE*, 9:1–13, 02 2014.

[37] B. Griffin. Code for 3d control design and simulation. `http://www.griffb.com/publications/`, July 2016.

[38] B. Griffin. Reinforcement learning-based feedforward control to improve walking over unknown terrain. YouTube Video: `https://youtu.be/9u48hzz87Aw`, July 2016.

[39] B. Griffin and J. Grizzle. Walking gait optimization for accommodation of unknown terrain height variations. YouTube Video: `http://youtu.be/OmhzbsCDN34`, September 2014.

[40] B. Griffin and J. Grizzle. Nonholonomic virtual constraints for dynamic walking. In *IEEE Conf. on Decision and Control*, pages 4053–4060, 2015.

[41] B. Griffin and J. Grizzle. Nonholonomic virtual constraints for dynamic walking. Online Video: `http://web.eecs.umich.edu/faculty/grizzle/vids/20150402NHVCSFPDemo.mp4`, 2015.

[42] B. Griffin and J. Grizzle. Walking gait optimization for accommodation of unknown terrain height variations. In *American Control Conference (ACC), 2015*, pages 4810–4817, Chicago, IL, USA, July 2015.

[43] B. Griffin and J. Grizzle. Nonholonomic virtual constraints and gait optimization for robust walking control. *The International Journal of Robotics Research* Submitted, 2016.

[44] B. Griffin and J. Grizzle. Nonholonomic virtual constraints and gait optimization for robust walking control. YouTube Video: `https://youtu.be/81I0H5d0tUM`, June 2016.

[45] J. A. Grimes and J. W. Hurst. The design of ATRIAS 1.0 a unique monopod, hopping robot. *Climbing and walking Robots and the Support Technologies for Mobile Machines, International Conference on*, 2012.

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

[47] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames. Models, feedback control, and open problems of 3d bipedal robotic walking. *Automatica*, 50(8):1955–1988, 2014.

[48] J. W. Grizzle, J. H. Choi, H. Hammouri, and B. Morris. On observer-based feedback stabilization of periodic orbits in bipedal locomotion. In *Proceedings of Methods and Models in Automation and Robotics (MMAR 2007)*, Szczecin, Poland, 2007.

[49] J.W. Grizzle, Jonathan Hurst, B. Morris, H. Park, and K. Sreenath. Mabel, a new robotic bipedal walker and runner. In *American Control Conference, 2009. ACC '09.*, pages 2030–2036, 2009.

[50] S. J. Hasaneini, C. J. B. Macnab, J. E. A. Bertram, and A. Ruina. Seven reasons to brake leg swing just before heel strike. In *Dynamic Walking 2013*, Carnegie Mellon University, Pittsburgh, Pennsylvania, June 2013. Dynamic Walking.

[51] 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 *IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, 2016.

[52] 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 *Intelligent Robots and Systems (IROS 2015), 2015 IEEE/RSJ International Conference on*. IEEE, 2015.

[53] 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*, pages 263–272. ACM, 2014.

[54] 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 (Cat. No.98CH36146)*, volume 2, pages 1321–1326. IEEE, 1998.

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

[56] D. G. Hobbelen and M. Wisse. Ankle actuation for limit cycle walkers. *The International Journal of Robotics Research*, 27(6):709–735, 2008.

[57] D. G. Hobbelen and M. Wisse. Swing-leg retraction for limit cycle walkers improves disturbance rejection. *Robotics, IEEE Transactions on*, 24(2):377–389, 2008.

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

[59] Y. Hurmuzlu, C. Basdogan, and J.J. Carollo. Presenting joint kinematics of human locomotion using phase plane portraits and poincar/'e maps. *Journal of Biomechanics*, 27(12):1495–1499, 1994.

[60] Y. Hürmüzlü and D. B. Marghitu. Rigid body collisions of planar kinematic chains with multiple contact points. *International Journal of Robotics Research*, 13(1):82–92, 1994.

[61] J. Hurst. Ri seminar: Designing robots to walk and run. YouTube Video: `https://www.youtube.com/watch?v=2NVFymqa8qY`, April 2015.

[62] M. Hutter, C. D. Remy, M. A. Hopflinger, and R. Siegwart. Slip running with an articulated robotic leg. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4934–4939. IEEE, 2010.

[63] D. J. Hyun, S. Seok, J. Lee, and S. Kim. High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the mit cheetah. *The International Journal of Robotics Research*, 33(11):1417–1445, 2014.

[64] A. Isidori. *Nonlinear Control Systems: An Introduction*. Springer-Verlag, Berlin, Germany, 3rd edition, 1995.

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

[66] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 239–246. IEEE, 2001.

[67] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4489–4496. IEEE, 2010.

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

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

[70] H. K. Khalil. *Nonlinear Systems - 3rd Edition*. Prentice Hall, Upper Saddle River, NJ, 2002.

[71] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[72] N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2619–2624. IEEE, 2004.

[73] S. Kolathaya and A. D. Ames. Achieving bipedal locomotion on rough terrain through human-inspired control. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pages 1–6. IEEE, 2012.

[74] J. Z. Kolter and A. Y. Ng. Policy search via the signed derivative. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[75] J. Z. Kolter and A. Y. Ng. Task-space trajectories via cubic spline optimization. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1675–1682. IEEE, 2009.

[76] F. H. Kong, A. M. Boudali, and I. R. Manchester. Phase-indexed ilc for control of underactuated walking robots. In *2015 IEEE Conference on Control Applications (CCA)*, pages 1467–1472, Sept 2015.

[77] 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*, 31(9):1094–1113, July 2012.

[78] P. Kormushev, B. Ugurlu, Sylvain Calinon, Nikolaos G Tsagarakis, and Darwin G Caldwell. Bipedal walking energy minimization by reinforcement learning with evolving policy parameterization. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 318–324. IEEE, 2011.

[79] A. D. Kuo. Stabilization of lateral motion in passive dynamic walking. *International Journal of Robotics Research*, 18(9):917–930, 1999.

[80] A. D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124:113–120, 2002.

[81] H. Lee, Y. Shen, C. Yu, G. Singh, and A. Y. Ng. Quadruped robot obstacle negotiation via reinforcement learning. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3003–3010. IEEE, 2006.

[82] D. Luenberger. Observers for multivariable systems. *IEEE Transactions on Automatic Control*, 11(2):190–197, Apr 1966.

[83] A. Majumdar, R. Vasudevan, M. M. Tobenkin, and R. Tedrake. Convex optimization of nonlinear feedback controllers via occupation measures. *The International Journal of Robotics Research*, page 0278364914528059, 2014.

[84] I. R. Manchester, U. Mettin, F. Iida, and R. Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, 2011.

[85] I. R. Manchester and J. Umenberger. Real-time planning with primitives for dynamic walking over uneven terrain. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4639–4646. IEEE, 2014.

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

[87] A. E. Martin, D. C. Post, and J. P Schmiedeler. The effects of foot geometric properties on the gait of planar bipeds walking under HZD-based control. *The International Journal of Robotics Research*, 33(12):1530–1543, 2014.

[88] MathWorks. Constrained nonlinear optimization algorithms. `http://www.mathworks.com/help/optim/ug/fmincon-interior-point-algorithm-with-analytic-hessian.html`, 2014.

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

[90] T. McGeer. Passive dynamic biped catalogue, 1991. In *Proc. 2nd Int. Symp. of Experimental Robotics, Toulouse*, pages 465–90, Berlin, 1991. Springer-Verlag.

[91] L. Menini and A. Tornambe. Reduced-order observers for the velocity estimation of nonlinear mechanical systems subject to non-smooth impacts. In *American Control Conference*, pages 2639–2644, 2002.

[92] J. Morimoto and C. G. Atkeson. Learning biped locomotion. *Robotics & Automation Magazine, IEEE*, 14(2):41–51, 2007.

[93] J. Morimoto, G. Zeglin, and C.G. Atkeson. Minimax differential dynamic programming: application to a biped walking robot. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1927–1932 vol.2, 2003.

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

[95] A. Y. Ng. *Shaping and policy search in reinforcement learning*. PhD thesis, University of California, Berkeley, 2003.

[96] Oregon State DRL. Atrias robot: Battery drain. YouTube Video: `https://www.youtube.com/watch?v=se0nSJMZKcY`, April 2015.

[97] A. Packard, U. Topcu, P. Seiler, and G. Balas. Help on sos. *IEEE Control Systems*, 30(4):18–23, 2010.

[98] H. Park. *Control of a Bipedal Robot Walker on Rough Terrain*. PhD thesis, University of Michigan, 2012.

[99] H. Park, K. Sreenath, J. W. Hurst, and J. W. Grizzle. Identification of a bipedal robot with a compliant drivetrain. *IEEE Control Systems Magazine*, 31(2):63 –88, april 2011.

[100] H. Park, P. Wensing, and S. Kim. Autonomous running jumps over obstacles in the mit cheetah 2. YouTube Video: `https://www.youtube.com/watch?v=_luhn7TLfWU`, May 2015.

[101] H.-W. Park, A. Ramezani, and J. W. Grizzle. A finite-state machine for accommodating unexpected large ground height variations in bipedal robot walking. *IEEE Transactions on Robotics*, 29(29):331–345, 2013.

[102] I. Park, J. Kim, J. Lee, and J. Oh. Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot 3: Hubo). In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 321–326. IEEE, 2005.

[103] J. Peters and S. Schaal. Policy gradient methods for robotics. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2219–2225. IEEE, 2006.

[104] D. C. Post and J. P. Schmiedeler. Velocity disturbance rejection for planar bipeds walking with hzd-based control. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4882–4887, Sept 2014.

[105] 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*, 54(8):1779–1793, August 2009.

[106] G.A. Pratt. Legged robots at mit: what's new since raibert? *Robotics Automation Magazine, IEEE*, 7(3):15–19, 2000.

[107] J. Pratt, M.C. Chee, A. Torres, P. Dilworth, and G. Pratt. Virtual model control: an intuitive approach for bipedal locomotion. *International Journal of Robotics Research*, 20(2):129–143, February 2001.

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

[109] J. E. Pratt, P. D. Neuhaus, M. Johnson, J. Carff, and B. T. Krupp. Toward humanoid robots for operations in complex urban environments, 2010.

[110] J.E. Pratt and R. Tedrake. Velocity-based stability margins for fast bipedal walking. In Moritz Diehl and Katja Mombaur, editors, *Fast Motions in Biomechanics and Robotics*, volume 340 of *Lecture Notes in Control and Information Sciences*, pages 299–324. Springer Berlin Heidelberg, 2006.

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

[112] M. H. Raibert. *Legged robots that balance*. MIT Press, Mass., 1986.

[113] 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*, 136(2), 2014.

[114] H. Razavi, A. M. Bloch, C. Chevallereau, and J. W. Grizzle. Restricted discrete invariance and self-synchronization for stable walking of bipedal robots. In *2015 American Control Conference (ACC)*, pages 4818–4824, July 2015.

[115] J. Rebula, F. Canas, J. Pratt, and A. Goswami. Learning capture points for humanoid push recovery. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 65–72, 2007.

[116] J. W. Roberts, L. Moret, J. Zhang, and R. Tedrake. Motor learning at intermediate reynolds number: experiments with policy gradient on the flapping flight of a rigid wing. In *From Motor Learning to Interaction Learning in Robots*, pages 293–309. Springer, 2010.

[117] M. T. Rosenstein and A. G. Barto. Reinforcement learning with supervision by a stable controller. In *American Control Conference, 2004. Proceedings of the 2004*, volume 5, pages 4517–4522. IEEE, 2004.

[118] A. Saccon, N. van de Wouw, and H. Nijmeijer. Sensitivity analysis of hybrid systems with state jumps with application to trajectory tracking. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 3065–3070, Dec 2014.

[119] C. O. Saglam and K. Byl. Switching policies for metastable walking. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 977–983. IEEE, 2013.

[120] C. O. Saglam and K. Byl. Quantifying the trade-offs between stability versus energy use for underactuated biped walking. In *IEEE/RSJ Conference on Intelligent Systems and Robots*. IEEE, September 2014.

[121] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent ASIMO: system overview and integration. In *IEEE/RSJ Conference on Intelligent Systems and Robots*, pages 2478–83, Lausanne, Switzerland, 2002.

[122] E. Schuitema, D. G. Hobbelen, P. P. Jonker, M. Wisse, and J. D.l Karssen. Using a controller based on reinforcement learning for a passive dynamic walking robot. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 232–237. IEEE, 2005.

[123] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim. Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3307–3312. IEEE, 2013.

[124] A. Seyfarth, H. Geyer, and H. Herr. Swing leg retraction: A simple control model for stable running. *Journal of Experimental Biology*, 206:2547–2555, 2003.

[125] C. Shih, J. W. Grizzle, and C. Chevallereau. From Stable Walking to Steering of a 3D Bipedal Robot with Passive Point Feet. *Robotica*, 30(07):1119–1130, December 2012.

[126] A. S. Shiriaev, L. B. Freidovich, and M. W. Spong. Controlled invariants and trajectory planning for underactuated mechanical systems. *Automatic Control, IEEE Transactions on*, 59(9):2555–2561, 2014.

[127] 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. *International Journal of Robotics Research*, 30(9):1170–1193, 2011.

[128] R. S. Sutton, A. G. Barto, and R. J. Williams. Reinforcement learning is direct adaptive optimal control. *Control Systems, IEEE*, 12(2):19–22, 1992.

[129] R. Tedrake, T. W. Zhang, and H. S. Seung. Learning to walk in 20 minutes. In *Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems*, 2005.

[130] R. L. Tedrake. *Applied optimal control for dynamically stable legged locomotion*. PhD thesis, Massachusetts Institute of Technology, 2004.

[131] S. Thrun and T. M. Mitchell. Lifelong robot learning. In *The biology and technology of intelligent autonomous agents*, pages 165–196. Springer, 1995.

[132] M. Vukobratovic and B. Borovac. Zero-moment point thirty five years of its life. *International Journal of Humanoid Robotics*, 01(01):157–173, 2004.

[133] S. Wang, W. Chaovalitwongse, and R. Babuska. Machine learning algorithms in bipedal robot control. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(5):728–743, 2012.

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

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

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

[137] M. Wisse, G. Feliksdal, J. Van Frankkenhuyzen, and B. Moyer. Passive-based walking robot. *Robotics & Automation Magazine, IEEE*, 14(2):52–62, 2007.

[138] M. Wisse, A. L. Schwab, R. Q. van der Linde, and F. C. T. van der Helm. How to keep from falling forward: Elementary swing leg action for passive dynamic walkers. *IEEE Trans. on Robotics*, 21(3):393–401, June 2005.

[139] W. Xi, Y. Yesilevskiy, and C. D. Remy. Selecting gaits for economical locomotion of legged robots. *The International Journal of Robotics Research*, 2015.

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

[141] R. Zhang, P. Vadakkepat, and C. Chew. An evolutionary algorithm for trajectory based gait generation of biped robot. In *Proceedings of the International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2003.

[142] H.-H. Zhao, W.-L. Ma, A. D. Ames, and M. B. Zeagler. Human-inspired multi-contact locomotion with AMBER2. In *Cyber-Physical Systems (ICCPS), 2014 ACM/IEEE International Conference on*, pages 199–210, 2014.

[143] C. Zhou. Robot learning with ga-based fuzzy reinforcement learning agents. *Information Sciences*, 145(1):45–68, 2002.