BIPEDAL WALKING ANALYSIS, CONTROL, AND APPLICATIONS TOWARDS HUMAN-LIKE BEHAVIOR

A Dissertation

by

KENNETH YI-WEN CHAO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Pilwon Hur
Committee Members,	John Buchanan
	Prabhakar Pagilla
	Sivakumar Rathinam
Head of Department,	Andreas Polycarpou

May 2019

Major Subject: Mechanical Engineering

ABSTRACT

Realizing the essentials of bipedal walking balance is one of the core studies in both robotics and biomechanics. Although the recent developments of walking control on bipedal robots have brought the humanoid automation to a different level, the walking performance is still limited compared to human walking, which also restricts the related applications in biomechanics and rehabilitation.

To mitigate the discrepancy between robotic walking and human walking, this dissertation is broken into three parts to develop the control methods to improve three important perspectives: predictive walking behavior, gait optimization, and stepping strategy. To improve the predictive walking behavior captured by the model predictive control (MPC) which is transitionally applied with the nonlinear tracking control in sequence, a quadratic program (QP)-based controller is proposed to unify center of mass (COM) planning using MPC and a nonlinear torque control with control Lyapunov function (CLF). For the gait optimization, we focus on the algorithms of trajectory optimization with direct collocation framework. We propose a robust trajectory optimization using step-time sampling for a simple walker under terrain uncertainties. Towards generating human-like walking gait with multi-domain (phases), we improve the optimization through contact with more accurate transcription method for level walking, and generalize the hybrid zero dynamics (HZD) gait optimization with modified contact conditions for walking on various terrains. The results are compared with human walking gaits, where the similar trends and the sources of discrepancies are identified. In the third part for stepping strategy, we perform step estimation based on capture point (CP) for different human movements, including single-step (balance) recovery, walking and walking with slip. The analysis provides the insights of the efficacy and limitation of CP-based step estimation for human gait.

DEDICATION

To my parents, and my sister.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Pilwon Hur for his support and guidance. I thank Dr. Hur for giving me the opportunity to continue on my study of bipedal robotics, and allowing me to participate in the internship opportunities to broaden my perspective. I also thank Dr. Sivakumar Rathinam, Dr. Prabhakar Pagilla and Dr. John Buchanan for their suggestions and comments of my research work.

I also want to thank Remo Pillat, the manager of the robotics team at MathWorks and Jerry Pratt, the lead researcher of the IHMC robotics lab, and all the other members for the great internship experiences. By learning the basics of software developing, communicating with different members, and seeing how different projects were planned and executed, I learned a lot of valuable experience towards to a software developer.

For my study of unified quadratic program-based controller, I thank Matthew Powell for his mentoring so that I can learn the constrained dynamics and quadratic program-based control which are the essentials of the bipedal robot walking control. I thank Ayonga Hereid for teaching me the first step to derive the mathematical model of the bipedal robot, and thank Eric Cousineau for giving me the chance to be the intern in the robotics team at MathWorks. For my study of trajectory optimization, I want to express my gratitude to Matthew Kelly, with his patient and detailed suggestions through emails, I got more familiar with the related code implementation and was able to resolve the optimization issues more efficiently. I also want to extend my thank to the members in Human Rehabilitation Group: I thank Dr. Han Yoon for listening to my concerns when I was facing some research difficulties and sharing his PhD experience with me. I appreciate the guidance of professional presentation and interesting discussions from Moein Nazifi, and I want to thank Yi-tsen Pan, it is nice to have a friend to share various experiences

and thoughts together to facilitate different ideas, and encourage each other through the entire PhD journey. I also want to thank Victor Christian Paredes Cauna, Woolim Hong, Namita Anil Kumar, Kenny Chour, and Christian Debuys for their help and discussions for my research work, experiments and writings, and I also thank Shawanee Patrick, Felipe Miftajov, and Veronica Knisley for their help and encouragements for my defense practice. I would like to thank the members in AMBER lab for their work of developing the mechatronics of bipedal robot AMBER 3. Their great care ensures that AMBER 3 can be operated in a nice condition within all these years.

In College Station, I am lucky to have many friends from Taiwan who are studying in different research areas so that we can share different stories and life experiences together. Thank you all for caring about me and the sincere encouragement.

Last but not least, I want to express my gratitude to my parents, who encourage me to pursuing the PhD for the subject I am passionate about, with their endless love and selfless support. I also want to thank my sister, Tiffany who I can always share my joy, happiness, worries, or regrets with. Thank her for the kind words and encouragement. I cannot achieve what I have done without the support of my families.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Dr. Pilwon Hur, Dr. Prabhakar Pagilla, and Dr. Sivakumar Rathinam of the Department of Mechanical Engineering, and Dr. John Buchanan of the Department of Health and Kinesiology.

All work conducted for this dissertation was completed independently by the student.

Funding Sources

Graduate study was supported by the Graduate Research Assistantship from Dr. Pilwon Hur, the Studying Abroad Scholarship from the Taiwan Government, the Internships at MathWorks (Robotics Team) and the Institute for Human & Machine Cognition (IHMC) Robotics Lab, and the Graduate Teaching Assistantship from the Mechanical Engineering Department at Texas A&M University.

NOMENCLATURE

CLF	Control Lyapunove Function
COM	Center of Mass
COP	Center of Pressure
COT	Cost of Transport
CP	Capture Point
GRF	Ground Reaction Force
HSM	Hermite-Simpson Method
HZD	Hybrid Zero Dynamics
ICP	Instantaneous Capture Point
LIPM	Linear Inverted Pendulum Model
LQR	Linear-Quadratic Regulator
MPC	Model Predictive Control
NLP	Non-Linear Program
ODE	Ordinary Differential Equation
PHV	Peak Heel Velocity
QP	Quadratic Program
RBD	Rigid Body Dynamics
RES-CLF	Rapidly Exponentially Stabilizing Control Lyapunov Function
SLIP	Spring Loaded Inverted Pendulum
ZMP	Zero-Moment Point

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Robots have become more and more common in our daily life. It is not surprised to see robots can vacuum and mop our floors, the drones can follow us and take aerial videos and photos from the location or perspective that we cannot easily reach. For the automobiles, the driving-automation reaches to the level that seems so futuristic that the automated systems can take control of accelerating, braking, and steering so that the driver can let off the steering wheel on high way – at least for a short while.

Except wheeled and flying robots, legged robot is another form of robotic locomotion that has greater adaptiveness to human-living environment. Compared to the wheeled robots, legged robot can traverse along unstructured terrain without the need of the continuous pathway, therefore can move or interact in the human environment (such as stair walking or ladder climbing). Legged robots in general also have larger payload than the flying robots. Among legged robots, bipedal robots and humanoids are suitable to perform human-robot collaboration and human services, as they possess the same locomotion type of humans'. However, those advantages come with their cost: as a highly articulated system, a bipedal robot's floating base limits the force and torque it can exert to maintain the balance, the energetically efficient gait can be the mixture of different actuation conditions, and fast reactions are required when robot's balance cannot be recovered via the original motion reference. Those challenges – although have been actively studying and exploring – make bipedal robots' performance still not close enough to their biological counterparts.

As part of our daily life, we use our vision to identify the path in front of us, and use it to guide our walking direction so that we will not stray out of the road. With years of learning and practice, we developed our walking gait such that we utilize heel-off (trailing leg) and knee-stretching to extend our step length, and handling the impact force with the foot moving downward after heel-strike. When we accidentally lose our balance, even without too many practices, we will try our best to make a step after a step till we do not feel the risk of fall any more.

This dissertation focuses on developing model-based methods in bipedal robotics to get more understanding about those features which can be observed in human gait: 1) predictive behavior, 2) gait optimization, and 3) the stepping strategy. By exploiting dynamics, planning and control, studying bipedal robots is also beneficial to the development of lower-limb wearable and rehabilitation devices for humans, which can potentially replace the wheel chair and walker, or help users to restore their mobility.

In the following sections, first the big picture of bipedal robot automation will be presented, and then the approaches which nicely capture those three features in the filed of the bipedal robotics will be introduced. The research objectives, the main topics of the dissertation, and the dissertation overview as well as the contributions will also be presented.

1.1 Bipedal Robot Automation as a Hierarchical Control System

For a bipedal robot to navigate in an environment autonomously, its control system can be illustrated as shown in Fig. 1.1. As the entire task is too complex to be handled within a single framework, the navigation task is usually broken down into three main components: 1) High-level motion planning, 2) Low-level tracking control, and 3) Model predictive control (MPC) to bridge the other two.

High-level motion planning. When an environment is given or is being perceived, the high-level motion planning is deployed to determine the sequence of the foot placement. This can be derived by the searching the environment as a grid-map using path-searching algorithms like A* with the collection of possible foot placements [7, 8], or the gait library solved by trajectory optimization [9, 10]. Usually this component take the longest computation time as the configuration space for a bipedal robot moving in an environment can

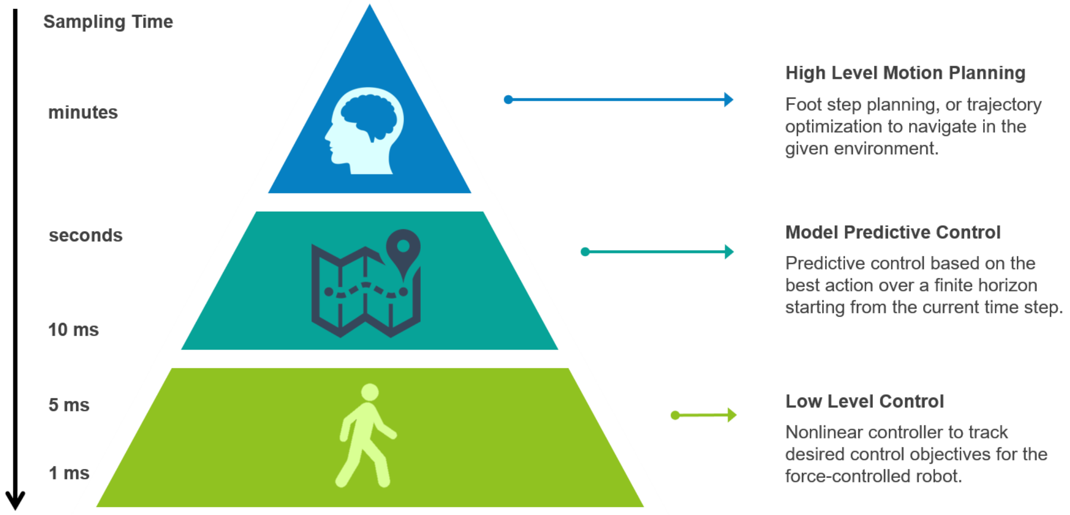

Sampling Time
minutes
seconds
10 ms
5 ms
1 ms
High Level Motion Planning
Foot step planning, or trajectory optimization to navigate in the given environment.
Model Predictive Control
Predictive control based on the best action over a finite horizon starting from the current time step.
Low Level Control
Nonlinear controller to track desired control objectives for the force-controlled robot.

Figure 1.1: Controller overview as a hierarchical control system.

be really complex, and the task potentially can be achieved in numerous ways. As a result, either the simplest model is adopted, or this component is running in an off-line manner.

Low-level tracking control. When the desired trajectory is planned, the low-level controller tracks the desired trajectory to achieve the walking motion. This needs to be achieved by continuously sensing the system's states and making the corresponding correction via control inputs. To control the robot to follow a trajectory precisely, it requires reliable software and hardware, and the precise model description (therefore Rigid Body Dynamics (RBD) is commonly used with the nonlinear controller design [11]). The sampling time is also crucial for the tracking performance (in general ranged in milliseconds).

Model predictive control. As the bipedal system is with the floating base, even with the good low-level controller and nice hardware, the tracking result can still deviate from the original plan, since the contact condition can be easily perturbed by stepping impact or slippage due to small obstacles on the terrain, or even vibrations from the joint control. This deviation can be accumulated step by step and cause the system reaches the infeasible states or loses the balance, therefore limits the reliability of the walking control performance. In this case, model predictive control with moderate re-planning speed can be a powerful tool to correct the system's behavior as the control input is derived from the best control sequence over a finite horizon, rather than just considering the state at the current time step. To solve the optimization with predicted states over the horizon on the flight, MPC is usually used with the model simpler than RBD.

With those building blocks, next we introduce several approaches of bipedal robot walking control studied in this dissertation, include the robot models, and how those approaches achieve bipedal walking.

1.2 Approaches of Bipedal Robot Walking Control

To utilize the hierarchical control system introduced in the previous section, a method of walking control basically determines how the walking motion is generated, based on the selected model. With different perspectives to reason walking balance and stability, each walking control method has its own benefits and limitations, which will be briefly introduced in this section.

1.2.1 Bipedal Robot Models

There are various bipedal robot models that can be used to describe the dynamics of a bipedal robot system. The main differences between these models are the dimensions of the state, the type of control inputs, and the required assumptions (usually for simplification) so that a model can represent a bipedal walking system.

Linear Inverted Pendulum Model (LIPM). Linear inverted pendulum model (Fig. 1.2) is one of the classical simplified model that has been well-studied in bipedal robotics because of its simplicity, and its effectiveness to generate 3D walking motion. LIPM in general has the following assumptions:

- The dynamics only considers the motion of center of mass (COM) and the effect from the ground reaction force (GRF).

- The COM height is constant. This is also the key assumption as it decouples the dynamics of the COM horizontal motion from the COM vertical motion.
- In general it is assumed the LIPM has a perfect surface contact to the ground, thus there is no rotation in the COM motion. This is also an important assumption for the walking control using zero-moment point (which will be introduced in Chapter 2).

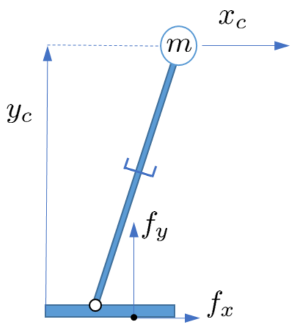


Figure 1.2: Schematic of the linear inverted pendulum model.

Though those assumptions help to greatly simplify the model and make it a lot easier to generate stable walking motion, it also make the system can only perform locomotion in a really restricted way. In addition, the generated walking using the LIPM may looks more unnatural, because the locomotion of humans and animals in general does not require those assumptions to be always hold.

Rigid Body Dynamics (RBD). Different from the LIPM, the rigid body dynamics aims to fully described the mechanical system with the very basic component – rigid body. RBD assumes each link of the robot model as a rigid body and is connected to other links with joints. In this way, a more complicated yet more accurate model can be derived to depict the system dynamics (as shown in Fig. 1.3).

Full-actuation vs. under-actuation. On one hand, when a system's degrees of freedom are equal to its actuated joint number, the system is called full-actuated as the entire configuration of the robot can be fully controlled. LIPM in Fig. 1.2 is an example of the

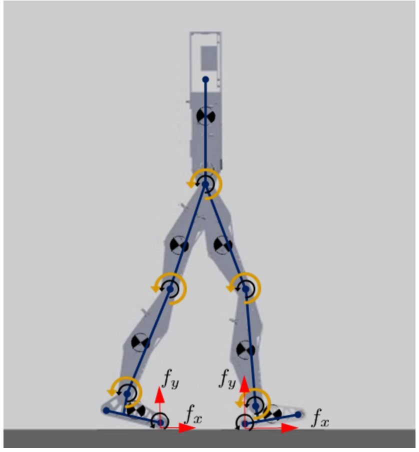


Figure 1.3: Schematic of the rigid body dynamics. The black circular arrows indicate the robot joints, and the yellow circular arrows indicate the actuators (motors) on the model.

full-actuated systems. On the other hand, when a system's degrees of freedom are more than its actuated joint number, the system is under-actuated. Take Fig. 1.3 for example, the system can not control the angle between the ground and the foot pad of the trailing leg since there is no actuator at the toe, therefore it can only affect that angle indirectly through the motors on other joints. Unlike LIPM, RBD can be used to describe different contact conditions including point and surface contacts, therefore is suitable to describe both full-actuated and under-actuated systems.

Feasible ground reaction force. For both surface and point contact, the resultant ground reaction force can be expressed with a normal force f_n and a tangential force f_t on an exertion point (e.g. in Fig. 1.2 $f_n = f_y$, and $f_t = f_x$). There are two important conditions to make a ground reaction force physically valid:

- The normal force should be always positive (i.e. pointing out of the contact surface.)
- Based on the Coulomb friction model, for a non-sliding contact, the ground reaction force should be inside the friction cone: $|f_t| \leq \mu f_n$, where μ is the friction coefficient.

1.2.2 Bipedal Robot Walking Control Methods

In this subsection we briefly introduce the state-of-the-art methods for walking control which are studied in this dissertation: zero-moment point, capture point, and hybrid zero dynamics.

Zero-Moment Point (ZMP). ZMP [12] is an important milestone in the development of bipedal robotics. The zero-moment point is the location where the ground reaction force can be expressed with zero-torque. By leveraging the concept of resultant force, it provides the *dynamic balance criterion* for the full-actuated legged system (e.g. system with flat-foot contact): When the ZMP is inside the support polygon (the convex hull of the foot contact area), the system will not tip over. With the LIPM, the momentum equation can be expressed as a linear equation of COM and ZMP. With the ZMP reference determined from the sequence of foot placement, the walking control problem becomes the COM planning problem, where using model predictive control (MPC) with the LIPM to plan the COM walking pattern is a classical method which has been widely used for decades [13, 14, 15]. However, applying MPC and low-level control in sequence has its own pitfall: the limitations of the over-simplified LIPM also limit the system's walking capability, therefore a lot of studies is also trying to generalize ZMP-based methods, such as methods using centroidal dynamics [16], or study using nonlinear simplified model [17]. Nevertheless, the dynamic balance criterion is still useful to ensure the walking balance while the system has surface contact to the ground.

Capture Point (CP). Extended from the ZMP-based method, capture point generalize it through the step-planning. Capture point [18] is the stepping location where the legged system can make a complete stop by stepping on it. Unlike the dynamic balance criterion that only quantifies the balance condition for the current stepping location, capture point can be generalized to N-step capture point for $N = 0, 1, 2, \ldots, \infty$ (the step location which will need to take N steps to make the system into a complete stop.) Because the CP-based methods focus on fast stepping and COM (re)planning, so in general it can used for both full-actuated and under-actuated systems. Using capture point with the simplified model (like LIPM) enables the legged system to have fast reactions (replanning) against undesired disturbances, therefore it is well-known to handle push-recovery and walking on uneven terrains[19, 20, 21]. To overcome the limitation from the oversimplified model, there are more and more studies focus on generalizing CP-based method with the nonlinear simplified model [17].

Hybrid Zero Dynamics (HZD). Compared to ZMP-based and CP-based methods, HZD-based methods are on the other side of the spectrum [22, 23, 24]. Having the direct root in the locomotion generation of the passive robots (e.g. compass gait), hybrid zero dynamics aims to solve a dynamically feasible walking trajectory which can be executed periodically for the system under the stepping impact – a typical example of a hybrid system (which contains the continuous dynamics and discrete event). Because both passive and under-actuated systems require to solve the walking trajectory which can fully/partially run with its natural dynamics (i.e. the unactuated dynamics), one common approach to solve the walking trajectory for passive or under-actuated robots is using *trajectory optimization*. Trajectory optimization is a mathematical method to formulate a nonlinear program to solve the walking trajectory which optimizes a target objective function while satisfying a set of constraints such as the dynamical feasibility and the gait periodicity under the stepping impact (the later is also termed *hybrid invariant condition*). By leveraging the natural dynamics and imposing energy consumption or control effort into an objective function, the generated gaits can be energetically efficient and usually look more natural. However, since this method heavily relies on the accuracy of the dynamic model, and lacks balancing mechanism for non-surface contact, it is more challenging for HZD-based methods to achieve 3D walking motion.

1.3 Research Objectives – The Big Picture

As we mentioned, the purpose of this dissertation is to get more understanding about three important features of human gait: predictive walking behavior, gait optimization, and stepping strategy. In this section, we will explain the big picture, including how these features connect to i) control hierarchy, ii) related walking control methods, and iii) research objectives (denoted as R#, e.g., R1, R2) to be investigated in this dissertation.

1.3.1 Predictive Walking Behavior

Both bipedal robotics [25, 13] and biomechanics studies [26, 27] have shown the significance of *predictive behavior* for walking (e.g. watching over few steps ahead during walking to make sure the walking motion can be executed properly). In the examples of bipedal robotics [25, 13], the predictive behavior can be reasoned as a process using MPC with low-level control in sequence – a classical example of ZMP-based walking control. Traditionally, with given foot placements, the ZMP-based walker first runs MPC with the LIPM to plan the COM trajectory, and then the planned COM trajectory along with the trajectories of the other end effectors are tracked using the low-level controller. However, there is one major pitfall for using those controller in sequence: the models used in MPC (the simplified model) and the low-level controller (which is in general the RBD) are not the same, which lead to inconsistency issue. Therefore the research objective is:

R1. Improving the model consistency between model-predictive control and low-level control to enhance the predictive walking control.

1.3.2 Gait Optimization

Since optimizing a bipedal walking gait requires to exploit a bipedal robot's dynamics as accurate as possible, the *gait optimization* using *trajectory optimization* – as one example of high-level motion planner – is usually performed in the offline manner. As we briefly mentioned, with an objective function, a trajectory optimization formulates a the mathematical problem of walking trajectory generation as a nonlinear program to optimize the objective function while satisfying a set of constraints of dynamical feasibility and periodicity of walking.

Among various methods of trajectory optimization in the literature, we mainly focus on the methods using *direct collocation* framework [28, 29, 30, 31]. Direct collocation method discretizes the trajectory of the system states and control inputs into discrete (collocation) points as independent decision variables, and solves the open-loop optimal control problem. The states and control inputs are related by the imposed constraints of the dynamic equations and the collocation constraints (which transcribe the states at the nearby collocation points as parameterized curves). Since most of the decision variables are only related to the adjacent ones, the direct collocation method makes the entire optimization can be solved with sparse Jacobian matrices (of its constraints and objective function). Therefore this method can efficiently generate complex walking behaviors for complex robot systems (e.g. HZD-based walkers).

Form the control perspective, human walking is complex because the walking gait contains both the full-actuated and under-actuated walking phases (domains), which is an example of walking with multiple-domain. Our ultimate goal is to develop trajectory optimization algorithms to generate robust, energetically efficient and adjustable walking gaits so that the proposed algorithms can be used for lower-limb wearable robots, including prosthesis and exoskeleton. For this purpose, we plan to generate energetically efficient walking trajectories for a bipedal robot AMBER 3 (Fig. 1.4) with the following research objectives:

R2. Improve the robustness of walking gait for uneven terrains.

R3. Improve the state-of-the-art of trajectory optimization algorithms to generate energetically efficient walking gait with multi-domain.

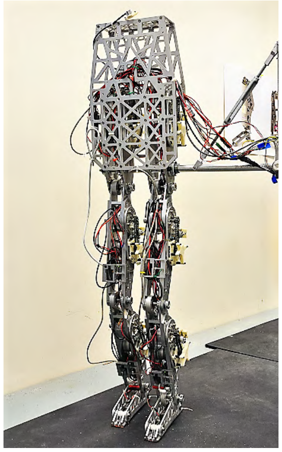


Figure 1.4: The human-sized planar bipedal robot AMBER 3. It is 148 cm tall, weights 33.4 kg, with actuated hip, knee, ankle joints and passive toe, heel joints. It is capable of performing walking with multiple domains (e.g. walking with foot rolling motion). © 2016 IEEE. Reprinted with permission from [1].

R4. Impose additional constraints and cost so that the gait can be adjustable (for customization) and more human-like.

R5. Generate gaits for various terrains, including flat ground, slopes and stairs.

1.3.3 Stepping Strategy

The scope of the study for stepping strategy is slightly different from the other two. In the control hierarchy, the stepping strategy can be treated as an emergent mode which will only be triggered when the desired trajectory generated from the high-level planner and MPC can no longer maintain the balance. Additionally, stepping strategy can also be used as a regular walking control method which works with MPC and the low-level controller. As Capture Point (CP)-based methods have already shown the capability in different studies [19, 20, 21], we focus on evaluating whether this method can be a potential tool for biomechanics and human rehabilitation. The research objectives are:

R6. Evaluate CP-based step estimation for human step-recovery from standing.

R7. Evaluate CP-based step estimation for different robotic walkers and human walking to exploit its performance.

R8. Evaluate CP-based step estimation for human walking with slip in different slip severities.

1.4 Dissertation Overview

1.4.1 Main Parts of the Dissertation

In this dissertation, to present how we achieved those research objectives, we split those studies into three main parts: i) Quadratic program-based walking controller design, ii) Trajectory optimization, and iii) Capture point-based method for human motion analysis. The scopes and the capabilities of the algorithms developed in those parts are summarized in Table 1.1.

Table 1.1: The summary of the scopes and capabilities of the algorithms developed in different parts in this dissertation.

Parts	QP-based controller	Trajectory optimization	Capture point -based analysis
Model	RBD + LIPM	RBD	LIPM
Torque saturation	✓	✓	✗
Balance criteria	✓	✗	✓
Feedback control	✓	✗	✗
Constrained optimal control	✓	✓	✗
Foot-rolling motion	✗	✓	✗
Step length estimation	✗	✗	✓
Human motion analysis	✗	✗	✓
Computation speed	online	offline	online

In the following subsections, we will introduce each of the three main parts in the dissertation, including the corresponding chapters, their relations to each research objective and our contributions.

1.4.2 Quadratic Program-based Walking Controller Design

The main purpose of this part is to achieve the research objective R1, which is described in Chapter 2. There are several potential issues about the model mismatch when applying MPC and the low-level control in sequence. First, for MPC with the LIPM (COM planning), although the dynamic balance (i.e. ZMP constraints) over the horizon can be imposed into the constrained MPC, the planned COM is based on simplified model therefore may not reflect the full dynamics of the bipedal robot. Second, for the nonlinear low-level control, although one can adopt a constrained optimal control to track the desired trajectory and impose the ZMP for the current time step, there is no guarantee that the state won't enter the region where the ZMP constraint in the next time step will be violated. To address those issues, by leveraging the fact that both the constrained MPC and constrained nonlinear control can be expressed as quadratic programs (QPs), we propose a QP-based controller design to combine both QPs into a single framework, with a synthesis equality constraint to equal the COM accelerations derived from the LIPM and RBD. In this way, the unified QP will simultaneously generate the COM motion (which satisfies the ZMP constraint over the horizon, and is with the feedback from the nonlinear RBD) and the control input (which can track alone the generated COM under torque saturation, dynamic balance, and Lyapunov stability constraints for the current time step).

1.4.3 Trajectory Optimization

For achieving research objectives (R2 – R5), Chapters 3 to 5 are the studies to explore different trajectory optimization algorithms with direct collocation framework.

To achieve objective R2 for the compass gait walker, we propose a robust trajectory optimization under terrain uncertainties, which is described in Chapter 3. In this work, by utilizing the structure of direct collocation method, the last few collocation points are used to sample the walking trajectory under terrain uncertainties – in this case, instead of sampling the walking trajectory with different step height similar to the works in [32, 33] (which complicate the trajectory optimization problem), we sample the walking trajectory with different step-time and design a robust cost function to improve the gait robustness without complicating the collocation framework.

To generate energetically efficient gait with multiple domains towards human-like motion, both Trajectory optimization through contact [30] and Hybrid Zero Dynamics (HZD) gait optimization [24] are the main methods we use to develop our works further. In Chapter 4, we modify the optimization through contact to generate human-like level walking for objectives R3 – R4. With more accurate transcription (using Hermite-Simpson method), we compared the generated level walking with different contact constraints, and we also compared the optimization results to the human data. In Chapter 5, to reduce the sensitivity of the optimization to the randomized initial guess, the HZD gait optimization is implemented, which covers the objectives R3 – R5. With the modified contact constraints, the optimization can be generally used on different terrains include flat ground, different slopes and stairs. To analyze the sensitivity of the HZD gait optimization to the initial guess, the optimization performance with the randomized initial guesses under different terrain profiles are also evaluated and compared. The details of the trajectory optimization algorithms developed in this dissertation are summarized in Table 1.2.

1.4.4 Capture Point-Based Method for Human Motion Analysis

The works of this part is described in Chapter 6. The CP-based step estimation for step-recovery (objective R6) was studied by comparing the estimated step location to the human experimental results from the literature [4, 6] and the estimation from the optimization

Table 1.2: The capabilities of the algorithms of trajectory optimization developed in this dissertation. All the methods we developed use the Hermite-Simpson Method (HSM) as the transcription method.

Topics	Robust trajectory optimization	Trajectory optimization through contact	HZD gait optimization
Model	RBD	RBD	RBD
Transcription method	HSM	HSM	HSM
Robustness under terrain uncertainties	✓	✗	✗
Multiple (contact) domains	✗	✓	✓
Contact sequence generation	✗	✓	✗
Sensitivity to initial guess	medium	high	low
Level walking	✗	✓	✓
Slope walking	✓	✗	✓
Stair walking	✗	✗	✓

using the simulation on a simplified model with MPC [5]. For the case of objectives R7 and R8, we compared the CP-based step estimation to the simulation data of robot walkers and the experimental data of human subjects. The results indicate that capture point can provide good estimations for human walking and walking with mild-slip (which is defined as the peak heel velocity (PHV) is $< 1.44m/s$ [34]).

1.5 Contributions

In this section, we summarize the contributions (denoted as C#, e.g., C1, C2) of all the studies introduced in the previous section.

Quadratic program-based walking controller design:

C1. Design a unified Quadratic Program (QP)-based controller design to integrate the elements from Model Predictive Control (MPC) for COM planning, and from rapidly exponentially stabilizing control Lyapunov function (RES-CLF) method. The resulting QP-based controller simultaneously solves for a COM trajectory that satisfies ZMP constraints over a future horizon while also producing joint torques consistent with instantaneous acceleration, torque, ZMP and RES-CLF constraints.

Trajectory optimization:

C2. Design a robust trajectory optimization using direct collocation with step-time sampling for terrain uncertainties. By utilizing the structure of direct collocation framework, the last few collocation points can be used to evaluate trajectory robustness and incorporated into the proposed robust cost function to improve the gait robustness.

C3. Improve the trajectory optimization through contact for bipedal robot AMBER 3 with more accurate transcription: Hermite-Simpson method. Compare the generated level walking gaits with different contact constraints and human data.

C4. With modified contact constraints, extend the HZD gait optimization for bipedal robot AMBER 3 to generate walking gaits on various terrains including flat ground and different slopes and stairs. Compare the gait behavior to the human walking, and analyze the optimization sensitivity to the randomized initial guesses for different walking tasks.

Capture point-based method for human motion analysis:

C5. Validate CP-based step estimation for human behaviors. Results suggest that it can provide good step-estimation for human step-recovery from standing, walking, and walking with mild slip (peak heel velocity $< 1.44m/s$).

2. UNIFICATION OF LOCOMOTION PATTERN GENERATION AND CONTROL LYAPUNOV FUNCTION-BASED QUADRATIC PROGRAMS*

2.1 Introduction

Numerical optimization plays an important role in the development of numerous walking control approaches as the mathematics used to model bipedal robot control systems are often constrained, nonlinear, high-dimensional and incorporate impulse effects due to collisions between the robot and the ground. The usage of optimization in the control of robot walking can be categorized into "offline" optimizations which solve for walking gaits before the robot is turned on and "online" optimizations which are solved while the robot is walking. Examples of successful usage of offline nonlinear optimization include the efficient design of the Cornell Ranger [35], control output parameterization establishing (hybrid) system stability through Hybrid Zero Dynamics (HZD) and Human-inspired Control [23, 22], and direct state and input trajectory optimization [30].

On the other end of the spectrum, online numerical optimization – in the form of Quadratic Programs – has become increasingly popular in the control of walking robots due to the fact that some QP-based controllers with affine constraints can be solved in real-time [36] and that the structure of a Quadratic Program is well suited to handle a diverse set of problems in robotic walking. For example, in locomotion pattern generation applications, Quadratic Programs can be used to solve Model Predictive Control problems to obtain center of mass (COM) trajectories consistent with Zero Moment Point constraints over a future horizon, as in [37, 13, 14, 15]. In this setting, the QP cost function is often setup to minimize the error between future values of the COM and desired reference

*This chapter is a slightly amended version of: © 2016 IEEE. Reprinted, with permission, from Kenneth Y. Chao, Matthew J. Powell, Aaron D. Ames and Pilwon Hur, "Unification of Locomotion Pattern Generation and Control Lyapunov Function-Based Quadratic Programs", American Control Conference, July 2016.

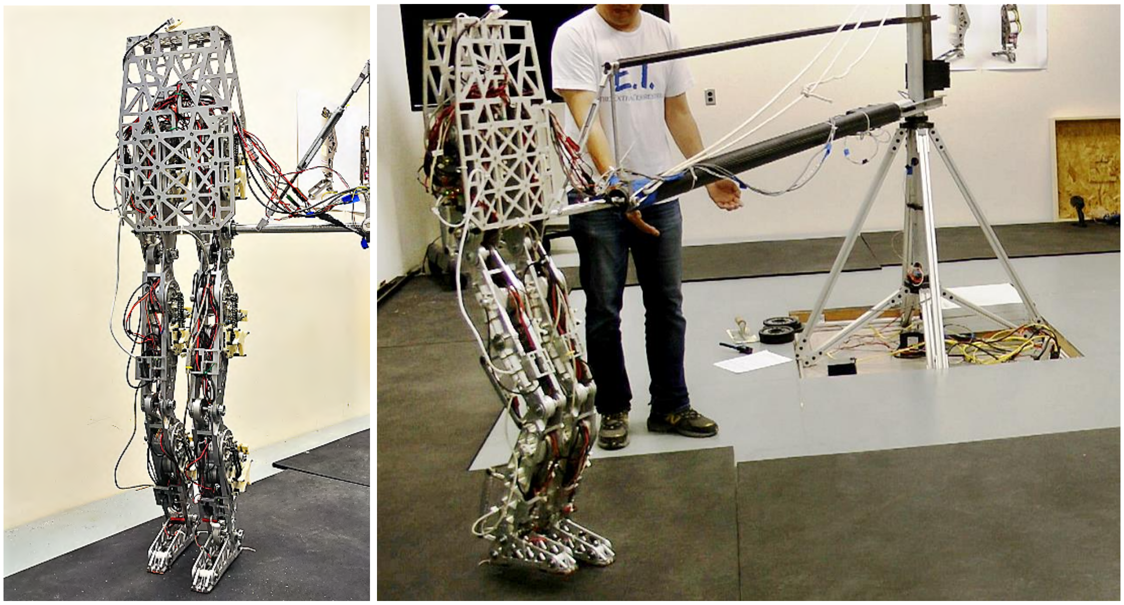


Figure 2.1: The human-sized planar bipedal robot: AMBER 3.

values. On the other hand, in the context of nonlinear systems, QPs can be naturally coupled with control Lyapunov functions (CLFs) to form an optimal controller guaranteed to stabilize outputs corresponding to walking [38, 39]. In this setting, the quadratic cost function minimizes actuation effort and the constraints encode instantaneous ZMP and torque limits on the full nonlinear system.

Inspired by optimization-based approaches to locomotion, the proposed method combines two QPs: an adaptation of the MPC proposed in [14] for planning center of mass trajectories with the Linear Inverted Pendulum (LIP) model and an adaptation of [40, 41] for locally exponentially stabilizing a control Lyapunov function for the full nonlinear dynamics of the robot. The connection point is an equality constraint imposed on the dynamics of the center of mass which enforces that the instantaneous horizontal COM acceleration is the same in both the nonlinear system and the LIP model. With this bridge in place, the unified QP enjoys the advantages of both QPs: it resolves control actions which locally stabilize nonlinear control system outputs while ensuring that these control actions are consistent with a forward horizon COM plan that satisfies ZMP constraints in the simplified model.

It is important to note that similar combinations of walking pattern generation methods and constrained, local nonlinear control have been proposed before. For example, in [42], the authors propose a similar QP which regulates the ZMP to zero over an infinite horizon using an optimal cost-to-go. In the present paper, however, the proposed controller solves a finite-time horizon MPC problem on the COM trajectory, which allows for both ZMP regulation and the enforcement of constraints on the evolution of the COM.

2.2 Controlling Robot Locomotion under ZMP Constraints

The Zero Moment Point (ZMP) is an important concept in the study of balance in robotic [12] and human locomotion [43]. For a legged robot with feet, the condition for dynamic balance (i.e. the robot not tipping over) is that the robot's ZMP lies inside the robot's support polygon. This section describes control methods for walking with ZMP constraints.

2.2.1 ZMP Constraints

As shown in Fig. 2.2, the ZMP position x_z in the sagittal plane can be expressed with the ground reaction normal force F_z and moment τ_y , e.g. $x_z = -\frac{\tau_y}{F_z}$ in single support. The ZMP constraints for dynamic balance can be described as:

$$a_\square \leq -\tau_y/F_z \leq b_\square \tag{2.1}$$

where $a_\square \in \{a_{SS}, a_{DS}\}$ and $b_\square \in \{b_{SS}, b_{DS}\}$ encode the largest moment arms of the support polygon in single support or in double support, as shown in Fig. 2.2. To satisfy *instantaneous* dynamic balance during walking, the inequality Eq. (2.1) on the ground reaction forces (GRFs) needs to be satisfied.

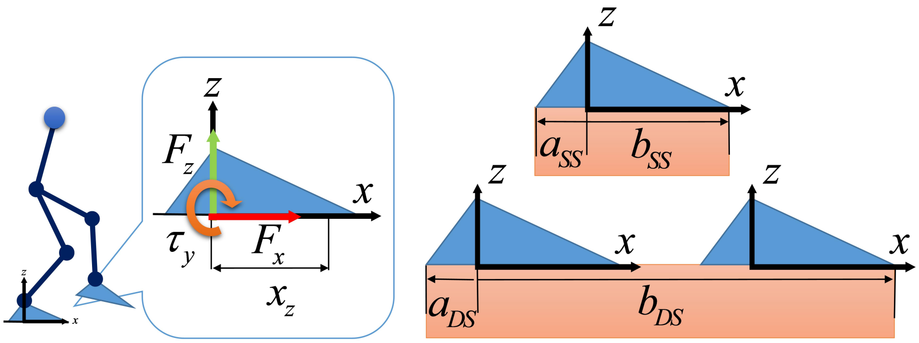


Figure 2.2: The ZMP position x_z, ground reaction forces, and the corresponding ZMP boundaries a and b in single support (SS) and double support (DS) are shown. Reprinted with permission from [1].

2.2.2 Nonlinear Robot Control System with ZMP Constraints

To achieve walking control with ZMP constraints and minimum control effort, a QP-based nonlinear controller with a Rapidly Exponentially Stabilizing Control Lyapunov Function (RES-CLF) [40] and force-based task [41] is adopted. This controller requires the full constrained dynamics described as the following form:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \begin{bmatrix} B & J_h^T \end{bmatrix} \begin{bmatrix} u \\ F \end{bmatrix} \triangleq \bar{B}(q)\bar{u} \tag{2.2}$$

where q is the generalized coordinate, $D(q)$ is inertia matrix, $C(q,\dot{q})$ is Coriolis matrix, $G(q)$ is gravity vector, J_h is Jacobian matrix of the contact constraint $h(q)$, B is the torque distribution matrix, F is the GRF vector ($F = [F_x, F_z, \tau_y]^T$ in the sagittal plane) and u is a set of actuator torques. Based on the extended input $\bar{u}$ which includes joint torques and GRFs, instantaneous dynamic balance can be satisfied by solving the following quadratic program:

$$\begin{aligned} \bar{u}^* = \underset{u}{\operatorname{argmin}} \quad & \bar{u}^T H_{CLF} \bar{u} + f_{CLF}^T \bar{u} && \text{(CLF-QP)} \\ \text{s.t.} \quad & \dot{V}_\varepsilon(x) \leq -\varepsilon V_\varepsilon(x) \\ & - bF_z \leq \tau_y \leq -aF_z && (2.3) \end{aligned}$$

where $x = [q, \dot{q}]^T$, H_{CLF} and f_{CLF} are the quadratic and linear objective function respectively. The first constraint establishes the exponential stability of output tracking where $V_\varepsilon(x)$ is a RES-CLF. The second inequality ensures that the instantaneous ZMP lies within the support polygon.

However, the second inequality Eq. (2.3) is not guaranteed to be solvable, i.e. the robot can enter states for which there is no feasible solution to the ZMP constraints Eq. (2.3). This limitation is one of the primary motivators for combining local QPs with COM trajectory planning methods. In the following section, we show how to pose a Quadratic Program which solves for COM trajectories that satisfies ZMP constraints in the linear inverted pendulum model.

2.2.3 Linear Inverted Pendulum Model for COM Trajectory Generation

To simplify the ZMP tracking problem, one common approach is to generate a COM trajectory with the linear inverted pendulum model which tracks a desired ZMP trajectory. Model Predictive Control (MPC) is one method which has been employed in the literature for pattern generation with the Linear Inverted Pendulum Model (LIP model) [14, 15, 44]. The LIP model assumes a constant center of mass height. The resulting equation of motion forms a simple expression relating the ZMP and the horizontal COM, x_c,

$$\ddot{x}_c = \frac{g}{z_0}(x_c - x_z) \triangleq \omega^2(x_c - x_z) \tag{2.4}$$

where z_0 is the constant COM height and g is the gravitational acceleration. To implement MPC with the LIP model in Eq. (2.4), the discretized state space form of LIP can be derived as shown:

$$x_{t+1} = \begin{bmatrix} 1 & \Delta T & 0 \\ \omega^2 \Delta T & 1 & -\omega^2 \Delta T \\ 0 & 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 \\ 0 \\ \Delta T \end{bmatrix} u_t \tag{2.5}$$

where $x_t = \begin{bmatrix} x_{ct} & \dot{x}_{ct} & x_{zt} \end{bmatrix}^T$, $u_t = \dot{z}_t$, and ΔT is the sampling time. With a given initial state x_{t_0} and a sequence of control inputs $\bar{U}$, the predicted sequence of states $\bar{X}$ for the next N time-steps can be expressed as $\bar{X} \; = \bar{A} \; \bar{X}_{t_0} + \bar{B} \; \bar{U}$ and $\bar{A}$ and $\bar{B}$ can be derived recursively from Eq. (2.5). The predicted states then can be used to formulate an MPC-based quadratic program for COM trajectory generation:

$$\begin{aligned} \bar{U}^* = \underset{\bar{U}}{\text{argmin}} \quad & \bar{U}^T H_p \bar{U} + f_p^T \bar{U} && \text{(MPC-QP)} \\ \text{s.t.} \quad & A_{iq,p} \bar{U} \leq b_{iq,p}, && (2.6) \end{aligned}$$

where $A_{iq,p}$ and $b_{iq,p}$ include constraints on the evolution of the COM and ZMP over an N time-step forward horizon. The advantage of COM generation with MPC is that it can be easily implemented in real-time. However, due to the simplification, there are some potential issues considering the implementation in full nonlinear dynamics, such as the fact that the control sequence $\bar{U}$ may not be feasible, or the generated COM trajectory for the simplified LIP model may not result in a feasible ZMP trajectory. These potential issues motivate a combined control method which takes advantage of the rapid pattern generation capabilities of the MPC-QP and ensures that the actual nonlinear control system satisfies instantaneous balance constraints.

2.3 Unification of Local Nonlinear Control and Walking Pattern Generation

In this section, we present the main formulation of the paper: a process for combining the nonlinear CLF-QP with the MPC-QP in a single control framework. Before the unification process is introduced, the setup of the CLF-QP and the MPC-QP, i.e. the construction of objective functions and constraints, will each be explained.

2.3.1 Control Lyapunov Functions

To realize ZMP-based locomotion, a local nonlinear controller in the form of the CLF-QP is used for tracking a set of control objectives [41]. For the controller construction, the rigid body equations of motion Eq. (2.2) can be expressed in the following general nonlinear control system form:

$$\dot{x} = f(x) + g(x)\bar{u}, \tag{2.7}$$

where $x = [q, \dot{q}]^T$. Input/output linearization [45] can be used to drive a set of control outputs $y(q) \triangleq y_a(q) - y_d(t)$ toward zero (where $y_a(q)$ are actual outputs, $y_d(t)$ are time-based desired outputs). Here, the input/output relation for the (relative degree two) outputs $y(q)$ is

$$\ddot{y} = L_f^2 y(x) + L_f L_g y(x)\bar{u} + \ddot{y}_d \triangleq L_f + \bar{A}\bar{u} + \ddot{y}_d, \tag{2.8}$$

where "L" is the Lie derivative operator and $\bar{A}$ denotes the decoupling matrix. Given a desired output dynamics $\ddot{y} = \mu$, a corresponding vector of joint torques and GRFs, $\bar{u}$, can be obtained through Eq. (2.8). In standard input/output linearization, this requires the inversion of $\bar{A}$, however, as mentioned in [41], Eq. (2.8) can be resolved implicitly via quadratic programming.

The goal in the design of μ is to drive $y \to 0$. This motivates consideration of a linearized system with coordinates $\eta = [y, \dot{y}]^T$ which can be expressed as $\dot{\eta} = F\eta + G\mu$. To exponentially stabilize η to zero with a convergence rate $\varepsilon > 0$, μ is designed to satisfy the following condition:

$$\dot{V}_\varepsilon(\eta) = L_f V_\varepsilon(\eta) + L_g V_\varepsilon(\eta)\mu \leq -\varepsilon V_\varepsilon(\eta), \tag{2.9}$$

where $V_\varepsilon(\eta) = \eta^T P_\varepsilon \eta$ is a RES-CLF, P_ε is obtained by solving equation (47) in [38], and $L_f V_\varepsilon(\eta) = \eta^T(F^T P_\varepsilon + P_\varepsilon F)\eta$, $L_g V_\varepsilon(\eta) = 2\eta^T P_\varepsilon G$. A CLF-based Quadratic Program (CLF-QP) of the form Eq. (2.3) is implemented to find the minimum control input μ that guarantees Lyapunov stability through the satisfaction of Eq. (2.9) and additional constraints.

2.3.2 CLF-QP Setup

This section describes the construction of the specific constraints and cost function considered for the local nonlinear CLF-QP of interest in this paper; for more details, see [41].

2.3.2.1 CLF-QP Constraints

The final form of the set of constraints to be used in the proposed CLF-QP variant is

$$A_{iq,CLF}\bar{u} \leq b_{iq,CLF}, \tag{2.10}$$

$$A_{eq,CLF}\bar{u} = b_{eq,CLF}, \tag{2.11}$$

where $A_{iq,CLF}$ and $A_{eq,CLF}$ are matrices and $b_{iq,CLF}$ and $b_{eq,CLF}$ are vectors of appropriate dimension. The subscripts iq and eq denote inequality and equality, respectively.

As discussed in the previous section, ZMP constraints Eq. (2.1) are included in the

(CLF-QP) optimization to ensure that the robot maintains dynamic balance. An additional constraint is included to ensure that the normal force applied to support foot is positive, i.e. $F_z \geq 0$. Note that the ZMP constraints Eq. (2.1) and the normal force constraint can be written as inequality constraints on $\bar{u}$ using the equations of motion Eq. (2.2), and thus be included in Eq. (2.10). Actuator saturation limits are likewise incorporated in Eq. (2.10) via the inequalities $-u_{max} \leq u \leq u_{max}$, where u_{max} is a vector of maximum allowable torques. Finally, a CLF constraint is to used to drive the control objectives $\eta \to 0$. However, as aggressive control objectives and conservative torque limits can lead to infeasible systems of inequalities, the CLF constraint is relaxed [41] by $\delta > 0$, resulting in

$$\dot{V}_\varepsilon(\eta) = L_f V_\varepsilon(\eta) + L_g V_\varepsilon(\eta)\mu \leq -\varepsilon V_\varepsilon(\eta) + \delta. \tag{2.12}$$

The relaxation δ will be minimized in the cost function of the corresponding CLF-QP. The CLF constraint Eq. (2.12) together with the ZMP, normal force and torque constraints comprise the inequality constraints Eq. (2.10).

The equality constraints Eq. (2.11) enforce holonomic constraints $h(q) = 0$ corresponding to contact(s) between the robot and the ground through a constraint on the acceleration

$$\ddot{h}(q, \dot{q}, \bar{u}) = 0. \tag{2.13}$$

The vector $h(q)$ includes the horizontal and vertical components of one foot in the single support phase, and both feet in the double support phase.

2.3.2.2 CLF-QP Cost Function

The CLF-QP cost function is designed to balance the minimization of the control μ and the relaxation δ to the CLF constraint in Eq. (2.12)

$$\underset{(\bar{u},\delta)}{\operatorname{argmin}} \quad p\delta^2 + \bar{u}^T \bar{A}^T \bar{A}\bar{u} + 2L_f^T \bar{A}\bar{u} \tag{2.14}$$

where $p > 0$ is a weighting factor. Note that $Eq.$ (2.14) encodes the goal of minimizing $\mu^T\mu$ through Eq. (2.8). The CLF-QP cost function Eq. (2.14) and constraints Eq. (2.10)–Eq. (2.11) are used in conjunction with elements of a walking pattern generation QP to form the unified QP described in Section 2.3.5.

2.3.3 Walking Control Objectives

This section describes the choice of control objectives, i.e., $y_a(q)$ and $y_d(t)$, for achieving ZMP-based walking in the nonlinear system Eq. (2.2). To reduce the differences between the LIP and the full nonlinear dynamic model, the height of the COM is regulated to a constant $z_0 > 0$, and the desired torso angle with respect to inertial frame is set to zero. In the single support phase, the desired orientation of the swing foot is set to zero (to ensure that the foot lands flat on the ground), and the desired horizontal and vertical components of the swing foot are smooth time-based polynomial functions with zero boundary velocities and accelerations. Finally, note that

$$\ddot{x}_c = L_f^2 x_c + L_f L_g x_c \bar{u}, \tag{2.15}$$

is the actual acceleration of the center of mass in the nonlinear system. To achieve forward walking, an equality constraint will be enforced on $\bar{u}$ to achieve a desired acceleration of the center of mass, i.e. $L_f^2 x_c + L_f L_g x_c \bar{u} = \ddot{x}_c^d$. The value of $\ddot{x}_c^d$ will be determined through the use of the LIP model for walking pattern generation, as described in the next section and Eq. (2.15) will subsequently be used as a bridge between the nonlinear robot dynamics and the LIP model.

2.3.4 LIP Model Predictive Control Setup

The CLF-QP – described by the constraints Eq. (2.10) and Eq. (2.11) and cost function Eq. (2.14) – provides a method of locally stabilizing control objectives in the full nonlinear robot dynamics while also ensuring the instantaneous ZMP constraints are satisfied. However, under the action of the CLF-QP alone, the robot can enter states for which there is no feasible solution to the ZMP constraints. This motivates the combination elements of the local nonlinear CLF-QP with elements of a Model Predictive Control QP for producing feasible ZMP trajectories over a forward horizon. The following sections describe the construction of the specific MPC-QP considered.

2.3.4.1 General MPC Setup

The MPC-QP solves a receding horizon problem using the discrete-time, LIP dynamics Eq. (2.5) with time-step ΔT. The target walking behavior consists of alternative phases of single and double support. The target duration of the single and double support phases are T_{SS} and T_{DS} seconds, respectively. The number of discrete points in the plan is fixed to be $N = (T_{SS} + T_{DS})/\Delta T$. Similar to $\bar{X}$ in Eq. (2.5), the predicted evolution of the ZMP, x_{zt}, COM, x_{ct}, and COM velocity, $\dot{x}_{ct}$, for the next N discrete points can be expressed as:

$$\begin{aligned}
\bar{X}_z &= \bar{A}_{zmp}\bar{X}_{t_0} + \bar{B}_{zmp}\bar{U} \\
\bar{X}_c &= \bar{A}_{com}\bar{X}_{t_0} + \bar{B}_{com}\bar{U} \\
\dot{\bar{X}}_c &= \bar{A}_{comV}\bar{X}_{t_0} + \bar{B}_{comV}\bar{U}
\end{aligned} \tag{2.16}$$

where $\bar{A}_{zmp}$, $\bar{B}_{zmp}$, $\bar{A}_{com}$, $\bar{B}_{com}$, $\bar{A}_{comV}$ and $\bar{B}_{comV}$ also can be derived recursively from Eq. (2.5). As these expressions are affine in $\bar{U}$, constraints on the evolution of ZMP and COM can be expressed as constraints on $\bar{U}$.

2.3.4.2 MPC Horizon Computation

As mentioned previously, the COM trajectory planner will implement a receding horizon. The model predictive control problem will solve 2 phases into the future. In general, this means the problem will have 3 domains: one for the completion of the current phase (with N_1 discrete points), one for the entire duration of the next phase (with N_2 discrete points) and one for the remainder (with N_3 discrete points), where $N_1 + N_2 + N_3 = N$. As the target walking consists of alternating phases of single and double support, the values N_1, N_2 and N_3 will change depending on the current phase. Specifically, at a point in time t during a single support phase, the number of discrete points in first domain is $N_{1,SS} = (T_{SS} - t)/\Delta T$, the number of discrete points for next domain is $N_{2,DS} = (T_{DS})/\Delta T$ and the third and final domain's discrete point number is $N_{3,SS} = t/\Delta T$. Similarly, at a point in time t during a double support phase, the numbers of discrete points for the three domains are $N_{1,DS} = (T_{DS} - t)/\Delta T$, $N_{2,SS} = (T_{SS})/\Delta T$ and $N_{3,DS} = t/\Delta T$ respectively.

2.3.4.3 MPC Constraints

An equality constraint is imposed on the MPC-QP to enforce that the center of mass reaches the position x_c^{goal} at the end of the trajectory with the terminal velocity $\dot{x}_c^{goal}$. Additionally, inequality constraints are imposed on the resultant ZMP trajectories to ensure that the ZMP lies within the support polygon throughout the duration of the plan.

$$\begin{aligned} A_{eq,p}\bar{U} &= b_{eq,p} \\ A_{iq,p}\bar{U} &\leq b_{iq,p} \end{aligned} \tag{2.17}$$

where

$$\begin{aligned}
A_{eq,p} &= \begin{bmatrix} [0_{N-1}, 1]\bar{B}_{com} \\ [0_{N-1}, 1]\bar{B}_{comV} \end{bmatrix} \\
b_{eq,p} &= \begin{bmatrix} x_c^{goal} - [0_{N-1}, 1]\bar{A}_{com}\bar{X}_{t_0} \\ \dot{x}_c^{goal} - [0_{N-1}, 1]\bar{A}_{comV}\bar{X}_{t_0} \end{bmatrix} \\
A_{iq,p} &= [\bar{B}_{zmp}, -\bar{B}_{zmp}]^T \\
b_{iq,p} &= [\bar{b} - \bar{A}_{zmp}\bar{X}_{t_0}, -\bar{a} + \bar{A}_{zmp}\bar{X}_{t_0}]^T
\end{aligned} \tag{2.18}$$

In Eq. (2.18), $\bar{a}$ and $\bar{b}$ both include three ZMP boundary sequences, which are determined by the three domains in the horizon, with the corresponding support phase (single support or double support) and step length (N_1, N_2, or N_3). For example, the frontal ZMP boundary of the horizon $\bar{a}$ will be $[\bar{a}_{SS}, \bar{a}_{DS}, \bar{a}_{SS} + 0.5L_{step}]^T$ if it is in single support, and $\bar{a}$ will be $[\bar{a}_{DS}, \bar{a}_{SS} + 0.5L_{step}, \bar{a}_{SS} + 0.5L_{step}]^T$ if it is in double support, where L_{step} is the step length.

2.3.4.4 MPC Cost Function

The cost function balances the goals of minimizing control effort, achieving ZMP trajectory tracking, and driving the COM position to the desired location for next stepping. This formulation is similar to the one used in [15]. The sequence of control inputs $\bar{U}$ then can be derived by solving the following optimization problem:

$$\begin{aligned}
\underset{\bar{U}^*}{\text{argmin}} \quad & \omega_1 \bar{U}^T\bar{U} + \omega_2|\bar{X}_z - \bar{X}_z^{goal}|^2 && (2.19) \\
\text{s.t.} \quad & \bar{a} \leq \bar{X}_z \leq \bar{b} && \text{(ZMP)} \\
& x_{c_{t_0+N}} = x_c^{goal} && \text{(COM)} \\
& \dot{x}_{c_{t_0+N}} = \dot{x}_c^{goal} && \text{(COM Vel.)}
\end{aligned}$$

where ω_1 and ω_2 are weighting factors, $\bar{X}_z^{goal}$ is the desired ZMP trajectory, x_c^{goal} and $\dot{x}_c^{goal}$ are the desired COM terminal location and velocity at $t = t_0 + N$ respectively, $\bar{a}$ and $\bar{b}$ are ZMP boundary vectors of the horizon. Using the equation in Eq. (2.16), the cost function in Eq. (2.19) can be expressed as follows:

$$\begin{aligned} \underset{\bar{U}^*}{\operatorname{argmin}} \quad & \frac{1}{2}\bar{U}^T H_p \bar{U} + {f_p}^T \bar{U} && \text{(MPC-QP)} \\ \text{s.t.} \quad & A_{eq,p}\bar{U} = b_{eq,p} && (2.20) \\ & A_{iq,p}\bar{U} \leq b_{iq,p} \end{aligned}$$

where

$$\begin{aligned} H_p &= 2\omega_1 I + 2\omega_2 \bar{B}_{zmp}^T \bar{B}_{zmp} \\ f_p &= 2\omega_2 [\bar{A}_{zmp}\bar{X}_{t_0} - \bar{X}_z^{goal}]^T \bar{B}_{zmp} \end{aligned} \qquad (2.21)$$

Note that the desired ZMP sequence $\bar{X}_z^{goal}$ and the desired terminal COM position x_c^{goal} are calculated based on a horizon which changes over time.

2.3.5 Main Result: Unified QP Combining Pattern Generation and ZMP-based Walking Control

Using the building blocks of the quadratic programs for pattern generation and ZMP-based locomotion with RES-CLF QP, the proposed controller synthesizes all elements into a unified quadratic program:

$$\begin{aligned}
&\underset{\bar{u}^*,\bar{U}^*,\delta^*}{\text{argmin}}\frac{1}{2}\begin{bmatrix}\bar{u}\\ \bar{U}\\ \delta\end{bmatrix}^T\begin{bmatrix}H_{CLF} & 0 & 0\\ 0 & H_p & 0\\ 0 & 0 & p\end{bmatrix}\begin{bmatrix}\bar{u}\\ \bar{U}\\ \delta\end{bmatrix}+\begin{bmatrix}f_{CLF}\\ f_p\\ 0\end{bmatrix}^T\begin{bmatrix}\bar{u}\\ \bar{U}\\ \delta\end{bmatrix} \qquad (2.22)\\
&\text{s.t.}\quad \begin{bmatrix}A_{eq,CLF} & 0 & 0\\ 0 & A_{eq,p} & 0\end{bmatrix}\begin{bmatrix}\bar{u}\\ \bar{U}\\ \delta\end{bmatrix}=\begin{bmatrix}b_{eq,CLF}\\ b_{eq,p}\end{bmatrix}\\
&\qquad\begin{bmatrix}A_{iq,CLF} & 0 & -1\\ 0 & A_{iq,p} & 0\end{bmatrix}\begin{bmatrix}\bar{u}\\ \bar{U}\\ \delta\end{bmatrix}\leq\begin{bmatrix}b_{iq,CLF}\\ b_{iq,p}\end{bmatrix}\\
&\qquad(L_f^2x_c+L_fL_gx_c\bar{u})\frac{z_0}{g}-x_c=-x_z
\end{aligned}$$

In Eq. (2.22), the LIP model equation in Eq. (2.4) is adopted as the QP synthesis constraint. The first term on left hand side is the COM acceleration expressed using input/output relation with full dynamics in Eq. (2.8). By solving the quadratic program above for each time step, the instantaneous torque input $\bar{u}$ for ZMP-based locomotion considering both output tracking and COM planning on-the-fly then can be derived.

2.4 Simulation Results

The unified controller was implemented in simulation on the model of AMBER 3[*], which is a human-sized, planar, and fully actuated bipedal robot (Fig. 2.1). Using the walking parameters listed in Table 2.1, the proposed unified controller was implemented in MATLAB, where the unified QP combining the nonlinear CLF-QP with the MPC-QP

[*] AMBER 3 was built in AMBER Lab led by Dr. Aaron Ames at Texas A&M University. Since July 2015, AMBER Lab has moved to Georgia Tech, and AMBER 3 has been maintained and operated in HUR Group led by Dr. Pilwon Hur at Texas A&M University.

Table 2.1: Important simulation parameters. Reprinted with permission from [1].

Parameter	Value	Parameter	Value
T_{SS}	2 s	T_{DS}	1 s
MPC sampling time ΔT	0.1 s	Length of MPC horizon	3 s
L_{step}	10 cm	Stride Height	5 cm

is solved at every time step.

Compared to the controller solving MPC-QP and CLF-QP in sequence, several adjustments of the unified QP controller has to be made to ensure that the system would not be over-constrained. The first and most important change was to remove x_c from the output vector in the unified framework. Since the resolved control input would also minimize the cost function in unified pattern generation, this soft ZMP tracking could implicitly provide larger flexibility for integration with other tasks than a strict ZMP tracking. Second, the COM terminal constraints which were originally for MPC stability were removed in the unified framework, since it would make the system over-constrained and the control input would lose continuity and cause chattering (as shown in Fig. 2.4 and Fig. 2.5), although

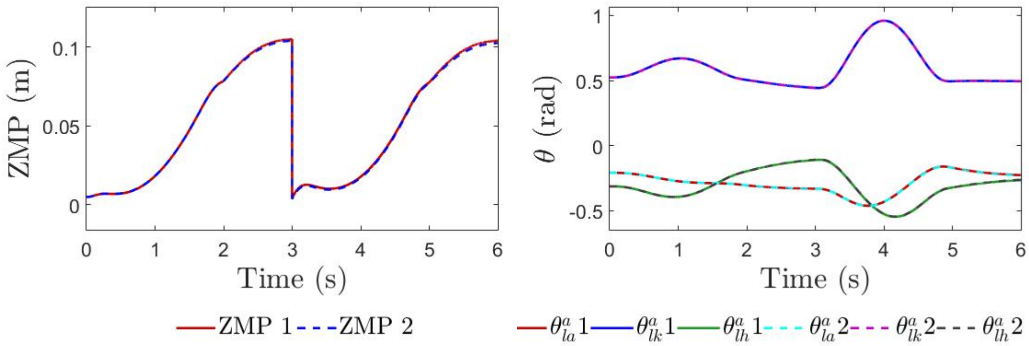


Figure 2.3: A comparison of ZMP trajectories (left) and joint tracking profiles (right) from two different simulations of the proposed method: in simulation (1) the unified QP with terminal constraints on the COM is used and in simulation (2) the terminal constraints are not used. Reprinted with permission from [1].

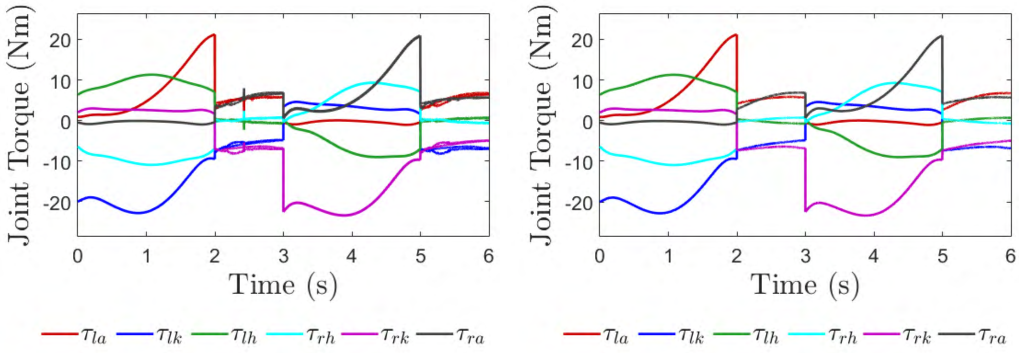


Figure 2.4: Joint torques from the simulation of the proposed unified QP with (left) and without (right) terminal constraints on the COM. Reprinted with permission from [1].

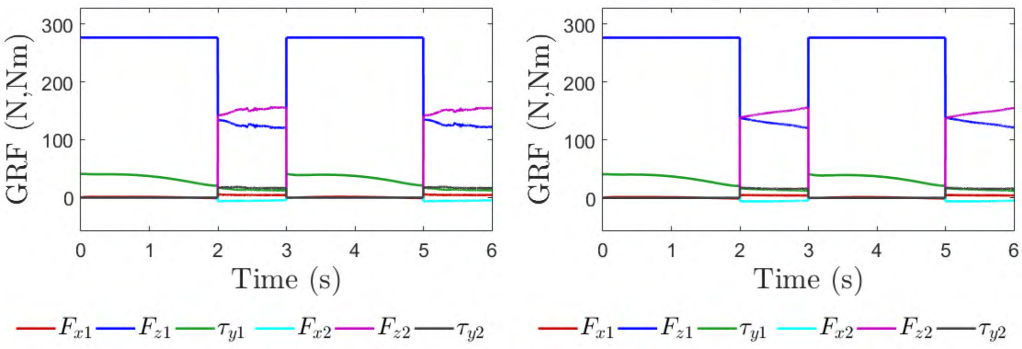


Figure 2.5: Ground reaction forces of the simulation using the proposed unified QP with (left) and without (right) COM terminal constraints. Reprinted with permission from [1].

the derived joint angle trajectories and ZMP patterns (Fig. 2.3) are similar. Last but not least, a impact map is adopted right after the single support phase in simulation, which will cause a discrete jump on COM velocity. The direct feedback of this COM velocity for updating x_{t_0} in real-time COM planning will easily cause the resolved x_c to diverge due to the large postimpact COM velocity. As a result, the feedback of postimpact COM velocity is assigned as zero to enforce the COM planned as free of impact, where the real impact effect in full dynamics is suppressed by the nonlinear controller as a perturbation.

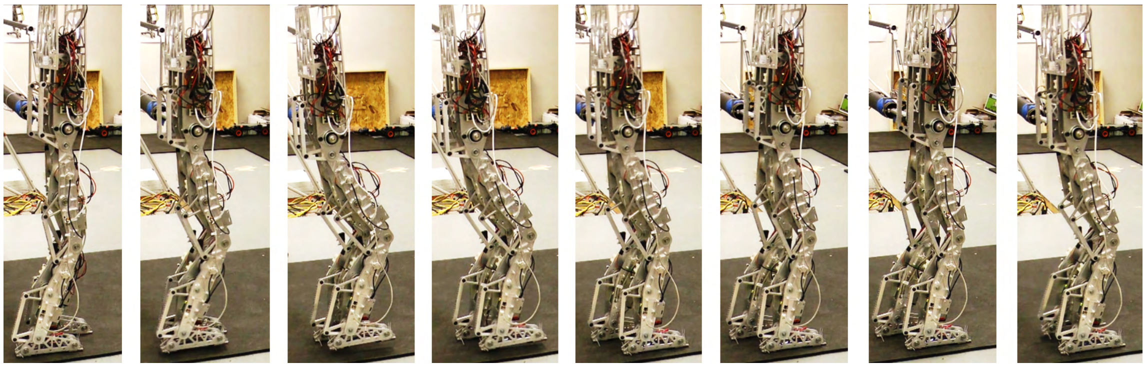


Figure 2.6: The walking tiles of a half gait cycle from a trajectory tracking experiment in which AMBER 3 took 383 steps without falling using trajectories produced by the proposed method. Reprinted with permission from [1].

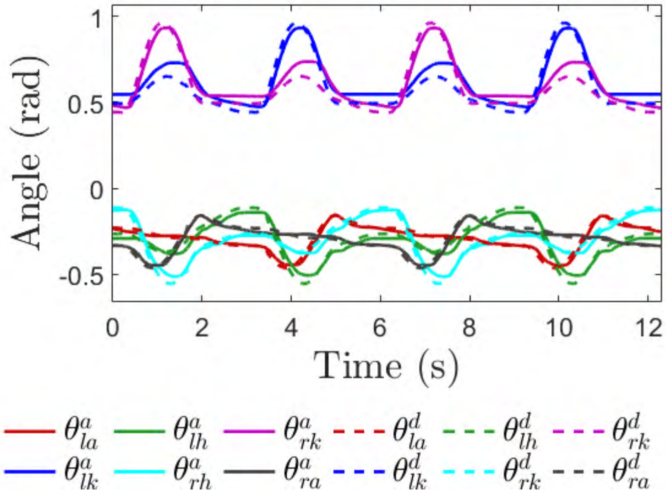


Figure 2.7: The joint tracking results from a trajectory tracking experiment in which AMBER 3 took 383 steps without falling using trajectories produced by the proposed method. Reprinted with permission from [1].

2.5 Experimental Result

The experiment was implemented in LabVIEW with C++ on AMBER 3. The walking motion is generated from simulation by using the proposed unified QP controller. Using this setup, AMBER 3 walked for two laps (about 383 steps); results from the corresponding experiments are shown in Fig. 2.6 and Fig. 2.7. Using nonlinear constrained optimization, the torque command was converted into position and velocity commands. Although real-time implementation of the quadratic program is still in progress, the current joint trajectory tracking and experiment video [46] are quite similar to the walking motion displayed in simulation.

2.6 Conclusions and Future Work

In this work, we presented a method which solves a single Quadratic Program which incorporates elements from Model Predictive Control (MPC) based center of mass planning methods and from rapidly exponentially stabilizing control Lyapunov function (RES-CLF) methods. The resulting QP-based controller simultaneously solves for a COM trajectory that satisfies ZMP constraints over a future horizon while also producing joint torques consistent with instantaneous acceleration, torque, ZMP and RES-CLF constraints. The method is developed for simulation and experimental study on a seven-link, planar robot AMBER 3. Future work entails completing a real-time implementation of the unified QP controller in C++. Robustness tests such as walking with disturbances, push recovery, or walking through uneven terrain then can be conducted.

3. A DIRECT COLLOCATION METHOD WITH STEP-TIME SAMPLING FOR ROBUST TRAJECTORY OPTIMIZATION OF BIPEDAL LOCOMOTION UNDER TERRAIN UNCERTAINTIES

3.1 Introduction

Generating efficient and agile dynamic walking of bipedal robots using trajectory optimization has been an active topic and attained more attentions in these years. Leveraged by the advancement of nonlinear optimization solvers and the sparsity of the Jacobian matrices of cost and constraint functions, trajectory optimization with direct collocation is able to incorporate complicated dynamic constraints and contact conditions, and scales to practical complex robotic system such as bipedal or humanoid robots [47, 31, 2]. However, even though the locally-optimal solution satisfies variable bounds, the dynamic and kinematic constraints for feasibility, the bipedal robots still easily lose their balances due to unmodeled dynamics, modeling errors, or perturbations from the environment. Various approaches have been proposed to address this problem, such as robust motion planning, robust controller design, or gait synthesis using gait library, just to name a few.

Among various methods to improve the robustness of robotic bipedal walking, the differential dynamic programming has been used in [48] for minimizing the robust cost under the worst-case disturbance (i.e. a MiniMax approach). For robust motion planning and control, there exist several methods for improving robustness by stepping strategy with whole body control, for example, the center of mass planning with Zero Moment-Point (ZMP) constraints for push recovery [14], and the control methods based on Capture Point [18, 19, 21]. Nguyen et al. [49] proposed a quadratic program-based controller with control Barrier function (CBF) to enforce robot to stay within the safe operational region. There also exist other approaches using gait library with switching controller design [9, 50] by integrating various limit cycles to synthesize more complicated walking behaviors.

On the other hand, some studies developed robust trajectory optimization algorithms for bipedal locomotion [32, 33] or other robot applications [51]. Griffin et al. [33] proposed a robust trajectory optimization with a robust cost based on the deviation of the states and control (normalized by the perturbed step-time) from the nominal trajectory. However, this optimization is based on direct shooting method and is hard to handle complicated systems. Dai et al. [32] proposed a method of trajectory optimization for the robust limit cycle generation. In this method, a robust LQR cost-to-go function related to the post-impact state and the pre-impact state was proposed, which is more suitable for underactuated robots. However, the LQR constraints imposed in the nonlinear program and the additional free variables used for sampling potential stepping heights make this approach hard to scale to high dimensional systems. Extended from trajectory optimization with direct collocation and dynamic programming, Manchester et al. [51] proposed a method which includes the bounds of the deviated states and control under the ellipsoidal disturbance estimated by the time-varying LQR. This approach scales well to the higher dimensional problem. However, the effect of the mode transition (i.e. impact) needs to be considered for bipedal robots especially the underactuated ones.

In this work, the proposed method aims to improve the robustness of the trajectory while keeping the complexity similar to the original trajectory optimization with direct collocation. In Section II, the related studies of the robust methods are first presented. In Section III, we introduce the robust cost using step-time sampling and the validation with SLIP running model. In Section IV, the main optimization framework is depicted. In Section V and Section VI, the simulation results and conclusions are presented respectively.

3.2 Optimization of the Robust Limit Cycle

Bipedal walking/running robots in general consist of both continuous (e.g. movement in the single support phase) and discrete (e.g. impact due to foot strike) behaviors, therefore can be modeled as hybrid systems. To find a limit cycle (the nominal periodic gait) of a given hybrid system, one common approach is to cast this problem as an open-loop optimal control problem and solve it using nonlinear optimization (i.e. trajectory optimization). Although this approach can generate walking gait for underactuated robots with complex dynamic behaviors, its open-loop nature make it challenging to impose cost or constraint to improve the robustness of the optimal solution, as the robustness is usually associated with the closed-loop controller design. In this section, the state of the art to find the robust limit cycle using trajectory optimization [32, 33] will be briefly introduced.

3.2.1 Problem Formulation for the Nominal Limit Cycle

A simple bipedal robot can be described as a hybrid system $\mathscr{H}$ (assuming it has only one mode (e.g. walking phase) and one mode transition):

$$\mathscr{H} = \begin{cases} \dot{x} = f(x,u), & \text{if } \phi(x,u) > h_F \\ x^+ = \Delta(x^-), & \text{if } \phi(x,u) = h_F \end{cases}$$

where $x = [q, \dot{q}]^T$ is the state of the system, $f(\cdot)$ is the state space equation of the robot. $\phi(\cdot)$ is a guard function to trigger the mode transition (where the foot-strike happens), e.g., the function that determines the foot height where h_F is the nominal step height. $\Delta(\cdot)$ is a function defined at the mode transition to map the pre-impact states x^- to the post-impact states x^+ based on momentum conservation.

The method for finding a limit cycle is widely used and extended from the studies of passive dynamic walking [52, 53, 54], just to name a few. By running the simulation with

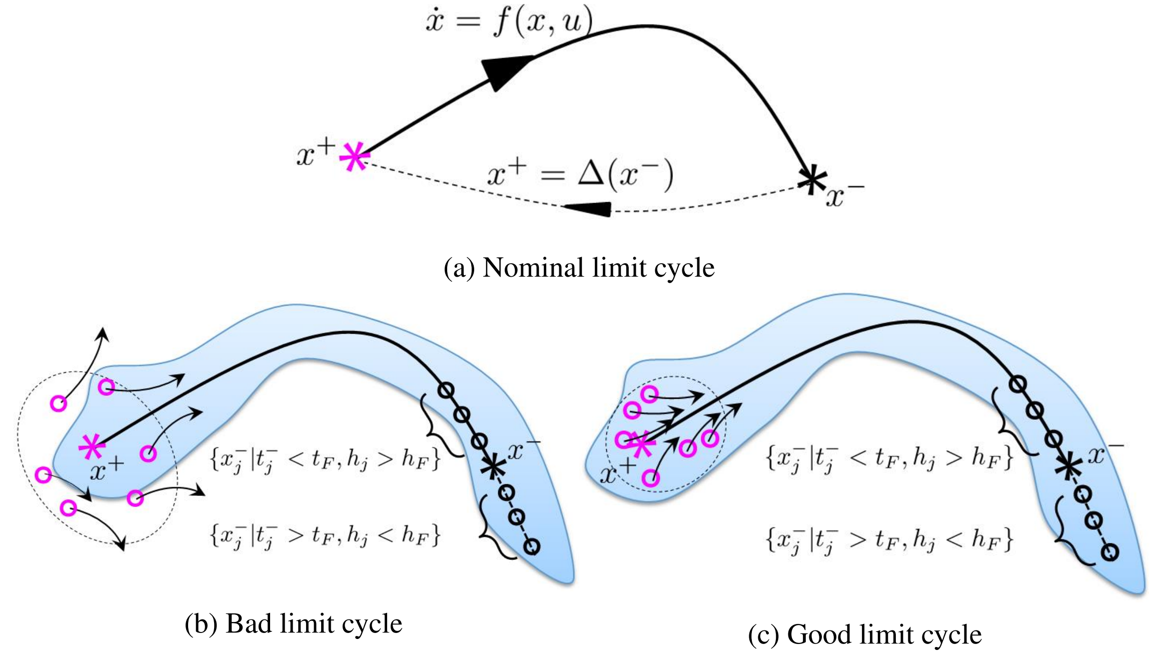
$f(\cdot)$ with the sequence of the states $x(\cdot)$ and control $u(\cdot)$, the optimization to find limit cycle can be formulated as:

$$\begin{aligned} \min_{x(\cdot),u(\cdot)} \quad & J_0(x,u) \\ \text{s.t.} \quad & x^+ = \Delta(x^-) \\ & \phi(x^-) = h_F \\ & c_{eq}(x,u) = 0,\, c_{iq}(x,u) \geq 0 \end{aligned}$$

where $x(\cdot)$ can be usually reduced to x_0 or a coefficient set $\alpha(\cdot)$ of the trajectory to express $x(\cdot)$. $J_0(x,u)$ is the objective function (e.g. $u^T u$ or cost of transport). An usual optimization incorporated with simulation is also referred to as *direct single shooting method*. The major downsides of this method is that when the optimization complexity increases, it is often hard to converge, or can be sensitive to the initial condition. To make the nonlinear program more well-posed, *direct collocation method* discretizes the states and control into a set of free variables. Instead of running a simulation, the equations of the state continuity and the system dynamics are added into the equality constraints $c_{eq}(x)$. With the discretization, a large but sparse nonlinear program can be formulated and solved efficiently.

3.2.2 Robustness of the Limit Cycle under Step Height Variation

Even though the robustness of the bipedal locomotion on uneven terrains could be challenging to quantify without considering the controller, there is an important insight we can get by checking the nominal limit cycle and its behavior under step height uncertainties [32]. The schematic of a nominal limit cycle and the mode transition of foot-strike is shown in Fig. 3.1(a). When a robot is walking on an uneven terrain, the pre-impact states will deviated from the x^- (the circle markers in Fig. 3.1(b) and (c)). Intuitively, if the


$\dot{x} = f(x, u)$
x^+
$x^+ = \Delta(x^-)$
x^-
x^+
x^-
$\{x_j^- | t_j^- < t_F, h_j > h_F\}$
$\{x_j^- | t_j^- > t_F, h_j < h_F\}$
x^+
x^-
$\{x_j^- | t_j^- < t_F, h_j > h_F\}$
$\{x_j^- | t_j^- > t_F, h_j < h_F\}$

(a) Nominal limit cycle

(b) Bad limit cycle

(c) Good limit cycle

Figure 3.1: The schematic of a nominal limit cycle, and the examples of 'Bad' and 'Good' limit cycles in terms of how far the post-impact states (due to step height uncertainties) deviated from the nominal limit cycle. The shaded region indicates the region of attraction of the trajectory.

post-impact states for all possible terrain heights are farther away from the nominal limit cycle (Fig. 3.1(b)), it is more likely that the robot will fall down because more post-impact states are out of the region of attraction (provided by the controller and trajectory). On the other hand, if all the post-impact states are closer to the limit cycle (Fig. 3.1(c)), then the system can be stabilized by the controller more easily. Therefore, this intuition is used to design the robust cost function for the robust trajectory optimization.

3.2.3 Robust Cost Function via Step Height Sampling

Assuming the profile of an uneven terrain is known and $H = \{h_j | 1 \leq j \leq K\}$ is a set of sampled heights with the fixed sampling increment dh, then the cost function which seeks the balance between the original objective $J_0(x, u)$ and the robust cost can be generally expressed as:

$$\begin{aligned} & J_0(x,u) + \omega \sum_{j=1}^{K} J_j(x_j^-, u_j) \\ \text{s.t.} \quad & \phi(x_j^-) = h_j, h_j \in H \end{aligned}$$

where $J_i(x_j^-, u_j)$ is the robust cost to measure the trajectory difference between the nominal limit cycle and the perturbed trajectory with step height h_j. By combining the optimization with time-varying LQR, Dai [32] proposed the following robust cost:

$$J_j = (x_j^+ - x^+)^T S(t_j)(x_j^+ - x^+), S = S^T, S > 0 \tag{3.1}$$

where J_i indicates the cost-to-go for the given initial condition $(x_j^+ - x_0)$ to the infinite horizon with the time-varying LQR control ($S(\cdot)$ can be derived by integrating the periodic Riccati equation backward for the step-time t_j), x_j^+ is the post-impact state for the step height h_j. Although this method provides a nice robust cost associated with the LQR controller design and it is using direct collocation for the nominal trajectory, the imposed constraints for the periodic Riccati equation, and the different sampled step heights with different time steps largely complicate the optimization. On the other hand, Griffin [33] proposed another robust cost function as:

$$J_j = \frac{1}{t_j} \int_{t=0}^{t_j} (||\delta x_j(\tau)||^2 + ||\delta u_j(\tau)||^2) \frac{\tau}{t_j} d\tau$$

where the square of the Euclidean distance between the normalized trajectory with the step height h_j and the nominal limit cycle is measured. This proposed method was validated with the experiment results using the planar bipedal robot MARLO. However, since it is based on the direct shooting method, it could be much more difficult for it to handle more complicated systems. These potential issues motivate us to propose the robust cost function using step-time sampling which takes advantage of the structure provided by the direct collation framework.

3.3 Robust Cost via Step-Time Sampling

Considering to sample the pre-impact states x_j^- near the nominal pre-impact state x^- (black circle markers) in Fig. 3.1(b) and (c) by utilizing the direct collocation method (where the states and controls are discretized with the fixed time step), it is natural to sample the step-time with fixed dt instead of sampling the step height with the fixed dh. In this section we will introduce the proposed robust cost using step-time sampling and the related robustness analysis with a Spring Loaded Inverted Pendulum (SLIP) running model for validation.

3.3.1 Using Collocation Points to Express the Robust Cost

By using the direct collocation, the states and controls are discretized as $x = [x_1, \ldots, x_N]$ and $u = [u_1, \ldots, u_N]$ which are corresponding to the time stamps $t = [t_1, \ldots, t_N]$. Assuming the objective is to evaluate the robustness of the last K collocation points (where K is an odd number, and the last $(K-1)/2^{th}$ state is the nominal pre-impact state x^-), inspired by the robust cost proposed by Dai (Eq. (3.1)), we propose the following objective function with the robust cost using step-time sampling:

$$\begin{aligned} & J_0(x,u) + \omega \sum_{j=N-K+1}^{N} J_j(x_j, u_j) \\ \text{s.t.} \quad & \phi(x_j) = h_j(t_j), t_j \in T \\ & J_j = (x_j^+ - x_1)^T P (x_j^+ - x_1), P = P^T, P > 0 \\ & x_j^+ = \Delta(x_j) \\ & T = \{t_j | N - K + 1 \leq j \leq N\} \end{aligned} \tag{3.2}$$

The main advantage of the robust cost using step-time sampling is that it utilizes the last K states x_j as the potential pre-impact states x_j^-, therefore no extra free variables are required. Also, instead of implementing controller within the optimization to get the $S(\cdot)$ like Eq. (3.1), we replace it with an user-defined weighting matrix P so that the optimization can still be solved in an open-loop manner (Note both the distance $(x_j^+ - x_1)$ and the weighting matrix P can be modified if the user has *a priori* knowledge about the system dynamics). On the other hand, in regards to the performance of robustness evaluation, the proposed robust cost also leads to two questions: 1) Can the robust cost using step-time sampling reflect the robustness of locomotion under step height variation? 2) Can this robust cost reflect the robustness of the trajectory without control? (Note in the previous studies [32, 33], the simulations or experiments for the robustness of the trajectories were tested with the feedback controllers. Since the trajectory optimization and control can be highly coupled to each other, this question is still remained open.) To seek the answers of those questions, in the following subsections, we will introduce the robustness analysis of the SLIP running model and compare it to the evaluated robust cost using proposed function Eq. (3.2).

3.3.2 SLIP Running Model and Stable Fixed Points on the Poincaré Section

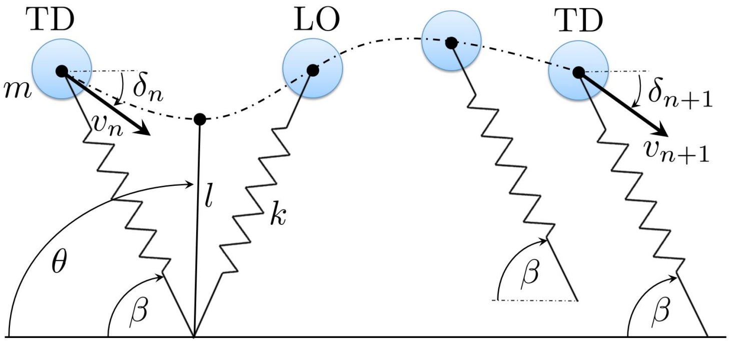


Figure 3.2: The schematic of a SLIP running model. TD indicates the touch-down event and LO indicates the lift-off event.

The SLIP model (Fig. 3.2) is a well-known model with the extensive studies of locomotion in biomechanics and robotics. It is simple (with only one point mass and the massless leg) while its high nonlinearity makes it analytically non-integrable. Additionally, it is also challenging to analyze the model with linearization or other approximation method due to the highly coupled dynamics. Following the dimension analysis of SLIP running model by Shen [55], the initial conditions (i.e. the fixed points at the Poincaré section defined at the touch-down) of the limit cycles can be defined as $x_0 = [l_0, \beta, v_0, \delta_0]$. Because in the simulation both the initial leg length l_0 and the touch-down angle β are used at the touch-down event, and the magnitude of the velocity v_0 for different steps should always be the same as the energy conservation assumed to be hold, the direction of center of mass (COM) movement δ_0 will be the main decision variable needs to be solved and checked for the stability. Across different dimensionless stiffness $\tilde{k}$, fixed points can be found by using direct single shooting method, as shown in Fig. 3.3. The stability of the limit cycle is identified by numerical Poincaré map analysis (The eigen values e_v of the Poincaré map for a stable fixed point need to be within the unit cycle). Once the stable fixed points are found, both the evaluation of the robust cost using step-time sampling and the robustness analysis with perturbed step heights can be implemented.

3.3.3 Robustness Analysis and Comparison to the Proposed Robust Cost Function

As indicated by the black arrows in Fig. 3.3, we simulated the SLIP running model at the stable fixed points along two directions: (a) where the touch-down angle is fixed, and (b) where the COM movement direction δ^* is fixed. To compare the robustness, we first ran the simulation for each fixed point with the randomized terrain height uncertainties ($h_j \in [-0.04, 0.04]$) and collected the number of steps before the SLIP model lost its balance. We ran the robustness test for 10 trials and then averaged the number of periodicity. Second, since the nominal step-time t_F can be derived from the simulation, we also evaluated the

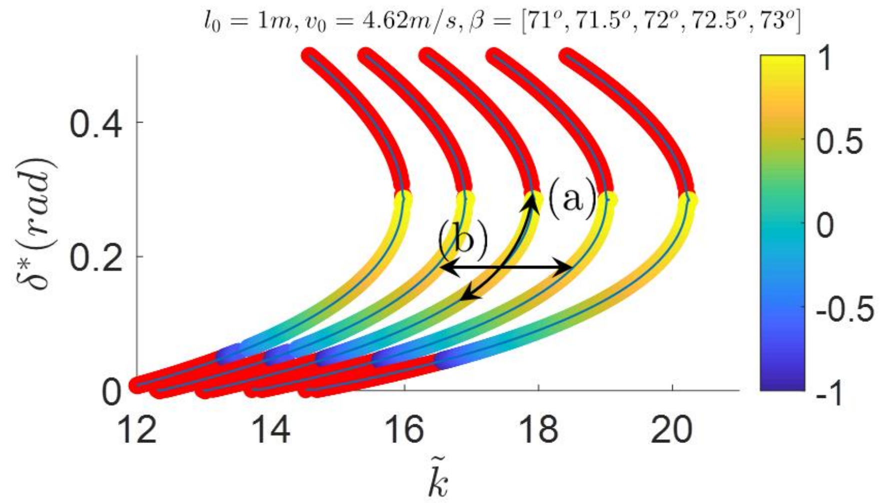


Figure 3.3: The fixed points for different touch-down angles (β) and the same $v_0 = 4.62m/s$. Each point indicates that there is a limit cycle of the SLIP running model. Its color indicates the e_v of its Poincaré map (Red indicates the fixed point is unstable).

same limit cycle with the step-time sampling using Eq. (3.2) with $K = 20$, $P = 100I$, $t_j \in t_F \pm 0.15s$. The robustness comparison for the fixed-point (a) and (b) are presented in Figs. 3.4 and 3.5.

In Fig. 3.4, it is shown that the node (in Table 3.1) with lower robust cost (node 5 and 6) have higher step numbers. Also, the nodes with higher robust costs (node 1 to 4) have lower step numbers except for node 7 and 8 which still have high step numbers (it could be the largest δ^*s of node 7 and node 8 lead to the largest jumping height therefore can easily jump over the tested heights compared to other nodes). Nevertheless, the region with low

Table 3.1: Stable fixed points of the SLIP running model along range (a) in Fig. 3.3 ($\beta = 72^o$, $v_0 = 4.62m/s$, $l_0 = 1m$).

Node	1	2	3	4	5	6	7	8
$\delta^*(rad)$	0.13	0.14	0.15	0.16	0.18	0.19	0.21	0.24
e_v	0.47	0.52	0.58	0.64	0.70	0.75	0.81	0.89
$\tilde{k}$	16.8	17.0	17.1	17.2	17.4	17.5	17.7	17.8

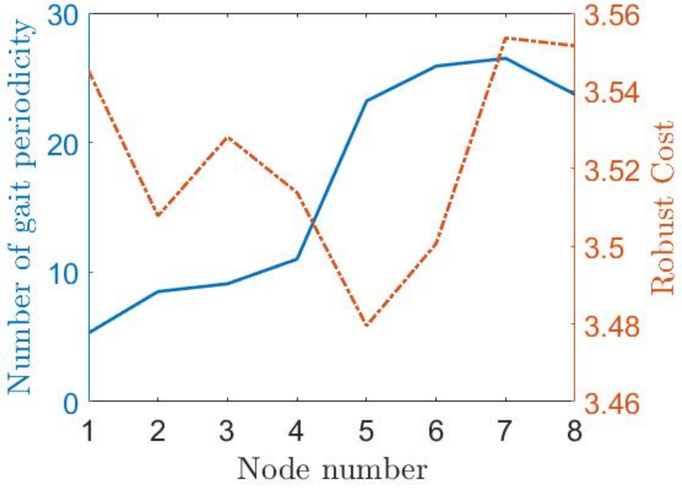


Figure 3.4: The gait periodicity under step height uncertainties versus the robust cost using step-time sampling (Eq. (3.2)) of fixed points listed in Table 3.1.

Table 3.2: Stable fixed points of the SLIP model along range (b) in Fig. 3.3 ($\delta^* = 0.2rad$, $v_0 = 4.62m/s$, $l_0 = 1m$).

Node	1	2	3	4	5	6	7	8
$\beta(^o)$	71.5	71.6	71.7	71.8	72.0	72.2	72.3	72.4
e_v	0.77	0.77	0.77	0.78	0.78	0.78	0.78	0.79
$\tilde{k}$	16.5	16.8	17.1	17.4	17.6	17.9	18.2	18.5

robust cost with the main objective function can still help to improve the robustness of the solution. For example, intuitively one would design J_0 to choose the fixed point with minimum $|e_v|$ because it has the fastest convergence speed (in the range (a) in Table 3.1, node 1 would be the optimal choice). While after adding the robust cost into the objective function, then it can help the optimizer to choose the node closer to the node 5. On the other hand, in Fig. 3.5 the result of the robust costs are consistent with the step numbers. It is also shown that the robustness is not sensitive to different touch-down angles (The nodes in Table 3.2). Therefore the optimizer with the proposed robust cost would be more effective to search along δ^* instead of β.

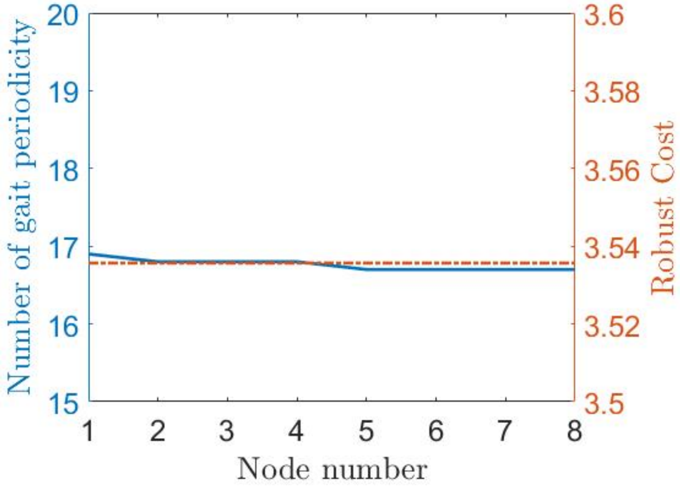


Figure 3.5: The gait periodicity under step height uncertainties versus the robust cost using step-time sampling (Eq. (3.2)) of fixed points listed in Table 3.2.

3.4 The Direct Collocation with Step-Time Sampling for Robust Trajectory Optimization

With the robustness analysis of the SLIP running model as part of the validation, in this section we present our main formulation of this paper: a trajectory optimization via direct collocation with uniform step-time sampling and an example of compass-gait generation (with control). By designing a tractable robust cost function and adjusting the path constraints without increasing the complexity of the original nonlinear program, this formulation also has the inherent benefits from the direct collocation method (sparse nonlinear program, less sensitive to the initial condition compared to direct shooting method, etc.).

3.4.1 Optimization Formulation

Assuming that the N knot (collocation) points $[x_1, \ldots, x_N]$ and $[u_1, \ldots, u_N]$ are discretized in the direct collocation, and the last K knot points are used (where N is an odd number, K is an even number)) for the step-time sampling (i.e. potential pre-impact states), the formulation of the nonlinear program with the proposed cost function in Eq. (3.2) can be expressed as:

$$\begin{aligned}
\min_{x,u,dt} \quad & J_0(x,u) + \omega \sum_{j=N-K+1}^{N} J_j(x_j,u_j) \\
\text{s.t.} \quad & x_F = x^- = x_{N-K/2}, t_F = (N-K/2)dt \\
& x_1 = x^+ = \Delta(x^-), x_j^+ = \Delta(x_j) \\
& \phi(x^-) = h_F \\
& J_j = (x_j^+ - x_1)^T P (x_j^+ - x_1), P = P^T, P > 0 \\
& g_i(x,u) = 0, \forall i = 2 \ldots N-1 \\
& t_{Fmin} \le t_F \le t_{Fmax} \\
& x_{min} \le x \le x_{max}, u_{min} \le u \le u_{max}
\end{aligned} \tag{3.3}$$

where dt is the sampling time. $g_i(x,u)$ is a set of equality constraints of *Hermite Simpson method* for the state continuity (i.e. the states are transcribed as a set of cubic splines):

$$x_i - \frac{1}{2}(x_{i+1} + x_{i-1}) - \frac{1}{8}h(f_{i-1} - f_{i+1}) = 0 \tag{3.4}$$

$$x_{i+1} - x_{i-1} - \frac{1}{6}h(f + 4f_i + f_{i+1}) = 0 \tag{3.5}$$

where f_i is $f(x_i,u_i) = \dot{x}_i$.

3.4.2 Robust Cost Function with Projection for Compass Gait Locomotion

As we mentioned in Section II, in the robust cost function, the distance $(x_j^+ - x_1)$ can be modified based on *a priori* knowledge. In this subsection we will briefly introduce the distance that is widely-used for the (periodic) bipedal locomotion. Considering the post-impact states in Fig. 3.3 again, if a post-impact state x_j^+ is far from the nominal x^+ while it is close to the limit cycle, then in this case it is better to define the distance as the distance of the x_j^+ projected to the closest point on the nominal limit cycle:

$$x_j^+ - \Pi(x_j^+) \tag{3.6}$$

where $\Pi()$ is the projection function*. To approximate the projection used in [32] for the compass gait (assuming the state variables of a knot point of the compass gait robot in Fig. 3.7 is $x_i = [q_{1i}, q_{2i}, \dot{q}_{1i}, \dot{q}_{2i}]^T$), we defined the projection as:

$$\Pi(x_j^+) = x_{i^*}, i^* = \min_i |x_j^+(1) - x_i(1)| \tag{3.7}$$

where this projection is to find the collocation point with the closest stance leg angle to the stance leg angle of x_j^+.

3.4.3 Optimization Results and Comparisons of Compass Gaits

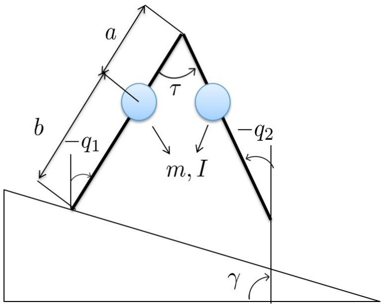
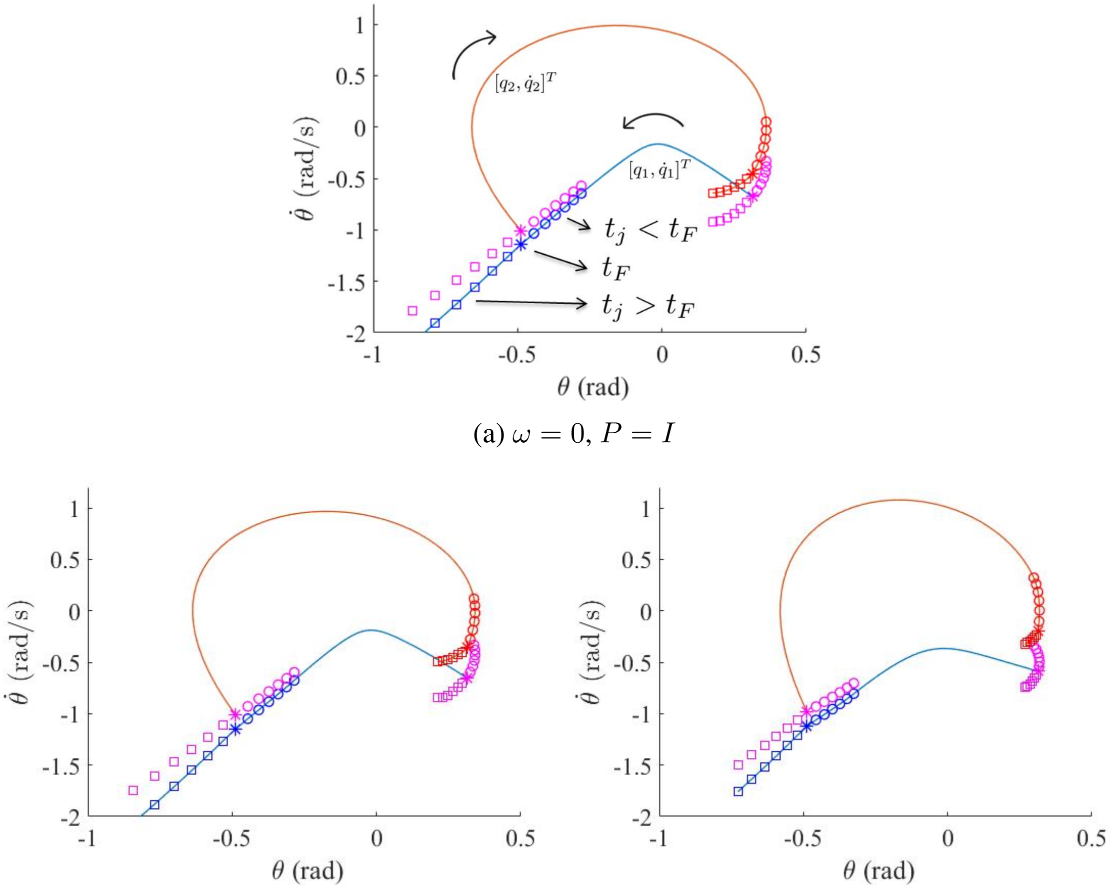


Figure 3.7: The schematic of a compass gait robot.

With the proposed optimization framework and the modified distance metric in Eqs. (3.6) and (3.7), we now applied it with the compass gait robot (Fig. 3.7, where the parameters are indicated in the Table 3.3) and see how the weighting factor ω of the robust cost will affect the optimal solution geometrically. The proposed optimization framework was implemented with the OptimTraj [56] which uses `fmincon()` as the nonlinear program solver

*To accommodate this modification in the trajectory optimization and walking control, researchers also use phase variable [23, 33] for the projection.



(a) $\omega = 0$, $P = I$

(b) $\omega = 100$, $P = I$

(c) $\omega = 500$, $P = I$

Figure 3.6: The phase portraits solved using Eq. (3.3) with the same set of stepping time sampling (in the range of $t_F \pm 10\% t_F$), the modified distance measure (Eqs. (3.6) and (3.7)), and different weighting ω for the robust cost function. The lines/curves spanned by post-impact states (pink markers) and pre-impact states of both legs (blue and red markers) become shorter towards to the nominal trajectory when ω is increased.

in MATLAB. As shown in Fig. 3.6, it is found that when the weighting ω is increased, the post-impact states (the pink markers) on both sides (of the swing and the stance leg) become closer to the nominal trajectory (at the expense of increased original cost where we use $J_0 = u^T u$). With the geometric validation, next we will introduce the robustness test of the generated compass gait with time-varying LQR control under step height uncertainties.

3.5 Compass Gait with Time-Varying LQR Control

3.5.1 Controller Implementation Details

An infinite horizon time-varying LQR controller was implemented with `ode45()` in MATLAB to simulate the compass gait walking control. With the optimal trajectory from the proposed optimization, the desired joint trajectories were parameterized as the polynomial function of time. At each time instance, the state feedback gain was calculated using `lqr()` in MATLAB with the linearized system and the associated Q and R (listed in Table 3.3). To accommodate terrain disturbances, two adjustments are made:

Projection of the perturbed post-impact states: As we mentioned in Section 3.4.2, it is better to project the perturbed post-impact state to the point on the nominal gait with the closest stance leg angle as the new initial condition. For the consistency this projection was also implemented in the controller as:

$$\tau = \frac{q_1 - q_{1min}}{q_{1max} - q_{1min}} t_F \tag{3.8}$$

$$\Pi(x) \triangleq x(\tau) = \begin{cases} x(\tau) & \text{if } \tau \in [0, t_F] \\ x(0) & \text{if } \tau < 0 \end{cases} \tag{3.9}$$

where τ is used as the new 'start time' for a new step.

Desired trajectory when $\mathbf{t > t_F}$: When the simulation time t is exceeding the nominal walking period t_F, the traditional way is to 'lock' the desired state as x_F:

$$x(t) = \begin{cases} x(t) & \text{if } t \in [0, t_F] \\ x(t_F) & \text{if } t > t_F \end{cases} \tag{3.10}$$

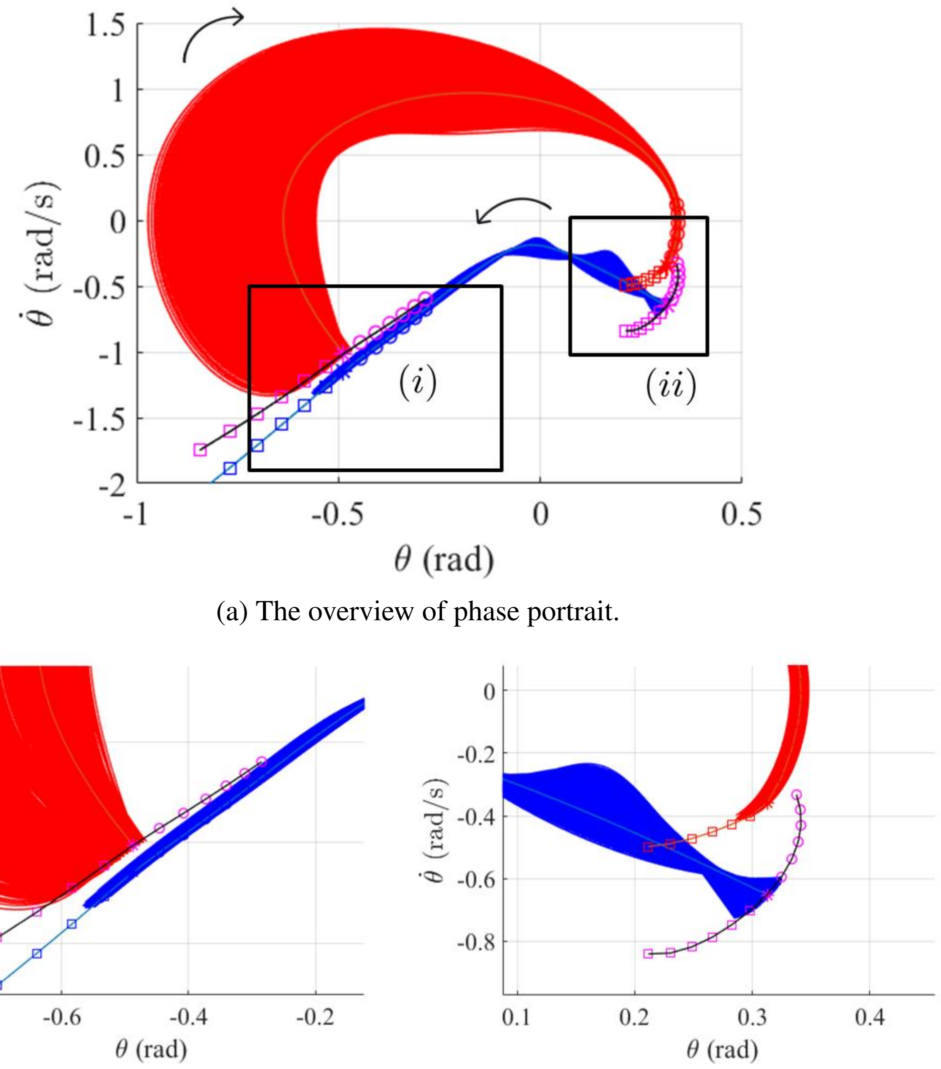
Although it seems a reasonable choice, the discontinuity caused by this locking can lead to instability. Another hidden merit of the proposed optimization is that the optimal solution also contains the optimal trajectory for $t > t_F$. Therefore compared to Eq. (3.10) it can provide smoother desired motion, and additional conditions can also be specified for the states with $t > t_F$ in the optimization if it is required.

Table 3.3: Important simulation parameters.

Parameter	Value	Parameter	Value
a	$0.2m$	Nominal slope angle	5^o
b	$0.8m$	Node number in opt.	61
m	$1kg$	Sampled step-time	$t_j \in t_F \pm 10\% t_F$
I	$1kg.m^2$	Range of slope	$[4.23^o, 9.23^o]$
R	0.01	Q	$[I, 0; 0, 0.01I]$

3.5.2 Result of Compass Gait Walking under Terrain Height Uncertainties

The simulation result is shown in Fig. 3.8. The range of slope was chosen based on the ratio of the range of h_j corresponding to t_j within the sampled step-time. In the optimization, the $\omega = 100$ and weighting matrix $P = I$ in the robust cost function were chosen with the Q and R for getting the better result. In this simulation, the compass gait robot walks 1000 steps without falling. In Fig. 3.8 it is also shown that although the distribution of post-impact states (the left part of the red trajectories and the right part of the blue trajectories) are diverse due to the randomized slope angle at every step, those trajectories converge nicely toward the nominal trajectory generated by our proposed method. This simulation result indicates that the generated trajectory with the time-varying LQR control can reject the disturbances caused by the terrain uncertainties.



(a) The overview of phase portrait.

(b) Closer view of region (i) in (a)

(c) Closer view of region (ii) in (a)

Figure 3.8: Walking 1000 steps on the terrain with uniform randomized slope angle $\in [4.23^o, 9.23^o]$ (Nominal slope angle: 5^o). It shows that the pre-impact states and post-impact states on the phase portraits (the end of red and blue trajectories) match the contour (the black lines connect the markers) predicted by the solution from the proposed robust trajectory optimization.

3.6 Conclusions and Future Work

In this work, we proposed a robust trajectory optimization using direct collocation. By utilizing the last N collocation points and the proposed robust cost, the robustness of the gait can be evaluated via step-time sampling. The robust cost based on step-time sampling was verified by the robustness test using passive SLIP running model (i.e. simulations with different initial conditions under terrain height uncertainties). With the proposed optimization framework, a compass gait bipedal locomotion was generated, controlled with time-varying LQR, and was capable of walking on the uneven terrain with slope uncertainties. Next, we plan to test it with wider range of physical constraints (e.g. with tighter torque saturation and adding contact force constraints), and with more complicated bipedal robots (e.g. 5-link knee-gait robot). On the other hand, without altering the direct collocation framework, the idea of step-time sampling can also be applied to the state and control with step-time deviations caused by other disturbances (e.g. incorporating the ellipsoidal disturbance estimated by a time-varying LQR controller in DIRTREL [51]). It will also be our next step in the future work.

4. GENERATING HUMAN-LIKE WALKING GAIT ON FLAT TERRAIN USING OPTIMIZATION THROUGH CONTACT[*]

4.1 Introduction

Generating dynamic walking gait of humanoid robots which is targeting on efficiency, agility and robustness is a challenging problem. For this motion generation problem, trajectory optimization is a powerful tool for solving the locally optimal trajectories for the dynamical systems which are potentially highly nonlinear. Among various methods in this field, trajectory optimization with direct collocation has gained more attention in recent years. In the field of biomechanics, the dynamic simulation using trajectory optimization is useful for studying neuromuscular coordination, predicting human behavior under various conditions [57, 58], or generating walking gait for lower-limb prosthesis control [59]. On the other hand, there are more and more applications using trajectory optimization for bipedal locomotion generation [47, 60], especially for bipedal robot controller design using Hybrid Zero Dynamics (HZD) scheme [61, 59, 31]. Although the trajectory optimization with direct collocation works well with HZD-based controller, this approach usually requires the specification of the contact sequence (or called domains/modes) as *a priori*. This specification can potentially make the problems much more complicated than needed since the combinatorics of the potential contacts need to be considered.

On the other side of the spectrum, some researchers have developed approaches that do not require domain knowledge specific to the target behavior. Mordatch et al. proposed the contact invariant optimization (CIO) [62] used for animations with simplified dynamic models. By introducing the contact-invariant cost and multiple optimization phases, this

[*] This chapter is a slightly amended version of: © 2017 IEEE. Reprinted, with permission, from Kenneth Y. Chao and Pilwon Hur, "A Step Towards Generating Human-Like Walking Gait via Trajectory Optimization through Contact for a Bipedal Robot with One-Sided Springs on Toes", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sept. 2017.

method optimizes over auxiliary decision variables which specify when and where the contacts are made, and can generate complex behaviors such as walking, climbing and handstand. Posa et al. [30] developed a unified framework termed *trajectory optimization through contact*, which has shown its capability to generate motion for high-dimensional systems with large number of modes, such as grasp planning, bipedal robot walking, or running. However, this local method will be affected largely by the choice of initial guess, and the accuracy of the numerical approximation using Euler method. In our work, for improving the accuracy and efficiency of the algorithm to generate walking gait under the similar optimization formulation, we first provide the modified framework and constraints for improving the numerical properties of the optimization formulation in Section III. Several schemes deigned for motion planning with better solution and better dynamical system description for the passive toes of the bipedal robot AMBER 3 (Fig. 4.1) are presented in Section IV. Results and conclusions are presented in Section V and Section VI respectively.

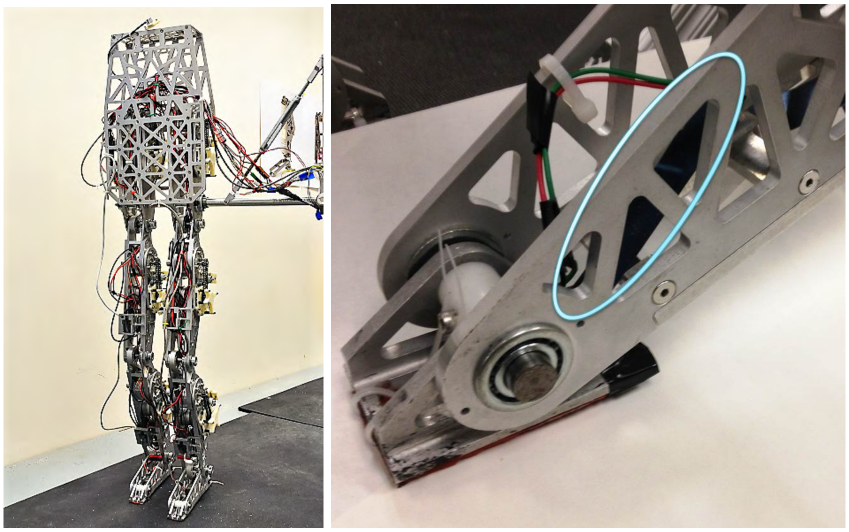


Figure 4.1: The human-sized planar bipedal robot AMBER 3 (left). It is 148 cm tall, weights 33.4 kg, with 6 active degree of freedoms at hip, knee and ankle joints, capable of performing walking with multiple contact domains (e.g. walking with foot rolling motion). It has passive toes (right) with the torsional springs (circled by the bright blue loop). Reprinted with permission from [2].

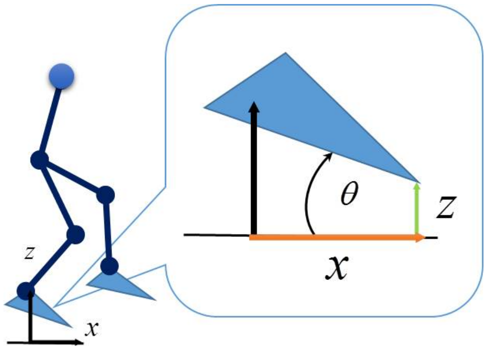


Figure 4.2: The schematic of a bipedal robot with a floating base. Reprinted with permission from [2].

4.2 Full Dynamics and Bipedal Locomotion

4.2.1 System Dynamics with Contact Constraints

The dynamics of a rigid body model with a floating base can be expressed as follows:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu + J^T\lambda \tag{4.1}$$

where q is the generalized coordinate that includes $[x, z, \theta]^T$ (Fig. 4.2), $D(q)$ is the inertia matrix, $C(q,\dot{q})$ is the Coriolis matrix, $G(q)$ is the gravity vector, J is the Jacobian matrix of the contact position $\phi(q)$ such that $J = \partial\phi/\partial q$, B is the torque distribution matrix, u is the control input, and λ is the contact force. Any potential contact point of the system can be described as $\phi(q) = [\phi_x(q), \phi_z(q)]^T$. $\phi_x(q)$ is tangential to the contact surface and $\phi_z(q)$ is the normal distance. A contact is made when $\phi_z(q)$ reaches zero.

For each potential contact point $\phi(q)$ with the contact force $\lambda = [\lambda_x, \lambda_z]$ and velocity $\gamma = J\dot{q} \triangleq [\gamma_x, \gamma_z]$, if the sliding contact is permitted, a set of *complementary constraints* [30] can be used to describe the Coulomb friction model:

$$\phi_z(q),\ \lambda_z,\ |\gamma_x| \geq 0 \tag{4.2}$$

$$\mu\lambda_z - |\lambda_x| \geq 0 \tag{4.3}$$

$$\phi_z(q)\lambda_z = 0 \tag{4.4}$$

$$(\mu\lambda_z - |\lambda_x|)|\gamma_x| = 0 \tag{4.5}$$

where μ is the friction coefficient. On the other hand, if sliding is not allowed, instead of Eq. (4.5), the constraint Eq. (4.6) needs to be satisfied:

$$\lambda_z|\gamma_x| = 0 \tag{4.6}$$

4.2.2 Domains of Bipedal Robot Walking

In our setup, the model of AMBER 3 has four potential contacts: the heels and toes on both feet. The contact conditions for each foot can be defined as: i) toe-off ($\phi_{z,toe} = 0,\ \phi_{z,heel} > 0$), ii) heel-contact ($\phi_{z,toe} > 0,\ \phi_{z,heel} = 0$), and iii) flat contact ($\phi_{z,toe} = 0,\ \phi_{z,heel} = 0$). A specific contact domain in the bipedal robot walking gait will be determined by the contact conditions of both feet.

4.2.3 Trajectory Optimization and Locomotion Generation for a System with Multiple Domains

A trajectory for a dynamic system can be treated as a set of state as a function of time $x(t)$ resulting from its initial condition $x(t_0)$ and control $u(t)$. As the name implies, *trajectory optimization* is a set of local methods for planning the optimal trajectory $x(t)$ with $u(t)$, and minimizing the objective cost over a horizon $t \in [0, T]$.

Among various approaches, *trajectory optimization with direct collocation* in general has nicer numerical properties than the *indirect method*. Thus, it can be used to solve complex problems such as a system with multiple contact domains. For handling this type of problems, most state-of-the-art techniques assume the contact sequence is known or specified, and then the contact condition is either inserted explicitly through constrained dynamics, or implicitly through dynamics constraints like Eq. (4.1). This approach works well with HZD schemes, where a set of boundary conditions between domains (especially the ones with impact maps based on inelastic collision) need to be satisfied [61, 59, 31]. However, several potential issues may arise with the increase of contact points, such as determining the optimal contact sequence, and the increasing restriction for searching trajectories with more and more impact maps, which may rule out some potentially feasible trajectories. Therefore, the *optimization through contact*, which treats the contact sequence as a part of the trajectory becomes a nice resolution for those issues, will be introduced in next section.

4.3 Trajectory Optimization through Contact with Direct Collocation

Inspired by the time stepping method [63] used for forward simulation, Posa et al. [30] proposed the *trajectory optimization through contact*. The main idea of the time stepping method is to discretize the system state and control to formulate the multi-contact dynamics as a Linear Complementarity Problem (LCP). In this way, only the contact force acting over a period will be considered, which eliminates the need to differentiate between continuous and impulsive forces. Similarly, using direct collocation with trapezoid method, the optimization through contact directly optimizes the given cost function over a set of free variables including discrete states, control inputs, contact forces, time step and other slack variables. The general trajectory optimization through contact can be stated as:

$$\begin{aligned} \underset{\mathbf{x}=[h,x_1,\ldots,x_N,u_1,\ldots,u_N,\lambda_1,\ldots,\lambda_N]}{\text{argmin}} \quad & g_f(x_N) + h\sum_{k=1}^{N} g(x_k, u_k) \qquad (4.7) \\ \text{s.t.} \quad & \mathbf{x_{min}} \le \mathbf{x} \le \mathbf{x_{max}} \\ & \mathbf{f_{min}} \le \mathbf{f}(\mathbf{x}) \le \mathbf{f_{max}} \end{aligned}$$

where h is the time step, x_k is the discretized state variables $[q_k; \dot{q}_k]^T$ at k^{th} time step, $g_f()$ represents the final cost, and $h \sum g()$ is the integral cost. $\{\mathbf{x_{min}}, \mathbf{x_{max}}\}$ and $\{\mathbf{f_{min}}, \mathbf{f_{max}}\}$ are the vectors corresponding to the lower and upper bounds of decision variables and constraints respectively. Though the direct collocation scheme largely increases the number of free variables and constraints, the well-posed nature of the problem as a large sparse Non-Linear optimization Problem (NLP) (with sparse Jacobian matrices of the cost function and constraints) allows nonlinear optimization solvers like IPOPT [64, 31] and SNOPT [65, 30] to solve the NLP efficiently.

However, despite the concise and unified framework that can automatically derive the contact sequence by solving the NLP in Eq. (4.7), there exist several issues. For example, general walking motion generation usually requires 20 to 50 collocation points for one half gait cycle (i.e. a single step), where the step size is in the order of 10^{-2} second. However, the step size required for time stepping method with more accurate simulation result is in millisecond [63]. Therefore, the time step size h may not be small enough for accurate dynamic simulation. As a result, a transcription method with higher accuracy, such as Runge-Kutta method or Hermite-Simpson method should be considered to decrease the integral error. In addition, the initial guess to this local method [30] and the related scheme for relaxation of the complementary constraints also need to be carefully handled for improving the optimization performance and the quality of the generated gait. In this section, we will focus on the transcription using Hermite-Simpson method with the corresponding constraint setup, followed by the introduction of the cost function and other important constraints in our modified optimization framework. The related constraint relaxation scheme and other adjustments will be introduced in the next section.

4.3.1 General Setup

For the discretization of all state variables, we set the time step as $h = T/N$, where $T > 0$ is the duration of a half gait cycle and $N = (2N_c + 1)$ is the number of collocation nodes. To use the Hermite-Simpson method to describe the relationship between the adjacent state variables, the number of *cardinal nodes* N_c needs to be selected first. In this case, the odd points $(x_1, x_3, \ldots, x_{2N_c+1})$ are the cardinal nodes where the time duration between any adjacent cardinal points can be arbitrarily chosen. For simplicity, the fixed time step h is used in our framework, so the duration between two adjacent cardinal points is $2h$. On the other hand, the even points called interior points $(x_2, x_4, \ldots, x_{2N_c})$ need to be placed at the center of two adjacent cardinal nodes.

4.3.2 Transcription for Direct Collocation Using Hermite-Simpson Method

In our modified framework, the Hermite-Simpson method is chosen for improving the accuracy of numerical approximation for kinematics and dynamics [31, 47]. The constraints of this method can be expressed as:

$$x_k - \frac{1}{2}(x_{k+1} + x_{k-1}) - \frac{1}{8}h(\dot{x}_{k-1} - \dot{x}_{k+1}) = 0 \tag{4.8}$$

$$x_{k+1} - x_{k-1} - \frac{1}{6}h(\dot{x}_{k-1} + 4\dot{x}_k + \dot{x}_{k+1}) = 0 \tag{4.9}$$

The physical meaning of the constraints above is that the k_{th} state variable and its time derivative x_k and $\dot{x}_k$ on the interior node (approximated as a cubic spline) should match the state variables and its time derivative evaluated through the system's kinematic and dynamic equations explicitly. Note for this Hermite-Simpson method with local compression [28], only the states of cardinal nodes belong to the free variables, since the state and its derivative of the interior node can be explicitly calculated based on the constraints stated above. This method is referred as *the compressed form of Hermite-Simpson method* [28].

4.3.3 The Implicit Constraint Expression with Extra NLP Variables

As per the previously mentioned explicit calculation (the acceleration $\ddot{q}_i = f(q_i, \dot{q}_i)$ from the system dynamics), the state on the interior node in Eqs. (4.8) and (4.9) is coupled with the states on adjacent collocation points. In addition, this calculation requires the inverse of the inertia matrix. Though the compressed form of Hermite-Simpson method using less decision variables (because the states of internal nodes are functions of cardinal nodes), it can limit the sparsity of the Jacobian matrix of constraints. To improve the sparsity, extra NLP variables, such as $\ddot{q}_k$ of the cardinal points and the interior points' states and accelerations (x_i and $\ddot{q}_i$) is introduced so that the set of free variables in Eq. (4.7) becomes:

$$\mathbf{x} = [h, x_1, \ldots, x_N, \ddot{q}_1, \ldots, \ddot{q}_N, u_1, \ldots, u_N, \lambda_1, \ldots, \lambda_N]$$

On the other hand, instead of explicitly calculating the $\ddot{q}_k$ for all nodes with the inverse inertial matrix, the dynamic constraints in Eq. (4.1) for each time step are inserted. Please refer to [28] for *Hermite-Simpson (Separated) method (HSS)* in Chapter 4 for further details and discussions.

4.3.4 Cost Functions and Constraints

Similar to the previous works using trajectory optimization with direct collocation [31, 30, 47], the mechanical cost of transport (COT) is used with an additional sum of torque squared with a small scalar factor ω as shown:

$$cost(\mathbf{x}) = \frac{1}{mgd} \sum_{k=1}^{N} \sum_{i} |u_{k,i} \dot{q}_{k,i}| + \omega \sum_{k=1}^{N} u_k^T u_k \tag{4.10}$$

where mg is the system total weight, and d is the total traveling distance of system's center of mass (COM). Empirically, the sum of torque squared can help to improve the generated gait. Except for the kinematic and dynamic constraints (Eqs. (4.1), (4.8) and (4.9)), other important constraints are summarized here:

Contact constraint. Depending on whether the sliding contact is allowed or not, either Eqs. (4.2) to (4.5) (referred as SACC: sliding allowed contact constraints) or Eqs. (4.2) to (4.4) and (4.6) (referred as NSCC: non-sliding contact constraints) need to be satisfied for all cardinal and interior points.

Periodic constraints. To generate the nominal walking gait, the periodic constraints are expressed as shown:

$$q_1 - Rq_N = 0 \tag{4.11}$$

$$\dot{q}_1 - R\dot{q}_N = 0 \tag{4.12}$$

$$\dot{x}_{1,COM} = \dot{x}_{N,COM} \tag{4.13}$$

$$z_{1,COM} = z_{N,COM} \tag{4.14}$$

$$\dot{z}_{1,COM} = \dot{z}_{N,COM} \tag{4.15}$$

$$x_{N,COM} \geq x_{1,COM} + d_{min} \tag{4.16}$$

where R is the relabeling matrix which switches the state variables at the joints on the left leg to the right and vice versa, d_{min} is the minimum moving distance of x_{COM} (the horizontal position of COM), and z_{COM} is the vertical position of COM.

Contact constraint of stance toe. Since the duration of toe contact is slightly longer (about 55% for a full gait cycle) than one half gait cycle in the human walking gait analysis, the following constraint can further simplify the optimization without altering the objective for human-like walking gait generation, i.e. the stance toe ($stoe$) is constrained as:

$$\phi_{stoe}(q_k) = 0 \tag{4.17}$$

for $k = [1, \ldots, N]$. This constraint can also help to eliminate some undesirable gaits, i.e. the walking motion which includes the hopping in the half gait cycle.

4.4 Running the Optimization Towards Generating Human-like Walking Gait

With the modified framework of the trajectory optimization through contact, we improve the numerical approximation accuracy, the sparsity of the Jacobian matrix about kinematic and dynamic constraints. In addition, the quality of the generated gait is improved by introducing the additional terms in cost function. However, sometimes it is still tricky to derive a high-quality gait by solving the NLP just a few times. There are several potential reasons for that. First of all, the choice of the initial guess can lead to different feasible gaits that satisfy all the constraints as mentioned in [30]. Naturally, more human-like initial guesses or the cost function for fitting human data may lead to a more desirable result. But, such approaches would be deliberately guiding the optimization towards human-like gait. Our objective, on the other hand, is to naturally generate a human-like gait through trajectory optimization with general constraints and initial guess that is easy to generate. We believe that such an approach would be applicable for prosthesis, orthosis, or exoskeletons, because the produced results would be favorable to both the robotic systems and the humans interacting with them. Second, dependent on sliding contact condition, the generated behavior from the same initial condition may vary a lot because the different contact constraints have different numerical properties. Last but not least, since humans inherently have more passive components compared to a pure rigid-body model, it may be helpful to introduce some virtual components to slightly alter the generated gait. In the following subsections, the series of adjustments and schemes for improving the generated gait will be introduced.

4.4.1 Choice of the Initial Guess

As per our objective, the ZMP-based flat walking gait is chosen as the initial guess. The reasons for this choice are: i) The ZMP-based method for walking motion generation is widely-used. ii) With the simple flat-contact condition and two domains (i.e. the single support and double support phases), it is relatively easy to derive a dynamically feasible trajectory using constrained dynamics. iii) The generation of desired trajectory for ZMP-based walking (e.g. the ZMP trajectory, end-effector trajectory, torso angle) is straightforward.

4.4.2 Choice of Contact Constraints

In our implementation with the ZMP-based walking gait as the initial guess, the generated gait with SACC and NSCC are quite different, as shown in the next section. For the gait with SACC, the foot clearance of the swing foot is relatively small; the foot is almost sliding along the ground until it makes a step. On the other hand, the gait with NSCC behave more like a passive walker, which has a slightly larger sway-up motion before the heel-strike.

4.4.3 Virtual Springs on Ankles for Inducing Heel-strike Motion

For the resulting gaits that have no obvious heel-strike even with a larger d_{min} in Eq. (4.16), one potential solution is to add virtual passive components to the system. Here we choose to add a torsional spring with a small stiffness k to the ankle joint to emulate the effect of the human Achilles tendon at the ankle (which prevents the foot from dropping even when the ankle is relaxed). The equation of motion then becomes:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu + J^T\lambda - kBB_kq \tag{4.18}$$

where BB_k is the spring torque distribution matrix for assigning the spring torques to the torque equations of ankle joints. With this setup, the resultant torque applied to the system becomes $u - kB_kq$. For mitigating the effect of introducing virtual elastic components, the second term in the cost function Eq. (4.10) can be modified as follows:

$$\omega \sum_{k=1}^{N} (u_k - kB_kq_k)^T(u_k - kB_kq_k) \tag{4.19}$$

4.4.4 Contact Constraints for One-sided Springs on Toes

For the bipedal robot AMBER 3, a set of torsional springs are attached on the passive toe joints (Fig. 2.1). The mechanical joint limit is designed that the torsional spring will activate only when the foot is in toe-off condition. Therefore it can be approximated as a contact point which has a one-sided torsional spring. An additional set of complementary constraints for the toe with one-sided spring then can be expressed as follows:

$$k_{toe}\theta_{toe} = (T_1 + T_2) - T^- \tag{4.20}$$

$$(T_1 + T_2)T^- = 0 \tag{4.21}$$

$$T_1T_2 = 0 \tag{4.22}$$

$$\phi_{z,toe}T_1 = 0 \tag{4.23}$$

$$\phi_{z,toe}, T_1, T_2, T^- \geq 0 \tag{4.24}$$

where the k_{toe} is the stiffness, T_1, T_2, and T^- are slack variables for the one-sided spring, where the real torque applied to the system through the active spring is the variable T_1. Under this setup, the equation of motion becomes:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu + J^T\lambda - J^T_{\theta,toe}T_1 \tag{4.25}$$

where the $J_{\theta,toe}$ is the Jacobian matrix of the toe orientation.

4.4.5 Relaxations on the Complementary Constraints

For solving this problem using SNOPT with sparse sequential quadratic programming, it has been reported that it is practically useful to temporarily relax the complementary constraints [30] as follows:

$$M(x),\ N(x) \geq 0 \tag{4.26}$$

$$M(x)N(x) = \epsilon \tag{4.27}$$

where ϵ is a small nonnegative constant. On the other hand, we also found when using IPOPT (based on a primal-dual interior point method) to solve this type of the problem, the relaxation is also required. Without the relaxation, the primal-dual barrier approach will drive $M(x)$, $N(x)$ away from the boundary and leads to the worse local solution and convergence of the optimization problem. Since ϵ is sensitive to the optimization problem, we empirically used a simple grid search in the range of 10^{-3} to 10^{-1}, with a smaller maximum iteration number of the solver for quickly choosing a ϵ for a better start.

4.4.6 A Kinematic-based Trajectory Optimization for Increasing the Foot Clearance

Another observation from the result of the optimization through contact is that the ground clearance of the swing foot can be very small, probably caused by the minimization of the objective function which contains cost of transport. In general it should be improved by inserting the contact constraint in the form of $\phi_z(x) - f(x) \geq 0$, instead of the constraint $\phi_z(x) \geq 0$ in Eq. (4.2). However, practically it might easily be compromised by the relaxation of complementary constraints. For increasing the foot clearance effectively with minimal effect on the original gait, a kinematic-based trajectory optimization for the swing leg trajectory is adopted as a post processing:

$$\begin{aligned} \underset{\mathbf{x}=[x_1,\ldots,x_N]_{swing}}{\text{argmin}} \quad & \sum (q_k - q_{k,ref})^T (q_k - q_{k,ref}) \qquad (4.28) \\ \text{s.t.} \quad & \phi_{z,toe} \geq f(\phi_{x,toe}) \\ & \phi_{z,heel} \geq f(\phi_{x,heel}) \\ & \text{kinematic constraints in Eqs. (4.8) and (4.9)} \end{aligned}$$

where $q_{k,ref}$ is the joint trajectory derived from optimization through contact, and $f(\phi_x)$ is a normal distribution function of the contact point's horizontal position.

4.5 Optimization Results and Related Comparisons

The formulated optimization with different constraints were solved using IPOPT with the linear solver `ma57`.Depending on the relaxation and the initial guess, the required computation time varied from 30 seconds to 10 minutes. For the common parameter setup applied for all the cases, $\omega = 10^{-3}$, $k = 10Nm/rad$, $d_{min} = 0.5m$. Except for the ZMP-based walking (ZMP) as the initial guess, other generated gaits for comparison include:

Table 4.1: The list of the modified costs, stride lengths and double support percentage values for the initial guess, and generated gaits with different contact constraints. Reprinted with permission from [2].

Gait type	ZMP	SACC	NSCC	OSS
Cost	0.577	0.048	0.049	2.664
Stride length	0.2m	1.10m	1.0m	1.0m
Double support percentage	33.33%	31.37%	35.48%	35.48%

the optimization using SACC (SACC), the optimization using NSCC (NSCC), and the optimization with one-sided spring constraints and NSCC (OSS) (as shown in Fig. 4.3 to Fig. 4.5). The main quantities for comparison are listed in Table 4.1.

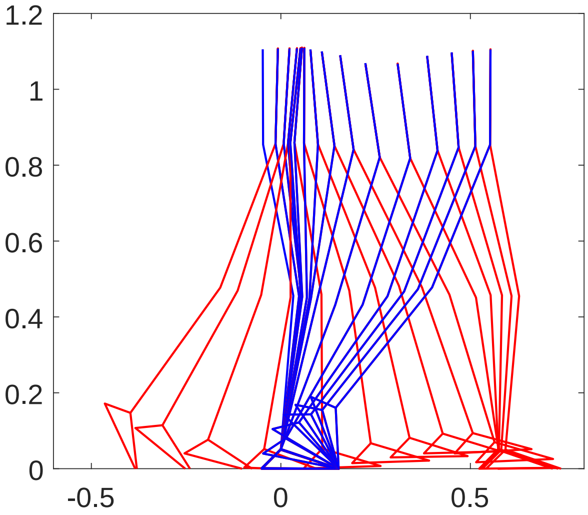


Figure 4.3: The walking tile of the generated gait with SACC. Reprinted with permission from [2].

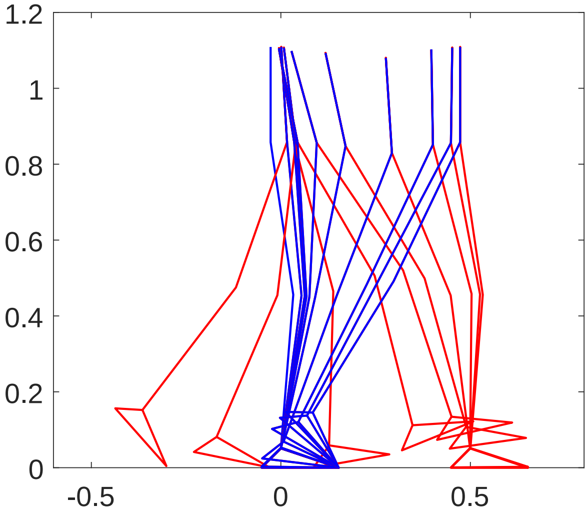


Figure 4.4: The walking tile of the generated with NSCC. Reprinted with permission from [2].

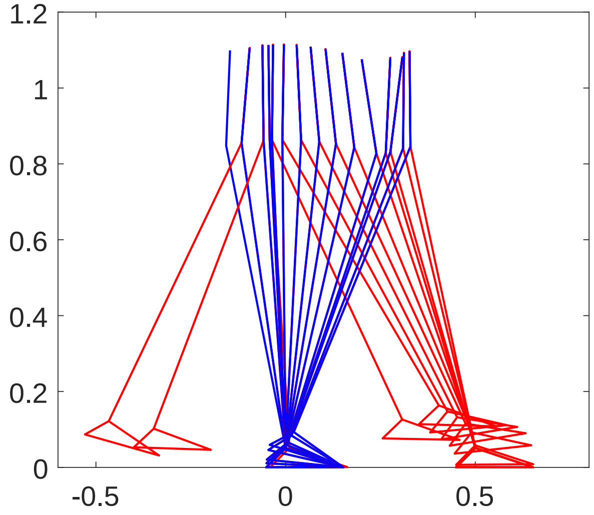
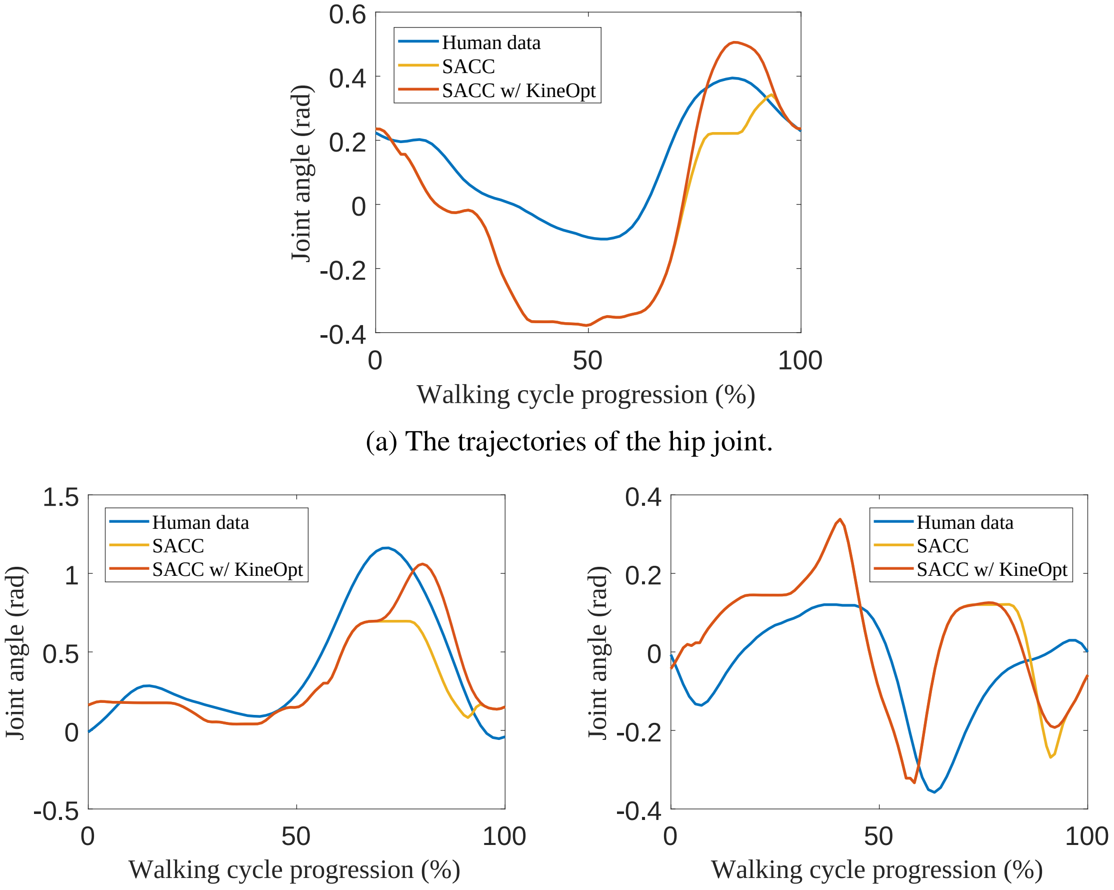


Figure 4.5: The walking tile of the generated gait with OSS. Reprinted with permission from [2].

The generated gaits with NSCC and SACC show that these two types of contact constraints can generate different gait characteristics. Thus the user can interchangeably switch among different constraints during the optimization process for getting a better result. In Fig. 4.6, the human gait, the gait with SACC before and after the kinematic optimization for increasing the swing foot clearance are compared. Although the discrepancies still exist, patterns of the gait with SACC and kinematic optimization are closer to the human ones. The differences observed in the ankle trajectories are larger than those of the knee and hip, but the concluding stage of the ankle trajectory with SACC and kinematic optimization is similar to the initial stage of the human ankle trajectory. Further adjustments of the introduced schemes are required to improve the phase difference here.

For the gait with one-sided spring constraints, although the constraints helps to decrease the toe-off angle (Fig. 4.5), the minimum cost was still quite high compared with the other gaits since there were more complementary constraints need to be satisfied or relaxed at the same time.



(a) The trajectories of the hip joint.

(b) The trajectories of the knee joint.

(c) The trajectories of the ankle joint.

Figure 4.6: The angular trajectory comparison between human data, gait SACC and gait SACC with the kinematic optimization. Reprinted with permission from [2].

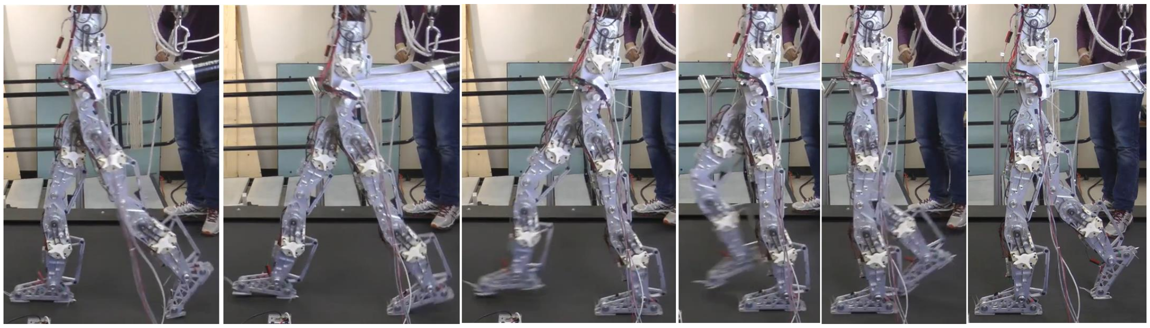


Figure 4.7: The walking tiles of the experiment using the bipedal robot AMBER 3 [3].

4.6 Conclusions and Future Work

In this work, we propose the modified optimization through contact, which uses Hermite-Simpson method to get a better approximation of the state trajectories. Several schemes such as virtual springs on ankles, additional constraints for the one-sided springs on toes, and the kinematic optimization to increase the foot clearance are provided to adjust the gait towards to more human-like motion. With the modified framework, a series of gaits with different constraints are generated and compared. To make the optimization with complementary constraints more tractable (e.g. for one-sided spring), other solvers using SQP method should be considered. Further adjustment for the parameters of the provided schemes is also required for more natural gait generation. On the other hand, further validations including more simulations and experiments are required for testing the stability and robustness of the gait. It is also important to measure the cost of transport (for the energetic efficiency) of the gait from the experiment, and compare it with the expected cost of transport from the optimization. Currently, the testing experiment (as shown in Fig. 4.7) is being undergone [3], and the artifacts from the support mechanism and treadmill will be resolved. We also plan to use this method for generating the trajectories for lower-limb prosthesis and exoskeleton.

5. GENERATING HUMAN-LIKE WALKING GAIT ON DIFFERENT TERRAINS USING HZD GAIT OPTIMIZATION

5.1 Introduction

Following the work presented in the previous chapter, the objective of this project is to generate energetically efficient walking gaits for bipedal robots on various terrains which are human-like. When it comes to a new terrain profile (assumed the profile is known), the contact constraints in the original optimization formulation need to be adjusted accordingly, which could potentially increase the infeasibility of the original initial guess. For a trajectory optimization method which is sensitive to the initial guess, that could increase the difficulties for solving walking motions on various terrains, because the users either need to generate different feasible initial guesses for different terrain profiles, or they need to take considerable time to adjust optimization parameters to get an working optimizer for each terrain condition. On the other hand, because human walking has really low Cost of Transport (COT) compared to the bipedal robots [35], the robot walking is desired to have the same contact sequence like human walking, which is with multiple domains (phases with different contact conditions). As a result, to determine the suitable approach for solving the walking gait with multiple domains in particular, we compared the capabilities of the states of the art: Trajectory optimization through contact [30] and Hybrid Zero Dynamics (HZD) gait optimization [24], as shown in Table 5.1.

Without the need of specifying the contact sequence, Trajectory optimization through contact (or Contact-implicit trajectory optimization), proposed by Posa et al. [30], is an approach which simultaneously optimizes all the states, controls and contact forces of all the potential contact points. By adopting the time-stepping method proposed by Stewart and Trinkle [63] which uses Euler method to discretize the states, the trajectory optimiza-

Table 5.1: Comparisons between HZD and Contact-implicit trajectory optimization.

Trajectory Optimization	HZD	Contact-implicit
Method	Direct collocation method	Direct collocation method
Objective function	Cost of transport	Cost of transport
Impact handling	Inelastic collision	Time-stepping method
Predefined contact sequence	Required	Not required
Sensitivity to initial guess	Relatively low	Relatively high
Discretization	Hermite-Simpson method	Trapezoid method

tion problem is formulated as a nonlinear optimization with complementary constraints. This time-stepping method only considers the integral of contact forces over a time step as an approximation of the effect from both the impulsive and continuous forces. However, this first-order approximation for states with relatively large time step (for bipedal walking generation is usually in the order of $10ms$) can cause unneglectable dynamic discrepancy. In addition, to make the optimization problem to be more well-posed, it has been reported that the relaxation of complementary constraints with nonlinear solvers such as SNOPT or IPOPT need to be done manually for a target environment [30, 2], therefore make it more difficult to be adapted to other terrain profiles efficiently. On the other hand, using the framework of controller design with hybrid zero dynamics, HZD gait optimization proposed by Hereid et al. [31, 59, 61, 24] relies on the pre-defined contact sequence. Although when the system is more complex, the combinatorics of the potential contacts grows exponentially and this method will be less practical [30], this approach works well for general walking motion generation. One important merit of this approach is that it is less sensitive to the initial guess (it has been reported it can be used with randomized initial guess [24]) and currently the implementation with third-order accuracy (by using Hermite-Simpson method for discretizing the states) has been achieved. As a result, to mitigate the effect caused by different initial guesses, we adopt the HZD gait optimization for walking motion planning, generalize it for various terrains, and propose several adjustments to make the walking motions more human-like.

5.2 Bipedal Locomotion as a Hybrid System

For a bipedal locomotion system, it is natural to describe it as a *hybrid system*, which contains both continuous and discrete dynamics. A **domain** (or a walking phase) in general is specified with a set of contact conditions across possible contact points. The continuous dynamics (like Eq. (4.1)) is used to describe the system behavior in a domain with the specified contact conditions. The discrete dynamics is used to describe the state transition from one domain to another, where the **guard** defines the subset of a domain to trigger the state transition (where at least one contact condition for a potential contact point is changed). A general bipedal locomotion (e.g. human walking) can be modeled as a *multi-domain hybrid system* as it has more than one domain.

Continuous Dynamics. Similar to Eq. (4.1), the dynamics of a rigid body model with a floating base can be expressed as follows:

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

where q is the generalized coordinate that includes $[x, z, \theta]^T$ (Fig. 4.2), $D(q)$ is the inertia matrix, $C(q,\dot{q})$ is the Coriolis matrix, $G(q)$ is the gravity vector, J is the Jacobian matrix of the contact position vector $\phi(q)$ such that $J = \partial\phi/\partial q$, B is the torque distribution matrix, u is the control input, and λ is the contact force. All active contact points of the system in a domain can be described as $\phi(q) = [\phi_x(q), \phi_z(q)]^T$, where $\phi_x(q)$ is tangential to the contact surface and $\phi_z(q)$ is the normal distance.

Discrete Dynamics. When the contact condition of the system is changed (e.g. a new contact is achieved or a existing contact breaks), the state of the system will have a discrete change, which can be represented as a *reset map* to project the system state at the guard to the state in the next domain. A classic example is the joint velocity change due to the impact induced by heel-strike (or foot-strike), which is a common phenomena across different walkers. Following the hypothesis listed in [66], there are three important assumptions for a bipedal locomotion system:

(i) The robot configuration is invariant under impact, i.e. $q^- = q^+$, where q^- is the pre-impact joint position and q^+ is the post-impact joint position.

(ii) The collision is inelastic, and the position of the new established contact is fixed during collision.

(iii) Following the second assumption, the momentum is conserved during the impact collision.

With those assumptions, the equations govern the kinematics and dynamics about *the new established contact* can be expressed as*

$$\begin{bmatrix} D(q^-) & -J_e^T \\ J_e & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ F_{impact} \end{bmatrix} = \begin{bmatrix} D(q^-)\dot{q}^- \\ 0 \end{bmatrix} \tag{5.2}$$

where the $D(.)$ is the inertia matrix (note $D(q^-) = D(q^+)$), J_e is the Jacobian matrix of the new established contact point, and F_{impact} is the integral of the impact force over the impact period. The first row of Eq. (5.2) is the momentum equation during impact under assumption (iii), and the second row is the velocity of the new-established contact position under assumption (ii). Note for the case where only the existing contact breaks, the state continuity will be hold (i.e. $q^- = q^+$ and $\dot{q}^- = \dot{q}^+$), as there is no impact force induced.

*From Eq. (5.2) the reset map to get $\dot{q}^+$ can be derived.

5.3 HZD Gait Optimization for Walking with Multiple Contact Phases

Following the work in [24], the method of trajectory optimization using direct collocation for hybrid systems will be introduced. Compared to the optimization through contact introduced in the previous chapter, the main difference of HZD gait optimization is that the contact sequence need to be specified, and then the trajectory in the previous optimization will be divided to several domains (the period of each domain also becomes free variable) and the bounds of dynamic constraints are specified according to the predefined contact conditions.

5.3.1 Hermite-Simpson Collocation

In the direct collocation framework using Hermite-Simpson method, all the joint variables q, $\dot{q}$, $\ddot{q}$ are discretized as nodes of cubic-splines. The derivative of the states at the interior node between two cardinal nodes expressed from the cubic-splines should match the derivative expressed via state-space equations of system kinematics and dynamics [31]. The constraints, assume the state at the node k as $x_k = [q_k, \dot{q}_k]^T$, the Hermite-Simpson collocation constraint can be expressed as:

$$H_{HSM}(x) = \begin{cases} x_k - \frac{1}{2}(x_{k+1} + x_{k-1}) - \frac{1}{8}h(\dot{x}_{k-1} - \dot{x}_{k+1}) & = 0 \\ x_{k+1} - x_{k-1} - \frac{1}{6}h(\dot{x}_{k-1} + 4\dot{x}_k + \dot{x}_{k+1}) & = 0 \end{cases} \tag{5.3}$$

where we impose this constraint with the *the compressed form of Hermite-Simpson method* [28]. Note this constraint is only applied on the collocation points within the same domain, the constraint to relate the boundary collocation points between domains will be introduced in Section 5.3.4.

5.3.2 Constrained Dynamics

At each collocation point x, given the joint state variables q, $\dot{q}$, $\ddot{q}$[*], the control u, and ground reaction forces λ at the active contact point(s) $\phi(q)$ where its Jacobian matrix is $J = \partial\phi/\partial x$, the constraints of constrained dynamics can be expressed as follows:

$$H_{CDym}(x) = \begin{cases} D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) - Bu - J^T\lambda = 0 \\ J\ddot{q} + \dot{J}\dot{q} = 0 \end{cases} \tag{5.4}$$

where the last equation is the time derivative of $J\dot{q} = 0$ (the velocity of the contact point).

5.3.3 Contact Sequence from Human Data

When a stable periodic gait with multiple domains reaches the steady state, the order of phases and the transitions in general will be fixed and periodic, therefore it enables us to use predetermined contact sequence to solve the HZD gait optimization. Similar to the sequence used in [24] derived from human data, we use the contact sequence as shown in the following figure:

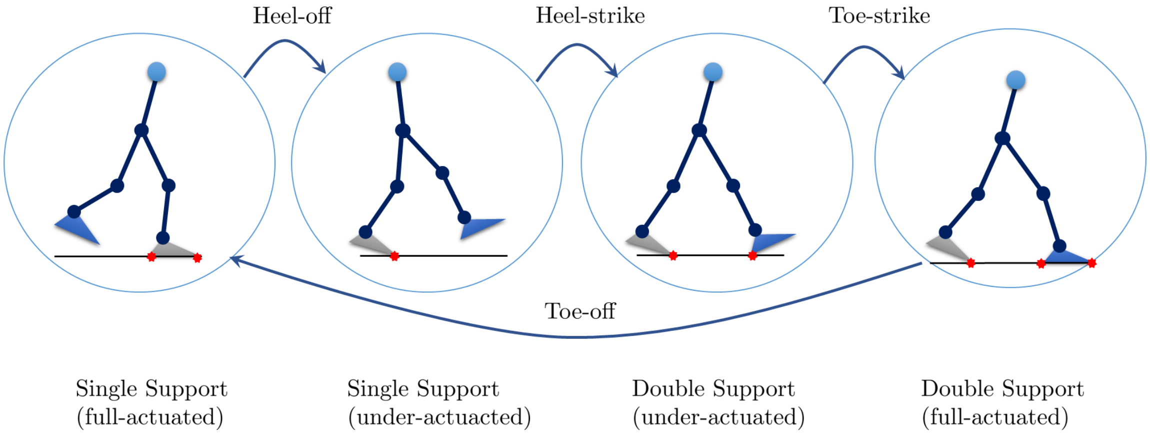


Figure 5.1: The schematic (a directed graph) of the contact sequence from human data.

Note since human walking is symmetric, in the hybrid optimization only the motion of the

[*] Starting from this section, for simplicity the subscript k for every free variables at node (collocation point) k is omitted, e.g. $x_k \to x$, $q_k \to q$.

first half walking cycle is solved, with the periodic constraints imposed accordingly.

5.3.4 Constraints Setup for Flat Terrain

In this section the other main building blocks for formulating HZD gait optimization for a multi-domain hybrid system will be introduced. Starting from the contact constraints, the method to impose contact dynamics for all potential contacts with given contact condition will be explained, and the periodic condition and the impact equations (Eq. (5.2)) to relate boundary collocation points between domains will be then introduced.

Contact Constraints Inspired by Optimization through Contact. For each active contact point $\phi(q)$ with corresponding contact force λ_x and λ_z, assuming the contact position is non-sliding (assumption (ii) described in Section 5.2), a set of equalities and inequalities can be used to describe the Coulomb friction model (for the 2D case):

$$\lambda_z \geq 0, \ \infty \geq \lambda_x \geq -\infty \tag{5.5}$$

$$\mu\lambda_z - |\lambda_x| \geq 0 \tag{5.6}$$

$$\phi_z(q) = 0 \tag{5.7}$$

$$J\dot{q} = 0 \tag{5.8}$$

where μ is the friction coefficient, $\phi_z(q)$ is the normal distance from the contact point to the contact surface and J is the Jacobian matrix of the position vector of the contact point. Note the constraints above is the special case of the contact constraints with complementary constraints by imposing Eq. (5.7) and Eq. (5.8). To simplify the optimization problem and improve its sparsity, we adopted a scheme similar to the one depicted in [30] to introduce a few slack variables to replace the absolute value and the normal velocity at the active contact points as follows:

$$H_{ActiveContact}(x) = \begin{array}{l} \lambda_z,\ \lambda_x^-,\ \lambda_x^+ \geq 0, \gamma = 0 \\ \begin{cases} \mu\lambda_z - \lambda_x^- - \lambda_x^+ \geq 0 \\ \phi_z(q) = 0 \\ \gamma - J\dot{q} = 0 \end{cases} \end{array} \tag{5.9}$$

where $\lambda_x^- + \lambda_x^+ = |\lambda_x|$ and $-\lambda_x^- + \lambda_x^+ = \lambda_x$. On the other hand, the constraints for inactive contact points can be expressed as :

$$H_{InactiveContact}(x) = \begin{array}{l} \lambda_z,\ \lambda_x^-,\ \lambda_x^+ = 0, \infty \geq \gamma \geq -\infty \\ \begin{cases} \mu\lambda_z - \lambda_x^- - \lambda_x^+ = 0 \\ \phi_z(q) \geq 0 \\ \gamma - J\dot{q} = 0 \end{cases} \end{array} \tag{5.10}$$

The constraints for inactive contact points seem redundant by intuition, but its insertion to the optimization actually greatly simplify the optimization formulation for the following reasons:

(a) This can simplify the formulation of both constrained dynamics and contact constraints because the only difference between the active and inactive contact constraints are their constraint and variable bounds. Therefore, the same the dynamic equations and the same contact constraints of all the potential contact points can be generally expressed in every domains.

(b) Similar to the the compressed form of Hermite-Simpson method [28], this method can slightly improve the sparsity of the optimization by introducing extra free variables.

(c) On the other hand, without changing the constraint expression, the contact constraint of the inactive contact point can also be modified to express the contact condition of the collocation point at the guard (when a contact point is about to achieved):

$$\lambda_z,\ \lambda_x^-,\ \lambda_x^+ = 0, \gamma_z \leq 0$$

$$H_{Guard}(x) = \begin{cases} \mu\lambda_z - \lambda_x^- - \lambda_x^+ = 0 \\ \phi_z(q) = 0 \\ \gamma_z - J_z\dot{q} = 0 \end{cases} \tag{5.11}$$

Boundary Constraints. As we introduced in the precious section about the discrete dynamics (Eq. (5.2)), the boundary constraints of collocation points between each domain can be expressed as:

$$\infty \geq F_{impact_x} \geq -\infty, F_{impact_z} > 0, F_{impact} = [F_{impact_x}, F_{impact_z}]^T$$

$$H_{Boundary}(x) = \begin{cases} q^+ - q^- = 0 \\ D(q^-)(\dot{q}^+ - \dot{q}^-) - J_e^T F_{impact} = 0 \end{cases} \tag{5.12}$$

Note the constraint $J_e\dot{q}^+ = 0$ is removed from the boundary constraints because it is already imposed as part of the contact constraint in the new domain in Eq. (5.9).

Periodic Constraints. The periodic condition is the slightly modified boundary constraint with an extra constraint as shown below:

$$\infty \geq F_{impact_x} \geq -\infty, F_{impact_z} > 0, F_{impact} = [F_{impact_x}, F_{impact_z}]^T$$

$$H_{Periodic}(x) = \begin{cases} R(q_{start}) - q_{end} = 0 \\ D(q_{end})(R(\dot{q}_{start}) - \dot{q}_{end}) - J_e^T F_{impact} = 0 \\ x_{com}(q_{end}) - x_{com}(q_{start}) \geq d_{min} \end{cases} \tag{5.13}$$

where R is the relabeling matrix to swap joint angles between legs, and d_{min} is the minimum horizontal traveling distance of the center of mass x_{com}.

5.3.5 Optimization Formulation

Assuming the target hybrid systems has N domains, M collocation points, define the set of free variable $X = \{q_i, \dot{q}_i, \ddot{q}_i, u_i, \lambda_i, \gamma_i, F_{impact_n}, \Delta t_n\}$ for all $i \in [1, 2, \dots M]$, $n \in [1, 2, \dots N]$. With the constraints introduced in the previous sections Eqs. (5.3), (5.4) and (5.9) to (5.13), the HZD gait optimization can be expressed as the following:

$$\begin{aligned} X^* = \underset{X}{\text{argmin}}\ & J(X) \\ \text{s.t.}\quad & x_{lb} \leq x \leq x_{ub} \\ & H_{HSM}(x) = 0 \\ & H_{CDym}(x) = 0 \\ & H_{Contact}(x) \\ & H_{Boundary}(x) = 0 \\ & H_{Periodic}(x) = 0 \end{aligned} \tag{5.14}$$

where $J(X)$ is the cost function, x_{lb} and x_{ub} are the lower bound and upper bound of X. $H_{Contact}(x)$ are the collection of contact constraints with stacked $H_{ActiveContact}(x)$, $H_{InactiveContact}(x)$, and $H_{Guard}(x)$ where the order is determined by the contact conditions of the domains. Because the contact constraints contain both equalities and inequalities so the right hand sides are omitted, please refer to Eq. (5.9) and Eq. (5.10) for the detailed expressions.

Cost Function. One of the popular choice of the cost functions for bipedal locomotion is the Cost of Transport (COT) [61, 30]: The cost function is the integral of the mechanical power divided by the total mass times the traveling distance, which can be approximated with Simpson's quadrature rule [24]:

$$J(X) = \frac{1}{m_{total} d} \sum_{n=1}^{N} J_n(X) \tag{5.15}$$

where m_{total} is the total mass, d is the horizontal traveling distance of the COM, and

$$J_n(X) = \sum_{j=1}^{M_n} \omega_j P(u_j, \dot{q}_j) \ , \tag{5.16}$$

$$\omega_i = \begin{cases} \frac{1}{6}\Delta t_n & \text{if } j = 1 \text{ or } j = M_n \\ \frac{2}{3}\Delta t_n & \text{else} \end{cases} \tag{5.17}$$

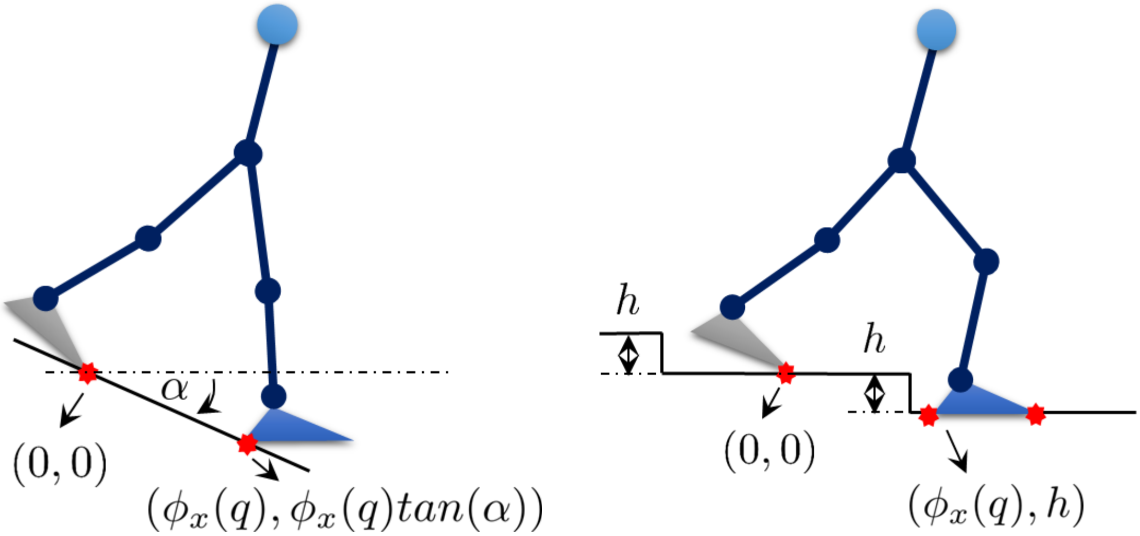
where M_n is the number of collocation points in a domain n ($n \in [1, 2, ..., N]$), and $P(.)$ is the summation of the absolute values of power consumption of all actuators (the reason to take absolute value is because the actuator cannot actually do negative work without power consumption, unless there is an elastic component to store the energy).

5.4 Modified Contact Constraints of HZD Gait Optimization for Different Terrains

In the original HZD gait optimization introduced in Eq. (5.14), it aims for solving multi-domain walking on the flat terrain. Therefore, the next step will naturally be extend this formulation to other terrains. Assuming the terrain profile is given, to generate walking gait for different terrains such as slope or stairs[†], it can be easily achieved by modifying the contact condition $\phi_z(q) = 0$ in and Eq. (5.9), Eq. (5.10) and Eq. (5.11). The following we will discuss the contact constraint modifications for both the periodic slope walking and walking on the stairs.

Slope walking. As shown in Fig. 5.2b (a), assuming the origin is at the toe of the trailing

[†] In this work we only focus on the periodic motion planning on slop or stairs, therefore the slope angle or stair height is assumed to be constant. But the optimization formulation can be readily extend to other terrain profile, where the periodic constraint need to be removed and additional terminal constraints for proper contact conditions need to be imposed.



(a) Slope walking

(b) Stair walking

Figure 5.2: The schematics of slope walking and stair walking.

leg and the slope angle is α, the slope heights for all active contact points can be derived based on the horizontal positions. As a result, the original constraint $\phi_z = 0$ can be modified as:

$$\begin{cases} \phi_z(q) - \phi_x(q)tan(\alpha) = 0 & \text{if a contact point is active} \\ \phi_z(q) - \phi_x(q)tan(\alpha) >= 0 & \text{if a contact point is inactive} \\ \phi_z(q) - \phi_x(q)tan(\alpha) = 0 & \text{if a contact point is inactive but at guard} \end{cases} \tag{5.18}$$

Stair walking. For the case of walking on the stairs, since the change of the stair height is discrete, therefore we need to check the direct graph to check the desired step height of a contact point. As shown in Fig. 5.1, the original constraint of contact point position in the normal direction as shown in Fig. 5.2b (b):

$$\begin{cases} \phi_z(q) - h = 0 & \text{if a contact point is active} \\ \phi_z(q) - h >= 0 & \text{if a contact point is inactive} \\ \phi_z(q) - h = 0 & \text{if a contact point is inactive but at guard} \end{cases} \tag{5.19}$$

where the stair height h is:

$$\begin{cases} -h & \text{for the contact points on the previous stair} \\ 0 & \text{for the contact points on the current stair} \\ h & \text{for the contact points on the next stair} \end{cases} \tag{5.20}$$

5.5 Additional Schemes Towards Human-like Motion

In Eqs. (5.14) and (5.18) to (5.20), the main work of this chapter – the fundamental framework of multi-domain hybrid locomotion on different terrains using HZD gait optimization has been introduced. In this section we introduce several schemes (in the form of additional constraints, cost function, or variable bounds) that are helpful to make the generated trajectory more human-like or improve the average converging speed in the optimization solving process.

Torso swaying. One common situation of the optimization result is that the locomotion sometimes will come with large torso swaying, which is usually undesired for human to perform upper body manipulation or have a steady view. This situation can be eased by constraining the torso angle of all collocation points within a smaller range:

$$\theta_{torso}^{max} \geq \theta_{torso}(q) \geq \theta_{torso}^{min} \tag{5.21}$$

Knee stretching. Another common difference between our optimization result and human walking is that sometimes the knee stretching of the optimization result (in the preswing on the trail leg or in the terminal swing on the lead leg) is not obvious. It could be because the robotic model does not have any elastic components, therefore it will not follow the trend like a walking spring-loaded inverted pendulum (SLIP) model to restore and releasing the energy during a walking cycle. To improve this, an additional cost can be added to the original cost $J(X)$ introduced in Eq. (5.15):

$$J_{kneeStretching} = \omega_{knee} \sum_{i=1}^{N} q_{i_{lknee}} + \omega_{knee} \sum_{i=1}^{N} q_{i_{rknee}} \tag{5.22}$$

where the subscripts $lknee$ and $rknee$ indicate the left and right knee separately.

Free variable bounds. During the tuning process for improving the optimization result, we found that the variable bounds can affect a lot for the speed of convergence. When there is no bound for the free variables, the solver could waste a lot of time to evaluate solution that is not actually desired or even physically feasible, while when the variable bound is too tight, it could make it more difficult for solver to find a solution because the interior method used in IPOPT will try to push the solution away from the variable bounds, which can also lead to convergence issues for more complicated problems. Those findings also match the observations presented in [24]. After checking the range of motion from human data, and other physical parameters of AMBER3, we choose the variable bounds to form the variable bound px_{lb} and px_{ub}, where p is a factor to enlarge the variable bound a bit to improve the convergence of the optimization solving process. All the variable bounds and the parameter p we used are listed in Table 5.2.

Foot clearance. In the optimization result, because the main objective is to minimize the COT, the foot clearance of the swing leg is usually pretty close to the ground, which is easy to get tripping, or is difficult to cross over small obstacles. Assuming the desired foot

Table 5.2: Important bounds for free variables

Parameter	Range	Parameter	Range
$q_{ankle}(rad)$	$[-0.3, 0.3]$	$u_i(Nm)$	$[-80, 80]$
$q_{knee}(rad)$	$[0, 1.4]$	$\lambda_i\ (N)$	$[-500, 500]_x$, and $[0, 500]_z$
$q_{hip}(rad)$	$[-0.5, 0.5]$	$\gamma_i\ (m/s)$	$[-100, 100]$
$q_{toeoff}(rad)$	$[0, 0.6]$	$\gamma_i\ (m/s)$ (guard)	$[-\infty, 0]$
$\dot{q}_i(rad/s)$	$[-20, 20]$	$F_{impact}\ (N)$	$[-\infty, \infty]_x$, and $[0, \infty]_z$
$\ddot{q}_i\ (rad/s^2)$	$[-20, 20]$	$\Delta t_n\ (sec)$	$[1e-5, 0.05]$
$\theta_{torso}\ (rad)$	$[-0.15, 0.15]$	p	1.2

clearance and its location is known, one can create a smooth profile by combing two cubic splines, as shown in Fig. 5.3 with the given (x_{start}, y_{start}), (x_{mid}, y_{mid}), and (x_{end}, y_{end}). Since the desired profile is above the line between (x_{start}, y_{start}) and (x_{end}, y_{end}), then if we replace the original terrain profile with this desired one, the locomotion with a larger foot clearance can be solved. This method can apply to basically any potential contact point if a more detailed gait modification is required.

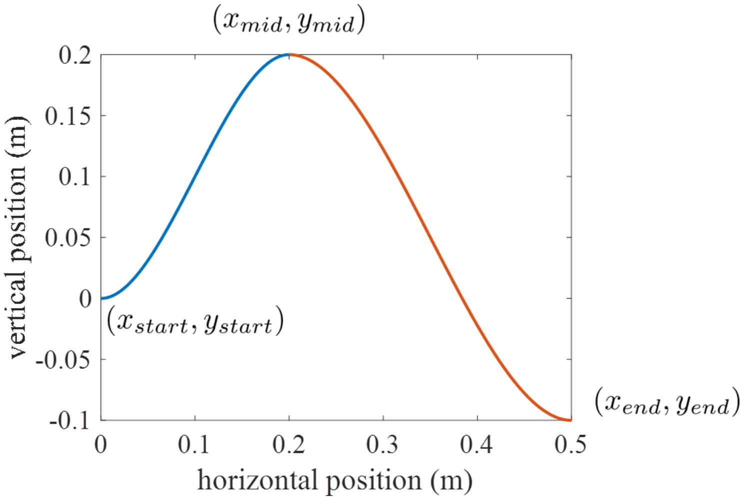


Figure 5.3: An example of a smooth curve combined by two cubic splines as the profile of the desired height of the foot clearance constraint.

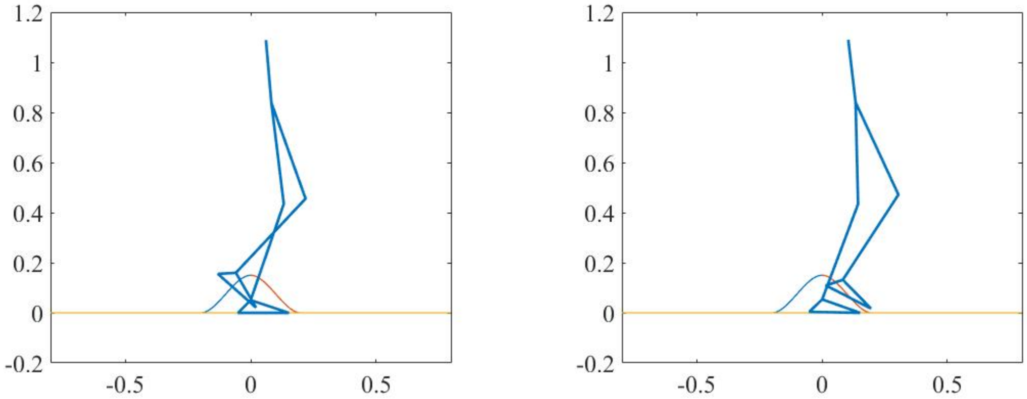


Figure 5.4: An example to use the desired height profile for the constraint of swing ankle (the top vertex of the triangle) height to increase the foot clearance.

To demonstrate how this work in a simpler way, we show an example where a height constraint is imposed to the swing ankle height $ankle_z(q)$ with a given smooth profile (combined with two cubic splines $a(x)$ and $b(x)$) as shown in Fig. 5.4, which can be expressed as follows:

$$\begin{cases} ankle_z(q) - a(x) \geq 0 & \text{if } x \geq x_{start} \text{ and } x \leq x_{mid} \\ ankle_z(q) - b(x) \geq 0 & \text{if } x \geq x_{mid} \text{ and } x \leq x_{xEnd} \\ ankle_z(q) \geq 0 & \text{else} \end{cases} \tag{5.23}$$

Since the created desired profile is continuous and both $a(x)$ and $b(x)$ are piecewise differentiable, therefore the analytical gradient of this constraint can also be calculated for the optimization solver. With this constraint setup, the generated locomotion as shown in Fig. 5.4 respects the constraint and keep the ankle height higher than the desired profile, therefore can enlarge the foot clearance.

5.6 Optimization Results

In this section, the optimization results of bipedal walking on flat ground, different slopes and stairs are presented. Starting from the optimization setup, we will introduce the optimization results for different terrains, including remarks about the robot configuration and torso angle limit. The result comparisons to human data and the related optimization sensitivity to the initial guess for each terrain will also be presented and discussed.

5.6.1 Optimization Setup

We use IPOPT [64] with the linear `ma57` solver and its MATLAB interface for the implementation of HZD gait optimization. In addition, we choose optimization option `optOptions.ipopt.hessian_approximation` as `'limited-memory'`, and `optOptions.ipopt.mu_strategy` as `'adaptive'`, and the optimization summary is shown in Table 5.3. To use IPOPT MATLAB interface, it is required to provide analytical Jacobian matrices in the form of sparse matrices. The symbolic expressions of the Jacobian matrices of the objective function and constraints are derived using Wolfram Mathematica and exported as C++ functions so that they can be used as MEX files to speed up the Jacobian matrix evaluation in MATLAB. Then the sparse matrices are created using the those MEX Files with the MATLAB build-in function `sparse()`.

Table 5.3: Details of the HZD gait optimization for bipedal robot AMBER 3.

Free variable number	2355	Constraint number	2721
Number of equality constraints	1927	Number of inequality constraints	794
Domain number	4 (Configurable)	Node number in domains	[21,21,21,5]
Objective function	Cost of transport	Jacobian sparsity	0.4%
Jacobian matrices	Analytical	Hessian matrices	Approximation

It is known that the direct collocation will greatly increases the number of free variables and constraints, which seems undesired for formulating a nonlinear program. However, there are a few noticeable advantages. First, similar to the advantage of the multi-shooting method over the single-shooting method in direct shooting, because the 'decision weights' are distributed to more variables, it eases the difficulties of solving the nonlinear program for more complex systems. Second, in the direct collocation framework, most of the free variables only affect the constraints of recent nodes, therefore the density of the Jacobian matrices in our application is really low (e.g. the sparsity pattern shown in Fig. 5.5), which can be solved efficiently with the sparse nonlinear program solver like IPOPT.

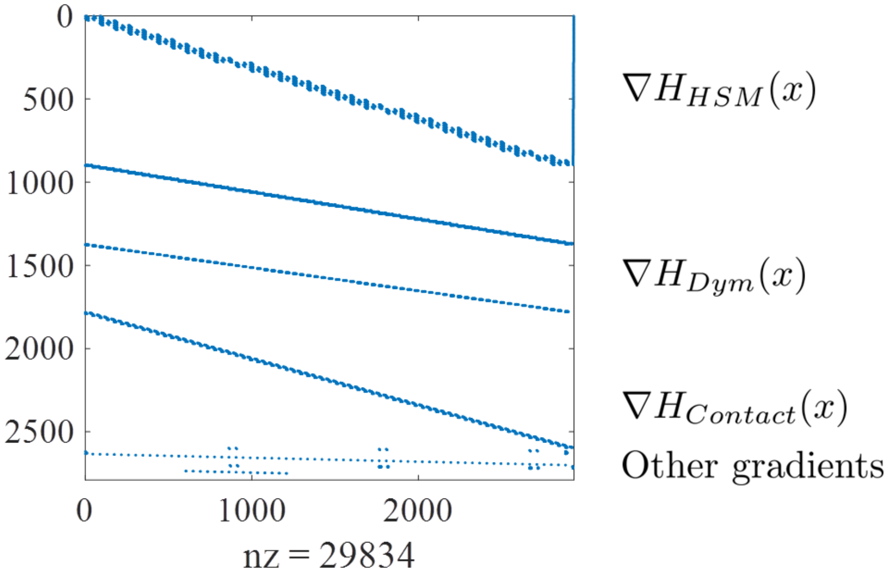


Figure 5.5: The sparsity pattern of the Jacobian matrix of the constraints. The makers indicate the nonzero elements.

5.6.2 Walking on Different Terrains

In this section, we present our main result – walking on different terrains (flat ground, ramp and stair) generated from the introduced optimization framework for AMBER 3. Note all the optimization results are the periodic gaits. Therefore it is assumed that the terrain profile (slope angle, stair height, or stair width) is consistent in the optimization.

Table 5.4: The summary of optimization results on different terrains

Terrain	COT	Step time (s)	Δt_n (s)
Flat ground	0.096	1.35	$[0.030, 0.030, 0.014, 1e-5]$
Ramp descent $(\alpha = -0.1rad)$	0.121	1.48	$[0.030, 0.030, 0.021, 1e-5]$
Ramp descent $(\alpha = -0.2rad)$	0.214	1.59	$[0.030, 0.030, 0.023, 0.023]$
Ramp descent $(\alpha = -0.2rad,$ $\theta_{torso} \in [\pm 0.15rad])$	0.529	1.44	$[0.030, 0.030, 0.019, 1e-5]$
Ramp ascent $(\alpha = 0.1rad)$	0.170	1.39	$[0.030, 0.030, 0.016, 1e-5]$
Ramp ascent $(\alpha = 0.2rad)$	0.266	1.58	$[0.030, 0.030, 0.022, 0.028]$
Stair descent $(h = -0.1m, w = 0.5m)$	0.356	1.20	$[0.030, 0.030, 0.006, 1e-5]$
Stair ascent $(h = 0.08m, w = 0.5m)$	0.357	1.16	$[0.030, 0.030, 0.004, 1e-5]$

To generate the optimization results, we ran the same optimization for each terrain with 200 randomized initial guesses, and then picked the solution with the lowest COT (the sensitivity of the initial guess for HZD gait optimization will be introduced and discussed in the next subsection). The cost of transport, walking speed, and time step in each domain (Δt_n) of the optimization results are listed in Table 5.4. The order of the domains and the corresponding contact conditions are depicted in Fig. 5.1. Also, the default torso angle range θ_{torso} for slope walking is set as $[-\alpha, \alpha]$ if it is not stated. Except the parameters explicitly mentioned or listed in Table 5.4 (in the first column from the left), the other bounds of decision variables listed in Table 5.2 are consistently used. In the following the walking tiles and remarks of walking on different terrains will be presented and discussed.

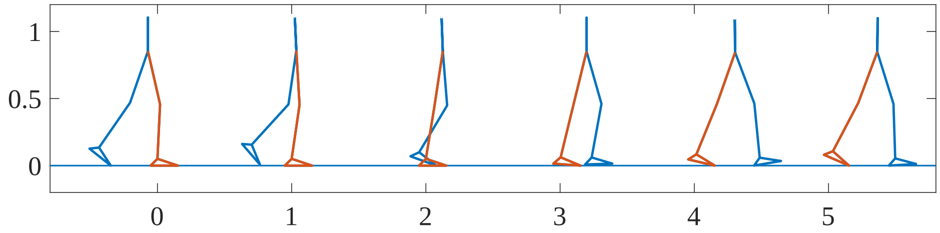
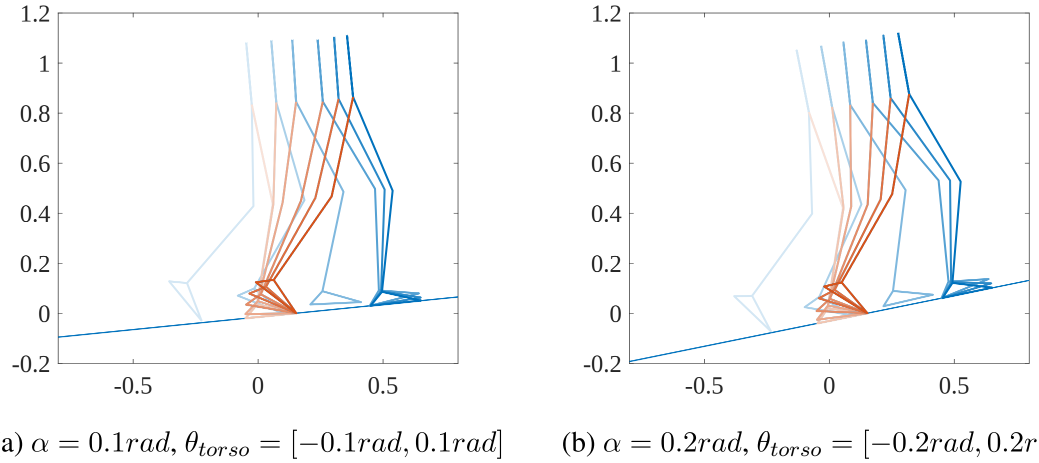


Figure 5.6: The walking tiles of the generated level-walking using HZD gait optimization.

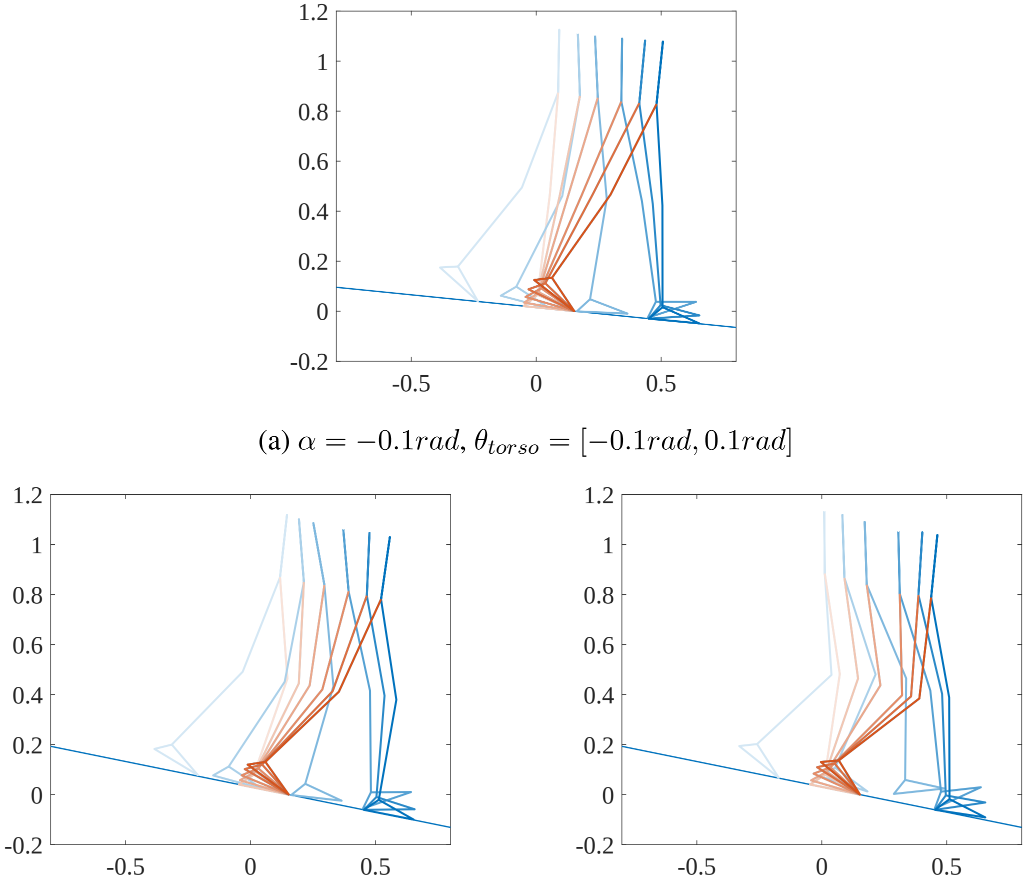
Flat-ground walking. Among walking on different terrains, level walking with multi-domain has the lowest COT 0.096. In the walking tiles shown in Fig. 5.6, it can show the walking gait is generated with the desired contact sequence. The extremely small forth Δt_n in Table 5.4 indicates that the toe-strike of the front leg and the toe-off of the trailing leg happen almost at the same time (as $1e-5$ is the lower bound of Δt_n).



(a) $\alpha = 0.1rad$, $\theta_{torso} = [-0.1rad, 0.1rad]$ (b) $\alpha = 0.2rad$, $\theta_{torso} = [-0.2rad, 0.2rad]$

Figure 5.7: The walking tiles of up-slope walking.

Slope walking. Compared to level walking, we can find that for slope walking (as shown in Fig. 5.8 and Fig. 5.7), the larger the slope angle ($|\alpha|$), the larger the COT. The optimization results in Table 5.4 also indicate that gait on ramp ascent with the same slope angle generally requires larger COT than the ramp descent. Another finding of the comparison



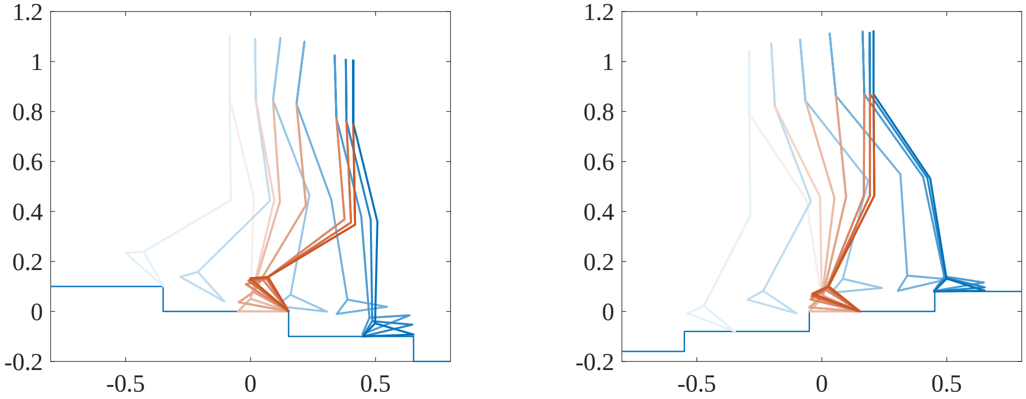
(a) $\alpha = -0.1rad$, $\theta_{torso} = [-0.1rad, 0.1rad]$

(b) $\alpha = -0.2rad$, $\theta_{torso} = [-0.2rad, 0.2rad]$ (c) $\alpha = -0.2rad$, $\theta_{torso} = [-0.15rad, 0.15rad]$

Figure 5.8: The walking tiles of down-slope walking.

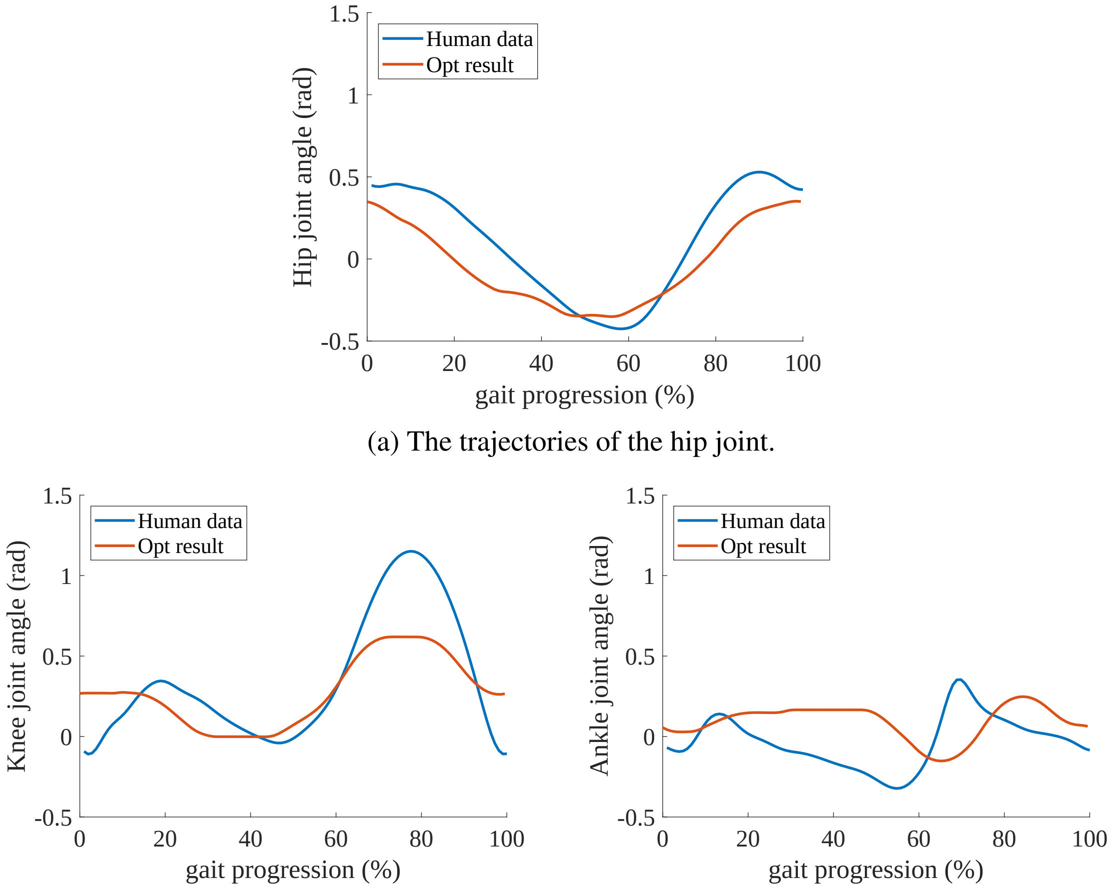
between walking result on ramp ascent and descent is that up-slope walking tends to have a smaller torso swaying, as the torso swaying on up-slope requires more energy to work against the gravity. In addition, we also noticed that when the slope angle becomes lager, the forth Δt_n in Table 5.4 becomes much greater than $1e-5$ which indicates the gait on a steeper slope requires longer time to transit the load from the trailing leg to the front leg. We also compared the optimization results of slope walking with the same slope angle but different torso angle limit, as shown in Fig. 5.8 (b) and (c). With a larger torso angle limit, the gait in Fig. 5.8 (b) is capable of swaying the entire body forward in the way similar to the level walking with a smaller COT, while with a smaller torso angle limit, the gait in Fig. 5.8 (c) needs to hold its torso upright which also limit its horizontal velocities, therefore requires the robot to lower the entire body vertically to reach the next stepping location, results in a more cautious gait with the higher COT.

Stair walking. Fig. 5.9 shows the optimization result of stair walking ascent and descent. One obvious difference of stair walking results from the other gaits is the third Δt_n in Table 5.4 becomes smaller so that the front foot can achieve flat contact more quickly. To demonstrate the capability of the optimization framework for stair walking, we only tried few terrain profiles with the similar motion range of sloped walking for several reasons. First, compared to the slope walking, stair walking is more complicated to be solved as the contact constraints for stair walking is not smooth. Second, because of the constraint complexity, the optimization is better to be solved with certain initial guess rather than the randomized one (e.g. the slope walking gait with similar height and step length). Third, the biomechanics study of stair walking [67] indicates that in stair walking the forefoot strikes the ground first, which is different from the sequence we used. To get better optimization results for stair walking, the contact sequence is required to be modified accordingly.



(a) Down-stair walking ($h = -0.1m$, $w = 0.5m$) (b) Up-stair walking ($h = 0.08m$, $w = 0.5m$)

Figure 5.9: The walking tiles of stair walking.



(a) The trajectories of the hip joint.

(b) The trajectories of the knee joint.

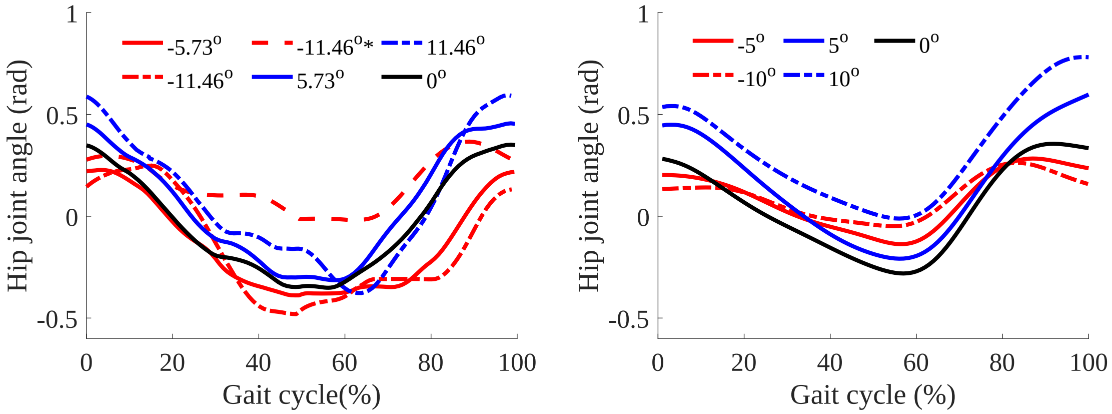
(c) The trajectories of the ankle joint.

Figure 5.10: The angular trajectory comparison between the optimization result of level walking and the human data.

5.6.3 Comparisons to Human Data

In this section we compare the optimization results of level walking and slope walking to human data, where the visualizations of the results with the 3D model of AMBER 3 and the comparisons with the recorded human walking videos are available in [68]. As shown in Fig. 5.10, the generated level walking and the human data are compared. Observed from both the walking tiles (Fig. 5.6) and joint trajectory comparison, we identified two sources of discrepancies. First, similar to the result of optimization through contact, the optimized gait tends to have lower foot clearance than human, which causes large differences of knee and hip angles in the gait progression $60\% - 90\%$. Second, compared to the human walking, the generated gait does not fully stretch the leg, which also causes the difference of knee and hip angle in the gait progression from 90% to 10% of the next step. This difference is larger compared to the result from optimization through contact as shown in Fig. 4.6. Possible reasons for this discrepancy include the effect of impact and the cost function we used. There is study has shown that when removing running shoes, human tends to shift foot-strike pattern from heel-strike to fore-foot strike to reduce the impact [69]. Since we assumed heel-strike is an inelastic collision which will cause the largest impact possible, in the optimization the gait will try to avoid fully stretching the leg before heel-strike so that it will not penalize by the high COT due to the large impact. This can also explain why the leg-stretching from the hybrid optimization result is smaller than optimization through contact, as the later one only considers the impact force integral over the time step and treats it as the regular force, therefore it can mitigate the effect from the inelastic impact dynamics. Those possible causes can also affect the difference in the ankle trajectory comparison, but the trends of optimization result and the human data, especially the later part (50% to 90%), are still look similar.

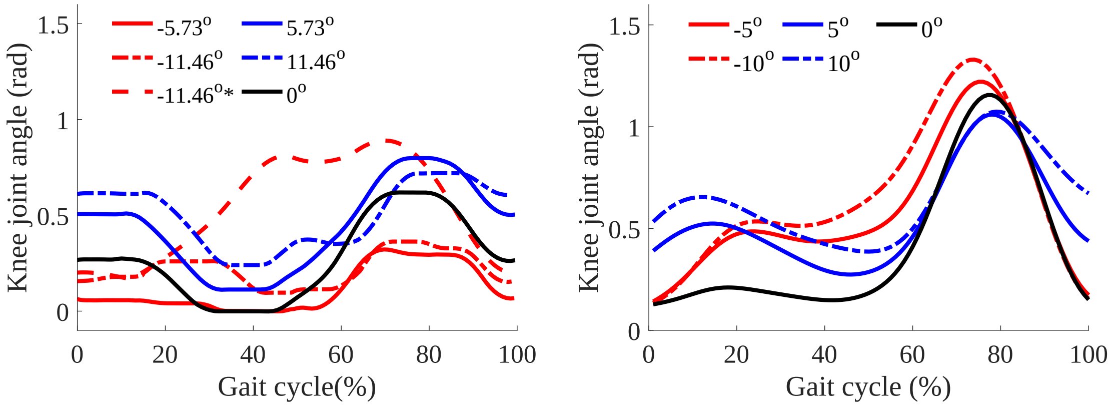
In Figs. 5.11 to 5.13, we compared the generated slope walking results to the human data. In general one major source of discrepancy is that we used almost the same parameter set for level walking and walking on different slopes and , including the range of time step Δt_n and minimum step length, which may not be the case for human gaits. In the hip trajectory comparison as shown in Fig. 5.11, the up-slope walking with slope angles 5.73^o and 11.46^o, and the down-slope walking with slope angle -11.46^o* (the one with smaller torso angle limit $\theta_{torso} \in [-0.15rad, 0.15rad]$) are more similar to the human ones. The discrepancies observed from other trajectories show that humans tend to use smaller hip range for down-slope walking. Comparing down-slope walking gaits 11.46^o and 11.46^o* (the one with smaller limit) in Fig. 5.11, the later looks more similar to human data, which



(a) Optimization results

(b) Human data

Figure 5.11: The hip joint trajectory comparison for walking on slopes. In the legend -11.46^o* indicates the down-slope walking with smaller torso angle range: $\theta_{torso} \in [-0.15rad, 0.15rad]$.

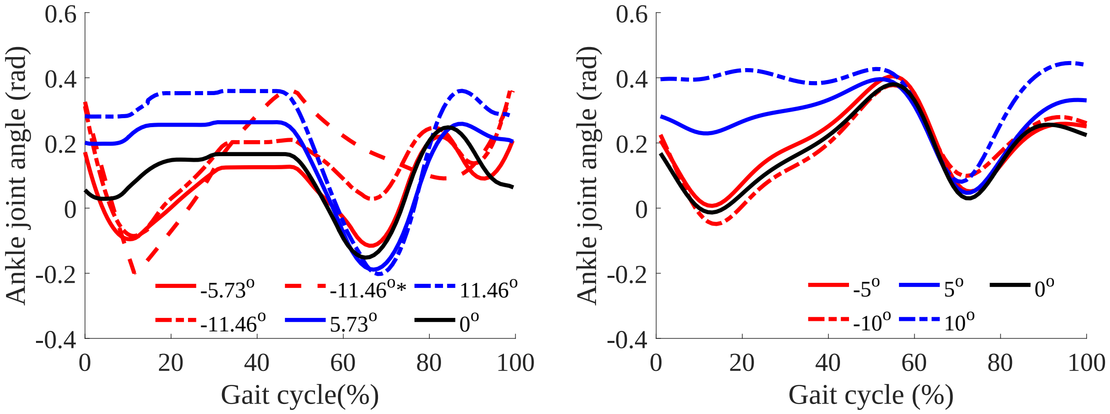


(a) Optimization results

(b) Human data

Figure 5.12: The knee joint trajectory comparison for walking on slopes.

may imply keeping torso upright is more important than having lower COT for down-slope walking. In the knee trajectory comparison (Fig. 5.12), the up-slope walking with slope angles 5.73^o and 11.46^o are more similar to the human ones. For the optimized down-slope walking, the segments near the beginning and the end are more close to the

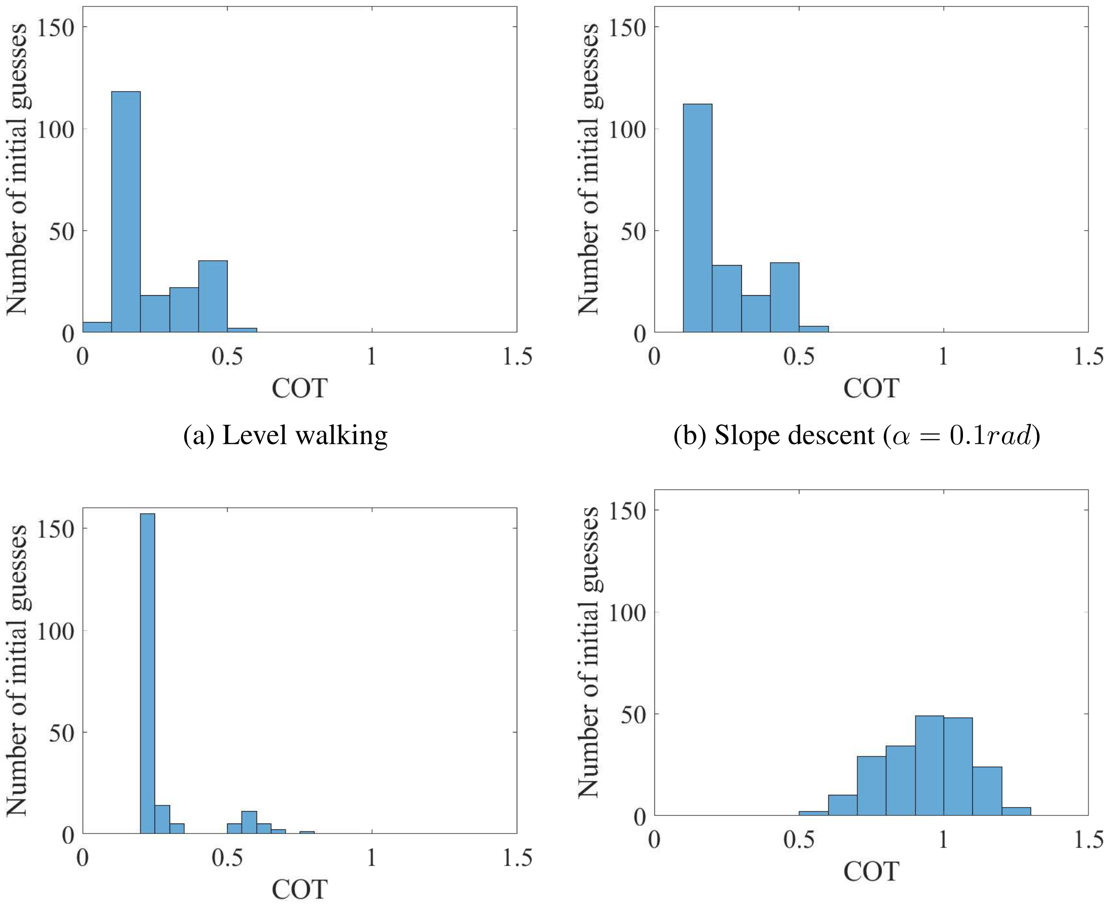


(a) Optimization results

(b) Human data

Figure 5.13: The ankle joint trajectory comparison for walking on slopes.

level walking, which are similar to the human data. In addition, -11.46^o* is slightly more similar to human data compared to other down-slope walking gaits. The cause of the main discrepancy in the duration of gait progression 40% to 60% is mainly because of the low foot clearance as the optimization minimizes the COT. From the human data we also noticed that human tends to have larger foot clearance when the slope angle of ramp decent goes larger, where there is no similar trend for up-slope walking gaits. In the ankle trajectory comparison (Fig. 5.13), the up-slope walking with slope angles 5.73^o and 11.46^o are still more similar to the human ones. For the down-slope walking, different from the comparisons of hip and knee angle, the walking gaits with slope angle -5.73^o and -11.46^o are more similar to human data compared to -11.46^o*, especially in the range of gait progression $0\% - 60\%$. The ankle angles of walking gaits with slope angle -5.73^o and -11.46^o are much larger than human data in the end of the gait cycle. This maybe because the robot requires a larger range of motion to move the foot downward quickly (in the beginning of the gait cycle) to reduce the effect from the heel-strike, as we already discussed that the impact effect of the inelastic collision could be larger than the collision in actual human walking.



(a) Level walking

(b) Slope descent ($\alpha = 0.1rad$)

(c) Slope descent ($\alpha = 0.2rad$)

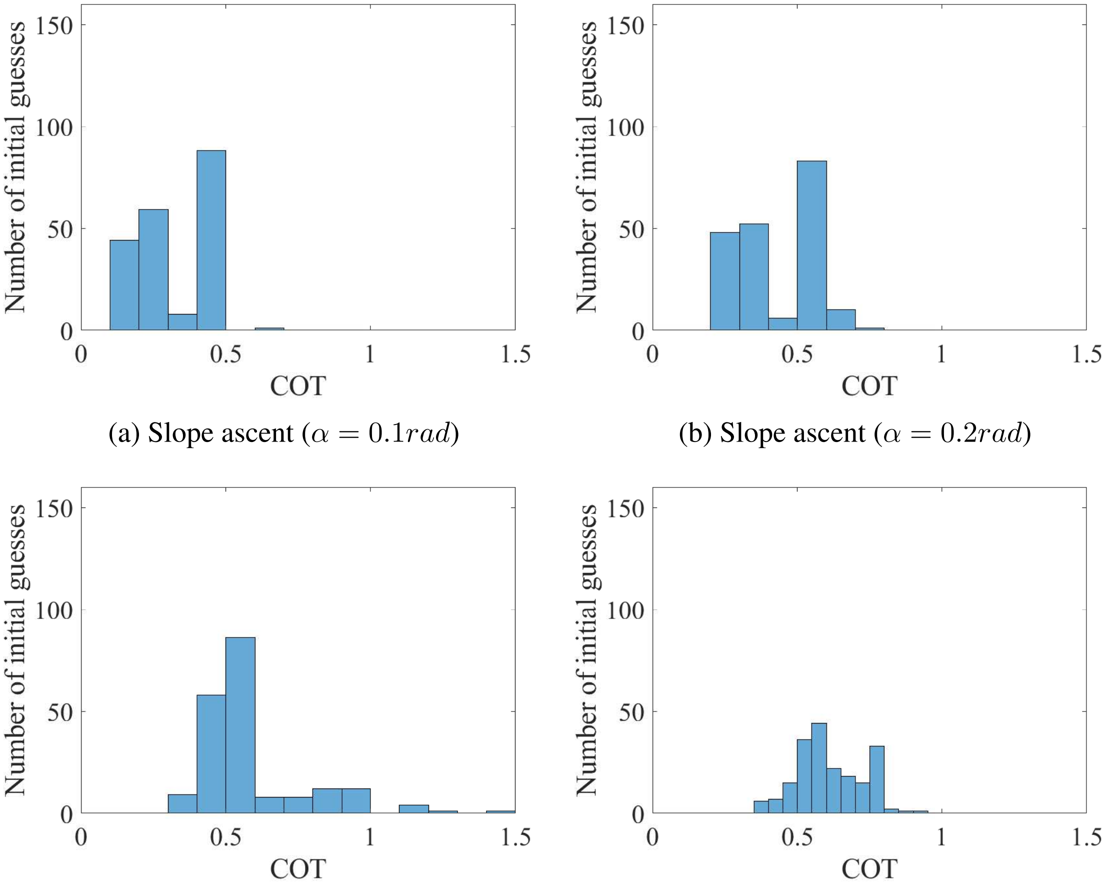
(d) Slope descent ($\alpha = 0.2rad$, $\theta_{torso} = [\pm 0.15rad]$)

Figure 5.14: The histograms of level walking and down-slope walking results.

5.6.4 Optimization Sensitivity to the Initial Guess

In this subsection, we present and discuss the optimization sensitivity to the initial guess. Because the optimization problem we formulated here is nonlinear and non-convex, when solving this nonlinear program with local optimization solver like IPOPT, it is likely the solver will return a local optimal solution rather than a global one. As a result, the initial guess can have significant effect on the optimization result.

Though in [24] it is mentioned that this optimization framework can be used with randomized initial guess, it is still not clear how well the randomized initial guess can be used for different terrains (i.e. different contact constraints) or variable bounds. To get more understanding towards to the optimization sensitivity to initial guess, in Fig. 5.14 and Fig. 5.15 we show the histograms of the optimization results to present the distribution of COT versus number of initial guesses (where we used 200 randomized initial guesses to generate 200 walking gaits for each terrain), which can be good indicators to show the optimization performance as well as the optimization complexity for each terrain. In Fig. 5.14, for the level walking, more than 100 initial guesses result in COTs lower than 0.2, which shows that the HZD gait optimization can work with randomized initial guess quite well. In addition, except Fig. 5.14 (d) we found that when the down-slope angle



(a) Slope ascent ($\alpha = 0.1rad$)

(b) Slope ascent ($\alpha = 0.2rad$)

(c) Stair descent ($h = -0.1m$, $w = 0.5m$)

(d) Stair ascent ($h = 0.08m$, $w = 0.5m$)

Figure 5.15: The histograms of up-slope and stair walking results.

increases, the more gaits are with the lower COT, which seems reasonable because the larger slope angle the gravity can provide more potential energy on different initialized walking gaits, and then a local optimal solution with lower COT might be easier to be solved. Conversely, when the walking gait need to walk against the gravity more (like Fig. 5.15 (a) and (b)) or the motion range is more limited (Fig. 5.15 (d)), the distribution becomes flatter. This indicates it is more difficult to get a better local optimal solution, therefore a better initial guess will be more important for those cases. The similar trend can also be found in stair walking as shown in Fig. 5.15 (c) and (d).

5.7 Conclusions and Future Work

In this work, we presented the modified HZD gait optimization framework to generate walking gaits on various terrains. With the modified contact constraints, this optimization framework can be generally applied to flat-terrain, slope or stair with different profiles – as long as the proper predefined contact sequence is given. We showed the optimization results on different terrains also compared those gaits. We found that the COT will increase when the step height either goes lower and higher (as the system needs to provide more work to decelerate the body, or lift it up to a certain height), and we also found that the torso angle limit is an main parameter to determine whether the down-slope walking is cautious or not. We compared the generated level walking and slope walking gaits to the human data, and discussed the similar trends, and possible reasons for the discrepancies. We also presented the histograms of optimization results with randomized initial guess, which indicate the performance and the complexities of optimization formulation for different terrain profiles and/or parameters. In the future work we plan to use the gaits generated from this optimization framework to build the gait library to synthesize more complex walking behaviors. We also plan to use this optimization framework with the human model and the lower-limb prosthesis to generate walking gaits for amputees.

6. CAPTURE POINT-BASED ANALYSIS ON STANDING, WALKING, AND WALKING WITH SLIPPING

6.1 Introduction

This project is for studying balance recovery which cannot be handled by low level control and trajectory optimization. When a disturbance is too large to recovery from a optimized reference for a bipedal system, fast reactions based on simple model-based methods are required. By the means of **stepping strategy**, human walking is even more resilient and can recovery the balance from unexpected disturbances (like a push, an uneven terrain, or a slippery floor). Although human stepping strategy had been identified as one of the main balance strategies in the literature of biomechanics [43], its underlying principle is still relatively not clear. In this work, to get more understanding about the human stepping strategy, we use the Capture Point(CP)-based methods originated from bipedal robotics to analyze human stepping behaviors. As a preliminary study, its ultimate goal is to apply the CP-based methods on the applications of controller design for human rehabilitation devices (e.g. sensory augmentation device [70]) to provide the additional sensory cue about the size of a proper step to the user, or directly control lower-limb wearable robots (e.g. prosthesis or exoskeleton) so that the user can perform proper stepping.

To interpret human balance strategies from standing/walking balance to stepping strategy, we can use model-based methods with inverted pendulum model. Among the different models, linear inverted pendulum model (LIPM) is one of the simplest while still capable of representing balance strategies with the connection between several important quantities in the literature of robotics and biomechanics: center of mass (COM), center of pressure (COP, or zero-moment point (ZMP) in a generalized sense) [13], and Instantaneous Capture Point (ICP) [19]. Depending on different concepts to achieve the balance, there are two different ways can explain the balance strategies: i) Balance margin and ii) Capture point.

6.1.1 Balance Margin

One way to maintain balance is starting by assuming the full control of the LIPM is available (therefore the system is **full actuated**, or equivalently, the system is with the foot having the flat-contact to the ground). As long as the full control is available, then it is possible for the system to exert torque to maintain balance. Therefore, the balance margin, as its name suggests, is the margin before the system loses the full-control (i.e. the flat-contact breaks). Originating from the criterion of standing (static) balance, different criteria can be derived for walking balance or stepping strategy. Except the flat-contact assumption, other important assumptions include:

- The ground reaction force (GRF), including the frictional force along the ground surface and the normal force, is large enough for the system to maintain the balance. It is also implied that no slip occurs.
- Although the ground reaction force at a contact point in general only contains translational forces, however, through the force distribution along the contact surface area (or the **support polygon**), it is still possible to exert torque to a contact point within the support polygon. Therefore the generalized GRF (e.g. $[f_x, f_y, \tau_z]$ in 2D case) is used through out this chapter.

Static balance: Used as a quasi-static approach, static balance assuming the COM movement is so slow that its dynamics appears to be static, and the force balance equation below must be hold (as shown in Fig. 6.1 (left)):

$$-mg + f_y = 0 \tag{6.1}$$

where m is the total mass of the system and in GRF $f_x = \tau_z = 0$. The static balance criterion, by assuming the support polygon (or the base of support, BOS) is $\in [x_{heel}, x_{toe}]$, then can be expressed as shown:

$$x_{heel} \leq x_c \leq x_{toe}$$

where the COM position is $[x_c, z_c]$ in the 2D case. Note the GRF needs to be exerted within the support polygon as a contact force. When x_c is outside the support polygon, the force outside the support polygon is equivalent to a GRF inside the support polygon with a moment, which will cause the system tipping and losing the balance because there is no other force/torque can balance out this induced moment. Since static balance criterion only requires the information of the COM position, it can be easily used. However, the quasi-static assumption greatly limits its application to more dynamic behaviors, such as walking or push-recovery by stepping.

Dynamic balance: Dynamic balance is extended from the static balance by considering the horizontal dynamics of COM, where the z_c is assumed to be a constant for the linear inverted pendulum model (LIPM). To achieve dynamic balance, i.e. keep the system not tipping over with the horizontal dynamics as shown in Fig. 6.1 (center), except Eq. (6.1), the torque balance also needs to be considered (with an introduced reference point $[x_{zmp}, 0]$ for the GRF where its τ_z is vanished):

$$-mg(x_c) + z_c m\ddot{x}_c + f_z x_{zmp} = 0 \tag{6.2}$$

$$\Rightarrow \ddot{x}_c = \frac{g}{z_c}(x_c - x_{zmp}), \text{ or} \tag{6.3}$$

$$\Rightarrow x_{zmp} = x_c - \frac{z_c}{g}\ddot{x}_c \tag{6.4}$$

The GRF reference point is referred as Zero-Moment Point (ZMP) [12], (basically equiv-

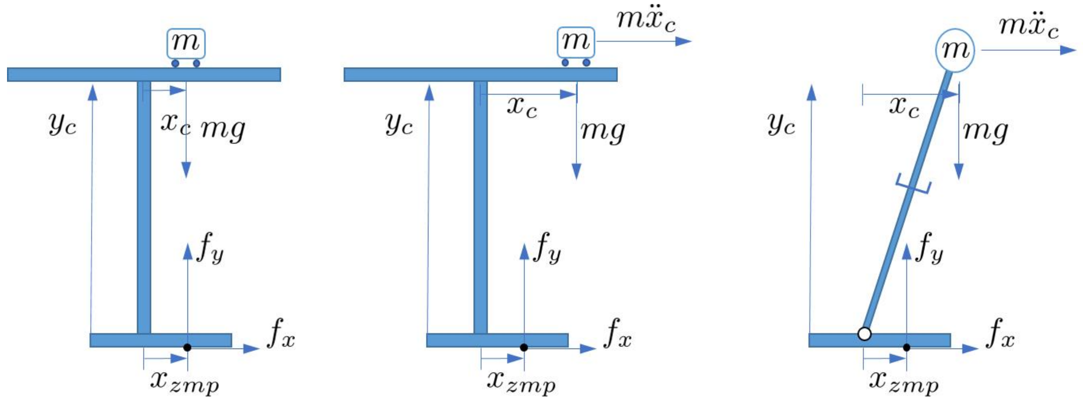


Figure 6.1: The famous cart-table model to describe the LIPM for static balance (left), dynamic balance (center), and the schematic of the linear inverted pendulum model (right).

alent to Center of Pressure (COP)). The system then is balanced as long as the Eqs. (6.1) and (6.3) hold. Following the concept for establishing the static balance criterion, the dynamic balance criterion can then be expressed as:

$$x_{heel} \leq x_{zmp} \leq x_{toe} \tag{6.5}$$

Note the above equation implies that x_c can be outside of the support polygon as long as the Eq. (6.5) holds, therefore it is more general and more suitable to be used for walking compared to the static balance criterion.

Extrapolated Center of Mass: In Eq. (6.3) if x_{zmp} is assumed to be a constant, then Eq. (6.3) becomes an ordinary differential equation (ODE) of x_c. Its analytical solution can be expressed as:

$$x_c(t) = x_{zmp} + (x_{c0} - x_{zmp})cosh(\omega t) + \frac{\dot{x}_{c0}}{\omega}sinh(\omega t) \tag{6.6}$$

where x_{c0} and $\dot{x}_{c0}$ is the initial condition of x_c, ω is $\sqrt{g/z_c}$. An additional assumption is made when considering the case that $\ddot{x}_c$ is non-positive (e.g. as a restoring force for recovering balance). Then by Eqs. (6.4) and (6.6):

$$\begin{aligned} \ddot{x}_c &\leq 0 \\ \Rightarrow x_c(t) - x_{zmp} &\leq 0 \\ (x_{c0} - x_{zmp})cosh(\omega t) + \frac{\dot{x}_{c0}}{\omega}sinh(\omega t) &\leq 0 \\ \Rightarrow x_{c0} + \frac{\dot{x}_{c0}}{\omega}tanh(\omega t) &\leq x_{zmp} \end{aligned} \tag{6.7}$$

As $-1 \leq tanh(\omega t) \leq 1$ for any t, therefore Eq. (6.7) becomes:

$$x_{c0} + \frac{\dot{x}_{c0}}{\omega} \leq x_{zmp} \tag{6.8}$$

where the left hand side is termed extrapolated center of mass (XCoM) [71, 72, 73]. There are two ways can interpret Eq. (6.8): i) The system's balance can be maintained if x_{zmp} is moving forward fast enough. ii) If XCoM exceeds the maximum x_{zmp} (e.g. x_{toe}), then the system becomes unbalanced. With those interpretations and Eq. (6.8), the distance between x_{zmp} and XCoM is used as a balance metric for studying human balance of standing, walking, and step recovery [71, 72, 73].

On the other hand, by considering $\ddot{x}_c$ is non-negative (as the restoring force when moving backward), then $x_{c0}+\frac{\dot{x}_{c0}}{\omega} \geq x_{zmp}$ can be derived. With the Eq. (6.8), an alternative criterion [14] of dynamic balance can also be derived as:

$$x_{heel} \leq x_{c0} + \frac{\dot{x}_{c0}}{\omega} \leq x_{toe} \tag{6.9}$$

where x_{heel} and x_{toe} are the boundary values of x_{zmp} as mentioned in Eq. (6.5).

6.2 Capture Point

Another way to keep balance is to make sure the system can step on a location which makes the system come to a *captured* state (a state at a trajectory which will reach the static equilibrium, i.e. a complete stop), where the stepping location is called a **capture point** (CP) [18]. Unlike the balance margin method, capture point does not require the system to be full-actuated, and can be derived and extended based on the LIPM with point feet. The capture point (CP)-based methods turns out to be useful for challenging balance tasks such as push-recovery or walking with line-contact for humanoid and bipedal robots [18, 20, 21]. Here we briefly introduced the variations of different types of capture points (which are proposed in [19]) we use in this study.

Instantaneous capture point of LIPM with point feet: With an arbitrary initial condition, if the system can place the next step instantaneously on a point which makes the system come to a completed stop, then this point is called an **instantaneous capture point (ICP)**. When the stepping actually happens, then the ICP becomes a capture point. For the LIPM with point feet, the ZMP location (x_{zmp} in Eq. (6.3)) coincides with the only contact point – its ankle position x_{ankle}:

$$\ddot{x}_c = \frac{g}{z_c}(x_c - x_{ankle}) \tag{6.10}$$

For this system the **orbit energy** E_{LIP} [74, 75] can be used to derive ICP as shown:

$$E_{LIP} = \frac{1}{2}(\dot{x_c}^2) - \frac{1}{2}\omega^2(x_c - x_{ankle})^2 \tag{6.11}$$

where $\omega = \sqrt{g/z_c}$. Consider the case the legged system will come to a rest and thus E_{LIP}

is zero. Assuming the ICP is in front of the current stance, then its location can be derived:

$$x_{ICP} = x_c + \frac{\dot{x}_c}{\omega} \tag{6.12}$$

Note that the equation of ICP is the same as the XCoM. Take the time derivative of Eq. (6.12) and substitute $\ddot{x}_c$ from the Eq. (6.10), then we can derive the first-order ODE of ICP and its solution (by assuming x_{ankle} is a constant):

$$\dot{x}_{ICP} = \omega x_{ICP} - \omega x_{ankle} \tag{6.13}$$

$$x_{ICP}(t_s) = e^{\omega t_s}(x_{ICP}(0) - x_{ankle}) + x_{ankle} \tag{6.14}$$

where $x_{ICP}(0)$ is the initial condition of ICP and t_s is the step time. This analytical expression of ICP in Eq. (6.14) can be used to estimate a stepping location if step time t_s is given.

N-step capture point: Extended from the concept of capture point, N-step capture point means the point that can make the system come to a stop after N steps. One step on this point cannot immediately bring the system to a complete stop, but it will bring the system into the region that requires fewer steps to achieve that (e.g. N-1 steps, N-2 steps, ..., 2 steps, 1 step). In the capture point-based analysis in the later section, the $\infty-$step is used to estimate the stepping location for walking and walking with slip, which can be calculated using the following equation:

$$x_{\infty step-ICP} = x_{ICP} - d_{\infty} \tag{6.15}$$

$$d_{\infty} = l_{max}\frac{e^{-\omega t_s}}{1 - e^{-\omega t_s}} \tag{6.16}$$

In Eq. (6.16), l_{max} indicates the maximum reachable distance and t_s is the step time. Please refer to [19] for the detailed derivation of d_∞.

ICP of LIPM with finite-sized foot: Considering that the legged system has finite-sized feet and makes a complete surface contact with the ground, without making the foot rotate with respect to the ground the ankle joint can exert limited torque to the system. In this case, modified from Eq. (6.2) with the new added ankle torque τ_y, the equation of motion can be expressed as:

$$\ddot{x}_c = \omega^2(x_c - x_{ankle}) - \frac{\tau_y}{mg} \tag{6.17}$$

$$= \omega^2(x_c - x_{CoP}) \tag{6.18}$$

where m is the total mass, g is the gravity acceleration, $x_{CoP} = x_{ankle} + \frac{\tau_y}{mg\omega^2}$, where the CoP indicates the center of pressure (COP). Similarly, the corresponding ODE of ICP and its solution can be expressed by simply substituting the x_{CoP} for x_{ankle} in Eq. (6.13):

$$\dot{x}_{ICP} = \omega x_{ICP} - \omega x_{CoP} \tag{6.19}$$

$$x_{ICP}(t_s) = e^{\omega t_s}(x_{ICP}(0) - x_{CoP}) + x_{CoP} \tag{6.20}$$

where the x_{CoP} is assumed to be a constant. Note for the LIPM with finite-sized foot, the set which contains 1-step CP is no longer a single point. Since any x_{CoP} inside the base of support can reach a valid 1-step CP according to Eq. (6.20), the set becomes a region which can be used as a metric for stability measurement which is called *1-step capturability*. This concept can also be extended to N-step capturability [19], where $N \in [0, \infty]$ including the cases of standing and nominal walking gait.

6.2.1 Human Balance Strategies and ICP

In biomechanics of human movement, there are three basic strategies to maintain the balance [43]: i) Ankle strategy, ii) Hip strategy, and iii) Stepping strategy. Similarly, for bipedal/humanoid robots modeled as a LIPM, Stephens [44] and Koolen et al. [19] also proposed the corresponding balance mechanisms based on balance margin and capture point respectively, which are summarized in the following table:

Table 6.1: Comparisons of the balance mechanism between the balance margin and capture point for human balance strategies.

Balance strategy	Balance margin	Capture point
Ankle	COP control with ICP constraint	ICP with fixed COP
Hip	CMP control	ICP with fixed CMP
Stepping	ICP or Hofmann stepping	ICP

where CMP is **centroidal moment pivot**. Extending the LIPM by attaching a flywheel as a reaction mass on the COM, the basic idea of CMP is to consider the torque balance at the COM. The torque generated by the flywheel (driven by hip joints) as an additional term in the torque balance equation Eq. (6.3), can also assist to enlarge the balance margin [44]. However, in this project we will not consider the effect of the angular momentum about the COM for the following reasons:

- The torque from the motion of reaction mass is hard to regenerate.
- The hip torque and range of motion during standing or walking can be limited.
- The stepping strategy can be used to deal with larger disturbance that ankle and hip strategy cannot handle.

From Table 6.1 it is also shown that ICP is a quantity can generalize the human balance strategies using simple LIPM, and it can be used to interpret stepping strategy from both the balance margin or capture point concepts. As a result, it inspires our work of capture point-based analysis on different human behaviors.

6.3 Method

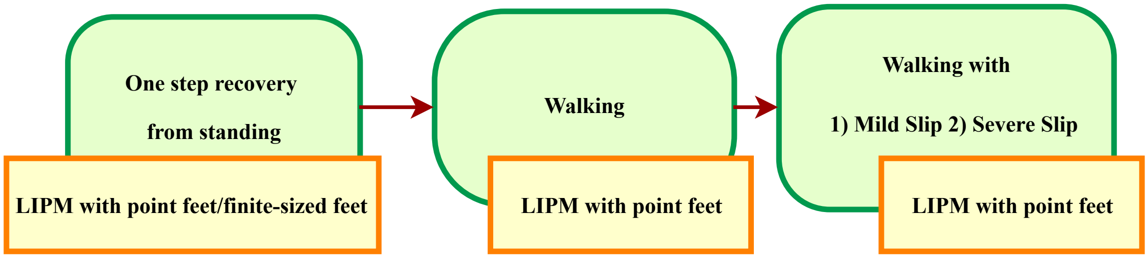


Figure 6.2: The analyzed tasks listed in the order of task complexity and the corresponding LIPM models for the CP-based step estimation.

To use ICP to study human balance with the focus on stepping strategy, we perform CP-based analyses for humans, i.e. using ICP for human step estimations on different tasks, including single step (balance) recovery from standing, walking, and walking with slip. As introduced in the previous section, different LIPM models are chosen for different tasks, which are shown in Fig. 6.2. Among those tasks, single step recovery is categorized as the 'stationary' task, as it is the step recovery from the perturbed standing posture. On the other hand, both walking and walking with slip are categorized as the 'non-stationary' tasks. Both the methods of the stationary tasks and the non-stationary tasks for the CP-based analyses will be introduced in the following subsections.

6.3.1 Step Estimation – Stationary Tasks

As the first step to study human stepping strategy, we perform the analysis for two stationary tasks. The analysis of stationary tasks are twofold: first is to validate CP-based analysis (by estimating the 1-step CP) for single-step recovery and compare it with the

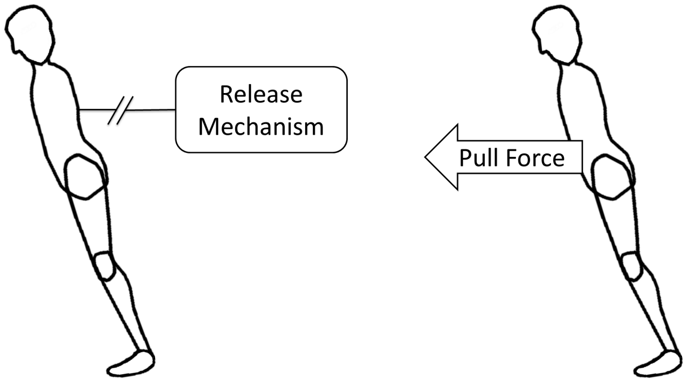


Figure 6.3: The schematics of stationary tasks: Single-step recovery from the forward lean (left) and single-step recovery from the combination of forward lean and pull force (right).

human experimental data [4, 6]. Second, we also compare the estimated 1-step CP to the simulation results for the same two tasks through the nonlinear optimization where the Model Predictive Control (MPC) with finite horizon was adopted [5]. Those stationary tasks (as shown in Fig. 6.3) are briefly described as follows:

Task (1) Subjects were instructed to perform a single-step recovery to regain the balance from the forward lean for 4 different inclination levels [4]. The corresponding step length, lean angle, reaction time and step time were all recorded. Note in [4] only the data from 10 young subjects is used for avoiding any age-related decay of recovery ability. For the 1-step CP estimation, since the COP is at the edge of the base of support (BOS) as the whole body is forward inclined and the support mainly relies on the tether and electromagnet which are connected to the chest harness, we assume that there is no ankle torque applied to the LIPM (i.e. Eq. (6.14) is used).

Task (2) Subjects were instructed to make a single-step to recover the balance as soon as one cannot sustain the standing posture against the two combinations of forward inclination and forward pull [6]. The corresponding step lengths, maximum lean angle, maximum pull force, reaction time and step time were all recorded. Note in

[6] only the data from the leaning trials is used. All the subjects from the leaning trails were asked to sustain the forward pull force which was gradually increased. As a result, for the 1-step CP estimation, we assume the ankle torque contributes to the force against the maximum pull force (i.e. Eq. (6.20) is used) before a step is made.

6.3.2 Step Estimation – Non-stationary Tasks of Humans and Robotic Walkers

Non-stationary tasks include walking and walking with slip where the CP-based quantities are used to estimate the step location for different data. For the human walking without and with slip, we used human experimental data from 20 healthy young adults, which were approved by institutional review board [76]. Subjects were informed that the surface would not be slippery for the first few trials. However, after two or three 'dry walk' trials, the surface was contaminated without notification to the subjects. Both marker data and force data were recorded, and we took one set of walking trial and walking with slip trial for every subject. For the walking with slip trails, subjects with Peak Heel Velocity (PHV) greater than $1.44m/s$ were categorized as the 'severe slippers' and the rest were 'mild slippers' [34]. In this study there were 12 mild slippers and 8 severe slippers. For details, please refer to [76].

On the other hand, we also use the CP-based quantities to estimate the stepping location of the robotic walking from robot simulations using different robot models and walking control methods. Since the system dynamics and control for those walking robots are well understood, they can help to validate CP-based step estimation's efficacy, and provide information for the discrepancy between the estimated and actual stepping locations. To make sure the robot models and control methods used in the analysis cover certain diversity in the literature of bipedal robotics, the following bipedal-robots are selected and used in the analysis:

- A passive compass gait (CG) robot which is composed of two linkages and point feet, can walk on down-slope periodically without any actuation.

- A 5-link active kneed-gait bipedal robot (the robot comprises torso, thighs and shanks with point feet, where the legs are controlled by the actuated knee and hip joints) walks on the level ground as an under-actuated system (KGUA) (i.e. without ankle actuation) or a full-actuated system (KGFA) (i.e. with ankle actuation). Human-inspired control with Partial Hybrid Zero Dynamics (PHZD) is adopted for both cases [77].

- A 7-link active bipedal robot (ZMP) (robot with torso, thighs, shanks and foot pads) walks on the level ground as a full-actuated system using ZMP-based walking motion generation and control [1].

Among the introduced CP-based quantities, $1-$step CP and $\infty-$step CP are used for the step estimation. In addition, to reduce the number of predetermined variables such as step time and maximum step length, we proposed a new quantity called Estimated Instantaneous Capture Point (EICP), which is defined as:

$$x_{EICP} = x_c + \frac{\dot{x}_{c,Averaged}}{\omega} \tag{6.21}$$

where $x_{c,Averaged}$ is the average walking speed. Because the non-stationary tasks are much more complicated than the stationary tasks, only the CP-based quantities (1-step CP, $\infty-$step CP and EICP) right before the heel-strike were calculated. By comparing the errors between the 1-step CP (Eq. (6.14)), $\infty-$step CP (Eq. (6.15)), EICP (Eq. (6.21)) with respect to the actual foot placements, the main objective of this analysis is to understand how well those CP-based quantities can estimate the stepping location for the walking tasks (for both humans and robots) and the tasks of walking with slip (humans only).

6.4 Result and Discussion

6.4.1 Step Estimation – Stationary Tasks

The step location estimations for the stationary tasks are shown in Figs. 6.4 and 6.5. In many cases, the 1-step CP estimation results are similar to the simulation results using MPC without considering the upper body inertia, and this also indicates the efficacy of 1-step CP estimation: given an initial condition of a simplified model and a predetermined step time, the step location then can be estimated without running any simulations using advanced control methods.

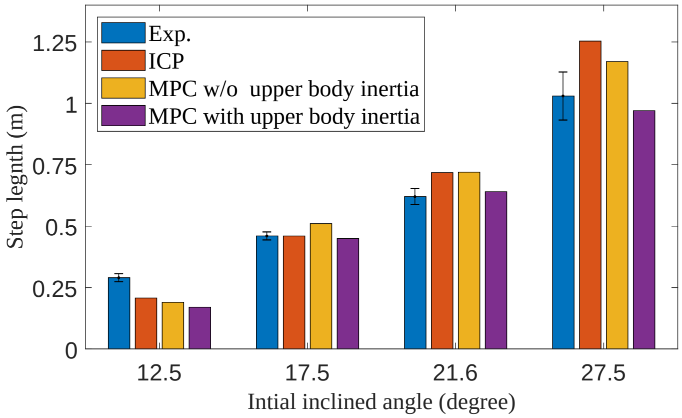


Figure 6.4: Step location comparison between ICP and results in [4, 5] for Task (1).

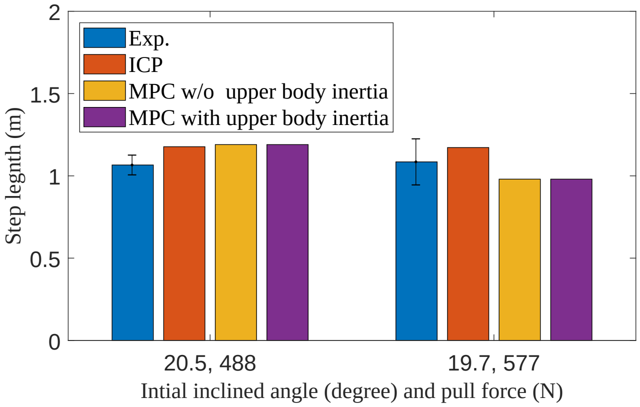


Figure 6.5: Step location comparison between ICP and results in [6, 5] for Task (2).

According to Figs. 6.4 and 6.5, the results of 1-step CP estimates the step location better with the inclination angles of 17.5 degree (Fig. 6.4), 19.7 degree (Fig. 6.5) and 20.5 degree (Fig. 6.5). Comparing the 1-step CP to the human data and MPC simulation with hip strategy (considering the upper body inertia), the largest error in Fig. 6.4 is the one with the largest inclined angle (27.5 degree) – where the estimated step length is larger than the actual ones. The fact that the adopted LIPM does not consider the effect from upper body motion could be the main reason for this discrepancy. On the other hand, the estimated step length in Fig. 6.4 is smaller than the actual ones for the inclination angle 12.5 degree. We postulate that although the inclination angle is small, subjects may still changed their strategies to the more conservative (safer) ones due to the psychological reaction based on the fear of fall.

6.4.2 Step Estimation – Non-stationary Tasks

For the non-stationary tasks, the main specifications of all walkers are listed in Table 6.2. For the actuation type, UA refers to under-actuated and FA refers to fully-actuated. For the terrain type, DS indicates down slope and LG indicates level ground. For the robotic walkers, the CG and KGUA have point feet, KGFA and ZMP have foot pads and both perform flat-feet walking (i.e. the foot always keeps horizontal during walking.)

Table 6.2: Parameters of the walkers (values in parentheses indicate the standard deviation).

	CG	KGUA	KGFA	ZMP	Human
Height (m)	0.90	1.0	1.0	1.5	1.73 (0.08)
Weight (Kg)	50	70	70	28.18	69.05 (12.02)
Speed (m/s)	0.6	1.36	1.18	0.07	1.39 (0.23)
Actuation type	Passive	UA	FA	FA	Mixed
Terrain type	4° DS	LG	LG	LG	LG

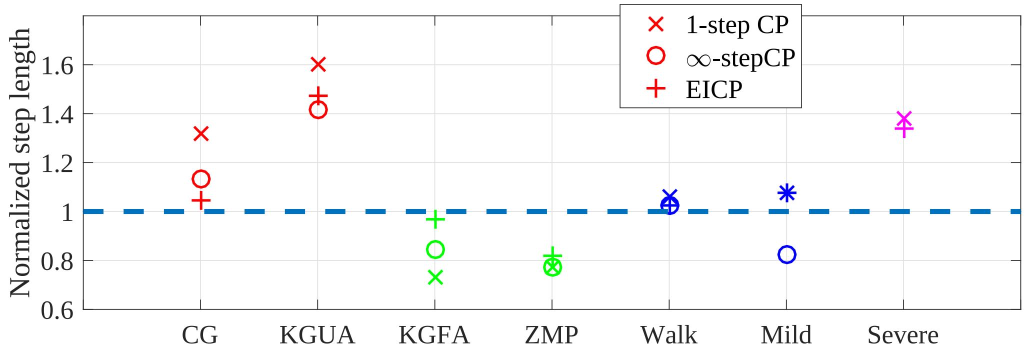


Figure 6.6: Estimation error of step location (normalized by step length) for different walkers and difference tasks.

Table 6.3: Estimation error of step location (normalized by step length) for different robotic walkers. The values in the parentheses indicate the standard deviation.

	CG	KGUA	KGFA	ZMP
1-step CP	0.327 (0)	0.603 (0)	-0.269 (0)	-0.225 (0)
$\infty-$step CP	0.133 (0)	0.417 (0)	-0.154 (0)	-0.225 (0)
EICP	0.048 (0)	0.473 (0)	-0.031 (0)	-0.182 (0)

Fig. 6.6 presents CP-based estimated step lengths that are normalized by step lengths for the robotic locomotion, human walking, and human walking with slip. The statistical results of estimation errors are listed in Tables 6.3 and 6.4. Fig. 6.7 to Fig. 6.9 show the normalized trajectories with respect to the normalized time (from step initiation to heel-strike). For comparison, all the position trajectories were normalized by the step length, and all the velocity trajectories were normalized by ω times step length which is similar to the dimensional analysis introduced in [19]. The reasons for the discrepancies in the

Table 6.4: Estimation error of step location (normalized by step length) for human walking, walking with mild slip, and walking with severe slip. The values in the parentheses indicate the standard deviation.

	Human walking	Walking with mild slip	Walking with severe slip
1-step CP	0.06 (0.033)	0.074 (0.038)	0.379 (0.102)
$\infty-$step CP	0.024 (0.028)	-0.174 (0.066)	> 1.00
EICP	0.023 (0.029)	0.078 (0.041)	0.341 (0.057)

estimation results (Fig. 6.6) are more diverse than the stationary tasks, so we categorize them into 4 different groups: un/under-actuated walkers, full-actuated walkers, human walking and walking with mild slip, and human walking with sever slip (please refer to 4 different color codes in Fig. 6.6).

Un/under-actuated walkers. In both the CG and KGUA cases (Fig. 6.6 and Table 6.3), $\infty-$step CP and EICP estimate the step location (slightly) better than 1-step CP. For the robots without ankle-actuation, the estimated CP-based step lengths are longer than the simulated ones. The COM velocity of CG increases (Fig. 6.7) before stepping, which

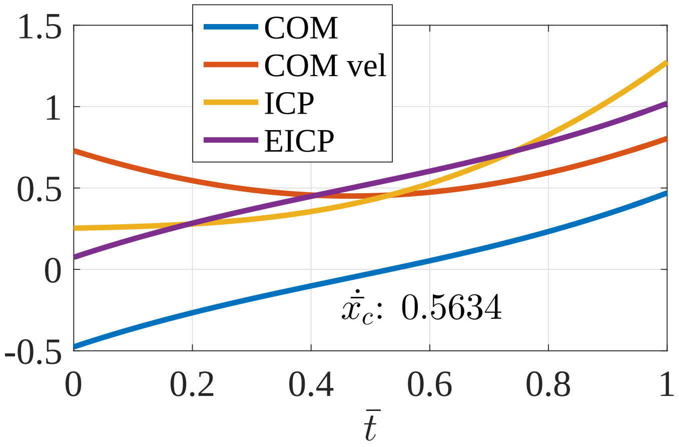


Figure 6.7: Normalized trajectories of COM, COM velocity, ICP, and EICP of the compass gait robot (CG) with respect to the normalized time , before the step is made (i.e. at $\bar{t} = 1$.)

moves like a free-swaying inverted pendulum.

On the other hand, the lack of considering the impact effect in calculating CPs could be the cause of those errors: in the CP derivation, if the energy dissipation induced by the foot strike is considered, the orbit energy in Eq. (6.11) should be smaller, which will result in a CP with a smaller step length. Th step estimation of KGUA in Table 6.4 is also over-estimated due to the same reason.

Fully-actuated walkers. For the fully-actuated robots (Fig. 6.6 and Table 6.3), the estimated step lengths are all shorter than the simulation ones. The main reason maybe that instead of performing free-swaying motion as a LIPM with point foot, the ankle joint torque dominantly contributes to the COM velocity in the flat-feet walking. The COM velocity of KGFA as shown in Fig. 6.8 decreases before stepping, which indicates that the ankle joint applies torque against the moving direction to modulate the COM velocity. Estimating the step location using EICP is better than $\infty-$step CP in both KGFA and ZMP cases and better than 1-step CP in the KGFA case. For the ZMP case, because the robot walks at a really slow speed $(0.2m/s)$, all CP-based estimation results are similar.

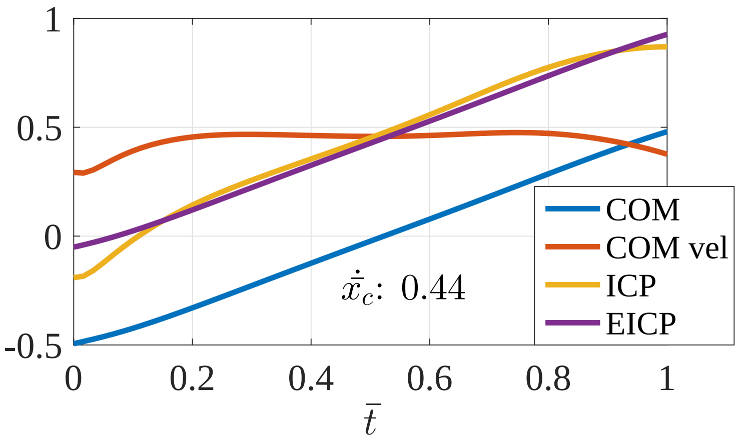


Figure 6.8: Normalized trajectories of COM, COM velocity, ICP, and EICP of the kneed-gait robot with actuated ankles (KGFA) with respect to the normalized time , before the step is made (i.e. at $\bar{t} = 1$.)

Human walking and walking with mild slip. The results shown in Figs. 6.6 and 6.9 and Table 6.4 indicate that both 1-step CP and EICP can estimate step length of human walking and walking with mild slip quite well. There are several possible reasons. First, human tends to utilize the passive dynamics while walking. As a result, the COM motion acts more like a free-swaying inverted pendulum than the full-actuated robots (the COM velocity in Fig. 6.9 is convex which is similar to the one in Fig. 6.7). Second, compared to un/under-actuated robots (Fig. 6.7), the human ankle joint helps to modulate the COM velocity, therefore the velocity profile (Fig. 6.9) in the beginning of a step $(0.0 - 0.2\ \bar{t})$ is more similar to the full-actuated robot as shown in Fig. 6.8. Third, unlike un/under-actuated robots, human tends to use foot-rolling motion to reduce the impact force and has a smoother load transition (e.g. the duration of the double-support phase is not infinitesimal). As a result, though the COM velocity in Fig. 6.9 still increases before stepping, the foot-rolling motion helps to reduce the impact, which make the system closer to the LIPM described in the CP-based methods.

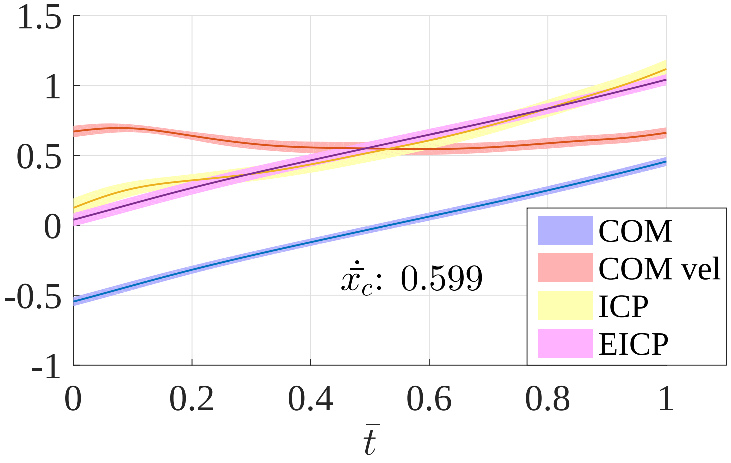


Figure 6.9: Normalized trajectories of COM, COM velocity, ICP, and EICP of human walking, before the step is made (i.e. at $\bar{t} = 1$.) The shaded areas indicate the regions within a standard deviation.

Compared to 1-step CP and EICP, $\infty-$step CP has larger discrepancy on estimating

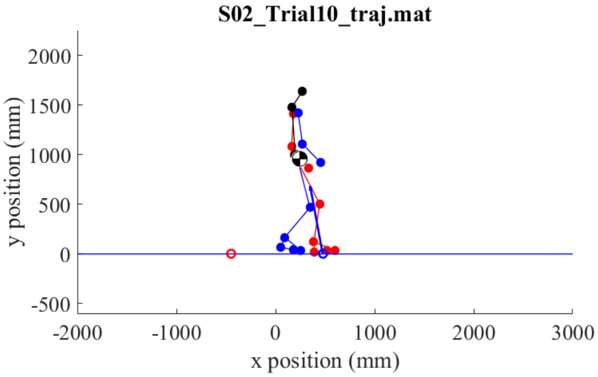


Figure 6.10: The snapshot of human walking with severe slip occurred at the leading leg (red) where the recovery step of the trailing leg (blue) was made behind the leading leg.

human walking with mild slip. A possible reason is that the normal step length is used as l_{max} in Eq. (6.16) for the $\infty-$step CP calculation due to the difficulty to determine the real maximum reachable range of humans with different kinematic configurations and contact conditions.

Human walking with severe slip. For estimating the foot placement for human walking with severe slip (Fig. 6.6 and Table 6.4), none of the CP-based quantities estimated the foot placement well. Fig. 6.10 shows a snapshot of the recovery step of a subject while the severe slip occurred, which was the general stepping reaction among subjects. Instead of making a step in front of the leading leg as the CP-based quantities suggested, the trailing leg (the leg colored blue), in most of the cases stepped on a nearby location behind the leading leg to reform the double support until the PHV decreased to an acceptable and safer value. We also found that the value of $\infty-$step CP calculated from Eq. (6.16) varied a lot and is not practical to be used for estimating foot placement in human walking with severe slip. The reason is that since in most of the cases the step time for severe slip is really small, which makes the $e^{-\omega t_s}$ close to 1 and thus the d_∞ can easily diverge to an infeasible value.

6.5 Conclusion and Future Work

In this CP-based analysis, we checked the efficacy of CP-based step estimation for different human tasks, including single step recovery, walking and walking with slip in different severities. It was inferred that the CP-based step estimation could be a useful tool for step estimation with certain modifications. Among the non-stationary and stationary tasks, the following factors should be considered for the modified CP-based foot placement estimation, which will be the future work: i) the effect of upper body motion, ii) the velocity change due to impact, and iii) the weak estimation for human walking with severe slip. In conclusion, the stepping location estimations using 1-step CP, $\infty-$step CP and EICP could provide reasonable estimations for human single step recovery, walking, and walking with mild slip. However, for walking with severe slip, CP-based method needs to be improved further for better stepping location estimations.

7. CONCLUSIONS

7.1 Summary

In this dissertation we present a series of works built upon different walking control methods and algorithms – to improve bipedal walking in the three important perspectives of human gait: predictive control, gait optimization towards robust and human-like motion, and step estimation of human locomotion tasks based on stepping strategy. The contributions are summarized in Section 1.5.

To improve the controller integration with model predictive control (MPC) and mitigate the model inconsistency between the simplified and full dynamic models, we designed a novel framework to unify middle level control (the MPC for COM planning) and low level control (QP-RES-CLF) which is capable of generating feasible walking gait for bipedal robot AMBER 3. For the trajectory optimization, we studied and developed several algorithms with direct collocation framework to improve the gait robustness and locomotion with multi-domain towards to human-like walking. A new robust trajectory optimization framework using step-time sampling was designed for simple walker under terrain uncertainties. Its performance was validated by the simulations of the SLIP running model, and the compass gait with time-varying LQR control. We improved the trajectory optimization through contact with more accurate transcription (Hermite-Simpson method), compared the generated level-walking gaits with different contact conditions and human data, and conducted the preliminary experiment on bipedal robot AMBER 3. We also generalized the contact constraints for HZD gait optimization, which enables the generation of multi-domain walking on flat ground, slopes and stairs. The generated results were compared with the human data where the similar trends and the potential reasons for the discrepancies were presented and discussed. In addition, we also studied the optimization sensitivity to randomized initial guesses for different terrains, which indicates the performance of the optimization framework as well as the optimization difficulties of different walking tasks. To get more understanding how humans choose their step location, we used CP-based step estimation, which suggests capture point can be a promising method to estimate step location for step-recovery from standing, walking and walking with mild slip (peak heel velocity $< 1.44m/s$).

7.2 Future Work

Towards to the ultimate goal, in order to contribute the control methods in bipedal robotics to the lower-limb wearable robots and rehabilitation devices, the current developed control methods and algorithms in trajectory optimization and stepping strategy open up several future directions for validations and extensions, which are listed below:

7.2.1 Trajectory Optimization

Experiment validations. With the generalized HZD gait optimization, it makes the motion planning of walking with multi-domain more tractable and easier to be adjusted with different parameters and additional constraint sets. One immediate research direction will be the experiment implementations on bipedal robot AMBER 3. In the premise that the RBD model is accurate enough, studying the discrepancies between the optimized results and the experiments will provide possible improving directions of the generalized HZD gait optimization framework.

Moderate modifications of model from RBD. The current optimization results are generated from rigid body dynamics model. The next step is to incorporate important elastic components existed in human body to the dynamic model, and use it to study their effect to the walking gait generated from the optimization as a potential research direction for both bipedal robotics and biomechanics.

Applications for lower-limb wearable robots. The current modified framework of generalized HZD gait optimization also enables the application for wearable robots like lower-limb prosthesis and exoskeleton, which can be achieved by replacing the AMBER 3 model with a model of amputee with a lower-limb prosthesis, or a human model with a lower-limb exoskeleton.

Gait library. With the capability of generating the walking gait for different terrain profiles, the generalized HZD gait optimization can be a potential tool to generate the key trajectories for a *gait library*: A collection of the low-dimension descriptions of optimized walking gaits (trajectories) within the range of specific gait parameters (e.g. the gait library for different slope angles, or level walking with different step lengths/speeds). This can be done by applying nonlinear regression to interpolate or extrapolate a new gait from the key trajectories. In this way, the robot's mobility can be improved by extending feasible gaits for more selections of gait parameters.

7.2.2 CP-based Step Estimation

CP-based controller design for aperiodic prosthetic walking. With the analysis result, one potential application is to design stepping controller for lower-limb prosthesis. Different from the approaches assumed the gait is periodic, this provides additional flexibility for the user when stepping on specific footholds or moving in a short distance is required.

CP-based controller design for walking rehabilitation device. Another potential application for CP-based method is to use it for the controller design of walking rehabilitation devices (e.g. the sensory augmentation device introduced in [70].) The step estimation potentially can be an useful indicator to warn the user when he/she reaches the state which has infeasible step length (i.e. has the risk of fall), or to provide sensory cue to the user for stepping on the proper location. It is also important to incorporate nonlinear model extended from the LIPM to capture variable COM height and step impact so that the nonlinear dynamics can also be considered for more accurate step estimation.

REFERENCES

[1] K. Y. Chao, M. J. Powell, A. D. Ames, and P. Hur, "Unification of locomotion pattern generation and control Lyapunov function-based quadratic programs," in *American Control Conference (ACC)*, pp. 3910–3915, Jul. 2016.

[2] K. Chao and P. Hur, "A step towards generating human-like walking gait via trajectory optimization through contact for a bipedal robot with one-sided springs on toes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4848–4853, Sept. 2017.

[3] "Experiment of walking with multiple domains on AMBER 3." `https://youtu.be/4QS9QBgkGrQ`.

[4] E. T. Hsiao-Wecksler and S. N. Robinovitch, "The effect of step length on young and elderly women's ability to recover balance," *Clinical Biomechanics*, vol. 22, no. 5, pp. 574–580, 2007.

[5] Z. Aftab, T. Robert, and P.-B. Wieber, "Balance recovery prediction with multiple strategies for standing humans," *PLoS ONE*, vol. 11, no. 3, pp. 1–16, 2016.

[6] K. E. Moglo and C. Smeesters, "The threshold of balance recovery is not affected by the type of postural perturbation," in *ISB XXth Congress - ASB 29th Annual Meeting*, Aug. 2005.

[7] P. Ling, K. Y. Chao, H. Huang, and J. Yan, "Footprint searching and trajectory design of a humanoid robot," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 259–264, Dec. 2012.

[8] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Online footstep planning for humanoid robots," in *IEEE International Conference on Robotics and Au-*

tomation (ICRA), vol. 1, pp. 932–937, Sept. 2003.

[9] S. Veer, M. Shafiee Motahar, and I. Poulakakis, “Generation of and switching among limit-cycle bipedal walking gaits,” in *IEEE Conference on Decision and Control (CDC)*, pp. 5827–5832, Dec. 2017.

[10] X. Da and J. Grizzle, “Combining trajectory optimization, supervised machine learning, and model structure for mitigating the curse of dimensionality in the control of bipedal robots,” *arXiv e-prints*, Nov. 2017.

[11] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[12] M. Vukobratović and B. Borovac, “Zero-moment point – Thirty five years of its life,” *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[13] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1620–1626, Sept. 2003.

[14] B. Stephens and C. Atkeson, “Push recovery by stepping for humanoid robots with force controlled joints,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 52–59, Dec. 2010.

[15] P. Wieber, “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 137–142, Dec. 2006.

[16] D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a humanoid robot,” *Autonomous Robots*, vol. 35, no. 2, pp. 161–176, 2013.

[17] T. Koolen, M. Posa, and R. Tedrake, “Balance control using center of mass height variation: Limitations imposed by unilateral contact,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 8–15, Nov. 2016.

[18] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, “Capture point: A step toward humanoid push recovery,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 200–207, Dec. 2006.

[19] 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 (IJRR)*, vol. 31, no. 9, pp. 1094–1113, 2012.

[20] 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 (IJRR)*, vol. 31, no. 10, pp. 1117–1133, 2012.

[21] G. Wiedebach, S. Bertrand, T. Wu, L. Fiorio, S. McCrory, R. Griffin, F. Nori, and J. Pratt, “Walking on partial footholds including line contacts with the humanoid robot atlas,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 1312–1319, Nov. 2016.

[22] J. W. Grizzle and E. R. Westervelt, “Hybrid zero dynamics of planar bipedal walking,” in *Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori*, pp. 223–237, Springer, 2008.

[23] A. D. Ames, “Human-inspired control of bipedal walking robots,” *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, 2014.

[24] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 370–387, 2018.

[25] P. Zaytsev, S. J. Hasaneini, and A. Ruina, "Two steps is enough: No need to plan far ahead for walking balance," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6295–6300, May 2015.

[26] J. S. Matthis and B. R. Fajen, "Visual control of foot placement when walking over complex terrain," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 40, no. 1, pp. 106–115, 2014.

[27] J. Matthis and M. Hayhoe, "The functional coupling of gaze and gait when walking over real-world rough terrain," *Journal of Vision*, vol. 16, no. 12, p. 766, 2016.

[28] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics (SIAM), 2010.

[29] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[30] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research (IJRR)*, vol. 33, no. 1, pp. 69–81, 2014.

[31] 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 (ICRA)*, pp. 1447–1454, May 2016.

[32] H. Dai and R. Tedrake, "Optimizing robust limit cycles for legged locomotion on unknown terrain," in *IEEE Conference on Decision and Control (CDC)*, pp. 1207–

1213, Dec. 2012.

[33] B. Griffin and J. Grizzle, "Nonholonomic virtual constraints and gait optimization for robust walking control," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 8, pp. 895–922, 2017.

[34] T. E. Lockhart, J. C. Woldstad, and J. L. Smith, "Effects of age-related gait changes on the biomechanics of slips and falls," *Ergonomics.*, vol. 46, no. 12, pp. 1136–1160, 2003.

[35] P. A. Bhounsule, J. Cortell, A. Grewal, B. Hendriksen, J. G. 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 (IJRR)*, vol. 33, no. 10, pp. 1305–1321, 2014.

[36] Y. Wang and S. P. Boyd, "Fast evaluation of quadratic control-Lyapunov policy," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 939–946, 2011.

[37] S. Faraji, S. Pouya, C. Atkeson, and A. Ijspeert, "Versatile and robust 3D walking with a simulated humanoid robot (Atlas): A model predictive control approach," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1943–1950, May 2014.

[38] A. D. Ames, K. Galloway, and J. W. Grizzle, "Control Lyapunov functions and hybrid zero dynamics," in *IEEE Conference on Decision and Control (CDC)*, pp. 6837–6842, Dec. 2012.

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

[40] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, “Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[41] A. D. Ames and M. Powell, “Towards the unification of locomotion and manipulation through control Lyapunov functions and quadratic programs,” in *Control of Cyber-Physical Systems: Workshop held at Johns Hopkins University*, pp. 219–240, Springer, 2013.

[42] S. Kuindersma, F. Permenter, and R. Tedrake, “An efficiently solvable quadratic program for stabilizing dynamic locomotion,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2589–2594, May 2014.

[43] D. A. Winter, *Biomechanics and Motor Control of Human Movement*. Wiley, 2009.

[44] B. Stephens, *Push Recovery Control for Force-Controlled Humanoid Robots*. PhD thesis, Carnegie Mellon University, 2011.

[45] S. S. Sastry, *Nonlinear Systems: Analysis, Stability and Control*. Springer, 1999.

[46] “ZMP-based walking experiment on AMBER 3.” `https://youtu.be/KNUo4AH2fv0`.

[47] M. Posa, S. Kuindersma, and R. Tedrake, “Optimization and stabilization of trajectories for constrained dynamical systems,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1366–1373, May 2016.

[48] J. Morimoto, G. Zeglin, and C. G. Atkeson, “Minimax differential dynamic programming: application to a biped walking robot,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1927–1932, Oct. 2003.

[49] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3D dynamic walking on stepping stones with control barrier functions," in *IEEE Conference on Decision and Control (CDC)*, pp. 827–834, Dec. 2016.

[50] Q. Nguyen, X. Da, J. W. Grizzle, and K. Sreenath, "Dynamic walking on stepping stones with gait library and control barrier functions," in *Workshop on Algorithimic Foundations of Robotics (WAFR)*, 2016.

[51] Z. Manchester and S. Kuindersma, "DIRTREL: Robust trajectory optimization with ellipsoidal disturbances and LQR feedback," in *Robotics: Science and Systems (RSS)*, Jul. 2017.

[52] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[53] A. Goswami, B. Thuilot, and B. Espiau, "Compass-like biped robot part I : Stability and bifurcation of passive gaits. [Research Report] RR-2996," *INRIA*, 2006.

[54] T. McGeer, "Passive dynamic walking," *The International Journal of Robotics Research (IJRR)*, vol. 9, no. 2, pp. 62–82, 1990.

[55] Z. Shen and J. Seipel, "A piecewise-linear approximation of the canonical spring-loaded inverted pendulum model of legged locomotion," *Journal of Computational and Nonlinear Dynamics*, vol. 11, no. 1, p. 011007, 2016.

[56] M. Kelly, "Optimtraj - Trajectory Optimization for MATLAB." `https://github.com/MatthewPeterKelly/OptimTraj`.

[57] M. Ackermann and A. J. van den Bogert, "Optimality principles for model-based prediction of human gait," *Journal of biomechanics*, vol. 43, no. 6, pp. 1055–1060, 2010.

[58] F. De Groote, A. L. Kinney, A. V. Rao, and B. J. Fregly, "Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem," *Annals of Biomedical Engineering*, vol. 44, no. 10, pp. 2922–2936, 2016.

[59] H. Zhao, A. Hereid, E. Ambrose, and A. D. Ames, "3D multi-contact gait design for prostheses: Hybrid system models, virtual constraints and two-step direct collocation," in *IEEE Conference on Decision and Control (CDC)*, pp. 3668–3674, Dec. 2016.

[60] W. Xi and C. D. Remy, "Optimal gaits and motions for legged robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3259–3265, Sept. 2014.

[61] A. Hereid, S. Kolathaya, and A. D. Ames, "Online optimal gait generation for bipedal walking robots using legendre pseudospectral optimization," in *IEEE Conference on Decision and Control (CDC)*, pp. 6173–6179, Dec. 2016.

[62] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 43:1–43:8, 2012.

[63] D. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with Coulomb friction," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 162–169, Apr. 2000.

[64] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[65] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.

[66] E. Westervelt and J. Grizzle, *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, 2007.

[67] R. Riener, M. Rabuffetti, and C. Frigo, "Stair ascent and descent at different inclinations," *Gait & Posture*, vol. 15, no. 1, pp. 32 – 44, 2002.

[68] "Human-like walking on various terrains – the gait visualizations compared with the human motions." `https://youtu.be/vQxLmPpmtMA`.

[69] D. E. Lieberman, M. Venkadesan, W. A. Werbel, A. I. Daoud, S. D'Andrea, I. S. Davis, R. O. Mang'eni, and Y. Pitsiladis, "Foot strike patterns and collision forces in habitually barefoot versus shod runners," *Nature*, vol. 463, pp. 531–536, 2010.

[70] Y. T. Pan, H. U. Yoon, and P. Hur, "A portable sensory augmentation device for balance rehabilitation using fingertip skin stretch feedback," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 1, pp. 31–39, 2017.

[71] A. L. Hof, M. G. J. Gazendam, and W. E. Sinke, "The condition for dynamic stability," *Journal of Biomechanics*, vol. 38, no. 1, pp. 1 – 8, 2005.

[72] A. L. Hof, R. M. van Bockel, T. Schoppen, and K. Postema, "Control of lateral balance in walking: Experimental findings in normal subjects and above-knee amputees," *Gait & Posture*, vol. 25, no. 2, pp. 250 – 258, 2007.

[73] A. L. Hof, S. M. Vermerris, and W. A. Gjaltema, "Balance responses to lateral perturbations in human treadmill walking," *Journal of Experimental Biology*, vol. 213, no. 15, pp. 2655–2664, 2010.

[74] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1405–1411, Apr. 1991.

[75] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 239–246, Oct. 2001.

[76] M. Nazifi, K. Beschorner, and P. Hur, “Correlation between slip severity and muscle synergies of slipping,” in *American Society of Biomechanics (ASB)*, Aug. 2016.

[77] A. D. Ames, “First steps toward automatically generating bipedal robotic walking from human data,” in *Robot Motion and Control*, pp. 89–116, Springer, 2012.