Control-Theoretic Approaches to Gait Termination in Bipedal Robots

by

Zahed Dastan

Bachelor of Science in Electrical Engineering, Sahand University of Technology, 2012
Master of Science in Electrical Engineering, K.N. Toosi University of Technology, 2015

A Dissertation Submitted in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy

in the Graduate Academic Unit of Electrical and Computer Engineering

Supervisor:	Jon Sensinger, Ph.D., Electrical and Computer Engineering
Examining Board:	Chris Diduch, PhD., Electrical and Computer Engineering Kush Bubbar, PhD., Electrical and Computer Engineering Nicholas Touikan, PhD., Mathematics and Statistics
External Examiner:	Manfredi Maggiore, PhD., Electrical and Computer Engineering, University of Toronto

This dissertation is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

April, 2025

Abstract

Gait is a fundamental aspect of legged locomotion, playing a crucial role in enabling bipedal robots to navigate diverse and unpredictable environments. However, the ability to safely and efficiently terminate gait is equally important, as it ensures the robot's stability when transitioning from motion to a standstill.

This thesis addresses the problem of gait termination in n-link, underactuated point-feet bipeds that use dynamic notion of stability, known as limit cycle walkers. Using the whole-body dynamics of the robot, I present a unified gait termination solution applicable across all such walkers. This result is achieved by introducing a dynamical system model specifically for gait termination, integrated with existing walking models through a stable switching control strategy. To stabilize this model, I explore the use of smooth controllers. My analysis reveals that while such controllers may achieve local stabilization, the presence of nonholonomic contact point constraints limits the size of the region over which smooth stabilization is possible. By applying Sontag's condition, I demonstrate that smooth control cannot ensure asymptotic stabilization over sufficiently large Regions of Attraction (ROA). However, I show that the system satisfies the Lie Algebra Rank Condition (LARC), indicating that it remains controllable. This controllability enables the design of a nonsmooth switching control strategy that, unlike computationally demanding optimization-based methods such as Model Predictive Control (MPC), primarily uses computationally efficient switching controllers and resorts to more intensive methods only when necessary to maintain constraints.

The proposed switching controller comprises two sub-controllers: a balancing controller and a Ground Reaction Force (GRF) controller. The GRF controller consists of two

components: one based on Nagumo's theorem for analytically enforcing constraints, and a numerical component for handling complex cases where analytical methods are inadequate. To improve the efficiency of this numerical component, I introduce a novel Differential Dynamic Programming (DDP) approach, tailored to handle inequality constraints involving both state and control variables. I analytically prove the stability of the proposed switching strategy. Simulations confirm that this approach effectively manages contact constraints for most of the trajectory, utilizing computationally intensive controllers only when required, thereby enhancing real-time performance of the control framework.

Dedication

To my parents, for their unconditional love, support, and encouragement throughout my academic journey. Your belief in me has been my greatest motivation.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Jon Sensinger, for his invaluable guidance, encouragement, and unwavering support throughout my PhD journey. His expertise and insight have been instrumental in shaping this research.

I gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC). Their funding made this research possible and supported me throughout my studies.

I would also like to thank the Institute of Biomedical Engineering (IBME) for providing a stimulating and supportive research environment. The resources and opportunities available at IBME have been crucial in advancing my work. I am especially grateful to my colleagues and friends at IBME for their support and for creating a collaborative and enjoyable research environment. Despite the disruptions caused by the COVID-19 pandemic, their collaboration and camaraderie were vital in maintaining progress during these challenging times.

I acknowledge the use of ChatGPT for refining the grammar of this thesis. This tool has been helpful in ensuring clarity and readability in my writing.

Finally, I would like to thank the administrative staff at the Electrical Engineering Department of the University of New Brunswick (UNB) for their assistance and support throughout my studies.

Table of Contents

List of Tables

List of Figures

List of Abbreviation

LARC	Lie Algebra Rank Condition
MPC	Model Predictive Control
GRF	Ground Reaction Force
DDP	Differential Dynamic Programming
CoM	Center of Mass
ZMP	Zero Moment Point
LIP	Linear Inverted Pendulum
SLIP	Spring-Loaded Inverted Pendulum
HZD	Hybrid Zero Dynamic
ROA	Region of Attraction
PD	Positive Definite
CDDP	Constrained Differential Dynamic Programming
IPDDP	Interior Point Differential Dynamic Programming
CLF	Common Lyapunov Function
MLF	Multiple Lyapunov Function

List of Symbols

$\mathbb{R}$	Set of real numbers
n	Number of links associated with the standing biped model
q	Generalized coordinates associated with the standing biped model
$\widehat{K}$	Kinetic energy
D	Inertia matrix
$\widehat{V}$	Potential energy
$\hat{L}$	Lagrangian
H	Matrix containing Coriolis and centrifugal terms
G	Matrix containing gravitational terms
B	Control input matrix
u	Control input
x	System states
$\mathcal{X}$	Domain of definition of system states
F^T	Tangential element of the ground reaction force
F^N	Normal element of the ground reaction force
Q	Set of allowable ground reaction forces
$\tilde{Q}$	Set of allowable states and control inputs for a given Q
t	time
t_g	The transition time from walking to standing
u_d	Controller bringing the biped into the double support phase
u_s	Standing controller

x^e	Equilibrium point
x_0	Initial state
m_i	The mass of link i
m	Total mass of the biped
lc_n	Distance to the center of mass for link n
q_w	Generalized coordinates associated with the walking biped model
n'	Number of links associated with the walking biped model
e_x, e_y	Coordinates of the end of the stance leg for walking model
$\bar{e}_x, \bar{e}_y$	Coordinates of the end of the swing leg for walking model
t_c	Time to command stopping order
g_a	Gravitational acceleration
μ	Coefficient of static friction
J	Cost functional
K_B	Balancing controller
K_G	Ground reaction force controller
T_S	Tangent cone
∇	Gradient operator
Z	Zero dynamics manifold
λ_{min}	Minimum eigenvalue of the matrix
λ_{max}	Maximum eigenvalue of the matrix
ΔT	Duration that the ground reaction force controller remains active

CHAPTER 1

1 Introduction

The field of robotic legged locomotion has seen significant advancements over the past few decades. Researchers have been driven by the need to create robots capable of navigating complex and unpredictable environments, which traditional wheeled robots struggle to handle. This introduction highlights the motivation behind the shift towards legged locomotion, the challenges that remain, particularly in gait termination, and the aims of this thesis.

1.1 Motivation

Since its invention around 3500 B.C., the wheel has been the dominant method of locomotion in human artifacts. From four-wheeled wagons to advanced motor vehicles, this simple mechanism has become ubiquitous, leading many to consider it the only effective method of locomotion for human-designed vehicles. However, there are inherent limitations to wheeled locomotion. Wheeled vehicles require flat or regular surfaces to move and function effectively, but over 50 percent of the Earth's surface does not meet this criterion [1], [2]. Additionally, these vehicles are impractical in living environments with obstacles like stairs and ladders. Although wheeled vehicles can overcome small obstacles, it comes at the cost of high energy consumption [3]. While these challenges may seem daunting, nature has already provided elegant solutions, as seen in the adaptability of animals and human legs.

Limitations associated with wheeled locomotion and the superiority of animal and human legs, with their multi-link structures capable of moving over rough and uneven terrains, have inspired scientists and engineers to adopt similar mechanisms in robots. Using legs enables robots to operate in harsh environments with discontinuous ground support by varying their leg configurations and selecting contact points based on the terrain [1]. This capability allows robots to navigate living areas with ease. Legged locomotion also facilitates movement on soft surfaces like sandy soil with less energy consumption, as legged robots deform the terrain less and can escape depressions more easily [3]. Furthermore, having multiple degrees of freedom enables legged robots to change direction without slipping and adjust their height according to task requirements.

Building on these advantages, significant progress has been made in the development of legged robots. Humanoid bipedal robots such as ASIMO by Honda [4], ATLAS by Boston Dynamics [5], MABEL from the University of Michigan [6], and Rabbit based in Grenoble, France [7], have garnered significant attention for their ability to mimic human walking. Research in legged locomotion has also led to the creation of robots with more legs, such as SPOT, a quadruped by Boston Dynamics [8], and the six-legged ASTEIRSK by Arai Robotics [9]. These advances highlight the potential of legged robots to perform complex tasks and operate in diverse environments, driving ongoing research and development in this field.

1.2 Gap in the Literature

Despite the considerable progress in robotic legged locomotion, particularly for underactuated bipeds that employ dynamic notion of stability, known as limit cycle

walkers, there remains a critical gap in the understanding and control of gait termination for these systems. Most existing research has concentrated on achieving dynamic walking stability through limit cycle behaviors, allowing robots to maintain balance via controlled falls between foot contacts. However, this focus on continuous locomotion has left the problem of stopping—gait termination—largely underexplored. Current approaches for gait termination often rely on simplified models or precomputed trajectories that do not adequately address the complexities of underactuated systems. This thesis targets this specific gap by developing an analytical control strategy for gait termination in underactuated bipeds that use limit cycle walking, aiming to enhance both the safety and efficiency of these systems.

1.3 Aims

I. Develop a unified gait termination method for point-feet bipeds that employ limit cycle walking.

II. Develop a fast-converging Differential Dynamic Programming (DDP) algorithm to handle contact point constraints.

III. Design a switching control strategy for upright equilibrium stabilization of point-feet bipeds, modeled as an Acrobot-like system with nonholonomic contact point constraints.

The primary focus of this research is to develop a unified gait termination solution for n-link, underactuated point-feet bipeds that operate based on a periodic notion of stability, maintaining balance not continuously, but from one foot strike to the next. This periodic stability is fundamentally different from the requirements of gait termination, which aims

to bring the robot to a single, stable equilibrium point. Transitioning from continuous motion to a static state presents significant challenges, as control structures optimized for periodic stability are not well-suited for achieving abrupt stops without risking instability or inefficiency.

To address this discrepancy, I propose a separate dynamical system model specifically for gait termination, modeled as an Acrobot-like system. The Acrobot is a simplified robotic model consisting of an open kinematic chain in the vertical sagittal plane, with two or more rigid links connected by revolute joints. It is an underactuated system, meaning that one of its degrees of freedom is not directly controlled by actuators.

The Acrobot-like model, characterized by its underactuated nature and ability to capture the essential dynamics of point-feet bipeds, is particularly suitable for modeling gait termination. It provides a simplified yet accurate representation of the robot's dynamics during the termination phase, allowing for a more targeted control strategy. By establishing a dedicated dynamical system for gait termination, I create a unified solution that applies across different types of controllers and walking dynamics, effectively managing the transition to a static state.

While continuous controllers may locally stabilize the separate dynamical model, my analysis shows that nonholonomic constraints, being velocity-dependent and nonintegrable, limit the extent of the stabilizable region. By applying Sontag's condition, I demonstrate that smooth stabilization is not possible over sufficiently large Regions of Attraction (ROA), which are critical in practical applications. However, I show that the system remains controllable under the Lie Algebra Rank Condition (LARC). This

controllability enables the design of a nonsmooth switching controller that can achieve stabilization across a broader and more practically meaningful portion of the state space.

As a key step toward developing this comprehensive switching controller, I focus on creating a fast-converging DDP algorithm to efficiently handle contact point constraints, which are crucial for the stability and balance of bipedal robots during gait termination. The proposed DDP approach optimizes control inputs to enhance convergence and reduce computational demands, ensuring rapid and accurate decision-making in real-time applications.

The final step in developing a comprehensive gait termination solution involves designing a switching control strategy to stabilize the upright equilibrium of the Acrobot-like model, which is subject to nonholonomic contact point constraints. This strategy combines both analytical and numerical methods to effectively manage the robot's dynamics from nonzero initial conditions, ensuring a stable upright posture. By utilizing switching controllers, which are well-suited for handling nonholonomic systems, I minimize computational demands while effectively managing contact point constraints. This approach provides a robust and efficient solution for maintaining the robot's stability during gait termination.

1.4 Thesis Outline

This thesis is organized to systematically explore and address the gait termination problem in n-link, underactuated point-feet bipeds that utilize periodic notion of stability.

Chapter 1: Introduction, offers a comprehensive overview of the motivation behind this research, highlighting the specific challenges of gait termination in bipedal robots. It also

outlines the research aims, setting the stage for the subsequent chapters and guiding the direction of the thesis.

Chapter 2: Literature Review offers a detailed examination of existing methods and strategies for gait termination in point-feet bipedal robots. The chapter covers key stability methods, including static stability, ZMP control, and periodic stability, as well as the unique challenges of gait termination in limit cycle walkers. It highlights both precomputed trajectories and real-time control strategies, evaluating their effectiveness and limitations. Additionally, the review explores methods for handling contact point constraints and emphasizes the role of switching systems and controllers, which are heavily used to address the gait termination problem by managing transitions between different control modes. This comprehensive review underscores the need for innovative approaches to address stability and contact constraints, setting the stage for the contributions of this thesis.

In Chapter 3: Unified Gait Termination for Limit Cycle Bipedal Robots, the thesis introduces a novel solution for gait termination. This chapter presents a separate dynamical system, modeled as an Acrobot-like system, specifically designed for terminating the gait of bipedal robots. It details the system's constraints, and the challenges involved in transitioning from dynamic walking to a stable standing posture. The discussion includes the difficulties of achieving stabilization to an isolated equilibrium while managing nonholonomic contact point constraints.

Chapter 4: Leveraging Control Inputs to Enforce Constraints in Differential Dynamic Programming for Nonlinear Optimization, describes the development of a fast-converging DDP algorithm. This algorithm is designed to handle both control and state constraints efficiently. The chapter explains how these constraints are integrated into the DDP

framework, enhancing convergence and computational efficiency. This approach is subsequently applied to effectively manage contact point constraints during gait termination. The material presented in this chapter has been accepted for presentation and publication:

Z. Dastan and J. Sensinger, “Leveraging Control Inputs to Enforce Constraints in Differential Dynamic Programming for Nonlinear Optimization”, in *63rd IEEE Conference on Decision and Control*, 2024.

Chapter 5: Switching Control Approach to Achieve Stable Standing in Point-Feet Bipeds, focuses on the design of a switching control strategy aimed at stabilizing the upright equilibrium of point-feet bipeds, using an Acrobot-like model. The chapter covers both the analytical and numerical components of the switching controller and discusses the specific conditions under which the strategy ensures stability during gait termination.

The thesis concludes with Chapter 6: Conclusion, which summarizes the key findings and contributions of the research. It discusses the innovative approaches proposed in the thesis and the significance of the work in advancing the field of robotic control systems, highlighting the practical implications of the proposed solutions. The chapter also identifies the limitations of the current methods and suggests directions for future research to further enhance gait termination strategies for bipedal robots.

CHAPTER 2

2 Literature Review

2.1 Introduction

Gait termination, a critical aspect of bipedal locomotion, presents unique challenges due to the complexity of maintaining stability during the transition from walking to a controlled stop. This literature review explores the existing research on gait termination strategies for bipedal robots, focusing on the key factors of stability methods, point-feet bipeds, contact point constraints, and switching controllers.

By examining the limitations and strengths of previous approaches, this review aims to identify gaps in the current state of the art and highlight promising avenues for future research. The insights gained from this review will inform the development of more advanced and reliable gait termination strategies, ultimately contributing to the realization of highly capable bipedal robots.

2.2 Key Stability Methods in Bipedal Robots

The field of robotic legged locomotion has advanced rapidly as researchers aim to develop intelligent machines capable of seamlessly integrating into human environments. While wheeled locomotion is highly efficient on flat and predictable surfaces, it can struggle in environments with gaps, obstacles, or uneven terrain. Legged systems, on the other hand, can navigate such environments more effectively by adapting their gait and using discrete footholds. This adaptability and efficiency, seen in human and animal movement, have

inspired the exploration of legged robots as a solution for achieving reliable and efficient locomotion in the real world.

One of the central challenges in achieving effective legged locomotion is maintaining stability, especially during movement across uneven or dynamic surfaces. This section examines three key methods developed to ensure stability in bipedal robots: static stability, Zero Moment Point (ZMP) control, and periodic stability, often termed dynamic stability. While these are not the only approaches to achieving stability, they represent foundational techniques that have profoundly shaped the evolution of bipedal robots.

2.2.1 Static Stability Method

One of the most fundamental approaches to maintaining stability in bipedal robots is static stability, which requires that the robot's Center of Mass (CoM) remains within the convex hull of all ground contact points, referred to as the support polygon. The support polygon is the smallest convex set containing all the contact points where the robot's feet touch the ground [10], as illustrated in Figure 2.1 for a bipedal robot. This approach led to the development of early robots like WABOT 1 [11] at Waseda University and the first active exoskeletons by Vukobratovic [12].

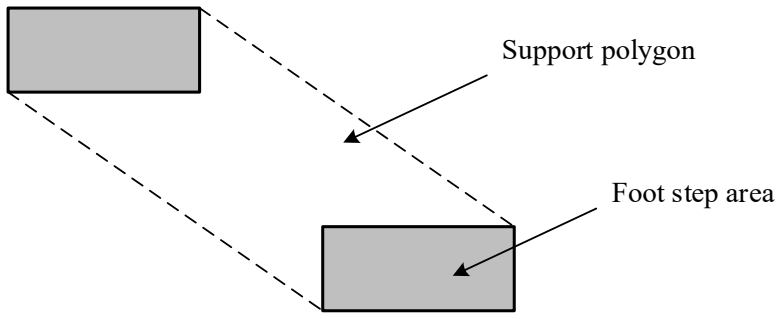


Figure 2.1. Support polygon in a bipedal robot [13].

Static stability is a widely used criterion for slow or stationary maneuvers, where dynamic forces such as momentum and inertia are minimal. In this mode, the robot can maintain stability simply by ensuring that its CoM remains within the support polygon. This criterion is simple to implement and computationally efficient, making it well-suited for early bipedal robots and robots performing slower, deliberate movements.

However, static stability has significant limitations in dynamic environments. As robots increase their walking speed or face uneven terrain, relying solely on the static position of the CoM becomes inadequate. Dynamic forces cause the CoM to shift rapidly, making it challenging to maintain stability without more advanced control strategies. Therefore, as the need for more agile and faster robots emerged, more advanced stability criteria were developed to address the shortcomings of static stability.

2.2.2 Zero Moment Point Stability

A major advancement in bipedal robot stability came with the introduction of the time-dependent ZMP criterion. ZMP was introduced to address the limitations of static stability and by far is one of the mostly adopted walking trajectory design algorithms [14]. By generating precomputed trajectories that account for dynamic forces, ZMP allows robots to maintain balance even during complex, dynamic movements.

The ZMP is the point on the ground where the sum of all moments acting on the robot is zero [14]. In practical terms, the ZMP criterion states that if the ZMP lies within the support polygon, the robot will maintain balance. This provides a more comprehensive understanding of balance, especially during dynamic maneuvers where the robot is moving [14]. Robots such as ASIMO by Honda, HRP series by AIST, TUlip by Dutch universities,

and WABIAN by Waseda university have successfully implemented this stability criterion [4], [10], [15]. These robots continuously adjust their body movements to ensure that the ZMP remains within the support polygon.

A defining feature of ZMP-controlled robots is their reliance on flat-footed walking to maximize the contact area between the foot and the ground. This larger contact patch helps distribute forces more effectively, keeping the ZMP within the support polygon to enhance stability. However, this approach results in unnatural, stiff gait patterns compared to the fluid heel-to-toe motion seen in human walking [16], [17].

While ZMP-controlled robots excel at static maneuvers with high robustness, they often lack the dynamic agility and energy efficiency found in natural locomotion, where the foot rolls and adjusts dynamically with each step. To further improve stability, ZMP-based robots tend to have larger feet, but this design also introduces awkwardness in their movement. The conservative nature of ZMP control favors stability over agility, limiting the robot's ability to execute fast, dynamic actions like running or turning quickly [18].

Despite its limitations, ZMP remains a highly effective stability criterion for many bipedal robots, especially those designed for environments that require robustness and balance on flat surfaces. ZMP's strength lies in its ability to handle disturbances and provide a stable walking gait in real-time, making it suitable for robots that need to operate in human environments, such as ASIMO. ZMP control is also favored in applications where precise movement is more important than dynamic agility, such as in industrial or service robots that need to carry loads or interact with humans in close proximity. These scenarios benefit from the balance and robustness provided by ZMP, as the robot's center of mass is tightly controlled, reducing the risk of falls [19], [20].

2.2.3 Periodic Notion of Stability

The limitations of early bipedal robots, particularly the unnatural and inefficient flat-footed walking patterns, led researchers to seek more dynamic and energy-efficient locomotion methods, ultimately giving rise to limit cycle walkers. Flat-footed walking, which focused on maximizing ground contact, typically led to awkward and constrained movements, falling short of the natural, seamless motion seen in human gait. These drawbacks, along with the inefficiencies in trajectory-based approaches like ZMP, spurred significant breakthroughs in the 1980s. Researchers began to exploit the natural nonlinear dynamics of robotic systems to achieve stable and efficient motion. Pioneering work by Marc Raibert at the MIT Leg Laboratory introduced hopping robots that could run and perform flips [21], demonstrating dynamic behaviors through models like the Spring-Loaded Inverted Pendulum (SLIP) [22]. These developments shifted the focus to a periodic notion of stability, where maintaining balance is achieved through stable, repetitive motions rather than relying on continuous monitoring or real-time adjustments [17], [23]. Concurrently, Tad McGeer's work on passive dynamic walking showcased that robots could achieve stable locomotion without full actuation [24], relying instead on their natural dynamics. Although these passive systems had limited ability to respond to disturbances, they underscored the potential benefits of underactuation in improving locomotion performance.

Despite significant advancements by the end of the 20th century, a noticeable gap persisted between the physical capabilities of robotic systems and the sophistication of controllers needed to fully utilize these capabilities. This gap was especially pronounced in the field of underactuated walking. The lack of formal methods that effectively harness the

inherently nonlinear dynamics of locomotion hindered the ability to maximize the robots' actuation potential. A pivotal advancement came in the early 2000s with the work of Jessy Grizzle and colleagues, who introduced the concept of virtual constraints [25]. Virtual holonomic constraints are dynamically imposed relationships between a robot's configuration variables, achieved through feedback control. Their purpose is to synchronize the movement of various links according to an internal gait timing variable, such as the hip position relative to the stance leg. This timing variable is chosen to be monotonically increasing during walking, allowing it to replace time as a parameter for commanding trajectories [26]. These constraints create low-dimensional invariant surfaces, known as zero dynamics manifold, within the continuous phase of the model [27]. By designing these virtual constraints to be hybrid invariant, applicable under both continuous and discrete dynamics, they developed the concept of Hybrid Zero Dynamics (HZD) [28]. The introduction of HZD enabled periodic motions to be generated naturally from the robot's interaction with the walking surface, rather than being driven by an external clock, similar to the stable periodic motion seen in a van der Pol oscillator and aligning closely with Raibert's pioneering work on the hopper [29]. Successful implementations of HZD on the RABBIT platform were demonstrated by Chevallereau et al. [7]. Another notable example is MABEL [6], a bipedal robot developed at the University of Michigan, which also utilizes HZD to achieve dynamic, efficient walking and running gaits. Building upon RABBIT, MABEL incorporates compliant elements to enhance energy efficiency and stability, making it a successful limit cycle walker capable of maintaining stability through periodic motions [30], [31]. The introduction of HZD marked a significant turning point in

formalizing control for underactuated walkers, enabling faster and more efficient algorithms for minimizing torque requirements and achieving stable, periodic orbits.

Grizzle, Plestan, and Abba were the first to establish the asymptotic stability of walking as a periodic orbit for an underactuated powered biped using formal analytical methods rather than simulations [23], [32]. Their feedback control design imposed virtual holonomic constraints during the single support phase, demonstrating stable walking in a three-link planar biped and providing crucial insights into achieving periodic stability in more complex robotic systems. This work was later extended by Chevallereau and colleagues to address both walking and running in underactuated bipeds [33].

More recently, the introduction of virtual nonholonomic constraints has enhanced the robustness of walking control for underactuated robots [26]. Unlike holonomic constraints, which rely solely on configuration variables, nonholonomic constraints incorporate velocity, allowing for dynamic posture adjustments. This approach was successfully demonstrated on MARLO [34], a 3D bipedal robot, which navigated uneven terrain using a single controller. These constraints mark a significant advancement in adaptive locomotion, enabling robots to better handle environmental disturbances and extend the limits of dynamic stability.

2.3 The Significance of Point-Feet Bipeds

The study of point-feet bipeds has garnered significant attention in bipedal robots due to their unique control challenges and potential to advance dynamic stability methods. With minimal ground contact, traditional stability frameworks like static stability or ZMP, commonly used for robots with actuated feet, are less effective for these systems [35]–[37].

Consequently, periodic stability, which relies on maintaining balance through repeated cyclical motions, has become the primary approach for point-feet bipeds [23]. This method aligns with their natural dynamics, where stability is achieved through controlled transitions between foot strikes, emphasizing their underactuated nature [1].

A key motivation for studying point-feet bipeds is their ability to mirror human walking mechanics. Human locomotion naturally involves phases of underactuation, particularly during the toe-off phase when parts of the foot lift off the ground, leveraging passive dynamics like gravity and inertia. Point-feet bipeds effectively model this natural underactuation, offering insights into more agile, energy-efficient, human-like gait [38].

Furthermore, point-feet bipeds simplify many control challenges, particularly in foot-ground interactions [39]. The absence of articulated feet and the smaller contact area reduce the complexities of managing large support polygons, making these robots an ideal platform for testing novel control strategies. Their simplicity allows researchers to focus on core dynamic stability techniques without the added complexity of multi-joint actuation. Once validated in point-feet bipeds, these strategies can be extended to more complex systems, such as humanoids with actuated feet [35].

Additionally, point-feet bipeds demonstrate their relevance in navigating complex terrains, where maintaining full-foot contact is often impossible. In real-world environments, such as rocky or uneven surfaces, robots must rely on smaller or intermittent contact points to maintain balance. With their focus on periodic stability and underactuated dynamics, point-feet bipeds are particularly suited to such environments, providing an efficient model for maintaining locomotion with minimal ground contact [17].

2.4 Gait Termination in Limit Cycle Walkers

Gait termination refers to the process of bringing a walking robot to a controlled stop. For limit cycle walkers, this is particularly challenging due to their reliance on periodic stability, which complicates the transition from motion to a stable halt. In the following sections, I will explore why gait termination presents such a significant challenge and examine the control strategies developed to address this issue.

2.4.1 Gait Termination as a Key Challenge for Limit Cycle Walkers

While significant progress has been made in developing control methods for dynamic walking, gait termination remains a major challenge. Successfully bringing a bipedal robot to a safe and controlled stop is essential for its overall functionality, allowing it to halt its walking or running gait smoothly and without instability [40]. This ability is important in various applications, including human-robot interaction, industrial manufacturing, and rescue operations [41], [42]. Robots that are unable to stop walking in a controlled way can pose a safety hazard to themselves and humans in their environment, potentially causing damage or injury [43]. In addition, the ability to terminate gait is crucial for optimizing the efficiency of bipedal robots as it allows them to conserve energy and perform tasks more efficiently by starting and stopping walking or movement as required [44]. Despite its importance, gait termination has received less attention. In part, this challenge is related to the fact that many bipedal robots have a narrow support polygon [17], [45], which renders robot stopping designs a challenging task.

A robot can stably stand on its legs when the projection of the CoM lies within the support polygon, the area formed by its contact points with the ground [46]. In static walking,

maintaining balance is relatively simple because the CoM is consistently kept within the support polygon, ensuring stability without requiring complex dynamic adjustments [11]. Additionally, stability methods that directly control the CoM, such as those used in ZMP control, allow for easier gait termination. In these methods, the primary goal is to design a CoM trajectory such that the ZMP remains inside the support polygon [47]. This controlled trajectory enables the robot to decelerate smoothly, reducing its walking speed until it stops with the CoM fully stabilized within the support polygon. By planning the CoM deceleration, the ZMP method ensures that the robot can halt in a controlled and stable manner, making gait termination more straightforward compared to more dynamic control methods.

In contrast, limit cycle walkers, which primarily involve underactuated robots [48], focus on maintaining periodic stability, where balance is achieved through stable, repetitive motions between foot strikes. During walking, the robot undergoes a controlled fall between each foot strike, relying on its momentum and transient stability to maintain balance [49]. This periodic approach is especially useful for agile, point-feet bipeds, where ZMP-based control methods are unsuitable due to their underactuated dynamics [17], [44]. For these robots, traditional ZMP control oversimplifies the system, failing to account for the nonlinear dynamics inherent in their walking patterns [17]. While periodic stability allows for efficient and adaptive walking, it also introduces significant complexity when attempting to bring the robot to a controlled stop.

It is important to recognize that the methods used to ensure stability in limit cycle walkers, such as the Poincaré map [23], are specifically designed to verify stability of a periodic solution. These methods differ fundamentally from those required to achieve static stability

during standing, where the system stabilizes at a single equilibrium point, and the projection of the CoM lies within the support polygon, as illustrated in Figure 2.2.

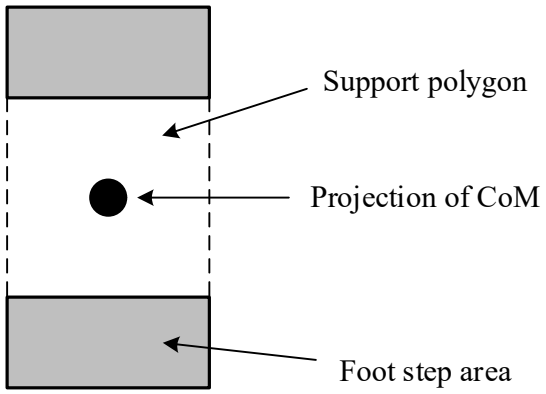


Figure 2.2. Projection of CoM within the support polygon to maintain static stability during standing.

This distinction helps explain the relative scarcity of studies directly addressing gait termination in limit cycle walkers. Among them, some studies have explicitly focused on gait termination, while others have indirectly tackled this issue by exploring control strategies for reducing or walking at zero speeds, which are closely related to the problem of stopping. Although these solutions come with certain limitations, they represent significant progress toward achieving controlled and stable gait termination. The following section will review these existing approaches, evaluating their effectiveness in managing the complexities of gait termination in dynamic walkers, and identifying areas where further improvements are needed.

2.4.2 Control Approaches for Gait Termination in Limit Cycle Walkers

To address the challenges of gait termination in limit cycle walkers, various control strategies have been developed, each aiming to improve the stopping performance of these robots. These approaches range from enforcing precomputed trajectories to employing

adaptive control methods that eliminate the need for offline planning. In the following sections, I explore these methods, beginning with precomputed trajectory enforcement.

2.4.2.1 Precomputed Trajectory Enforcement

One common method for improving the stopping performance of dynamic walkers involves enforcing precomputed trajectories, where the robot follows a set of paths generated offline. This approach has become central to several speed regulation and gait termination methods [50]–[52]. Kajita et al. [53] used a simple biped model with a rigid mass and two massless legs to generate trajectories for different phases of walking: gait initiation, move forward, and termination, based on the conservation of the potential energy of the body.

Another offline trajectory-based approach that has gained traction is stepping in place, which has been explored as a way to bring limit cycle walkers to a controlled stop. In recent work, Xingye et al. [54] considered an underactuated biped model and formulated a set of trajectories using Beziér polynomials to achieve varying walking speed from 0 m/s (i.e., stepping in place) to the top walking speed of 0.8 m/s and then return to stepping in place. In work by Gong et al. [38], a full dynamic model of the Cassie bipedal robot in the form of Lagrange was used to create a continuum of gaits for various walking speeds ranging from -0.5 m/s to +1 m/s. This library of gaits then was imposed via the virtual constraints technique which enabled Cassie to initiate walking followed by a standing phase, though with stepping in place action [55]. In addition, they developed a distinct standing mode for Cassie, but it was not integrated into the walking motion and had to be initiated separately. While these methods effectively reduce walking speed and transition to stepping in place,

they do not constitute true gait termination. The robot continues making corrective movements to maintain balance rather than executing a sudden, commanded stop.

Westervelt et al. [51] introduced a hybrid control strategy for managing speed transitions in planar bipeds that does not rely on stepping in place. Their approach combines event-based PI control with switching between a continuum of gaits, each associated with specific regions of attraction. The PI controller adjusts key parameters during discrete events (such as foot strikes) to regulate speed, while the switching mechanism ensures that the robot remains stable as it transitions between different gaits. This method enables continuous, real-time adjustments to the robot's gait, allowing for smooth transitions across a range of walking speeds, including the ability to achieve zero walking speed. As the robot's speed decreases, it gradually loses forward momentum and eventually enters a rocking back-and-forth phase before coming to a complete stop. However, like the stepping in place methods, this approach does not provide true gait termination, where the robot halts immediately in response to a command. Instead, it relies on natural deceleration, which, while effective for speed control, falls short of the human-like ability to command an immediate and controlled stop. Bhounsule et al. [50] proposed a solution aimed specifically at speed regulation for running limit cycles, using exponentially stabilizing discrete control Lyapunov functions to manage smooth transitions between running speeds. Their system successfully transitioned between speeds from 2 m/s to 5 m/s through region-of-attraction-based switching. However, this approach does not address stopping or achieving zero walking speed.

2.4.2.2 Adaptive Control Strategies Eliminating Offline Trajectory Planning

While precomputed trajectories have been effective in managing speed reduction and transitions in bipedal robots, none of these methods fully achieve gait termination in the traditional sense. Moreover, they lack the adaptability to handle changing environments and often fail to account for unexpected events. To address these shortcomings, more adaptive control strategies have emerged, eliminating the reliance on offline trajectory planning. For instance, a successful approach to the speed control of the bipeds with the ability to create a zero forward speed has been initiated by Raibert [21] for the hopping machine, namely, foot placement. This method estimates a position for the foot in the next step based on the current CoM speed. However, since a closed-form expression relating the forward foot placement to the net forward acceleration is not available, this method relies on the approximations of the exact solutions. An extension of Raibert's method to bipeds has been proposed in [56], [57] where a simplified SLIP model was used to design a controller for the ATRIAS robot with the ability to vary walking speed down to 0 m/s, though, walking in place.

Another approach to speed regulation in bipedal robots focuses on harnessing inherent mechanical properties rather than relying heavily on computational methods. As a prominent example, Hobbelen and Wisse [58] achieved speed regulation by modifying parameters such as ankle push-off, step length, and upper body pitch, leveraging the robot's natural dynamics to efficiently control speeds between 0.24 m/s and 0.68 m/s. While their approach enabled smoother speed adjustments, it did not involve walking at zero speed.

Geng [48] introduced the first application of MPC to regulate walking speed in a limit cycle walker. His method used a two-level control architecture, where MPC at the higher-level

adjusted parameters of a lower-level state machine-based controller responsible for generating the walking gait. By predicting future gait dynamics and optimizing control inputs in real time, his system successfully managed walking speeds between 40 cm/s and 85 cm/s while maintaining stability. However, the computational demands of MPC-based methods, due to the need for continuous real-time optimization, make them less practical for scenarios where rapid response times or limited computational resources are critical.

Although this research focuses on limit cycle walkers with point feet, it is important to consider Boston Dynamics' Atlas [59], as it is widely regarded as a benchmark for dynamic legged locomotion. While the company is not forthcoming in sharing detailed information about Atlas's control strategies, insights from their webinar [60] provide valuable glimpses into its operation. Building upon the early approaches to MPC, Boston Dynamics has significantly advanced the application of MPC and whole-body dynamic control in legged locomotion. To mitigate the computational heaviness of solving optimization problems online in MPC, Atlas utilizes a combination of precomputed optimized trajectories and real-time adjustments to perform high-speed, dynamic tasks such as running, jumping, and executing complex parkour movements.

The MPC in Atlas works by taking these precomputed trajectories and adjusting them in real time to respond to environmental changes and maintain balance. It predicts future states over a short time horizon and optimizes control inputs to minimize discrepancies between the desired and actual motions. This approach allows Atlas to adapt its movements on the fly, handling external disturbances and adjusting foot placement and timing to ensure stability during complex maneuvers, without the full computational burden of solving complex optimization problems entirely online. However, it is important to note that

generating these precomputed trajectories requires substantial computational effort offline, and the effectiveness of Atlas's MPC heavily relies on this extensive offline computation.

2.5 Handling Contact Point Constraints in Bipedal Robots

One of the key challenges in gait termination and overall bipedal robots' control is handling contact point inequality constraints. This problem has been widely addressed, in particular, in the case of optimization-based techniques that are widely used in robotics applications such as DDP [61], each with its unique approach to embedding these constraints into the optimization process [62], [63]. Penalty methods [64], for example, add significant costs for deviating from constraints to the objective function, effectively deterring solutions that violate these limits. This approach simplifies the computational process by treating a constrained optimization problem as an unconstrained one. However, this method faces challenges such as potential numerical instability (ill-conditioning) [61] and the difficulty of selecting appropriate penalty parameters.

On the other hand, barrier functions [65]–[69] are employed to create nearly insurmountable 'barriers' at the edges of feasible regions, ensuring that the optimization process remains within the bounds of constraint adherence. A notable instance of this methodology is found in [70], where control barrier functions are employed to enforce the nonnegative normal component of the Ground Reaction Force (GRF) within the framework of an optimization-based controller [39]. This approach involves the derivative of the normal GRF, which becomes increasingly complex when utilizing the full Lagrangian model of the robot [35], as opposed to [70], which employs a simpler SLIP model.

While barrier functions are adept at guiding the solution to stay within the feasible space, they also require meticulous parameter tuning to avoid ineffective constraint enforcement or excessively slow convergence [71]. Interior point methods [65] offer another technique for handling constraints, maintaining the search within the feasible region by utilizing barrier functions. These methods progress from the interior towards the boundary of the feasible region and require a mechanism to effectively vary the perturbation parameter during the optimization process. Additionally, they typically do not benefit from warm starts, which can be a limitation depending on the application.

Active set methods [15] present a different approach by embedding the constraints directly into the optimization process, allowing for the exploitation of warm starts and potentially reducing the number of iterations required. These methods necessitate that the constraints be explicitly dependent on the control variables, which may limit their application in scenarios where constraints are dependent on state variables [72]–[74].

A similar approach involves using barrier functions to modify an existing controller for a system so that it satisfies certain conditions, commonly known as safety conditions [75], [76]. This method also relies on optimization techniques, but there is no guarantee that the resulting controller will be stable [77]. A well-known and widely adopted application of barrier functions is within the structure of MPC, which generates control actions that do not violate dynamic and contact point constraints [78], [79]. However, this method comes with the cost of heavy computations, making it unsuitable when energy efficiency and space limitations are critical. To manage these heavy computations, researchers often use simpler robot models like Linear Inverted Pendulum (LIP) [80], [81], or employ techniques such as using a fast low-level controller in conjunction with a slower MPC update to

compensate for the slower update rates with MPC [82]. Despite these efforts, implementing whole-body MPC that utilizes full-body dynamics remains a significant challenge, and researchers continue to work on making it faster and more practical for real-world applications [83].

As precomputed trajectories can be used for gait termination, they are also utilized to handle contact point constraints. These trajectories, designed offline, ensure that the robot never violates the constraints if followed precisely [23], [50], [51]. However, these methods assume certain initial conditions and environmental parameters. Once these trajectories are set, they cannot adapt to unforeseen changes or disturbances in the environment in real time. Additionally, they heavily rely on the accuracy of the robot's model and the environment. Any discrepancies between the model used for trajectory generation and the actual system can lead to significant performance degradation.

2.6 Switching Systems

Switching systems, a class of hybrid dynamical systems, have become a versatile tool for addressing complex control problems, particularly in applications that require handling abrupt changes in dynamics or tasks [84]–[86]. The key feature of these systems is that they switch between different subsystems depending on certain conditions, which can be based on time, the system's state, or external factors. This switching actions introduces a hybrid nature to the system, where the continuous evolution of the system's state is interrupted by discrete events that cause the system to jump between different subsystems [84]. In simpler terms, think of a switched system as a machine with multiple operating settings (or modes), like a car that can switch between different gears. In each gear, the car

behaves differently, and the control system needs to manage when to switch gears to optimize performance. In a similar way, a switched system decides when to switch from one set of dynamics (rules governing its behavior) to another, with the goal of maintaining stability, performance, or achieving some specific tasks [87].

To illustrate this concept, consider a simple switching linear system with $x(t) \in \mathbb{R}^n$ representing the state of the system at time t, and two different subsystems, each represented by its own linear system:

I) The subsystem 1 is described by the equation $\dot{x}(t) = A_1 x(t)$, where $A_1 \in \mathbb{R}^{n \times n}$ is a matrix that defines the behavior of this subsystem.

II) The subsystem 2 is governed by $\dot{x}(t) = A_2 x(t)$, where $A_2 \in \mathbb{R}^{n \times n}$ represents a different dynamic behavior.

The switching between these two subsystems is controlled by a signal $\sigma(t)$, which specifies which subsystem is active at any given time

$$\dot{x}(t) = A_{\sigma(t)} x(t). \tag{1}$$

Here, $\sigma(t) \in \{1,2\}$ determines whether the system follows the dynamics of subsystem 1 or 2 at time t.

The study of switching systems spans multiple disciplines, with different communities approaching the subject from distinct perspectives. In computer science, the focus is often on the discrete aspects of the system, simplifying the continuous dynamics [88]. Key concerns in this area include ensuring well-posedness, performing simulations, and verifying system behavior. In contrast, researchers in systems and control theory typically view switching systems as continuous systems with switching dynamics, placing more

emphasis on analyzing the continuous state. For these researchers, the primary challenges lie in stability analysis and control design. This latter perspective is the one emphasized and utilized in this thesis [89].

2.6.1 Classification of Switching Systems

One way to categorize switching systems is by the conditions that trigger transitions between subsystems. These transitions can be either state-dependent, where they are triggered by the system's current state, or time-dependent, where they occur based on predefined time intervals.

In state-dependent switching systems, the transitions between subsystems are governed by the system's current state. The switching occurs when the state variables cross certain thresholds or reach predefined regions in the state space. This type of switching is often employed in scenarios where the system's dynamics need to adapt to changing conditions, such as in control systems that adjust behavior based on real-time feedback from sensors [90].

For instance, in the simple switched linear system described earlier, if the switching is based on the system's state $x(t)$, I might define a state-dependent switching rule such that

$$\sigma(t) = \begin{cases} 1 & if\ x(t) \in R_1 \\ 2 & if\ x(t) \in R_2 \end{cases} \tag{2}$$

where R_1, R_2 represent different regions of the state space. The system switches between subsystems 1 and 2 depending on whether the state $x(t)$ is within region R_1 or R_2. This approach is useful when the dynamics need to react to the internal state of the system, such

as in adaptive controllers that adjust their actions based on the current operating conditions [91].

In contrast, time-dependent switching systems rely on a predefined schedule or external timer to trigger transitions between subsystems. The switching signal $\sigma(t)$ is determined by the passage of time, often following a periodic or aperiodic pattern, without direct influence from the system's state.

For the same switched linear system, if the switching occurs at regular time intervals, the time-dependent switching rule could be expressed as

$$\sigma(t) = \begin{cases} 1 & if\ t \in [0, T_1) \cup [T_2, T_3) \cup \cdots \\ 2 & if\ t \in [T_1, T_2) \cup [T_3, T_4) \cup \cdots \end{cases} \quad (3)$$

Here, the system alternates between subsystems 1 and 2 according to specific time intervals T_1, T_2, ... regardless of the system's state.

Switching systems can be categorized not only by the conditions that trigger switching action, such as state-dependent or time-dependent switching, but also by the nature of control over the switching process, which can be either autonomous (uncontrolled) or controlled. While state- and time-dependent switching describe when the system transitions based on internal states or predefined time intervals, autonomous vs. controlled switching refers to who or what triggers the switch. In autonomous switching, the transitions happen automatically, often in response to external disturbances or system failures, without direct intervention from a controller or designer. This contrasts with controlled switching, where the system designer or controller deliberately manages when and how switching occurs, often optimizing performance or maintain stability. For example, a state-dependent system can switch autonomously when the state crosses a certain threshold, or it can switch under

controlled conditions where the system monitors state variables and switches based on set criteria. Similarly, time-dependent switching can be autonomously pre-programmed to occur at regular intervals or controlled to adapt the timing based on evolving system dynamics [89], [92].

2.6.2 Switching Controllers

Switching controllers are a class of control strategies employed to address situations where a single continuous controller is either inadequate or impractical. These controllers select from a set of control laws, switching between them based on predefined conditions. This approach is often crucial in systems where a single continuous control law cannot manage all operating conditions, or where optimal performance can only be achieved by employing a combination of controllers. Figure 2.3 illustrates the structure of a typical switching controller, where the decision mechanism governs the transition between different controllers. The act of switching between control laws alters the dynamics of the closed-loop system, classifying systems with switching controller as a subset of switching systems. Consequently, their performance and stability can be analyzed using well-established techniques from switching system theory [93], [94].

One of the most well-known examples of switching controllers is sliding mode control, which has been widely adopted by control practitioners, often without recognizing that it falls under the category of switching systems. This method has been successfully implemented in various applications, primarily due to its ability to effectively handle uncertainties and disturbances across a wide range of systems [95]–[97]. For instance, [98] demonstrates the effectiveness of sliding mode control in robotic manipulators. By using a

fast terminal sliding mode surface and integrating the super-twisting method, the authors address the challenge of trajectory tracking in the presence of parameter perturbations and external disturbances. Similarly, [99] presents a multi-input multi-output sliding mode control strategy for a gait rehabilitation exoskeleton, ensuring both accurate joint trajectory and compliance control. This application highlights the robustness of sliding mode control in nonlinear, multi-variable environments, particularly in systems with highly nonlinear actuators like pneumatic muscles.

Another well known switching controller is gain scheduling. This method involves switching between different linear controllers, so the system meets specific predetermined performance criteria [100]. [101] applies gain scheduling to standing balance of humanoid robots, dynamically adjusting feedback gains based on perturbation characteristics, demonstrating improved efficiency and stability. [102] uses a nonlinear gain-scheduled approach for a 10-DOF bipedal robot, adapting control gains between single and double support phases, resulting in accurate tracking and improved computational efficiency.

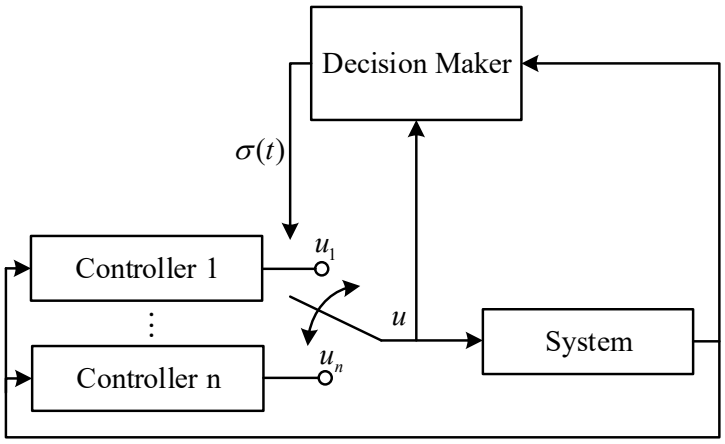


Figure 2.3. Block diagram of a switching controller.

More general forms of switching controllers have also been applied in various applications [91]. In power systems, [103] proposes adaptive switching control strategies that dynamically adjust control modes to handle disturbances like voltage and frequency fluctuations in large-scale grids. [104] introduces a novel switching strategy for Si/SiC hybrid switches, which optimizes both efficiency and reliability. The proposed method dynamically adjusts driving voltage and switching sequences based on the current load to minimize power loss and electrical stress. In three-phase voltage source converters, [105] presents a switching predictive control method that stabilizes switching frequencies and manages system delays, further improving robustness and performance.

In robotics, more advanced forms of switching controllers have been employed to address varying operational conditions, system uncertainties and disturbances. [106] discusses adaptive PID switched control to ensure precise tracking under varying loads in robotic manipulators. To manage unknown constraints in robotic systems, [107] introduces an enhanced decentralized robust adaptive control method that combines decentralized control with sliding mode techniques to handle unknown constraints while ensuring stability.

For bipedal robots, [108] presents a switching control design for accommodating large step-down disturbances, ensuring the robot transitions smoothly between walking and shock-absorbing modes to maintain stability. Further improving the adaptability of humanoid robots on uneven surfaces, [109] proposes a position-based impedance control and force control in a switched framework. Initially, the impedance control softens the impact when the robot's foot touches the ground, absorbing disturbances caused by irregular surfaces. If necessary, the control switches to force control to ensure the foot applies the correct supporting force before taking the next step. The control system is

implemented on the Nao robot, a commercial humanoid with a closed control architecture. The experimental results demonstrate improved walking stability on gravel, showcasing the benefits of integrating both control strategies.

One of the main applications of switching controllers, which will be extensively discussed and utilized in Chapter 3 and Chapter 5, is controlling nonholonomic systems, systems that cannot be managed effectively with conventional continuous controllers [110]. Perhaps the most well-known example is the control of car-like robots, which has been extensively explored in several studies [111]–[113]. For instance, [114] proposes a fuzzy gain scheduling strategy combined with a time-varying fuzzy sliding mode controller to tackle the challenges of nonholonomic constraints and input saturation. This approach reduces the complexity of traditional gain scheduling by integrating fuzzy logic to manage local path tracking, achieving precise and robust parallel parking performance. Similarly, [115] presents a state-dependent switching control strategy for a car-like mobile robot, where the system is divided into several second-order subsystems, each stabilized sequentially using finite-time controllers. This method ensures smooth transitions between phases such as stopping, moving towards a target, and adjusting orientation, demonstrating its effectiveness in managing nonholonomic constraints with robustness and practical stability.

Another well-known application of switching controllers in nonholonomic systems is the control of a knife edge, which can slide only in the heading direction, and has important applications in robotic surgery [116], [117]. Other notable examples include the control of a vertical wheel rolling without slipping on a plane surface [118], [119], and underwater vehicles moving on a horizontal plane [115]. Controlling underwater vehicles, especially

underactuated ones, presents a unique challenge due to the second order nonholonomic constraints imposed by their unactuated dynamics. These vehicles often operate with limited actuator capabilities, making direct control over certain movements, such as lateral drift, unattainable. Switching controllers effectively manage these complexities by coordinating different phases of motion, such as stabilizing orientation and regulating velocities, enabling precise navigation in the challenging conditions of underwater environments.

In summary, switching controllers offer a powerful solution for managing complex systems that are difficult or impossible to control with traditional continuous control methods. Their ability to switch between different control laws based on system conditions allows for enhanced flexibility and performance across a wide range of applications, including nonholonomic systems, robotic manipulators, and autonomous vehicles. These controllers effectively handle the dynamic challenges of underactuated systems, nonholonomic constraints, and external disturbances, enabling precise control where continuous controllers would fail.

CHAPTER 3

3 Unified Gait Termination for Limit Cycle Bipedal Robots

3.1 Introduction

In the context of limit cycle walkers, controllers are typically designed to maintain a periodic concept of stability, aligning with the cyclic nature of the robot's gait. This periodic stability is fundamentally different from the requirements of gait termination, which aims to bring the robot to a single, stable equilibrium point. The transition from a state of continuous motion to a static state poses substantial challenges, as the underlying control structures optimized for periodic stability are not well-suited for achieving abrupt stops without risking instability or inefficiency.

It is important to clarify that gait termination can be viewed as a two-step process: the first step involves transitioning the biped from dynamic walking into a double support phase, while the second step focuses on stabilizing it to an upright equilibrium. This thesis specifically addresses the second step. I assume that the robot has already come to a double support phase and aim to design control strategies that stabilize it from this configuration.

To address the challenges specific to the second phase of gait termination, this chapter proposes the development of a separate dynamical system model specifically for gait termination. The employed model is an n-link, one degree of underactuation planar biped [23] described by ordinary differential equations obtained by applying Lagrangian mechanics [49], which leverages all available degrees of freedom and actuators with fewer simplification assumptions compared to models like LIP and SLIP, enabling a more

accurate representation of the robot's dynamics. The rationale behind this approach is twofold: firstly, it acknowledges that the existing controller structures, being inherently designed for maintaining periodic motion, are ill-equipped to handle the transition to a static state. This is due to their inability to adapt to the sudden absence of dynamic motion without extensive modification or complete redesign. Secondly, by establishing a separate dynamical system dedicated solely to gait termination, I can create a unified solution that is applicable across different types of controllers and walking dynamics. This dedicated system allows for the design of control strategies that are specifically tailored to manage the complexities of stopping from various walking patterns [120], [121]. This separation not only simplifies the control design process by isolating the termination phase but also enhances the overall effectiveness of the robot's operational capabilities.

Furthermore, to seamlessly integrate this separate dynamical system with the existing walking system, I employ a switching control strategy [89]. This strategy effectively 'glues' the termination system to the walking part, enabling a smooth and controlled transition between dynamic walking and stationary standing. By using switching control, the system can instantaneously switch between the control strategies optimized for walking and those designed for termination, depending on the robot's immediate needs.

Building upon this integration, I explore the extent to which stability can be guaranteed for this separate dynamical model using analytical controllers, which are favored for their reduced computational demands compared to conventional numerical methods [61], [78]. These controllers leverage explicit mathematical expressions to facilitate control actions, significantly easing the computational load, which is crucial for real-time applications in robotics.

While continuous controllers theoretically exist that could stabilize the separate dynamic model developed for gait termination, my analysis demonstrates that, due to the system's constraints, these controllers are inadequate if stabilization over a sufficiently large ROA is desired. The contact point constraints, which are applicable not only during the walking phase but also throughout the gait termination and balancing phases, manifest as velocity-dependent nonintegrable nonholonomic inequality constraints, thus forming a nonholonomic system [122], [123]. Extensive research has been devoted to the control of nonholonomic systems and the formulation of nonholonomic dynamics [124]–[128]. However, this type of constraint is distinctly different from those encountered in scenarios such as controlling a knife edge moving in point contact on a plane surface [119] or defining nonholonomic virtual constraints for robot control [26], [129]. In my system, the nonholonomic behavior presented in the form of inequalities precludes the incorporation into the system dynamics to obtain a reduced-order differential equation [110]. While continuous controllers may achieve local stabilization near the equilibrium, using Sontag's condition, I show that smooth stabilization is not possible in sufficiently large ROA. This is a critical insight, as large ROAs are necessary in practical applications like gait termination, where the system must recover from a broad range of initial conditions.

Despite the challenges posed by nonholonomic constraints that limit the effectiveness of continuous controllers, my analysis provides a crucial contribution by demonstrating an alternative path to stability. Through rigorous analysis, I have formally established the system's controllability using the LARC [130], showing that the system remains controllable wherever the constraints do not fully block available control inputs. This finding is significant as it indicates that, although smooth feedback stabilization is

infeasible over large ROA, other methodologies such as switching or time-varying controllers offer viable solutions [89]. My extensive work in establishing this controllability lays the foundation for developing these alternative control strategies in the next chapter.

3.2 Robot Model and Constraints

This section introduces the robot model and the required constraints that will be used for the gait termination control development in this chapter. This class models a point-feet biped during the standing phase. The model is called "point feet" because it has two pointed feet that are used to support the weight of the model and provide stability during movement. This is in contrast to some other bipedal models that may have more rounded or flat feet. Additionally, I am using a full-dimensional dynamic model that captures the complete dynamics of the robot. This approach provides a more detailed and accurate representation of the biped's behavior, compared to simplified models such as LIP, inverted pendulum on a flywheel, or the SLIP [35]. By employing the full dynamic model, I can better account for the complexities of gait termination and contact point constraints.

3.2.1 Standing Biped Model

The point-feet biped model considered is similar to the walking model introduced in [23] except it is assumed that two legs are in contact with the surface in an identical configuration. This assumption allows me to view the biped as an Acrobot-like robot [120], [121] but with the distinction that it is not attached to the standing surface. To maintain name brevity, I will refer to this Acrobot-like robot as simply Acrobot throughout the chapter. The model is an open kinematic chain in a vertical sagittal xy plane and is

composed of $n \geq 2$ articulated planar weighted rigid bodies with revolute joints. Figure 3.1 illustrates such a configuration for $n = 3$. The robot has identical legs connected at a common point called the hip. It is supposed that the connection of two links is independently actuated except at the pivot, thus the system has one degree of underactuation. It is assumed that actuators cannot generate impulsive torques and their output remains continuous during the entire motion.

To derive the dynamic equations of the biped, I employ the Lagrangian method. I define the generalized coordinates as $(q_0, q_1, \dots, q_{n-1})^T$, where q_0 is measured from the horizontal, and $(q_1, \dots, q_{n-1})^T$represent the relative angles between two adjacent links as shown in Figure 3.1 for $n = 3$. Let $\widehat{K} = \frac{1}{2}\dot{q}^T D(q)\dot{q}$ be the quadratic kinetic energy in the Lagrangian where $D(q)$ is the Positive Definite (PD) inertia matrix containing $(q_1, \dots, q_{n-1})$ only. It is supposed that gravity is the only existing force field. This assumption gives rise to a potential energy $\hat{V}$ that depends only on the configuration variables [23].

The Lagrangian is defined as $\hat{L} = \widehat{K} - \hat{V}$, and Lagrange equations are employed to derive the dynamic equation of the biped

$$D(q)\ddot{q} + H(q, \dot{q}) + G(q) = Bu, \tag{4}$$

where $q = (q_0, q_1, \dots, q_{n-1})^T$, $H(q, \dot{q}) \in \mathbb{R}^n$ is the matrix containing Coriolis and centrifugal terms, $G(q) \in \mathbb{R}^n$ contains gravitational terms, $B = \begin{bmatrix} 0 \\ I_{n-1} \end{bmatrix} \in \mathbb{R}^{n \times n-1}$ and $u \in \mathbb{R}^{n-1}$ is the input torque vector generated by the actuators at the active joints. Note that

$q \subseteq [0,2\pi)^n$ corresponds to the physically reasonable configuration of the biped (i.e., all links remain above the surface).

The dynamic equation (4) is written in the state space form as

$$\dot{x} = f(x) + g(x)u, \tag{5}$$

where $x \coloneqq (q, \dot{q})^T, f(x) = \begin{bmatrix} \dot{q} \\ -D^{-1}(H + G) \end{bmatrix}$, and $g(x) = \begin{bmatrix} 0 \\ D^{-1}B \end{bmatrix}$. The domain of definition of x is the manifold given by

$$\mathcal{X} \coloneqq \{x \coloneqq (q, \dot{q})^T | q \subseteq [0,2\pi)^n, \dot{q} \in \mathbb{R}^n \}. \tag{6}$$

In what follows, I will discuss the constraints and assumptions of the introduced biped model.

3.2.2 Biped Model Constraints and Assumptions

According to Newton's first law of motion, objects continue to be in a state of rest or uniform motion unless some external forces act upon them [131]. In the case of a biped, Newton's first law means that the sole action of joint actuators internal to the system cannot influence the global position and orientation unless some part of the robot comes into contact with the environment [132]. Figure 3.1 shows that a biped interacts with the environment through the end of its leg where it experiences limited tangential F^T and normal F^N elements of the Ground Reaction Forces (GRF) [133]. The set of allowable GRF known as friction cone or contact point constraints is given by

$$Q = \{(F^T, F^N) \in \mathbb{R}^2 | F^N \geq 0, |F^T / F^N| \leq \mu\}, \tag{7}$$

where $\mu \in \mathbb{R}^+$ denotes the coefficient of static friction. The corresponding set of permissible states and control inputs, which ensures that the physical constraints of static friction are met at the interaction points between the robot and the ground, is given by

$$\tilde{Q} = \{(q, \dot{q}, u)^T \in \mathbb{R}^{3n-1} | (F^T, F^N) \in Q\}. \tag{8}$$

Note that F^T and F^N are functions of generalized coordinates q, their derivatives $\dot{q}$, and control input u [39], i.e., $F^T(q, \dot{q}, u), F^N(q, \dot{q}, u)$. However, for brevity, these dependencies have been omitted here.

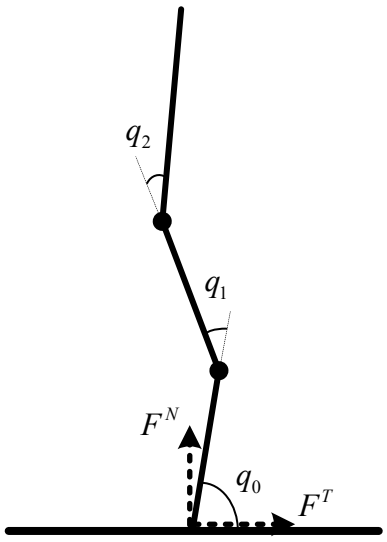


Figure 3.1. 3-link Acrobot-like model representing a standing biped with two legs held together.

Having established the interaction of the biped with the environment and delineated the constraints dictated by the GRF, I now introduce the following assumption, which focuses on the dynamics of the actuators.

Assumption 1. The actuators are subject to the following characteristics:

I. Actuators cannot generate impulses and, therefore, can be ignored during the impact phase [23].

II. They exhibit low-pass behavior, filtering out high-frequency components from the input signals.

3.3 Transitioning to Stable Standing

The process of gait termination [40] begins by positioning the walking robot in a configuration that aligns with a natural standing posture, modeled as the Acrobot represented in Figure 3.1. This transition involves two critical actions: first, moving the robot to a double support phase by placing the swing leg next to the stance leg; and second, activating a standing controller to bring the body to an upright equilibrium point and capture any residual momentum. Preserving the biped's stability during these actions is crucial. The following sections will provide a detailed explanation of each of these actions.

3.3.1 Driving the Robot to a Double Support Phase

In this subsection, I begin the process of transitioning from walking to standing at which both legs achieve the same configuration in double support at $t = t_g$. See the Chapter Notes at the end of this chapter for details regarding walking nomenclature prior to this point. At time $t = t_g$, the biped experiences impulsive forces resulting from the impact with the ground, and it is assumed that neither slip nor rebound occurs at the end of the swing leg due to these forces [23]. Note that at $t = t_g + \epsilon_g, \epsilon_g \to 0$, when the walking biped is at double support phase, GRF distribution between the two legs is an indeterminate problem [39]. However, I further assume that the feedback controller maintains the swing and stance legs in identical configurations for $t > t_g$ and GRF are equitably distributed between the

two legs. Consequently, the walking biped for $t > t_g$ in Figure 3.4 can be modeled by the Acrobot shown in Figure 3.1. This assumption simplifies the model, facilitating the demonstration of key concepts presented herein.

Now let $F_{\epsilon_g}^T$, $F_{\epsilon_g}^N$ represent GRF immediately after the impact, i.e., at $t = t_g + \epsilon_g$, then based on the no-slip no-rebound assumption mentioned in [23], I have

$$\left(F_{\epsilon_g}^T, F_{\epsilon_g}^N\right) \in Q, \tag{9}$$

which ensures that transitioning from walking to the double support occurs without the loss of stability. For later developments, I further assume that

$$dist\left(\left(F_{\epsilon_g}^T, F_{\epsilon_g}^N\right), \partial Q\right) > \epsilon_Q, \tag{10}$$

where $\epsilon_Q > 0$, and $dist(.)$ is the point to set distance given by

$$dist(\mathrm{z}, \partial Q) = \inf_{\mathrm{w} \in \partial Q} \|\mathrm{z} - \mathrm{w}\|, \tag{11}$$

where z is a vector of appropriate dimension. Condition (10) implies that $\left(F_{\epsilon_g}^T, F_{\epsilon_g}^N\right)$ is strictly inside the set Q. After achieving stable double support and subsequently modeling the biped as the Acrobot, it is necessary to switch to the standing controller that drives the system trajectories to the single equilibrium point. In what follows, I show that this switch also maintains stability.

3.3.2 Changing the Walking Controller to the Standing Controller

After positioning the walking biped in a configuration that can be modeled as the Acrobot, I am ready to switch to the standing controller. Before delving into the specifics

of this transition through Lemma 2, I reference a foundational Lemma that underpins sufficient condition for the continuity of solutions of an ordinary differential equation.

Lemma 1 [134]**:** *Assume an initial value problem defined by*

$$\dot{x} = \bar{g}(t, x, \mu_{\bar{g}}), \;\; x(t_0) = y_0, \tag{12}$$

where $\bar{g}: \mathbb{R}^n \to \mathbb{R}^n$, $\mu_{\bar{g}} \in \mathbb{R}^m$ *and* y_0 *denotes the initial condition. If* $\bar{g}$ *is continuous in* $t, x, \mu_{\bar{g}}$, *and Lipschitz in* x *with Lipschitz constant independent of* t *and* $\mu_{\bar{g}}$, *then* $x(t, \mu_{\bar{g}}, y_0)$ *is continuous in* $(t, \mu_{\bar{g}}, y_0)$ *jointly.*

With this Lemma in place, I can more confidently analyze the dynamics of switching to the standing controller, as elucidated in the following lemma.

Lemma 2. *Let* $u_d(t)$ *denote the controller responsible for bringing the biped into the double support configuration and let* $u_s(t)$ *be the standing controller. Suppose that the switch from* $u_d(t)$ *to* $u_s(t)$ *occurs immediately after impact with the ground, i.e., at* $t = t_g + 2\epsilon_g$, *then*

$$\left(F^T_{2\epsilon_g}, F^N_{2\epsilon_g}\right) \in Q, \tag{13}$$

where $F^T_{2\epsilon_g}, F^N_{2\epsilon_g}$ *represent GRF immediately after the transition to the standing controller.*

Proof:

To prove this lemma, I need to show that $\left(F^T_{2\epsilon_g}, F^N_{2\epsilon_g}\right)$ can get arbitrarily close to $\left(F^T_{\epsilon_g}, F^N_{\epsilon_g}\right)$ by selecting a sufficiently small ϵ_g. Note that F^T, F^N are continuous functions of $q, \dot{q}$, and u. Therefore, if I show that for every $0 < \delta_q, \delta_{\dot{q}}, \delta_u \ll 1$, there exists $\epsilon_g > 0$ so that

$$\begin{aligned} \left|q\left(t_g+\epsilon_g\right)-q\left(t_g+2\epsilon_g\right)\right| &< \delta_q, \\ \left|\dot{q}\left(t_g+\epsilon_g\right)-\dot{q}\left(t_g+2\epsilon_g\right)\right| &< \delta_{\dot{q}}, \end{aligned} \tag{14}$$

and

$$\left|u\left(t_g+\epsilon_g\right)-u\left(t_g+2\epsilon_g\right)\right| < \delta_u, \tag{15}$$

then $\left(F^T_{2\epsilon_g}, F^N_{2\epsilon_g}\right)$ will be sufficiently close to $\left(F^T_{\epsilon_g}, F^N_{\epsilon_g}\right)$ for $\epsilon_g \to 0$, and (13) holds true. The continuity of u immediately establishes (15). To establish (14), I must show that solutions $q(t)$ and $\dot{q}(t)$ of the differential equation (4) remain continuous over the interval $\left[t_g+\epsilon_g, t_g+\ 2\epsilon_g\right]$.

Drawing upon the conclusions reached in Lemma 1, I aim to apply its results to establish certain properties of the right side of (5) denoted by $F \coloneqq f+gu$, crucial for my proof here. First, I need to show that F is continuous and second, it satisfies the Lipschitz condition with the Lipschitz constant independent of u. To begin, I will establish the continuity of F by examining its individual components. Recall that the vector-valued function F has the following structure

$$F = \begin{bmatrix} \dot{q} \\ -D^{-1}(q)P(q,\dot{q}) + D^{-1}(q)Bu \end{bmatrix}, \tag{16}$$

where $P(q,\dot{q}) \coloneqq H+G$. Note that $D^{-1} = \frac{1}{\det(D)} adj(D), \det(D) \neq 0$ can be written in the general form of

$$D^{-1} = \frac{1}{\bar{A} + \bar{B} sin(\Gamma^T q)\cos(\Delta^T q)} \begin{bmatrix} \xi_{1,1} & \cdots & \xi_{1,n} \\ \vdots & \ddots & \vdots \\ \xi_{n,1} & \cdots & \xi_{n,n} \end{bmatrix}, \tag{17}$$

where $\xi_{i,j} = \alpha_{i,j} + \beta_{i,j} \sin\left(\bar{\gamma}_{i,j}^T q\right) \cos\left(\delta_{i,j}^T q\right), \alpha_{ij}, \beta_{ij}, \bar{A}, \bar{B} \in \mathbb{R}$, and $\bar{\gamma}_{ij}, \delta_{ij}, \Gamma, \Delta \in \mathbb{R}^{n\times 1}, i, j = 1, \dots, n$. Furthermore, $P(q, \dot{q})$ can be expressed in general form as

$$P(q, \dot{q}) = \begin{bmatrix} \vartheta_{1,1} + \bar{h}_{1,1}(q, \dot{q}) \sin\left(\bar{\eta}_{1,1}^T q\right) \cos\left(\bar{\lambda}_{1,1}^T q\right) \\ \vdots \\ \vartheta_{n,1} + \bar{h}_{n,1}(q, \dot{q}) \sin\left(\bar{\eta}_{n,1}^T q\right) \cos\left(\bar{\lambda}_{n,1}^T q\right) \end{bmatrix}, \tag{18}$$

where $\vartheta_{i,1} \in \mathbb{R}, \bar{h}_{i,1}(.): \mathbb{R}^{2n} \to \mathbb{R}$ is continuous, $\bar{\eta}_{i,1}, \bar{\lambda}_{i,1} \in \mathbb{R}^{n\times 1}, i = 1, \dots, n$.

Each element of (17) and (18) is a continuous function of $q, \dot{q}$ in its domain of definition. Therefore, in light of the fact that the product and quotient of continuous functions are also continuous ($D(q)$ is symmetric PD and nonsingular), I can conclude that (16) consists of continuous functions thus defining a continuous vector-valued function of $q, \dot{q}$. Additionally, based on Assumption 1, u remains continuous even if $u_d\left(t_g + \epsilon_g\right) \neq u_s\left(t_g + 2\epsilon_g\right)$, therefore F is a continuous function of the control input.

At this point, I have established the continuity of F, and it remains to show that it satisfies the Lipschitz condition with the Lipschitz constant independent of u. To proceed, I assume that $u, q, \dot{q}$ are bounded variables, and express the Lipschitz condition as

$$|F\ (x, u) - F\ (z, u)| \leq L|x - z|, \tag{19}$$

where L is a constant which will be determined shortly.

Let $\widetilde{K}$ be the maximum value of all partial derivatives $\left|\frac{\partial F_i}{\partial x_j}\right|$, or $\left|\frac{\partial F_i}{\partial x_j}\right| \leq \widetilde{K}, for\ i, j = 1, \dots, 2n$. Note that $\widetilde{K}$ always exists due to the continuity of partial derivatives of F and

boundness of $u, q, \dot{q}$ signals in the domain of interest. Then for fixed values of x, z, u, and for variable $\hbar$

$$\frac{d}{d\hbar}[F_i(x + \hbar z, u)] = \sum_{j=1}^{2n} \frac{\partial F_i}{\partial x_j}(x + \hbar z, u) z_j. \tag{20}$$

Therefore, using the mean-value theorem applied to the function $F_i(x + \hbar z, u)$ of the variable $\hbar \in [0,1]$, I have

$$F_i(x + z, u) - F_i(x, u) = \sum_{j=1}^{2n} \frac{\partial F_i}{\partial x_j}(x + \bar{\omega} z, u) z_j, \tag{21}$$

for some $\bar{\omega} \in [0,1]$. Squaring and using the Schwarz inequality to the right side, I get

$$|F_i(x + z, u) - F_i(x, u)|^2 \leq \sum_{j=1}^{2n} \left|\frac{\partial F_i}{\partial x_j}\right|^2 \sum_{j=1}^{2n} |z_j|^2 \leq 2n\widetilde{K}^2 |z|^2. \tag{22}$$

Summing and taking square roots over all components yield

$$|F\ (x + z, u) - F\ (x, u)| \leq 2n\widetilde{K}|z|, \tag{23}$$

which gives the Lipschitz condition with $L = 2n\widetilde{K}$ independent of u, and completes the proof of continuity of solutions■

To illustrate the key phases of the transition from limit cycle walking to stable standing, Figure 3.2 shows the sequence of events. The system begins in the walking phase until the stopping command is issued at $t = t_c$. After t_c, the robot places its swing leg next to the stance leg in preparation for the double support phase. At $t = t_g$, the swing leg impacts the ground, initiating a double support phase with impulsive GRF, causing the states to be

discontinuous. However, due to the no-slip, no-rebound condition, (9) is satisfied. By $t_g + \epsilon_g$, the GRF become fully distributed across both legs. During the interval $(t_g + \epsilon_g, t_g + 2\epsilon_g)$, since ϵ_g tends to zero and the GRF are strictly inside the set Q, they remain within the stable region. The final transition occurs at $t_g + 2\epsilon_g$ when the standing controller $u_s(t)$ is activated to stabilize the system at the upright equilibrium point.

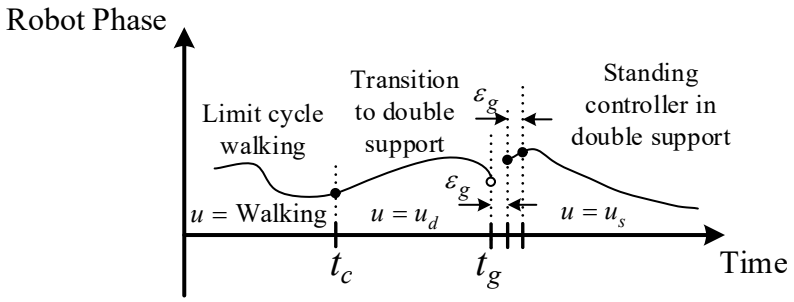


Figure 3.2. Time series representation of the transition from limit cycle walking to stable standing. The line shows how the continuity of states changes across each phase of the process.

Remark 1. It is imperative to highlight that the impulse effect manifests solely at $t = t_g$. According to Lemma 2, the transition from $u_d(t)$ to $u_s(t)$ only begins after $t = t_g$. Consequently, no additional impulsive phenomena are encountered during this transition.

In what follows, I will delve into the assessment of the standing controller u_s characteristics required in the presence of contact point constraints. The next section will explore the necessary attributes and design considerations that u_s must embody to effectively manage these constraints, ensuring the controller is tailored to maintain stability while adhering to the dynamic and physical limits imposed by the robot's interaction with its environment.

3.4 Stabilization to an Isolated Equilibrium with Contact Point Constraints

By following the procedure outlined in section 3.3, I successfully transformed the biped from walking to a standing configuration. The resulting model of the biped while considering the contact point constraints is given by

$$\dot{x} = f(x) + g(x)u, \tag{24}$$

$$(x, u) \in \tilde{Q}. \tag{25}$$

I now define the stabilization problem as follows: given the nonlinear control affine system (24)-(25), find the analytic feedback control law u_s that can asymptotically drive system trajectories to the upright equilibrium $x^e = (\pi/2, 0, \dots, 0)$ from the initial condition x^0, without the biped falling over. The class of controllers to stabilize the Acrobot to an upright equilibrium point is abundant [120], [121], [135]. However, my focus here extends to evaluating whether these continuous smooth analytical controllers are sufficiently effective in the presence of the constraints outlined in (25).

To effectively tackle this challenge, I analyze two core aspects: First, I examine the nature of the contact point constraints (25) to determine if they are nonholonomic and nonintegrable, affecting the fundamental approach to control design. Second, I assess the feasibility of achieving smooth stabilization given these constraints, focusing on whether stabilization over a sufficiently large ROA is possible. Using Sontag's condition, I show that the presence of these constraints introduces holes in the state space that prevent the ROA from being contractible, thereby precluding smooth stabilization in practically relevant regions.

3.4.1 Nonholonomic Nonintegrable Characteristic of the Contact Point Constraints

At this point, it is important to realize the nature of constraint (25) as it fundamentally distinguishes the type of control problem encountered here [110]. First, observe that (25) describes an inequality constraint, in contrast to an equality constraint [136], which has been widely addressed in the literature [137]–[139]. Second, I need to determine if it is reduceable (integrable) to a holonomic constraint or not (nonholonomic).

A constraint is holonomic if it can be expressed solely as a function of configuration variables (e.g., positions) and time, meaning the motion is confined to a lower-dimensional submanifold of $\mathbb{R}^n$. In contrast, nonholonomic constraints cannot be reduced to such a form; they typically depend on velocities and introduce nonintegrable conditions. A constraint is considered integrable if it can be rewritten as an equality involving only the configuration variables and time, independent of the system's velocities. Conversely, nonintegrable constraints inherently depend on the velocities, making them more complex to handle. A detailed discussion regarding the peculiar properties of systems with nonholonomic constraints is provided in [136].

To check the integrability of (25), let m_i be the mass and (x_i, y_i) denote the position of the CoM of link i in the Cartesian coordinates, then F^N is written as

$$F^N = \sum_{i=1}^{n} m_i \frac{d^2\big(y_i(q)\big)}{dt^2} + g_a \sum_{i=1}^{n} m_i \tag{26}$$

where g_a is the gravitational acceleration. Let $\ddot{Y}_c(q) \coloneqq \sum_{i=1}^{n} m_i \frac{d^2(y_i(q))}{dt^2}$, and $m \coloneqq \sum_{i=1}^{n} m_i$, then substitute (26) into $F^N \geq 0$ and integrate twice to get

$$Y_c(q) \geq -\frac{1}{2} g_a m t^2 + \hat{C}_1 t + \hat{C}_0, \tag{27}$$

where $\hat{C}_0, \hat{C}_1$ are some constants in $\mathbb{R}$. Although it may be argued that (26) is integrable to holonomic constraints, however, note that (26) cannot be derived from (27) because inequalities do not necessarily hold true under differentiation. Therefore, (25) describes a set of nonholonomic constraints that depend on velocity terms. This set of nonholonomic constraints alters the nature of controllers suitable for addressing the gait termination problem.

3.4.2 Checking the Necessary Condition for Smooth Stabilization in a Sufficiently Large Neighborhood of the Equilibrium

While local stability of (24)-(25) may be achievable using existing smooth controllers for the Acrobot [140], [141], this type of stability does not specify the extent of the region from which convergence to the equilibrium point is guaranteed. However, for practical applications, it is crucial that the system exhibits stability across a sufficiently large neighborhood. In this section, I will demonstrate that in the presence of contact point constraints (25), smooth controllers are only effective in a limited neighborhood of the equilibrium point x^e. To lay the groundwork for this discussion, consider the following definition.

Definition 1 (*Region of attraction of an equilibrium point*). *Consider the system* $\dot{x} = \tilde{f}(x, u)$ *with equilibrium point* x^e. *Let* $\phi(t, x_0)$ *denote the solution of this system starting from* x_0 *under the feedback law* $u = \varrho(x)$. *Then the region of attraction of* x^e *is defined as*

$$\Omega = \left\{ x_0 \in \mathcal{X} \mid \lim_{t \to \infty} \phi(t, x_0) = x^e \right\}. \tag{28}$$

The ROA exhibits specific topological properties when only the continuous class of the feedback law $u = \varrho(x)$ is considered. The following theorem elucidates these properties.

Theorem 1 (*Sontag's condition*) [142]. *Let x^e be a locally asymptotically stable equilibrium point of $\dot{x} = \tilde{f}(x, u)$ under the continuous feedback law $u = \varrho(x)$. Then the set Ω given by* (28) *is open and contractible to x^e.*

This theorem clarifies that the ROA of any smoothly stabilized equilibrium point must necessarily be free of any topological complexities. Specifically, the ROA should not contain any holes, gaps, or other discontinuities that could obstruct a smooth and continuous reduction of the region to the equilibrium. However, this crucial requirement is not met in the presence of nonholonomic contact point constraints (25), which introduce such disruptive complexities. Application of Theorem 1 to the bipedal model described by (24)-(25) leads to the following proposition:

Proposition 1. *The nonlinear bipedal model described by* (24) *cannot be stabilized to a single equilibrium point using a smooth feedback control law in any ROA containing obstructions introduced by nonholonomic contact point constraints* (25).

In order to demonstrate the relevance of this proposition to the model I am exploring, it is sufficient to demonstrate that there are specific regions within the neighborhood of x^e where no control input u can simultaneously satisfy both $F^N \geq 0, |F^T/F^N| \leq \mu$ in (7) simultaneously. To illustrate this point, I provide a numerical example that clearly shows the existence of such regions within the state space.

Consider (24)-(25) with the parameters enumerated in Table 3.1, where the acceleration of gravity is 9.8 m/s^2 and $\mu = 0.3$. This 5-link model is commonly utilized to represent a simplified version of a walking human or humanoid robot in the sagittal plane.

Table 3.1. Mass and length parameters for three link standing biped

	Link 1	Link 2	Link 3
Length (m)	0.4	0.4	0.3
Mass (Kg)	6.4	13.6	12.0

Figure 3.3 demonstrates the identified "blocked" regions within the velocity space $(\dot{q}_0, \dot{q}_1, \dot{q}_2)$ for $q_0 = -0.69, q_1 = 0.87, q_2 = -0.69$, which is a neighborhood of the upright equilibrium $(q_0, q_1, q_2) = (\pi/2,0,0)$. It is evident that any ROA encompassing these blocked regions inherently lacks the property of contractibility required for smooth stabilization, as stipulated by Theorem 1.

The analysis and numerical example presented in this section clearly demonstrate the inherent limitations of smooth feedback control laws in stabilizing (24)-(25) with a large ROA. Throughout this thesis, unless otherwise stated, stability refers to stabilization over a large ROA, one that extends beyond a small neighborhood of the equilibrium and holds practical value. This limitation necessitates a shift towards exploring and implementing nonsmooth control strategies to ensure stability across broader regions of the state space.

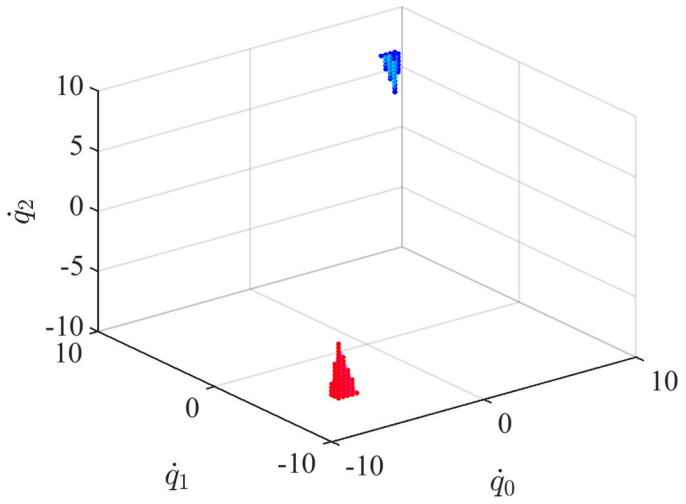


Figure 3.3. Visualization of blocked regions in velocity space $(\dot{q}_0, \dot{q}_1, \dot{q}_2)$ for $q_0 = -0.69, q_1 = 0.87, q_2 = -0.69$.

3.5 Controllability Condition

In the previous section, I demonstrated that constraint (25) is nonintegrable, meaning that it does not confine the motion of the system to any proper submanifold of $\mathbb{R}^n$. I also showed that any ROA containing blocked regions introduced by these constraints cannot be smoothly stabilized.

This section delves deeper into the dynamics of the system to reassess its controllability in any desired neighborhood of the equilibrium point. I will show that the system's state space, under the influence of this nonholonomic constraint, remains sufficiently expansive and satisfies the LARC [130]. Control systems of the form (24)-(25) satisfying the LARC are classified as completely nonholonomic [89].

3.5.1 Lie Algebraic Approach to Controllability of Nonlinear Systems

In contrast to linear systems, where in almost all cases controllability is equivalent to the Kalman rank condition, in nonlinear systems, controllability is a multifaceted concept with various interpretations depending on the context of the control system in question. While global controllability is often too stringent a requirement to handle, particularly for nonlinear systems, working with local concepts allows for an achievable analysis of control capabilities. It is important to note that my work primarily addresses the system's behavior in a local context, and unless specified otherwise, all terms and conditions discussed should be understood as pertaining to local controllability.

Central to my exploration of controllability is the concept of accessibility, a cornerstone principle within control theory. Broadly, a state x' is said to be accessible from x if there exist piecewise controls that enable the system to move from x to x' over a finite time [143]. Accessibility by nature is not necessarily symmetric, a formal symmetric definition is defined as weak accessibility:

Definition 2. *A state x' is defined as weakly accessible from x if, and only if, there exists a finite set of control inputs that can steer the system from x to x' and another, possibly distinct, set of control inputs that can steer the system from x' back to x. The set of points weakly accessible from x is denoted by $\mho(x)$.*

Weak accessibility implies the existence of a bidirectional path between the two states, although the paths for each direction may be different.

Definition 3. *A system is weakly controllable at x if $\mho(x)$ is a neighbourhood of x.*

Given a state space endowed with a topological structure, if a system is weakly controllable in each of a sequence of overlapping open subsets that cover a larger open subset, then by

synthesizing the control strategies valid within these overlapping neighborhoods, one can establish weak controllability across the entire larger open subset.

An effective method to tackle the controllability of nonlinear systems involves treating the control system as a set of vector fields, characterized by control input. Specifically, the vector field associated with (24) is given by

$$\mathcal{V} = \{f(x) + g(x)u | u \in \mathbb{R}^{n-1}\}. \tag{29}$$

Given an initial state x and the vector field X in $\mathcal{V}$, the state evolution following the path of X for duration δt is often represented using the Lie group exponential map

$$x' = \exp(X\delta t)\, x. \tag{30}$$

If Y is another vector field in $\mathcal{V}$, the state evolution following X and then Y for duration δt is

$$x'' = \exp(Y\delta t) \exp(X\delta t)\, x. \tag{31}$$

Note that if I reverse the order of following the vector fields X, Y, I might end up at a different configuration

$$\hat{x}'' = \exp(X\delta t) \exp(Y\delta t)\, x, \tag{32}$$

where generally, $x'' \neq \hat{x}''$, and it highlights that the vector fields X, Y do not commute. It is straightforward to check that for small δt, $x'' - \hat{x}''$ is approximated by

$$x'' - \hat{x}'' \approx \delta t^2 [X, Y], \tag{33}$$

where $[X, Y]$ is the Lie bracket of two nonlinear vector fields [89]. The term $\delta t^2 [X, Y]$ suggests that secondary motions can be induced in the direction of the Lie bracket $[X, Y]$,

which are not directly achievable through the actions of X or Y alone. Additionally, exploring higher-order iterated Lie brackets such as $[X,[X,Y]]$ yields even more complex motions. The study of higher order Lie brackets is critical as it lays the foundational framework for the control Lie Algebra associated with $\mathcal{V}$, which is essential for understanding the system's controllability properties [27].

Definition 4. *The control Lie Algebra associated with* $\mathcal{V}$ *and denoted by* $\mathfrak{g}(\mathcal{V})$ *is the smallest subalgebra that contains* $\mathcal{V}$*, generated by all the linear combinations of vector fields in* $\mathcal{V}$ *and all their Lie brackets.*

Focusing now on a particular state $x \in \mathcal{X}$, $\mathfrak{g}(\mathcal{V})(x)$ denotes the subspace of tangent vectors spanned by the vector fields of $\mathfrak{g}(\mathcal{V})$ at x. If $\mathfrak{g}(\mathcal{V})(x)$ is full rank at x, it is said that (24) satisfies the LARC at x. When this condition holds true for all $x \in \mathcal{X}$, the system is considered to satisfy the LARC.

Lemma 3 [143]. *The system* (24) *is weakly controllable at* x*, if it satisfies the LARC at* x*. Furthermore, it is weakly controllable if this is true for all* $x \in \mathcal{X}$.

Note that for driftless systems, where $f(x) = 0$ in (24), satisfying the LARC ensures full controllability of the system. However, for systems with drift, meeting the LARC only implies weak controllability, not necessarily full controllability [130].

3.5.2 Controllability with Nonholonomic Contact Point Constraints

Building on my previous analysis of the LARC and its implications for nonlinear systems, this subsection delves into how the contact point constraints, as specified in (25), influence the system's weak controllability. Note that (25) can be interpreted as imposing a constraint

solely on the control input u, for each state $x \in \mathcal{X}$. Specifically, it defines the set of admissible controls at x

$$U(x) \coloneqq \{u \in \mathbb{R}^{n-1} | (x,u) \in \tilde{Q}\}. \tag{34}$$

Theorem 2. *Assume that the system dynamics given by* (24) *satisfies the LARC at* x*, and* $U(x) \neq \emptyset$*. Then* (24) *with constraint* (25) *is weakly controllable at* x*.*

Proof:

Consider two vector fields $X_1(x,u_0)$ and $X_2(x,u_1)$ associated with two distinct control inputs u_0 and u_1 in $\mathcal{V}$. Using the bilinearity of the Lie bracket, I obtain

$$[X_1,X_2] = [f,f] + \sum_{i=1}^{n-1}[f,g^i u_1^i] + \sum_{i=1}^{n-1}[g^i u_0^i, f] + \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}[g^i u_0^i, g^j u_1^j], \tag{35}$$

where g^i denotes the ith column of g. Now with the fact that the Lie bracket of a vector field with itself is zero, (35) simplifies to

$$[X_1,X_2] = \sum_{i=1}^{n-1}(u_1^i - u_0^i)[f,g^i] + \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} u_0^i u_1^j [g^i, g^j]. \tag{36}$$

Given that $U(x) \neq \emptyset$ and continuity of contact point constraints, I can select $u_0, u_1 \in U(x)$ such that $u_1^i - u_0^i$ and $u_0^i u_1^j$ are nonzero for $i,j = 1,\dots,n-1$. This condition ensures that the Lie bracket $[X_1,X_2]$ is nontrivial, contributing constructively to the span of the control Lie algebra.

Since the scalar coefficients u_0^i and u_1^i merely scale the resulting vector fields (this is also the case for any higher order Lie brackets) without changing the span of the control Lie

algebra, the system continues to satisfy the LARC at x. Therefore, according to Lemma 3, it is weakly controllable at x■

Theorem 2 posits that the system dynamics specified in (24) satisfy the LARC at x. To show this analytically, it is required to compute high order Lie brackets and determinants. However, these calculations are notably complex, and so far, a general recurrence formula for the determinants remains elusive. Therefore, I present a numerical example using the actual values of the RABBIT robot [7] to demonstrate this case. By examining several specific points and patching the resulting open sets, I can infer broader conclusions about the system's controllability within a larger region [143]. Despite these efforts, it is crucial to recognize the inherent limitations of such numerical examples—they illustrate possible scenarios rather than providing comprehensive proof applicable in all cases.

Example: I analyze a 5-link model, commonly utilized to represent a simplified version of a walking human or humanoid robot in the sagittal plane. Following the methodology outlined in section 3.3, the biped has been configured into the double support phase, modeled akin to an Acrobot-like robot. The parameters defining this configuration, listed in Table 3.1, are selected to reflect the characteristics of the RABBIT robot with legs held together [120]. In this setup, $g(x) \in \mathbb{R}^{6\times 2}$, and $f(x) \in \mathbb{R}^{6\times 1}$.

To determine if the system satisfies the LARC, I analyze the rank of the distribution

$$\Delta_L(x) = span\left\{\left[g^{i_1}, \left[g^{i_2}, \dots, \left[g^{i_{k-1}}, g^{i_k}\right] \dots\right]\right]\right\}, \tag{37}$$

where $k \geq 1$, represents the order of the iterated Lie bracket and $0 \leq i_1, i_2, \dots, i_k \leq 2$ with $g^0 = f$. Due to computational complexity, iterated Lie brackets for $k \geq 3$ are impractically large; therefore, I restrict my analysis to $k \leq 2$. Calculations are performed using the

Python SymPy library, which offers a significant speed advantage in symbolic computations compared to MATLAB.

My results show that, except for the condition where $\dot{q} = 0$, $\Delta_L(x)$ is full rank within a sufficiently large neighborhood around the equilibrium point x^e, confirming that the bipedal robot satisfies the LARC. Thus, based on Theorem 2, as long as $U(x) \neq \emptyset$ in this neighborhood, the system remains weakly controllable. Therefore, alternative control strategies such as switching control or time-varying feedback are viable for successfully driving the system states towards the equilibrium point [89].

3.6 Conclusion

I introduced a unified gait termination solution for limit cycle walkers that are only periodically stable. To address the challenges of gait termination, I utilized a separate dynamical system model specifically for this task, integrating it with existing walking models through a stable switching control strategy. Given that numerical methods like MPC are computationally heavy, I investigated the use of analytical controllers. This chapter highlights a crucial difference in achieving stability through analytical controllers during gait termination, where contact point constraints make it impossible to use a continuous controller to stabilize an equilibrium point.

The study revealed that the true nature of gait termination is to stabilize a single equilibrium point, fundamentally distinct from generating periodic solutions. This observation is significant as it suggests that in the presence of nonholonomic contact point constraints, a general solution to the gait termination problem, without requiring offline trajectory generation, is to use a separate controller for standing. This represents a paradigm shift

from existing robot stopping approaches that achieve gait termination without changing the walking controller structure.

My findings indicate that although the system's constraints prevent smooth stabilization over sufficiently large regions, the system remains controllable. This suggests that alternative solutions such as switching or time-varying controllers can effectively manage gait termination.

I recognize the inherent limitations of my assumption that the feedback controller maintains both swing and stance legs in identical configurations for $t > t_g$ and ensures an even distribution of the GRFs between the two legs. While this assumption might seem an oversimplification and not entirely representative of real-world complexities, it provides a useful approximation that facilitates the computation of GRFs based on joint angles, velocities, and accelerations, which is instrumental in the development of my results.

In advancing this work, the focus will shift to the design of a switching controller that can effectively handle the complexities introduced by the contact point constraints. This will be crucial for fully realizing the potential of the proposed approach to gait termination, ensuring stability and control under various conditions.

3.7 Chapter Notes

3.7.1 Walking Biped Model

The point-feet biped model considered is an open kinematic chain that walks in a vertical sagittal xy plane and is composed of $n' \geq 2$ articulated planar weighted rigid bodies with revolute joints. It is supposed that the connection of two links is independently actuated except at the pivot thus the system has one degree of underactuation. Figure 3.4 illustrates

this model for $n' = 5$. The walking motion consists of successive phases of single and double support [49]. The single support describes the motion of the robot with only one leg on the walking surface whereas, during the double support phase, both legs are in contact with the surface. In this model, the double support takes place in an infinitesimal length of time as an impact between two rigid bodies. This impact causes a discontinuity in the velocity component of the state and leaves the configuration unaffected [144]. A complete list of assumptions for the walking biped model that is considered here can be found in [23].

3.7.2 Walking to Standing Transition

To characterize the walking to standing transition, Cartesian coordinates are added to the end of the stance and swing legs as demonstrated in Figure 3.4.

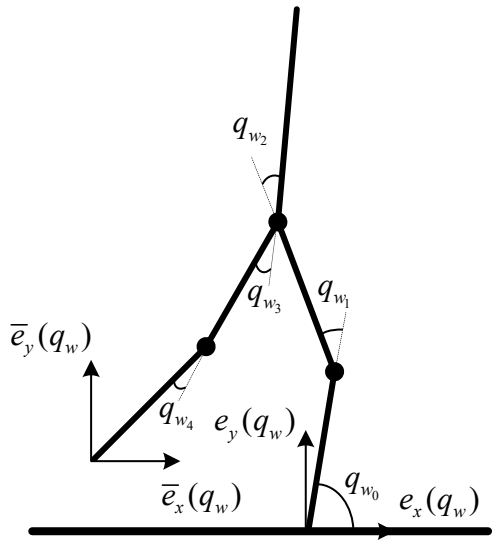


Figure 3.4. 5-link point-feet walking robot model.

Let $\left(e_x(q_w), e_y(q_w)\right)^T$ and $\left(\bar{e}_x(q_w), \bar{e}_y(q_w)\right)^T$ represent the coordinates of the end of the stance and swing legs, respectively. Once the stop order is commanded at $t = t_c$, the biped places the swing leg next to the stance leg at $t = t_g$ described by

$$\begin{bmatrix} e_x\left(q_w(t_g)\right) \\ e_y\left(q_w(t_g)\right) \end{bmatrix} = \begin{bmatrix} \bar{e}_x\left(q_w(t_g)\right) \\ \bar{e}_y\left(q_w(t_g)\right) \end{bmatrix}. \tag{38}$$

It is assumed that $t \in (t_c, t_g]$ contains only one impact with the ground occurring at $t = t_g$. Note that for the scope of this thesis, I have not designed the controller necessary to make the transition from walking to double support. Developing such a controller presents a compelling avenue for future research and would further enhance the comprehensiveness and applicability of the walking biped model.

CHAPTER 4

4 Leveraging Control Inputs to Enforce Constraints in Differential Dynamic Programming for Nonlinear Optimization

4.1 Introduction

The challenge of achieving stable gait termination in bipedal robots, particularly when dealing with nonholonomic contact point constraints, necessitates the development of innovative control strategies. In Chapter 3, I highlighted the limitations of conventional continuous controllers in this context, demonstrating the need for a new approach, specifically, a switching control strategy proposed in this thesis.

As a key step toward developing this comprehensive switching controller, this chapter introduces a novel DDP algorithm specifically designed to effectively handle both state and control constraints, with a unique capability to manage state constraints indirectly through the control inputs. By ensuring fast convergence and effectively managing these critical constraints, the proposed DDP serves as a foundational element for the control framework that will be developed in the subsequent chapter.

DDP has become a cornerstone in nonlinear system optimization, particularly in robotics where precise motion control is paramount [74], [145]. DDP distinguishes itself through an iterative approach based on Bellman's principle, optimizing input trajectories without direct parameterization of state trajectories [146]. Its algorithmic efficiency and ability to handle complex dynamics with high accuracy make DDP particularly suitable for real-time applications and scenarios requiring rapid decision-making. Additionally, the adaptability

of DDP to various control and dynamics models extends its applicability beyond traditional robotics, into areas such as autonomous vehicles [147], [148], aerospace [149], and energy systems [150], demonstrating its broad utility in tackling a wide range of optimization problems. However, incorporating inequality constraints directly into the DDP framework poses a significant challenge, particularly when dealing with complex scenarios that require stringent adherence to safety and operational limits. This limitation necessitates approaches to modify or extend DDP to extend its effectiveness and applicability across a diverse range of constrained optimization problems [151].

The foundational work in [74] advances DDP by incorporating box inequality constraints on control inputs through active set methods. This methodology integrates control constraints into the optimization framework without compromising the original DDP algorithm's convergence properties, critical for real-time applications demanding prompt decision-making. Its efficiency is evident in handling multiple smaller Quadratic Programming (QP) problems rather than a singular, larger one, markedly lowering computational complexity. Additionally, its capability to reach the solution in a single iteration when the initial point and the optimum share the same active set underscores its computational effectiveness.

Although the work in [74] can efficiently handle inequality *control* constraints, it does not extend the method to constraints involving *state*, instead relying on the fact that due to the feasibility property of indirect methods, inequality constraints on the state can be handled when enforcing regularity conditions. More recent studies, however, have cited the lack of box-bounded constraints of states as a limitation [72]. My work accordingly introduces two

novel extensions to the framework established in [74] to extend box-bounded constraints to state:

I. I propose a formulation that incorporates inequality constraints involving both control and state variables.

II. I devise a variant specifically for inequality constraints that are functions of state variables alone.

By employing these extensions, I use the robust features of the method outlined in [74], broadening the applicability to a wider spectrum of constrained optimization scenarios. I showcase the efficacy of the proposed methods through several simulations, demonstrating their practical utility and advantages in a variety of settings.

4.2 Overview of Differential Dynamic Programming

In this section, I briefly recall the derivation and implementation of unconstrained DDP. More details can be found in [145].

Consider a system characterized by its continuous-time behavior

$$\dot{x} = f_c(x, u), \tag{39}$$

where $x \in \mathbb{R}^n$ denotes the state of the system, $u \in \mathbb{R}^m$ is the control input and $f_c: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the system dynamics. For analytical and computational tractability in control applications, this continuous-time system can be approximated by its discrete-time equivalent

$$x_{i+1} = f(x_i, u_i) \tag{40}$$

where $x_i \in \mathbb{R}^n$ represents the state of the system at the discrete time step i, and $u_i \in \mathbb{R}^m$ denotes the corresponding control input.

The optimal control problem is defined by the cost functional

$$J(X, U) = \Phi(x_N) + \sum_{i=0}^{N-1} L(x_i, u_i), \tag{41}$$

where $U \coloneqq \{u_0^T, \dots, u_{N-1}^T\}$, $X \coloneqq \{x_0^T, \dots x_N^T\}$, $L(x_i, u_i)$ represents the running cost at each time step i for the state x_i and control u_i, and $\Phi(x_N)$ is the terminal cost at the final state x_N. The objective is to determine the optimal control sequence U that minimizes the cost functional J subject to the system dynamics given by (40) with a given initial state x_0.

The Value function, $V(x)$ quantifies the minimal cost-to-go from any given state x_i to the final state under an optimal control policy. Formally, the Value function at time step i in a finite-horizon optimization framework is defined as

$$V_i(x_i) = \min_{u_i} \{L(x_i, u_i) + V_{i+1}\big(f(x_i, u_i)\big)\}, \tag{42}$$

with the terminal condition given by $V_N(x_N) = \Phi(x_N)$. The objective of DDP is to find local optimal solution for (41) by expanding both sides of (42) about a nominal trajectory $\bar{X}, \bar{U}$.

Let $Q_i(\delta x_i, \delta u_i)$ denote the change in the argument of the right-hand side of (42)

$$Q_i(\delta x_i, \delta u_i) = L(x_i + \delta x_i, u_i + \delta u_i) + V_{i+1}\big(f(x_i + \delta x_i, u_i + \delta u_i)\big). \tag{43}$$

Note that Q encapsulates the immediate cost associated with small deviations, δx and δu, in the state and control vectors, respectively. The second-order expansion of Q is

$$Q_i(\delta x_i, \delta u_i) \approx Q_i + Q_{x,i}^T \delta x_i + Q_{u,i}^T \delta u_i + \frac{1}{2}\delta x_i^T Q_{xx,i} \delta x_i + \delta x_i^T Q_{xu,i} \delta u_i + \frac{1}{2}\delta u_i^T Q_{uu,i} \delta u_i, \tag{44}$$

with

$$\begin{aligned} Q_{xx,i} &= L_{xx,i} + f_x^T V_{xx,i+1} f_x + V_{x,i+1} \cdot f_{xx}, \\ Q_{xu,i} &= L_{xu,i} + f_x^T V_{xx,i+1} f_u + V_{x,i+1} \cdot f_{xu}, \\ Q_{uu,i} &= L_{uu,i} + f_u^T V_{xx,i+1} f_u + V_{x,i+1} \cdot f_{uu}, \\ Q_{u,i} &= L_{u,i} + f_u^T V_{x,i+1}, \\ Q_{x,i} &= L_{x,i} + f_x^T V_{x,i+1}. \end{aligned} \tag{45}$$

Now the optimal control perturbation δu_i^* for some δx_i is obtained by minimizing (44)

$$\delta u_i^* = k_i + K_i \delta x_i, \tag{46}$$

where $k_i = -Q_{uu,i}^{-1} Q_{u,i}$ and $K_i = -Q_{uu,i}^{-1} Q_{ux,i}$. By substituting (46) back into the quadratic model of the Value function V_i, I get

$$\begin{aligned} V_{x,i} &= Q_{x,i} - Q_{xu,i} Q_{uu,i}^{-1} Q_{u,i}, \\ V_{xx,i} &= Q_{xx,i} - Q_{xu,i} Q_{uu,i}^{-1} Q_{ux,i}. \end{aligned} \tag{47}$$

The backward pass in DDP starts by setting the Value function to the terminal cost $V(x_N) = \Phi(x_N)$, and proceeds to calculate its derivatives [74]. It then iteratively computes the optimal control perturbations and the corresponding updates to the Value function, as encapsulated in equations (46) and (47). Upon completion of the backward pass, the forward pass applies the derived control sequence to update the system states. The iterative process continues until convergence criteria—such as a sufficiently small change in the control sequence or the cost-to-go—are met, indicating that an optimal or near-optimal control policy has been found.

4.3 Integrating Constraints Into DDP

This section delineates the theoretical underpinnings of the proposed extensions to DDP for handling inequality constraints. My methodology is divided into two primary extensions, each designed to address specific types of inequality constraints and broaden the applicability of DDP in constrained optimization scenarios.

4.3.1 Integrating Inequality Constraints Involving Control and State Variables

My first extension is motivated by integrating inequality constraints of the form

$$h(x, u) \geq 0, \tag{48}$$

where $h \in \mathbb{R}^P$, into the DDP framework. These constraints are commonly encountered across a variety of applications such as robotics [152], and vehicle control [153], to name a few.

In contrast to [74], who assumes a fixed boundary of u along the entire trajectory, my method introduces an iterative adjustment of control input boundaries, a process that is confined to the backward pass phase of DDP. This adjustment is based on the current nominal trajectory, x_i, for $i = 0, \ldots, N$, succinctly formulated as

$$h(x_i, u) \geq 0 \; for \; i = 0, \ldots, N. \tag{49}$$

Such constraints inform the dynamic establishment of control input boundaries

$$u_i \in \left[u_i^{min}(x_i), u_i^{max}(x_i)\right], \quad i = 0, \ldots, N. \tag{50}$$

where $u_i^{min} = \begin{bmatrix} u_i^{1,min} \\ \vdots \\ u_i^{m,min} \end{bmatrix}, u_i^{max} = \begin{bmatrix} u_i^{1,max} \\ \vdots \\ u_i^{m,max} \end{bmatrix}$. Given these dynamically adjusted boundaries, I then articulate the optimization problem at each timestep as follows

$$\begin{aligned} &\min_{\delta u_i} Q\,(\delta x_i, \delta u_i) \\ &\text{subject to} \quad u_i^{min}(x_i) \le u_i + \delta u_i \le u_i^{max}(x_i). \end{aligned} \tag{51}$$

For the sake of compactness and clarity in later discussions, I will omit the explicit dependence on x from $u_i^{min}(x)$ and $u_i^{max}(x)$.

I can then address this optimization problem by employing the method proposed in [74] with the extension of having a state-informed varying box constraint of u rather than the conventional non-state informed constant boundary of u. This approach modifies the traditional DDP algorithm to directly incorporate control input constraints into the optimization procedure. Utilizing Tassa et al. method offers distinct advantages. Notably, one of its key features is the capability to converge to the optimal solution within a single iteration if the initial point shares the same active constraint set with the optimum. This efficiency is a direct consequence of employing the Projected-Newton class of algorithms, which are adept at handling simple constraints through the projection operator. These algorithms facilitate a dynamic adjustment process where the search point is continuously clamped to respect multiple constraints, thereby enabling a highly efficient and targeted approach to constraint management within the DDP framework.

Note that my approach streamlines the process of incorporating inequality constraints by specifically satisfying the constraints through the control inputs. The constraint (49) is used to determine the bounds for the control input u, rather than simultaneously considering the

state and control spaces. This approach significantly reduces complexity as it circumvents the need to resolve the constraints in the higher-dimensional joint space of (x, u) [72]. In formal terms, assuming that the system's dynamics are given by (40), which can be thought of as a vector field in the state space x parametrized by u, my approach involves selecting control directions so that the resulting trajectory inherently satisfies (49).

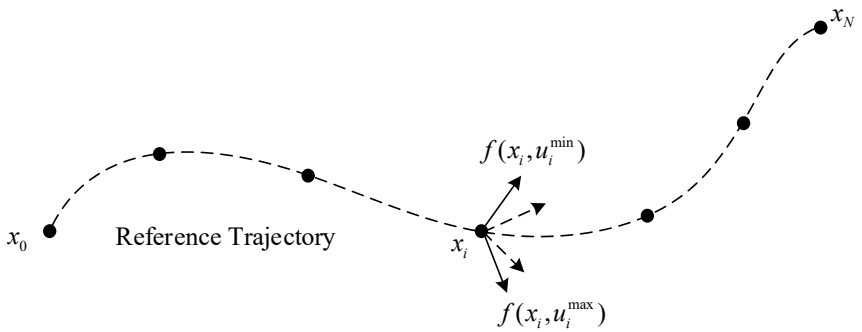


Figure 4.1. Calculating u_i^{min} and u_i^{max} during backward pass at each point x_i of the reference trajectory, which yields range of vectors $f(x_i, u_i), u_i \in \left[u_i^{min}, u_i^{max}\right]$. Modest adjustments along these vectors ensure (49) is still satisfied.

To visualize the practical implementation of this methodology, consider the following process. Assume that from some initial condition x_0, and initial control input u_0, a reference trajectory has been generated during the forward pass, depicted by dashed line in Figure 4.1. During the backward pass, starting from x_N, for each point x_i along this trajectory, I compute u_i^{min} and u_i^{max}, resulting in a set of vectors given by $f(x_i, u_i), u_i \in \left[u_i^{min}, u_i^{max}\right]$. The algorithm then selects the optimal δu_i to minimize Q, ensuring compliance with the constraints. In the subsequent forward pass, the reference trajectory is updated according to these optimized control inputs, reflected in Figure 4.2 with a solid line.

It is worth mentioning that as long as the adjustments to u are modest and $h(x,u)$ is smooth, the updated trajectory will continue to fulfill the inequality constraint

$$h(x_i + \delta x_i, u_i + \delta u_i) \geq 0 \; for \; i = 0, \dots, N. \tag{52}$$

By focusing on adjusting u within its allowable range to ensure compliance with the constraints, I bypass the intricacies that arise from the interdependencies between x and u. This targeted strategy not only simplifies the computational procedure but also aligns with the intuitive understanding that control inputs are often the direct levers for ensuring constraint satisfaction.

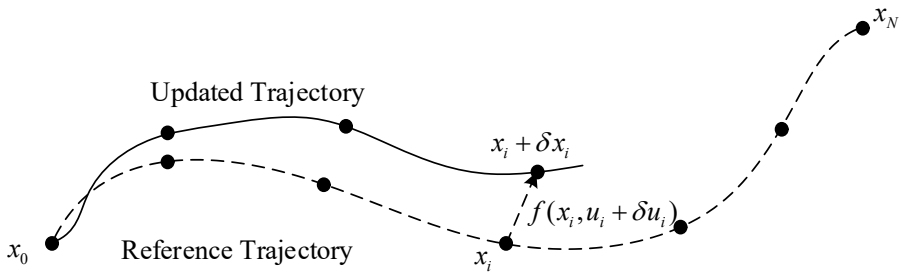


Figure 4.2. Updating the reference trajectory during the forward pass.

4.3.2 Constraints Without Explicit Control Input

In this subsection, I broaden the scope of my method to accommodate constraint solely dependent on the state, denoted as

$$h(x) \geq 0, \tag{53}$$

which does not explicitly involve the control input u. To integrate such constraints within my framework, I turn to the concept of relative degree from nonlinear control theory.

Assume (53) exhibits a relative degree one with respect to one control input. This assumption implies that the time derivative of $h(x)$ denoted as $\dot{h}(x)$ and given by

$$\dot{h}(x) = \frac{\partial h}{\partial x}\dot{x}, \tag{54}$$

transforms with the system dynamics (4), into

$$\dot{h}(x) = \frac{\partial h}{\partial x}f(x,u), \tag{55}$$

which explicitly depends on u. Now to enforce (53) within the context of my DDP framework, I construct the differential equation

$$\dot{h}(x) + \alpha h(x) \geq 0, \tag{56}$$

where $\alpha > 0$ is a constant that can be determined through iterations, ensuring that the constraint is not violated. Substituting $\dot{h}(x)$ from (55) yields

$$\frac{\partial h}{\partial x}f(x,u) + \alpha h(x) \geq 0. \tag{57}$$

This formulation introduces a condition that dynamically adjusts the system's behavior to maintain compliance with the constraint. Specifically, if $h(x)$ were at risk of becoming negative, the positive contribution from $\alpha h(x)$ encourages a corrective increase in $\dot{h}(x)$, directing $h(x)$ back towards positive values. This approach ensures the system's states remain within the permissible region by dynamically enforcing the constraints without the need for explicit control input in the constraint equation.

With the reformulated constraint now implicitly involving the control input u, I can directly apply the methodology developed in the previous subsection to my current problem. By

translating the state-dependent constraints into a form that includes the control input u indirectly, I extend the DDP framework to handle a broader range of constraints.

4.4 Simulation Results

In this section, to assess the performance of my proposed DDP algorithm, I conducted a series of simulations across two distinct scenarios: inverted pendulum [65], and a nonholonomic 2D car [74]. My objective was to benchmark the proposed DDP algorithm against two well-established methods: the Constrained DDP (CDDP) [72] and the primal–dual interior-point DDP (IPDDP) [65]. For all simulations, the initial control input u_0 is selected by generating small random values, providing a diverse starting point for the optimization process. These simulations are designed to test the algorithm's capability in handling different dynamic systems and control challenges. To effectively evaluate and compare these algorithms, while recognizing the importance of iteration time, I selected the number of iterations as my primary performance metric [65]. This choice is rooted in providing a hardware-independent assessment of each algorithm.

4.4.1 Inverted Pendulum

In my first simulation, I evaluate the effectiveness of my proposed DDP method in controlling an inverted pendulum model, in comparison with the CDDP and the IPDDP methods. The system is characterized by the states $X = [\phi, \omega]^T$, and following discrete-time dynamic equations

$$\begin{aligned} \phi_{i+1} &= \phi_i + h\omega_i, \\ \omega_{i+1} &= \omega_i + hsin(\phi_i) + hu_i, \end{aligned} \tag{58}$$

where ϕ is the angle, ω is the angular velocity, and u denotes the control input. The task is to drive the pendulum from an initial state $X_0 = [-\pi, 0]^T$ to $X_{final} = [0,0]^T$, over a control horizon of $N = 500$, and $h = 0.05$. I incorporate a state-dependent constraint to maintain the angular velocity below a certain threshold

$$\omega < 1. \tag{59}$$

Given that (59) has a relative degree of one, I employ (57) with $\alpha = 0.1$ for my proposed DDP algorithm. The running cost and the terminal cost used for the simulations across all three algorithms are defined as

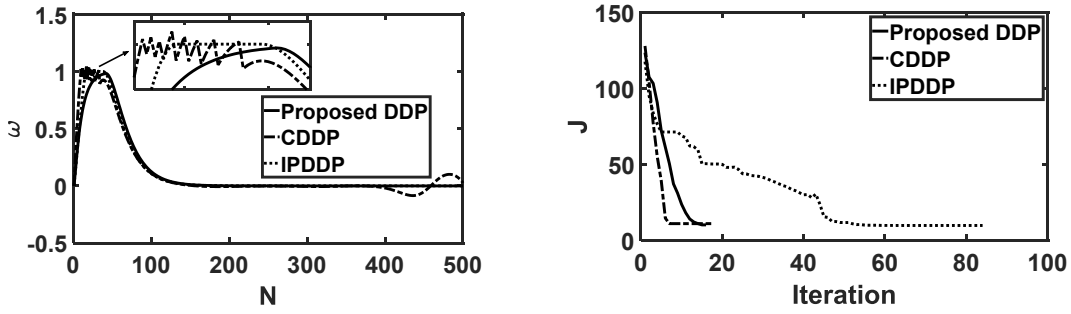
$$L(X, u) = 0.025(\phi^2 + \omega^2 + u^2), \tag{60}$$

$$\Phi(X) = 5(\phi^2 + \omega^2). \tag{61}$$

Figure 4.3a illustrates the trajectories of the angular velocity, ω, as a function of the control horizon N. It is observed that while all three algorithms successfully drive ω close to 0, the CDDP algorithm's trajectory shows fluctuations near the constraint boundary, which is not desirable. Figure 4.3b presents the evolution of the cost function J with respect to the number of iterations. Here, the proposed algorithm and CDDP demonstrate rapid convergence within fewer than 20 iterations, specifically 8 for CDDP and 16 for the proposed method. In contrast, IPDDP requires nearly 60 iterations to stabilize.

Notably, my proposed algorithm achieves a final cost of 10.11 with an approximate ω_{final} of 0, outperforming CDDP, which concludes with a final cost of 10.92 and an ω_{final} of approximately 0.02. The IPDDP method attains the best final cost of 9.72, albeit with a higher iteration count, yet still arriving at an ω_{final} close to 0. These results affirm the

robustness of my proposed method in efficiently reaching the target state while also adhering to the state-dependent constraint imposed on the system.



(a) (b)

Figure 4.3. Comparative analysis of DDP algorithms on the inverted pendulum model. (a) Angular velocity (ω) trajectories. (b) Cost function (J) convergence.

4.4.2 2D Car

In the subsequent example, I turn my attention to the application of the proposed DDP algorithm on a 2D car navigation scenario. The car is modeled with a state vector $X = [x, y, \theta, v]^T$, delineating the car's planar position (x, y), angle θ, and front wheel velocity v. The control inputs are represented by $u = [\omega, a]^T$, which consist of the steering angle ω and acceleration a. The dynamics of the car are given by

$$\begin{aligned} x_{i+1} &= x_i + b\cos\theta_i\,, \\ y_{i+1} &= y_i + b\sin\theta_i, \\ \theta_{i+1} &= \theta_i + sin^{-1}\left(\frac{h}{d}sin(\omega_i)\right), \\ v_{i+1} &= v_i + ha_i, \end{aligned} \tag{62}$$

where $b = d + hv\cos(\omega) - \sqrt{d^2 - h^2v^2\sin^2(\omega)}$. The constants are set as $d = 2.0$, representing the distance between the rear and front axles, and $h = 0.03$, which represents the time step.

4.4.2.1 Comparison with CDDP

The navigation task involves maneuvering the car from the initial state $X_0 = [1,1,3\pi/2\,,0]^T$ to the destination $X_{final} = [0,0,0,0]^T$, circumventing a circular obstacle centered at $[-2,-3]$ with a radius of 1 defined by

$$(x+2)^2 + (y+3)^2 \geq 1. \tag{63}$$

Note that (63) has a relative degree of one; therefore, I utilize (57) with $\alpha = 0.1$ for the proposed DDP. The horizon is set to $N = 500$, with steering angle constraint of $-\pi/2 \leq \omega \leq \pi/2$ and acceleration constraint of $-8 \leq a \leq 8$ for both algorithms. The cost function employed for both CDDP and the proposed DDP is composed of a running cost

$$L(X,u) = 10^{-3}r(x,0.1) + 10^{-3}r(y,0.1) + 10^{-2}(\omega^2 + 0.01a^2), \tag{64}$$

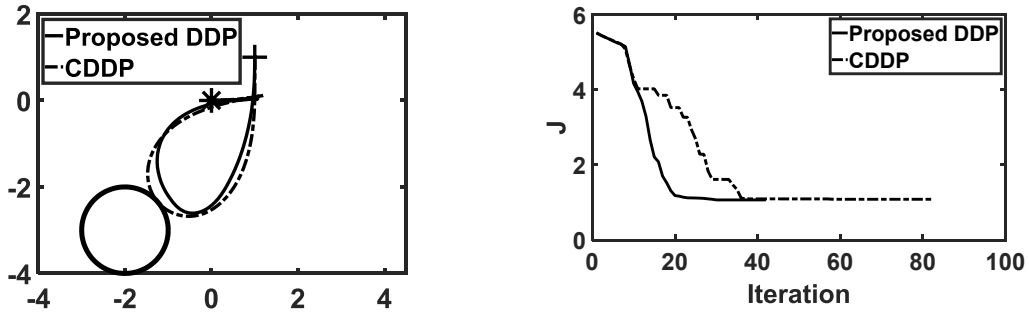
and the terminal cost

$$\Phi(X) = 0.1r(x,0.01) + 0.1r(y,0.01) + r(\theta,0.01) + 0.3r(v,1), \tag{65}$$

where $r(x,p) = \sqrt{x^2+p^2} - p$ represents the Huber-type loss function [74].

Figure 4.4a illustrates the trajectories generated by both the proposed DDP and the CDDP algorithms as they navigate towards the destination while avoiding the circular obstacle. The effectiveness of both methods is demonstrated by their successful avoidance of the obstacle and their convergence towards the final desired state. Specifically, the final state achieved by the CDDP algorithm is $X = [0.007,0.003,0.002,-0.228]^T$, while the proposed DDP method reaches a final state of $X = [0,0.006,0.002,0]^T$. Furthermore, the cost function J for both proposed and CDDP algorithms are depicted in Figure 4.4b, offering a quantitative measure of their performance. Notably, the proposed DDP algorithm

exhibits a faster convergence, requiring only 25 iterations to reach a steady value, as opposed to the 45 iterations needed by the CDDP method. Despite this, the CDDP algorithm demonstrates a slight advantage in minimizing the total cost, achieving an optimal cost of 1.086 compared to 1.067 for the proposed DDP method.



(a) (b)

Figure 4.4. Performance comparison in 2D car navigation for the proposed and CDDP algorithms. (a) Trajectory paths of the proposed and CDDP algorithms. (b) Cost function (J) convergence.

4.4.2.2 Comparison with IPDDP

In this subsection, I present a comparative analysis of my proposed DDP algorithm with the IPDDP method. The evaluation is carried out in a scenario involving navigation through a field with three circular obstacles. The navigation task commences from the initial state $X_0 = [0,0,0,0]^T$ and aims to reach the destination $X_{final} = [3,3,\pi/2\,,0]^T$. The environment comprises three circular obstacles located at $[1,1],[2.5,2.5],[1,2.5]$, each with a radius of 0.5. Due to the relative degree of one presented by the obstacles, I apply (57) with distinct α parameters— $\alpha_1 = 0.2, \alpha_2 = 0.01$, and $\alpha_3 = 0.02$—for each corresponding constraint. The horizon is set at $N = 200$, with the steering angle ω constrained within $-\pi/2 \leq \omega \leq \pi/2$.

The running cost for both the proposed DDP and IPDDP algorithms is defined as follows

$$L(X,u) = 10^{-4}r(x,0.1) + 10^{-3}r(y,0.1) + 10^{-4}(\omega^2 + 0.01a^2). \quad (66)$$

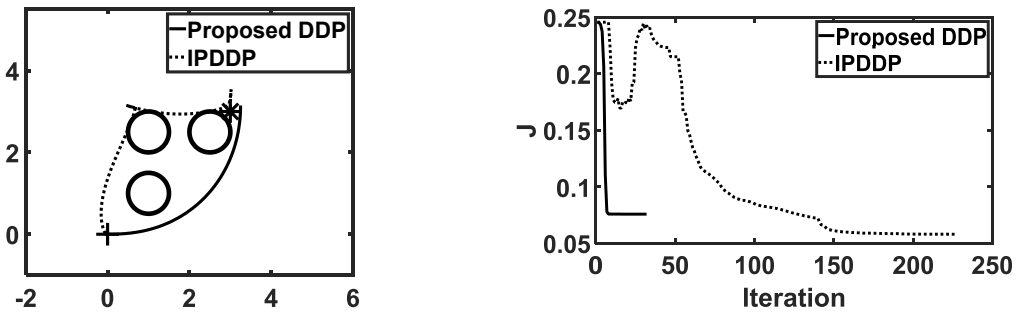
The terminal cost is given by

$$\Phi(X) = 0.06\big(0.1r(x,0.01) + 0.1r(y,0.01) + r(\theta,0.01) + 0.3r(v,1)\big). \quad (67)$$

This cost function has been selected to accommodate the changed obstacle structure, ensuring that both the proposed DDP and IPDDP algorithms converge efficiently.

Figure 4.5a displays the paths computed by the proposed DDP and IPDDP algorithms. While both methods adeptly navigate around the obstacles and converge towards the target, it is noteworthy that the proposed DDP algorithm yields a more direct path compared to the circuitous route taken by the IPDDP. The final state reached by the IPDDP is $X = [3,3.01,1.57,-0.03]^T$, while the proposed method achieves $X = [3.25,3.15,1.57,0.41]^T$. These results suggest that the IPDDP algorithm has a slight edge in reaching the precise X_{final}.

Nonetheless, when examining the cost function J, depicted in Figure 4.5b, it is observed that the proposed algorithm converges significantly faster, in only 22 iterations, compared to 160 iterations for the IPDDP. Though the IPDDP achieves a slightly lower final total cost of 0.0615 compared to the 0.0761 of the proposed method.



(a) (b)

Figure 4.5. Performance comparison in 2D car navigation for the proposed and IPDDP algorithms. (a) Trajectory paths of the proposed and IPDDP algorithms. (b) Cost function (J) convergence.

4.5 Conclusion

This chapter contributes to the field of nonlinear system optimization by exploring the use of control inputs to directly enforce inequality constraints within DDP. I have introduced two variants of this extended DDP framework, targeting constraints associated with both state and control variables, and those specific to state variables alone. Through simulations on an inverted pendulum and a nonholonomic 2D car, my method was compared against established techniques such as CDDP and IPDDP, demonstrating its feasibility and potential advantages in certain aspects of convergence rate and trajectory efficiency.

CHAPTER 5

5 Switching Control Approach to Achieve Stable Standing in Point-Feet Bipeds

5.1 Introduction

In Chapter 3, it was demonstrated that gait termination in underactuated point-feet bipeds that use dynamic notion of stability, known as limit-cycle walkers, can be reduced to the problem of stabilizing an Acrobot-like system subject to contact point constraints. Using Acrobot, an n-link planar robot with one passive joint, renders the control problem tractable and serves as an idealized model for analyzing the stability challenges in bipedal gait termination. However, it was shown that existing controllers for the Acrobot [120], [154], which typically rely on smooth, continuous control inputs, cannot stabilize the system in a sufficiently large ROA due to the presence of nonholonomic constraints. Despite this limitation, it was demonstrated that the system remains controllable as it satisfies the LARC [155]. This means that, while smooth stabilization in sufficiently large ROA is not possible, other control strategies, such as switching, and time-varying controllers can be employed to overcome this limitation [156]

In this chapter, I focus on designing an effective switching controller, which has proven successful in addressing stability challenges in various nonholonomic systems [89], [119], [157]. By applying a switching controller, I aim to extend this approach to the Acrobot-like model, addressing the gait termination problem in point-feet bipeds previously tackled in Chapter 3. Unlike computationally intensive optimization-based methods for enforcing

nonholonomic contact point constraints, my theoretical approach employs a switching method to stabilize the upright equilibrium point of the Acrobot-like model while utilizing the full Lagrangian model of the robot.

I use Nagumo's theorem in conjunction with an existing analytical controller for the Acrobot [120] to enforce constraints and achieve asymptotic stability for the overall system. While my approach primarily focuses on computational efficiency, it strategically resorts to more computationally intensive methods only when necessary to maintain stability. One of the primary advantages of employing switching controllers is their ability to facilitate analytical proofs of stability for the closed-loop system. By leveraging the well-established stability properties of a feedback linearization-based controller for the Acrobot, I will analytically demonstrate the stability of the resulting switching controller. My simulation results further illustrate that this approach effectively satisfies contact point constraints while reducing computational demands, thereby ensuring robust and efficient performance.

As demonstrated in Chapter 3, the challenge of stabilizing a unique equilibrium solution of (24) in the presence of contact point constraints (25) in a sufficiently large ROA remains elusive when relying solely on time-invariant smooth feedback. Given this constraint and the fact that (24)- (25) is controllable, I now transition to the core of this chapter: the design of the switching control strategy to effectively stabilize the upright equilibrium x^e [156].

My methodology involves the development of a hybrid control system that combines continuous and discrete control actions through a switching mechanism. This controller is designed to switch between different sub controllers—each tailored to specific operational conditions of the robot. By leveraging the existing controllers developed for the Acrobot,

I can use this well-established knowledge to stabilize the upright equilibrium point effectively, avoiding the need for parameter tuning or issues related to the ill conditioning often associated with numerical solutions [65]. This approach ensures that I capitalize on proven control strategies while addressing the unique challenges posed by the nonholonomic constraints in my system [120], [121].

My approach involves systematically switching between two control systems:

I. **Balancing controller**: Directs the robot trajectories towards the equilibrium point.

II. **GRF controller**: Two distinct GRF controllers are used to ensure the nonholonomic constraint (25) is not violated, one based on Nagumo's theorem and the other on DDP.

To illustrate how the balancing and GRF controllers operate in coordination, assume we have a simplified two-dimensional system with state variables (x_1, x_2), where the equilibrium is located at the origin. Suppose this system is subject to contact point constraints, and the goal is to steer the system from an initial state x_{init} to the equilibrium while strictly respecting these constraints.

Starting from an initial state $x_{initial}$, the system is first driven by the balancing controller, represented by the green dashed line in Figure 5.1. This controller reduces a Lyapunov function and drives the state toward equilibrium, as long as the constraints remain satisfied. When the trajectory approaches a region where contact constraints are close to being violated, control switches to the Nagumo-based GRF controller, which analytically enforces admissible directions of motion. In Figure 5.1, one such direction is shown by the blue arrow; however, these directions are recalculated dynamically at every state.

If the system enters a region where analytical constraint enforcement is no longer possible, typically due to stronger nonlinearities or tighter constraints, the strategy switches to the DDP-based GRF controller, shown by the red dashed line. This controller uses DDP algorithm to numerically compute an optimal trajectory that advances the states closer to the equilibrium point while respecting contact point constraints. To ensure computational efficiency, the terminal state of the DDP algorithm is determined by the balancing controller.

Once the system returns to a constraint-satisfying region, the balancing controller resumes control and continues driving the system toward the equilibrium. A key result established in this work is that the zero dynamics manifold, shaded region in Figure 5.1, lies entirely within the constraint-satisfying set. Once the system reaches this manifold, contact point constraints are inherently satisfied, and no further switching is required. The balancing controller can remain active indefinitely and complete the convergence to the upright equilibrium.

5.2 Asymptotic Stabilization of the Equilibrium with State-Dependent Switching Feedback

This section presents the design of the balancing controller and the Nagumo-based GRF controller, with a focus on how they contribute to system stability and constraint enforcement. The design of the DDP-based GRF controller will be detailed separately in Section 5.3.

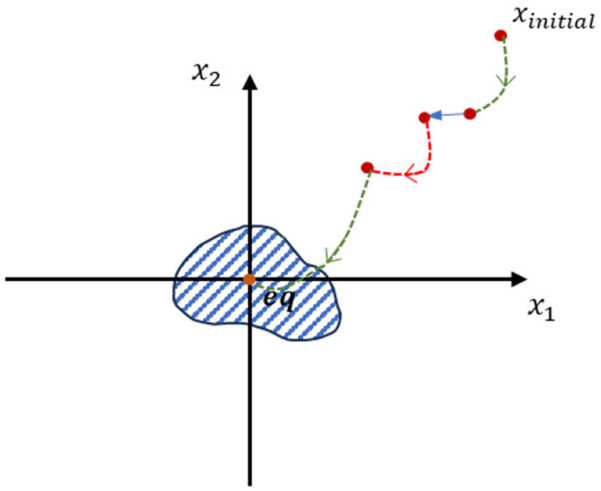


Figure 5.1. Illustration of how the two sub-controllers operate on a system with two states. Green dashed line, blue arrow, and red dashed line represent the balancing controller, Nagumo-based GRF controller, and DDP-based GRF controller, respectively. The shaded region denotes the zero dynamics manifold of the balancing controller.

5.2.1 Analysis of the Balancing Controller

For the challenge of stabilizing (24) without the contact point constraint (25) at the upright position, the scientific community has put forth a multitude of approaches, reflecting the breadth and depth of control theory's potential solutions. These range from pseudo-linearization techniques [140], [141], to methods of quadratic stabilization discussed in [154]. In light of this variety, my analysis will focus particularly on the methodology proposed in [120], which has garnered attention for its robustness in achieving stabilization at the upright position. This choice is motivated by the proven effectiveness of the partial feedback linearization approach, which offers a compelling balance between complexity and performance [156].

This methodology is grounded on the formulation of a set of outputs defined as

$$y(x) = \begin{bmatrix} h_1(x) \\ \vdots \\ h_{n-1}(x) \end{bmatrix}, \tag{68}$$

with a vector relative degree of $(3,3,\dots,3)$. These outputs are judiciously chosen so that by driving them to zero, the system exhibits the desired behavior. In applying the input-output linearization method, I differentiate the output $y(x)$ three times, allowing the control input to explicitly appear as follows

$$\dddot{y} = L_f^3 y(x) + L_g L_f^2 y(x) u, \tag{69}$$

where $L_g L_f^2 y(x)$ is the invertible decoupling matrix. The application of the control input of the form

$$K_B = -\left(L_g L_f^2 y(x)\right)^{-1} \left(L_f^3 y(x) + K_0 y(x) + K_1 \dot{y}(x) + K_2 \ddot{y}(x)\right), \tag{70}$$

results in a linear system on the outputs

$$K_0 y(x) + K_1 \dot{y}(x) + K_2 \ddot{y}(x) + \dddot{y}(x) = 0. \tag{71}$$

Here, constant matrices K_0, K_1 and K_2 are chosen so that $y(x)$ converges to zero [28].

For later developments, a PD quadratic Lyapunov function for (24) with the control signal (70) is defined as

$$V = \eta^T(x) P \eta(x), \tag{72}$$

where $\eta(x) = \begin{bmatrix} y(x) \\ \dot{y}(x) \\ \ddot{y}(x) \end{bmatrix}$, and P is a PD matrix. It is noted that the specific definition of outputs gives rise to a one-dimensional zero dynamics manifold, which can be described as

$$Z = \{x \in \mathcal{X} | y(x) = 0, \dot{y}(x) = 0, \ddot{y}(x) = 0\}. \tag{73}$$

It is important to note that $V = 0$ indicates that the system's states reside on the zero dynamics manifold Z. I encourage readers seeking a profound understanding of this strategy to refer to the detailed discourse presented in [120].

5.2.2 Development and Analysis of the Nagumo-Based GRF Controller

A key aspect of my contribution to the current research is the development of Nagumo's based GRF controller, an essential component in my devised switching control strategy. This tool has been designed to uphold the nonholonomic constraint detailed in (25). Crafted as a novel element, this GRF controller operates synergistically with the balancing controller, presenting a solution capable of addressing the intricate challenge of steering the bipedal robot to an upright equilibrium in the presence of contact point constraints. In what follows, I provide an examination of the GRF controller starting with an overview of some preliminary results.

Let $\phi(t, x^0)$ be the solution of (24) for the initial condition x^0 and a feedback controller of the form $u = \xi(x)$. Define the set S as

$$S \coloneqq \{x \in \mathcal{X} | (x, \xi(x)) \in \tilde{Q}\}. \tag{74}$$

I then present the following definition:

Definition 5. (*Positive Invariance*) *The set S is said to be positively invariant if*

$$x^0 \in S \to \phi(t, x^0) \in S \; \forall t > 0. \tag{75}$$

Note that for the contact point constraint (25) *to be satisfied during the entire motion, it suffices to render S positively invariant. This problem is addressed by defining the tangent cone and applying Nagumo's theorem, both of which I now describe.*

Definition 6. (*Tangent Cone*) *Given the set S, the tangent cone at each point x in S is defined as*

$$T_S(x) = \left\{ z : \liminf_{\tau \to 0} \frac{dist(x + \tau z, S)}{\tau} = 0 \right\}. \tag{76}$$

Using (76) to calculate $T_S(x)$ is not straightforward. Hence, a more tractable approach is presented as follows.

Remark 2. Given two vectors v_1, v_2, the inner product of these two vectors is

$$\langle v_1, v_2 \rangle = \|v_1\| \|v_2\| cos\theta, \tag{77}$$

where θ is the angle between two vectors. For vectors whose θ is between -90 and 90, the inner product will be a positive value. This fact can be used to determine $T_S(x)$ at each point $x \in \mathcal{X}$

$$T_S(x) \coloneqq \left\{ \zeta \in \mathbb{R}^{2n} | \langle \nabla S, \zeta \rangle > 0, \|\zeta\| \in \left[0, \max_h \left(S(x + h\zeta) \in \tilde{Q} \right) \right] \right\}, \tag{78}$$

where ∇ denotes the gradient operator. Notice that for $x \in int\{S\}, T_S(x) = \mathbb{R}^{2n}$. Therefore, $T_S(x)$ is meaningful only for $x \in \partial S$, where ∂S denotes the boundary of S..

Theorem 3. (*Nagumo's theorem*) [158]: *Consider the system given by* (24) *with the desired control action* $u = K_G(x)$. *Then* S *is positively invariant for* (24) *if the following condition holds*

$$f(x) + g(x)K_G(x) \in T_S(x), \forall x \in \partial S. \tag{79}$$

In what follows, I will demonstrate how $K_{\mathrm{G}}(x)$ from (79) can work in conjunction with the balancing controller K_B to stabilize the upright equilibrium x^e without violating the contact point constraints specified in (25).

5.2.3 Stabilization of the Acrobot Using the Switching Controller

To establish the conditions for switching between the GRF and balancing controllers that lead to the asymptotic stability of (24)-(25), I need to analyze the stability of the system, as not every switch between two controllers guarantees stability. In fact, as illustrated in example 2 of [87], even switching between subsystems that are individually asymptotically stable does not necessarily result in overall system stability. Consequently, the stability analysis of switching systems remains a complex and challenging problem that has been the subject of several research [89], [159]. One common approach is the dwell time method, where stability is ensured by defining a minimum amount of time that the system must remain in a particular subsystem before switching to another [160]. However, this method can be overly restrictive when the purpose of switching is to choose the subsystem that best meets certain performance criteria. In my case, I aim to use switching to prevent the system from violating contact point constraints, making the dwell time approach less applicable.

An alternative strategy to ensure system stability involves identifying a single Common Lyapunov Function (CLF) that meets stability conditions across various subsystems. A

more nuanced approach is the Multiple Lyapunov Function (MLF) method, where a distinct Lyapunov function is constructed for each subsystem [92]. This method guarantees stability as long as the set of these functions satisfies a certain condition [161]. However, these strategies require each subsystem to have a stable and well-defined Lyapunov function, a requirement not met by the GRF controller.

To implement the switching control strategy, it is essential to understand the dynamics of the underlying system. Before delving into the main theorem on the asymptotic stability of (24)-(25), which employs the principles of the CLF and MLF methods via the Lyapunov function from the balancing controller, I must address a foundational aspect of the system dynamics. Specifically, it is essential to analyze the effects of nonholonomic contact point constraints on the zero dynamics manifold, Z. This analysis is pivotal as it demonstrates that when the Lyapunov function V equals zero—indicative of states residing on the zero dynamics manifold—the contact point constraints are inherently satisfied. This fact establishes a crucial premise that underpins the conditions for asymptotic stability in Theorem 4.

To confirm that contact point constraints are inherently satisfied, I verify that the projection of Z onto (F^T, F^N) plane does not intersect the boundaries of (25) given by

$$\left(\frac{F^T}{F^N}\right)^2 \leq \mu^2 \,. \tag{80}$$

I further adopt the realistic assumption that $F^N \leq F^N_{max}$, acknowledging the physical limitation that actuators cannot generate infinite force. Figure 5.2 illustrates the triangle specified by (80) in the (F^T, F^N) plane with $F^N \leq F^N_{max}$ and $F^N \geq 0$. Note that

$$\left(F^T(x^e), F^N(x^e)\right) = (0, mg), \tag{81}$$

where m represents the total mass of the robot, defined as the sum of the masses of all n links. This corresponds to the upright equilibrium point, where the tangential force F^T is zero, and the normal force F^N is equal to the gravitational force acting on the total mass of the robot.

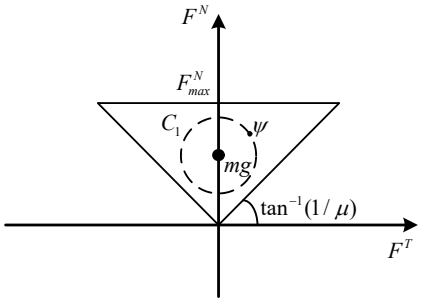


Figure 5.2. Triangle specified by (80) with $F^N \leq F^N_{max}$ and $F^N \geq 0$ (solid line), C_1 circle inside the triangle (dashed line).

Lemma 4. *Let the map of Z in (F^T, F^N) plane given by the set*

$$\Pi := \left\{\left(F^T(x), F^N(x)\right) | x \in Z\right\}. \tag{82}$$

Furthermore, define C_1 as the circle centered at $(0, mg)$ with radius $\hat{R}$, constructed such that it does not intersect with the triangle, and denote the set of all points inside C_1 by

$$\tilde{C_1} := \left\{(F^T, F^N) | (F^T)^2 + (F^N - mg)^2 \leq \hat{R}^2\right\}. \tag{83}$$

Assuming the minimum value of V denoted by ψ on C_1 is nonzero, the following conditions are satisfied:

$$\Pi \cap \partial\tilde{C_1} = \emptyset, \tag{84}$$

$$\Pi \subset \tilde{C}_1. \tag{85}$$

Proof:

To establish (84) by contradiction, suppose that $\Pi \cap \partial\tilde{C}_1 = \left(F^T(x^*), F^N(x^*)\right)$. This scenario is illustrated in Figure 5.3. Given that x^* is on the circle, by definition, $V(x^*) \neq 0$. However, since $x^* \in \Pi$, this implies $V(x^*) = 0$, leading to contradiction and thereby concluding that $\Pi \cap \partial\tilde{C}_1 = \emptyset$.

To prove (85), note that $F^T(x)$ and $F^N(x)$ are continuous functions, and Π is a connected set. Therefore, from the fact that $(0, mg) \in \Pi$, it follows that $\Pi \subset \tilde{C}_1$ ■

Having established in Lemma 4 that states on the zero dynamics manifold inherently satisfy the contact point constraints, I am now ready to introduce the main stability theorem.

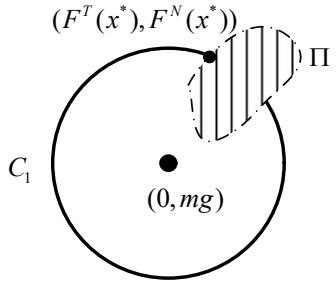


Figure 5.3. Set Π intersects $\partial\tilde{C}_1$ at $\left(F^T(x^*), F^N(x^*)\right)$.

Theorem 4. *Let V in* (72) *serve as the Lyapunov function for the balancing controller K_B. Consider t_{2i}, and t_{2i+1}, $i = 0,1, \ldots$ as the activation instances for the balancing and GRF controllers respectively. Assume*

I. The duration that the GRF controller remains active does not exceed a sufficiently small ΔT, specifically

$$t_{2i} - t_{2i-1} \leq \Delta T, \quad i = 1,2, \ldots \tag{86}$$

II. The Lyapunov function V decreases for $t \in [t_{2i}, t_{2i+1}), i = 0,1, \ldots$, and may increase or decrease for $t \in [t_{2i-1}, t_{2i}), i = 1,2, \ldots$.

Under these conditions, the control law is given by

$$u = \begin{cases} K_B & if \quad x \in int\{S\}, \\ K_G & if \quad x \in \partial S, \end{cases} \tag{87}$$

renders (24)-(25) *asymptotically stable, provided that*

$$V(t_{2i+2}) - V(t_{2i}) < 0, i = 0,1, \ldots \tag{88}$$

Proof:

I initiate the proof by visually illustrating the condition of the theorem through the evolution of the Lyapunov function V as depicted in Figure 5.4. To establish stability in the sense of Lyapunov, it is noted that while V is continuous, it is only piecewise differentiable due to switching events. The system dynamics $f(x)$ in (24) is Lipschitz continuous. As a result, the switching sequence depicted in Figure 5.4 satisfies the global Lipschitz condition

$$\max_{t \geq 0} \{\dot{V}(t)\} \leq \varphi, \tag{89}$$

where $\varphi = \max\{\phi_i\}$, and ϕ_i represents the maximum absolute value of $\dot{V}(t)$ within differentiable intervals $[t_{2i-1}, t_{2i})$. It then follows that

$$V(t) \leq V(t_{2i-1}) + \varphi \Delta T, \tag{90}$$

for $t \in [t_{2i-1}, t_{2i})\ i = 1,2, \ldots$. Furthermore, it is observed that V is decreasing for $t \in [t_{2i}, t_{2i+1}), i = 0,1, \ldots$, and

$$V(t_1) \geq V(t_3) \geq \cdots \geq V(t_{2i-1}), i = 1,2, \ldots \tag{91}$$

This observation supports the conclusion that $x(t)$ consistently remains within the bounded level curves of V given by $V(t_{2i-1}) + \varphi\Delta T, i = 1,2, \ldots$, over the time interval $t \in [0, \infty)$. Thus, I assert that the system is stable in the sense of Lyapunov.

To demonstrate asymptotic stability, it is noted that since V is decreasing for $t \in [t_{2i}, t_{2i+1}), i = 0,1, \ldots$, it follows that

$$V(t_{2i+1}) \leq V(t_{2i}),\ \ i = 0,1, \ldots \tag{92}$$

Considering that (88) and (92) are decreasing positive sequences, for sufficiently large i, they converge to some limit $\varrho \geq 0$

$$\lim_{i \to \infty} V(t_{2i}) = \lim_{i \to \infty} V(t_{2i+1}) = \varrho. \tag{93}$$

This condition, in turn, implies that for sufficiently large i, $\dot{V}$ approaches zero. However, the condition $\dot{V} = 0$ holds true only if $V = 0$. Therefore, the only possible value for ϱ as i goes to infinity is zero. As V approaches zero for large i, system states converge to the zero dynamics manifold Z. According to Lemma 4, contact point constraints are inherently satisfied on Z, thus eliminating the need for switching actions. Consequently, the balancing controller remains continuously active, guiding the system states toward the equilibrium point■

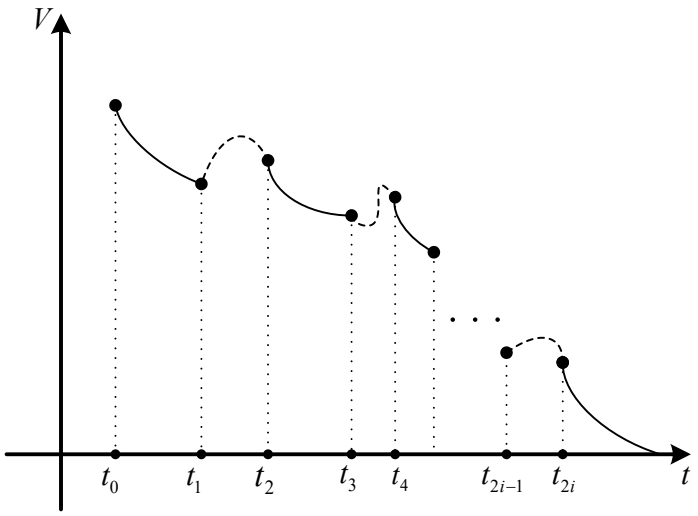


Figure 5.4. Evolution of the Lyapunov function V when K_B (solid line) and K_G (dashed line) are active to satisfy condition (88).

Remark 3. It is crucial to understand that the smoothness of the vector fields $f(x)$ and contact point constraints ensures that the switching actions terminate even before the system reaches the zero dynamics manifold, i.e., when $V > 0$.

5.3 Designing the Switching Strategy in the Case of Quadratic Lyapunov Functions

In the previous section, I introduced the balancing and GRF controllers and, through Theorem 4, specified the condition necessary to ensure the asymptotic stability of (24)-(25) by strategically switching between these controllers. As a critical component of Theorem 4, achieving asymptotic stability requires satisfying the condition outlined in (88). This section focuses on meeting this requirement for the quadratic Lyapunov function described in (72) by carefully selecting the control gain K_G in (79). I will develop a strategy that considers ΔT, the maximum duration the system can remain under the control of the GRF controller, as a general case to cover all scenarios up to this worst-case scenario.

To begin my development, let me consider the derivative of the Lyapunov function V to be of the form

$$\dot{V} = -\eta(t)^T Q\eta(t), \tag{94}$$

where Q is a PD matrix, presumably calculated from the Lyapunov equation [162]. From the Rayleigh-Ritz inequality, it follows that

$$\lambda_{min}(P)\|\eta(t_{2i})\|^2 \leq V(t_{2i}) \leq \lambda_{max}(P)\|\eta(t_{2i})\|^2, \tag{95}$$

where, $i = 0,1, \ldots,$ $\lambda_{min}(P)$ and $\lambda_{max}(P)$ represent the minimum and maximum eigenvalues of P, respectively. Also, from Grönwall's inequality

$$\dot{V}(t) \leq -\frac{\lambda_{min}(Q)}{\lambda_{max}(P)} V(t), \tag{96}$$

which leads to

$$e^{\frac{\lambda_{min}(Q)}{\lambda_{max}(P)}(t_{2i+1}-t_{2i})} V(t_{2i+1}) \leq V(t_{2i}). \tag{97}$$

By assuming $\omega = e^{\frac{\lambda_{min}(Q)}{\lambda_{max}(P)}(t_{2i+1}-t_{2i})}$ and employing the Rayleigh-Ritz inequality again, I find that

$$\omega\lambda_{min}(P)\|\eta(t_{2i+1})\|^2 \leq \omega V(t_{2i+1}) \leq \omega\lambda_{max}(P)\|\eta(t_{2i+1})\|^2, \tag{98}$$

$$\lambda_{min}(P)\|\eta(t_{2i+1} + \Delta T)\|^2 \leq V(t_{2i+1} + \Delta T) \leq \lambda_{max}(P)\|\eta(t_{2i+1} + \Delta T)\|^2, \tag{99}$$

where $t_{2i+2} = t_{2i+1} + \Delta T$. Then, if I have

$$\lambda_{max}(P)\|\eta(t_{2i+1} + \Delta T)\|^2 \leq \omega\lambda_{min}(P)\|\eta(t_{2i+1})\|^2, \tag{100}$$

or

$$\left\|\frac{\eta(t_{2i+1}+\Delta T)}{\eta(t_{2i+1})}\right\| \leq \sqrt{\frac{\omega\lambda_{\min}(P)}{\lambda_{\max}(P)}}, \tag{101}$$

based on equation (96), it follows that

$$V(t_{2i+1}+\Delta T) \leq \omega V(t_{2i+1}) \leq V(t_{2i}), \tag{102}$$

which is (88). I then examine which ζ vectors in (78) satisfy inequality (101). To simplify the analysis, I assume that for small ΔT, $\dot{x}$ remains constant

$$\dot{x} = \zeta \in T_S(x) \quad \forall t \in [t_{2i+1}, t_{2i+1}+\Delta T). \tag{103}$$

Assuming a small ΔT allows me to estimate $\eta(t_{2i+1}+\Delta T)$ as

$$\eta(t_{2i+1}+\Delta T) \approx \eta(t_{2i+1}) + \dot{\eta}\Delta T, \tag{104}$$

which can be substituted into (101) to obtain

$$\|\eta(t_{2i+1}) + \dot{\eta}\Delta T\| \leq \sqrt{\frac{\omega\lambda_{\min}(P)}{\lambda_{\max}(P)}}\|\eta(t_{2i+1})\|. \tag{105}$$

Given that $\dot{\eta} = \frac{\partial\eta}{\partial x}\dot{x}$, $\dot{x} = \zeta$, the above inequality can be used to calculate ζ

$$\left\|\eta(t_{2i+1}) + \frac{\partial\eta}{\partial x}\zeta\Delta T\right\| \leq \sqrt{\frac{\omega\lambda_{\min}(P)}{\lambda_{\max}(P)}}\|\eta(t_{2i+1})\|. \tag{106}$$

However, to further simplify the calculations, consider the following inequality

$$\|\eta(t_{2i+1}) + \dot{\eta}\Delta T\| \leq \|\eta(t_{2i+1})\| + \|\dot{\eta}\|\Delta T, \tag{107}$$

which simply states that the change in η over a small time ΔT is bounded by its initial value plus its rate of change over ΔT. If I now impose the condition

$$\|\eta(t_{2i+1})\| + \|\dot{\eta}\|\Delta T \leq \gamma\|\eta(t_{2i+1})\|, \tag{108}$$

where $\gamma = \sqrt{\frac{\omega\lambda_{\min}(P)}{\lambda_{\max}(P)}}$, then (105) holds true. From (108), it directly follows that

$$\|\dot{\eta}\| \leq \left(\frac{\gamma - 1}{\Delta T}\right)\|\eta(t_{2i+1})\|. \tag{109}$$

Similarly, use $\dot{\eta} = \frac{\partial \eta}{\partial x}\dot{x}$, $\dot{x} = \zeta$ to arrive at

$$\left\|\frac{\partial \eta}{\partial x}\zeta\right\| \leq \left(\frac{\gamma - 1}{\Delta T}\right)\|\eta(t_{2i+1})\|, \tag{110}$$

which provides the criterion for selecting ζ vectors from $T_S(x)$ such that (101) is satisfied. To complete the calculation, I need to determine K_G from (110). Given the smoothness of the functions $f(x)$, $g(x)$ and $\eta(t)$ in their respective regions of definition, and the assumption that they can be approximated by constants over sufficiently small intervals, a constant K_G can be chosen such that

$$\left\|\frac{\partial \eta}{\partial x}(f + gK_G)\right\| \leq \left(\frac{\gamma - 1}{\Delta T}\right)\|\eta\|. \tag{111}$$

Note that as long as the ζ vector, determined from (110), resides within the tangent cone T_s, (111) ensures that condition (88) is consistently satisfied at each switching event. This compliance is essential for maintaining the closed-loop system asymptotic stability.

Control strategy when ζ lies outside the tangent cone T_S: In some cases, the vector ζ determined from (110) may not lie within the tangent cone $T_s(x)$, meaning there is no guarantee that at all switching times, $T_s(x)$ will contain the direction in which the Lyapunov function decreases. This condition prevents me from using (111) to determine

control input at certain times. To address this challenge, I propose an effective approach that maintains a low computational burden.

Although it is not possible to decrease Lyapunov function at certain switching times, I can still leverage the information provided by the balancing controller. Assume that the vector ζ does not lie within the tangent cone at time $t = t_D$, then for some $\Delta t \ll 1$, the state given by

$$x_f \approx x(t_D) + [f(t_D) + g(t_D)K_B(t_D)]\Delta t, \tag{112}$$

indeed, corresponds to a state where the Lyapunov function V has a smaller value. Therefore, it is only necessary to steer system states to this point using methods that can incorporate contact point constraints. In this work, I use the DDP algorithm, which is particularly well-suited for nonlinear systems due to its iterative nature and its capability to handle complex dynamics with both state and control constraints [145]. Given that (24)-(25) is controllable, DDP can consistently provide a solution. This is because controllability guarantees the existence of a path between any two points in the state space, even though this path may not always be smooth [130].

In this method, first, the continuous dynamic equation (24) is discretized as follows

$$x_{k+1} = f_d(x_k, u_k), \tag{113}$$

where $x_k \in \mathbb{R}^{2n}$ represents the state of the system at the discrete time step k, and $u_k \in \mathbb{R}^{n-1}$ is the corresponding control input. Then a cost function is defined

$$J = \Phi(x_N) + \sum_{k=0}^{N-1} L(x_k, u_k), \tag{114}$$

where N is the control horizon, $\Phi(x_N)$ is the terminal cost, and $L(x_k, u_k)$ is the running cost. DDP minimizes J by performing a series of forward and backward passes until it reaches a locally optimal control policy [65]. This algorithm has been extended in the literature to handle inequality constraints both on control and state variables [62], [64], [71]. To apply DDP to my problem, I interpret contact point constraints as a constraint solely on the control input u, for each state $x \in X$

$$U(x) \coloneqq \left\{u \in \mathbb{R}^{n-1} \middle| (x, u) \in \tilde{Q}\right\}. \tag{115}$$

Within this framework, the control signal K_D generated by the DDP must belong to U, i.e., $K_D \in U$. Consequently, the DDP variant I employ is designed to efficiently manage these control input constraints. To this end, I employ the DDP algorithm introduced in Chapter 4. This new DDP method dynamically adjusts the control input boundaries during the optimization process, ensuring rapid convergence and effective constraint management. This method ensures that the system transitions smoothly to the desired state x_f, and if necessary, temporarily allows an increase in the Lyapunov function V. By leveraging the fast convergence properties of DDP, my approach provides a local and efficient numerical solution, reducing the need for extensive fine-tuning of parameters.

I define Δt_D as the duration for which the DDP is active. After this interval, it is ensured that

$$V(t_D + \Delta t_D) < V(t_D). \tag{116}$$

The switching strategy proposed in this chapter, which dynamically selects between K_B, K_G, and K_D is detailed in Algorithm 1. Figure 5.5 illustrates the evolution of the Lyapunov

function V through this switching strategy until it converges to zero. The combination of these three controllers not only ensures the asymptotic stability of (24)-(25), but also minimizes the use of numerical methods and employs an efficient numerical method to reduce the computational burden. In practice, the execution of Algorithm 1—including constraint checks, gain selection, and potential DDP computation—can introduce delays in real-time applications. During this period, a low-level PID controller that directly controls the actuators is used to control joints. Although this low-level control layer is not explicitly modeled in this thesis, it represents an important practical consideration for deploying this strategy on real robotic hardware.

It is worth mentioning that the use of DDP in my method offers several advantages. Given that at each step, the target state x_N is in close proximity to the initial point, the convergence of the DDP algorithm is consistently fast. This proximity allows the system to reach a stable state without needing to solve all the way to the equilibrium point x^e.

Algorithm 1. The proposed switching method to stabilize (24)-(25).

while $x \neq x^e$ **do**
set $u = K_B$
if (25) is not satisfied
find T_s from (78)
find K_G from (111)
if K_G satisfies (79)
set $u = K_G$
else
find x_f from (112)
set $x_N = x_f$ in (114)
use the proposed DDP with J from (114) and find K_D considering (115)
set $u = K_D$
end if
end if
end while

Additionally, the robustness of the proposed DDP in handling contact point constraints means that extensive fine-tuning of parameters may not be necessary, further simplifying the implementation process. I will demonstrate these advantages with detailed simulations in the following section, showcasing the practical effectiveness of my proposed method.

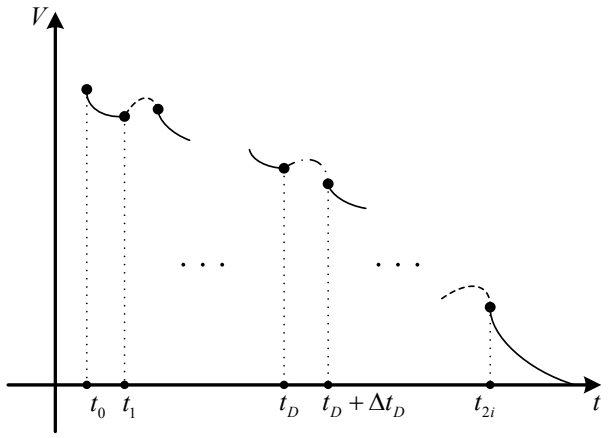


Figure 5.5. Evolution of the Lyapunov function V when K_B (solid line), K_G (dashed line), and K_D (dash-dot line) are active, including scenarios where ζ does not reside within the tangent cone T_S.

5.4 Simulation Results

In this section, I apply the proposed switching strategy outlined in Algorithm 1, providing an overview of its implementation. I then present and analyze the simulation results, offering a detailed discussion of the system's performance.

I focus on the 5-link model, which is commonly used to represent a simplified version of a walking human or humanoid robot in the sagittal plane. It is assumed that the framework outlined in Chapter 3 has been applied to bring the biped into the double support phase, allowing me to model it as an Acrobot-like robot with nonzero initial conditions, illustrated in Figure 3.1. The parameters defining this robot are enumerated in Table 3.1, chosen to

mirror the characteristics of RABBIT [7] with legs held together [120]. The acceleration of gravity is taken as 9.8 m/s^2 and $\mu = 0.3$. The simulations were conducted on a system equipped with an Intel i9-14900K CPU.

The objective is to navigate the robot to the designated upright equilibrium point, represented by $(q_0, q_1, q_2) = (\pi/2,0,0)$starting from the initial configuration $(q_0^0, q_1^0, q_2^0) = (1.39,0.52,-0.69)$, and initial angular velocities $(\dot{q}_0^0, \dot{q}_1^0, \dot{q}_1^0) = (0.5,-0.8,-1.2)$, with all measurements in radians. To account for actuator physical constraints, I model them using a first-order transfer function, given by

$$u_{filtered} = \frac{30}{s+30}u, \tag{117}$$

which indicates that the actuators exhibit a low-pass behavior. I use $u_{filtered}$ in my simulations to accurately reflect these dynamics.

For the balancing controller $K_B(x)$ in (70), parameters $K_0 = \begin{bmatrix} 120 & 0 \\ 0 & 120 \end{bmatrix}, K_1 = \begin{bmatrix} 74 & 0 \\ 0 & 74 \end{bmatrix}, K_2 = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}$, and $K = 6$, defined in example *C* of [120], are considered. To specify a PD matrix P for the Lyapunov function (72), consider the PD matrix $Q = I_6$, where I_6 denotes the identity matrix, and solve the Lyapunov equation to obtain

$$P = \begin{bmatrix} 0.43 & 0 & -0.5 & 0 & -1.06 & 0 \\ 0 & 0.43 & 0 & -0.5 & 0 & -1.06 \\ -0.5 & 0 & 1.06 & 0 & -0.5 & 0 \\ 0 & -0.5 & 0 & 1.06 & 0 & -0.5 \\ -1.06 & 0 & -0.5 & 0 & 10.98 & 0 \\ 0 & -1.06 & 0 & -0.5 & 0 & 10.98 \end{bmatrix}. \tag{118}$$

Note that the existence of a solution to the Lyapunov equation is guaranteed due to the stability of the system.

To check the condition of Lemma 4, the C_1 circle with $\hat{R} = 84.67$ centered at $(0,313.6)$ in (F^T, F^N) plane is chosen such that it does not intersect the triangle specified by (80), and $F^N_{max} = 700\ N$. The minimum value of V on the C_1, ψ, can be determined by solving the Lagrangian equation

$$\bar{L} = V + \lambda\left[(F^T)^2 + (F^N - mg)^2 - \hat{R}^2\right], \tag{119}$$

where λ is the Lagrange multiplier. I calculate $\psi = 0.57$, thereby confirming that on the resulting zero dynamics manifold Z, the contact point constraint (25) is satisfied. Therefore, Algorithm 1 can be successfully applied to stabilize the system.

For the DDP algorithm, I let $N = 30$ and choose $\Phi(x_N)$ and $L(x_k, u_k)$ in (114) as follows

$$\Phi(x_f) = \sum_{i=1}^{6} c_f^i \left(\sqrt{\left(x_N^i - x_f^i\right)^2 + {p_f^i}^2} - p_f^i\right), \tag{120}$$

where,

$$\begin{aligned} c_f &= [0.5,1,1.25,1.5,0.75,2.5], \\ p_f &= 0.01 \times [1,1,1,1,1,1], \end{aligned} \tag{121}$$

and

$$L(x_k) = \sum_{k=0}^{N-1} \sum_{i=1}^{6} c_x^i \left(\sqrt{\left(x_k^i - x_f^i\right)^2 + {p_x^i}^2} - p_x^i\right), \tag{122}$$

where

$$c_x = 0.1 \times [0.5,0.1,0.2,0.3,0.1,0.4], \tag{123}$$

$$p_x = 0.1 \times [1,1,1,1,1,1].$$

Note that i superscript shows the ith element of the vector. I did not specify any constraint on u. I keep J the same for the entire motion.

Figure 5.6 shows a stick animation of the biped reaching the upright equilibrium; a video demonstration is also provided at [163]. This figure was included to better represent the motion of the robot until the equilibrium point, which appears to be natural. Figure 5.7 demonstrates the convergence of joint angles and velocities during the stabilization process. The figure shows that the joints converge to their desired values and all signals remain smooth and bounded.

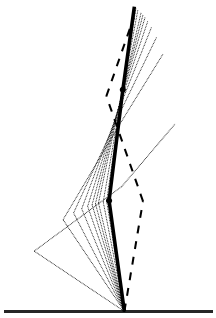


Figure 5.6. Stick animation of the biped reaching the upright equilibrium. The final and initial configurations are represented by solid and dashed lines, respectively.

Figure 5.8 compares the contact point forces with and without the proposed switching controller and with the only use of K_B, showing that the proposed switching strategy successfully maintains stable contact between the robot and the ground by keeping the contact point forces inside the triangle and satisfying the constraint (25). Finally, Figure 5.9 illustrates the joint torque profiles during the stabilization process. The plot shows that the joint torques are bounded and converge to zero after reaching the equilibrium point.

During the interval $[0.1, 0.3]$ seconds, the GRF controllers were activated 11 times, with the K_G controller engaged 6 times and the K_D controller engaged 5 times. Table 5.1 presents the number of iterations required and the corresponding computation time (in seconds) for each activation of the K_D controller. Notably, the DDP algorithm consistently converged in fewer than 20 iterations, underscoring the effectiveness of selecting x_f from equation (112).

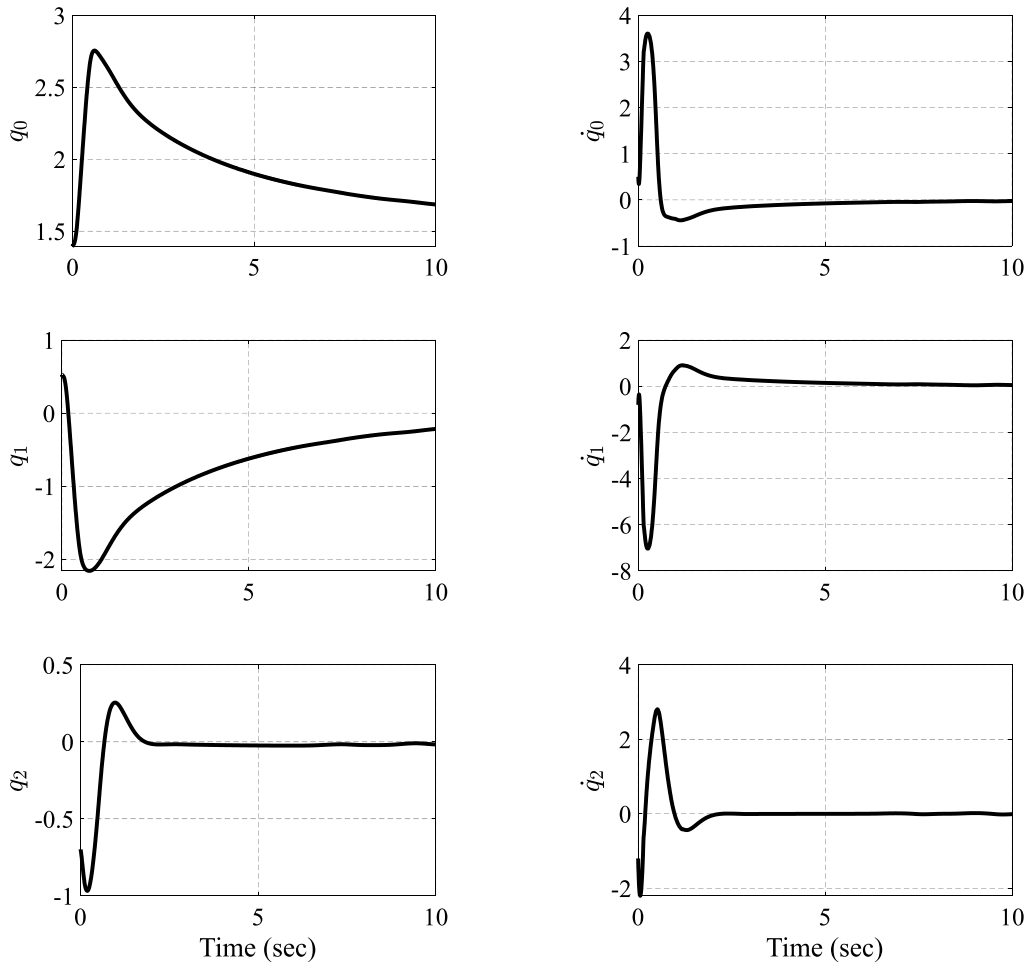


Figure 5.7. Stabilization to the upright equilibrium. The figure demonstrates the convergence of joint angles and velocities.

While the contact point constraint (25) was at risk of being violated early in the trajectory, the numerical method was employed only when necessary, ensuring minimal computational overhead. Beyond 0.3 seconds, the system has maintained stability without further switching, allowing the balancing controller K_B to govern the trajectory up to 10 seconds without additional computational intervention. It is worth noting that the majority of the running time in my DDP implementation is attributed to dynamically calculating the boundary of u at each step. This overhead could be significantly reduced with more efficient algorithms, further enhancing the method's overall performance.

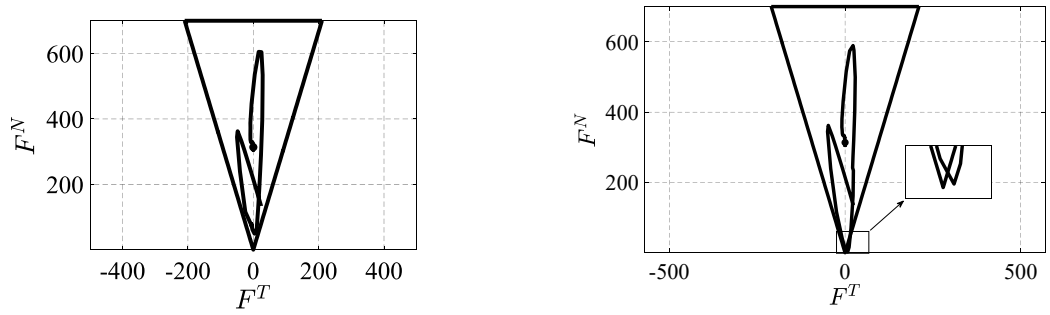


Figure 5.8. Comparison of contact point forces with (left) and without (right) the application of the switching method outlined in Algorithm 1. The plot illustrates that the proposed switching strategy effectively ensures that the contact point forces remain inside the triangle and satisfy the constraint (25).

To further assess the effectiveness of my proposed switching controller, I conducted a comparative analysis using only the DDP algorithm without the switching mechanism. In this scenario, the terminal state was set to $x_N = x^e$, and the algorithm's performance was evaluated. Notably, the DDP algorithm did not converge when the horizon was initially set to $N = 30$, which was previously effective when using my switching controller with $x_N = x_f$. To achieve convergence, it was necessary to extend the horizon significantly to $N = 800$. This adjustment resulted in a substantial increase in simulation time, extending to 72

seconds for a single iteration. This is considerably longer than the time required with my switching controller, underscoring the computational efficiency of my approach.

Table 5.1. Number of iterations and computation time for each activation of the K_D controller

Activation	1	2	3	4	5
Iteration	14	15	18	10	9
Time (s)	0.65	0.32	0.59	0.31	0.25

While I acknowledge that faster DDP algorithms can be achieved, as demonstrated in [82], where an MPC for whole-body dynamics using DDP was implemented on the TALOS robot at 60 Hz, it is important to note that their approach utilizes the Pinocchio software library for fast computations of dynamics and derivatives.

In contrast, my goal here is to provide a fair comparison between the two methods under similar conditions, without the additional optimizations that Pinocchio offers. This approach allows me to more directly compare the differences in computational efficiency and performance between the DDP algorithm alone and my proposed switching controller.

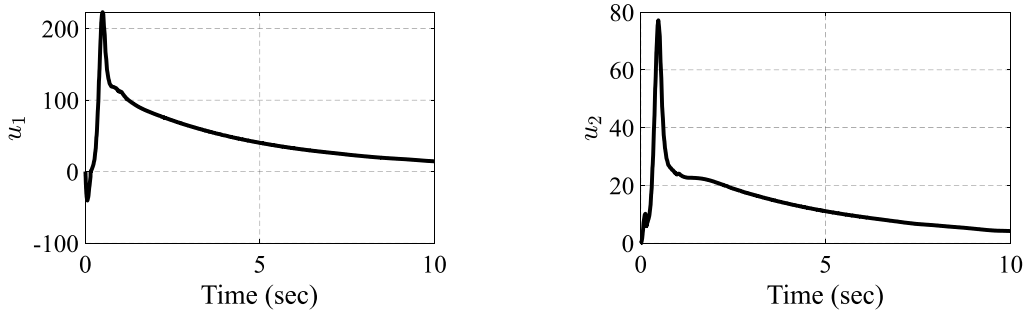


Figure 5.9. Joint torque (N.m) profiles during the stabilization.

5.5 Conclusion

It has been demonstrated that the gait termination problem in point-feet limit cycle walkers can be reduced to stabilizing an Acrobot-like model with contact point constraints at the

upright equilibrium. Building on this foundation, the current chapter addresses the challenge of achieving stable standing by introducing a novel switching control approach. My methodology combines switching controllers with Nagumo's theorem and existing analytical controllers for the Acrobot, ensuring the asymptotic stability of the closed-loop system, even in the presence of nonholonomic constraints. While I utilized the controller proposed by [120], it is important to note that other existing controllers could also be employed [135], [164], [165].

By employing switching controllers, I can leverage the inherent stability of existing controllers without the need for modification. This approach also reduces computational demands by activating more complex control algorithms only when necessary, optimizing resource usage. My strategy significantly lowers the computational burden compared to traditional optimization-based methods by adopting an "intervene when necessary" approach. This not only simplifies the control process but also allows me to analytically verify the stability of the system.

The findings from this study offer new possibilities for managing nonholonomic constraints in walking problems. My approach effectively handles the complex dynamics and constraints of bipedal locomotion, providing a robust and efficient framework that can be extended to other robotic systems. This work highlights the potential of using switching controllers to manage the intricate dynamics of bipedal robots, paving the way for further research and innovation in this field.

CHAPTER 6

6 Conclusion

6.1 Summary

The primary goal of this research was to address the gait termination problem in n-link, underactuated point-feet bipeds that utilize dynamic notion of stability, known as limit cycle walkers. Traditional control strategies designed for maintaining periodic stability are not suitable for gait termination, which requires transitioning the robot to a single, stable equilibrium point. To bridge this gap, I developed a unified gait termination solution by introducing a separate dynamic model, based on an Acrobot-like system, specifically designed for this purpose and distinct from models used for maintaining periodic stability. My findings indicate that continuous controllers are inadequate for stabilizing this dynamical model over sufficiently large ROA, as the nonholonomic contact point constraints violate the topological conditions required for smooth stabilization, as formalized by Sontag's theorem. However, the Acrobot-like model satisfies the LARC, demonstrating its controllability in all regions not fully blocked by constraints. Building on this foundation, I designed a switching control strategy to stabilize the upright equilibrium of point-feet bipeds, effectively managing the transition from dynamic walking to stationary standing while accommodating the challenges imposed by nonholonomic constraints.

The proposed switching strategy consists of a balancing controller to stabilize the upright equilibrium point of the Acrobot-like model and a GRF controller to enforce contact point

constraints. My GRF controller, developed using Nagumo's theorem and a fast-converging DDP algorithm, ensures that the computationally intensive algorithm is employed only when necessary, thereby minimizing the heavy computational demands typically associated with numerical methods such as MPC. This approach enhances the control strategy's effectiveness for real-time applications. Furthermore, I analytically proved the stability of the switching strategy under certain assumptions, which is crucial for ensuring reliable and predictable robot behavior during gait termination, especially in real-world scenarios where maintaining stability is critical.

Overall, this research presents a comprehensive framework that integrates both analytical and numerical methods, ensuring stable and efficient gait termination strategies for bipedal robots. It provides valuable insights and contributions to robotic control systems, advancing current methodologies and setting the stage for future innovations.

6.2 Innovation

- **Unified gait termination solution for point-feet bipeds utilizing limit cycle walking:**

  The major innovation of this work is the development of a unified gait termination framework for n-link underactuated point-feet bipeds, applicable regardless of the method used to generate periodic motions. This flexibility significantly improves the generalizability of the proposed gait termination strategy, making it suitable for a wide range of robotic platforms and systems.

- **Full-dimensional dynamic modeling:**

This research utilizes a full-dimensional dynamic model, which captures the complete dynamics of the robot, including all degrees of freedom and contact point constraints. This approach contrasts with simplified models such as LIP or SLIP. By employing the full dynamic model, I provide a more accurate and realistic representation of the robot's behavior, capturing the complexities of its interactions and constraints. As a result, the controller developed in this work is closer to real-world applications, offering more reliable performance in practical scenarios where simplified models may fail to account for the intricate dynamics of the system.

- **Demonstrating the limitations of continuous controllers for point-feet bipeds:**
  Another key contribution of this work is the demonstration that continuous controllers are insufficient for stabilizing a point-feet biped in the double-support phase due to nonholonomic contact point constraints. Using Sontag's condition, I show that these constraints introduce topological obstructions—such as non-contractible regions in the state space—that prevent smooth stabilization over sufficiently large ROA. This finding highlights a fundamental limitation of traditional continuous control methods and underscores the need for alternative approaches, such as switching controllers.
- **Lie algebra rank condition for controllability:**
  I have analytically demonstrated that point-feet biped in double support satisfies the LARC, proving that despite the presence of nonholonomic constraints, the system remains controllable. This contribution establishes a solid theoretical foundation for employing a switching controller to stabilize the system.
- **Switching controllers to enforce contact point constraints:**

To address the limitations of continuous controllers, I introduced the use of switching controllers to enforce contact point constraints. This method reduces the need for computationally expensive numerical controllers, while ensuring stability. I have also analytically proven the stability of the system under this switching control strategy, further demonstrating its effectiveness in handling the robot's complex dynamics.

- **Development of a fast-converging DDP algorithm:**

  A key innovation of this research is the development of a fast-converging DDP algorithm, specifically designed to efficiently handle inequality constraints. This algorithm utilizes control inputs to ensure a faster convergence, making real-time control more feasible. I have successfully applied this algorithm in managing contact point constraints during gait termination, ensuring stability and efficient performance.

6.3 Significance

This research significantly enhances the adaptability of bipedal robots by offering a unified solution for gait termination, applicable across different walking algorithms and robotic platforms. The use of a separate dynamical system for gait termination simplifies the control design process and extends the operational capabilities of robots, allowing them to perform complex maneuvers more safely. Moreover, the development of a novel DDP algorithm and switching control strategy improves the real-time performance and stability of bipedal robots, contributing valuable insights to robotics and control systems.

A key innovation of this work is the approach to enforcing contact point constraints using switching controllers, which not only ensures stability during gait termination but can also be employed in other critical areas such as push recovery in robots. This versatility highlights the broader applicability of my method beyond gait termination, offering a robust framework for enhancing robot resilience in various dynamic scenarios.

Additionally, this research addresses a critical gap in the literature by focusing on nonholonomic constraints in the context of gait termination, an area that has been largely underexplored. By advancing my understanding and control strategies for these constraints, this work lays the groundwork for future innovations in robotic mobility and stability.

6.4 Limitations

The current research is based on certain assumptions that simplify the control model. Specifically, I assumed that the employed controller maintains both swing and stance legs in identical configurations for $t > t_g$ and ensures an even distribution of the GRFs between the legs. While this assumption facilitates the computation of GRFs based on joint angles, velocities, and accelerations, it does not fully represent the complexities of real-world scenarios. Future work will focus on relaxing this assumption and designing stabilizing controllers for the standing model that respect contact point constraints while driving the robot to its equilibrium point.

Additionally, the use of the DDP algorithm introduces some computational overhead, which may limit its practicality for real-time applications. To address this, I plan to explore alternative methods rooted in sub-Riemannian geometry, potentially reducing reliance on DDP. By leveraging sub-Riemannian distance—defined as the shortest path along curves

tangent to horizontal subspaces defined by nonholonomic constraints—I aim to develop analytical controllers that naturally respect these constraints, minimizing or eliminating the need for computationally intensive optimization methods.

Another limitation of the current work is that computation time has not been explicitly accounted for in the controller design. Although the use of primarily analytic methods keeps computation time relatively short, the robot may still move beyond safe boundaries during the brief delay between state measurement and control application. In the current framework, this issue is mitigated using local control laws to regulate the system during these short intervals. However, a more comprehensive treatment, either through experimental validation on a physical bipedal robot or through formal robustness analysis, is needed to fully address this concern.

6.5 Next Steps

Future research will focus on several key areas to overcome the limitations identified. Firstly, I will relax the current assumptions about leg configurations and GRF distribution to develop more realistic stabilizing controllers that fully respect contact point constraints. Secondly, I will investigate the feasibility of using sub-Riemannian distance to guide the design of control strategies, aiming to simplify and optimize the control process for bipedal robots. This approach could enable me to maintain stability with lower computational costs, making the strategy more efficient for real-time implementation. Additionally, I will explore multistep stabilization strategies, where the robot takes steps to manage its momentum, and analyze the range of initial conditions corresponding to different walking speeds where my algorithm is effective. This exploration will provide greater insight into

the robustness and adaptability of my approach, further enhancing its applicability in various real-world scenarios. In real applications, the system must be robust to parameter variations as accurate measurements of these parameters may not always be feasible. This aspect of robustness, while not fully explored in this research, will be a significant focus of future work.

Bibliography

[1] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.

[2] S. Aoi, T. Tanaka, S. Fujiki, T. Funato, K. Senda, and K. Tsuchiya, "Advantage of straight walk instability in turning maneuver of multilegged locomotion: a robotics approach," *Sci. Rep.*, vol. 6, p. 30199, 2016.

[3] M. G. Bekker and M. G. Bekker, "Mechanics of off-the-road locomotion," *Proc. Inst. Mech. Eng. Automob. Div.*, vol. 16, no. 1, pp. 25–44, Jan. 1962.

[4] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, vol. 3, pp. 2478–2483.

[5] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Auton. Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[6] J. W. Grizzle, J. Hurst, B. Morris, H.-W. Park, and K. Sreenath, "MABEL, a new robotic bipedal walker and runner," in *2009 American Control Conference*, 2009, pp. 2030–2036.

[7] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. C. De Wit, and J. Grizzle, "Rabbit: A testbed for advanced control theory," *IEEE Control Syst. Mag.*, 2003.

[8] E. Guizzo, "By leaps and bounds: An exclusive look at how Boston dynamics is redefining robot agility," *IEEE Spectr.*, vol. 56, no. 12, pp. 34–39, 2019.

[9] T. Takubo, T. Arai, K. Inoue, H. Ochi, T. Konishi, T. Tsurutani, and Y. Hayashibara, "Integrated Limb Mechanism Robot ASTERISK," *J. Robot. Mechatronics*, vol. 18, no. 2, pp. 203–214, 2006.

[10] D. Hobbelen, T. De Boer, and M. Wisse, "System overview of bipedal robots flame and tulip: Tailor-made for limit cycle walking," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2486–2491.

[11] R. Brooks, "Humanoid robots," *Commun. ACM*, vol. 45, no. 3, pp. 33–38, 2002.

[12] M. Vukobratovic, D. Hristic, and Z. Stojiljkovic, "Development of active anthropomorphic exoskeletons," *Med. Biol. Eng.*, vol. 12, pp. 66–80, 1974.

[13] P. Vadakkepat and D. Goswami, "Biped locomotion: stability, analysis and control," *Int. J. Smart Sens. Intell. Syst.*, vol. 1, no. 1, pp. 187–207, 2008.

[14] M. Vukobratović and B. Borovac, "Zero-moment point—thirty five years of its life," *Int. J. Humanoid Robot.*, vol. 1, no. 01, pp. 157–173, 2004.

[15] H. Lim, Y. Yamamoto, and A. Takanishi, "Control to realize human-like walking of

a biped humanoid robot," in *IEEE International Conference on Systems, Man and Cybernetics.*, 2000, vol. 5, pp. 3271–3276.

[16] Z. Li, X. Cheng, X. Bin Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2811–2817.

[17] S. Gupta and A. Kumar, "A brief review of dynamics and control of underactuated biped robots," *Adv. Robot.*, vol. 31, no. 12, pp. 607–623, 2017.

[18] R. M. Alexander, "Optimization and gaits in the locomotion of vertebrates," *Physiol. Rev.*, vol. 69, no. 4, pp. 1199–1227, 1989.

[19] T. Sugihara and T. Yamamoto, "Foot-guided agile control of a biped robot through ZMP manipulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4546–4551.

[20] A. I. Haldar and N. D. Pagar, "Predictive control of zero moment point (ZMP) for terrain robot kinematics," *Mater. Today Proc.*, vol. 80, pp. 122–127, 2023.

[21] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[22] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Trans. Robot.*, 2023.

[23] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Trans. Automat. Contr.*, vol. 48, no. 1, pp. 42–56, 2003.

[24] T. McGeer, "Passive dynamic walking," *Int. J. Rob. Res.*, vol. 9, no. 2, pp. 62–82, 1990.

[25] F. Plestan, J. W. Grizzle, E. R. Westervelt, and G. Abba, "Stable walking of a 7-DOF biped robot," *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 653–668, 2003.

[26] B. Griffin and J. Grizzle, "Nonholonomic virtual constraints and gait optimization for robust walking control," *Int. J. Rob. Res.*, vol. 36, no. 8, pp. 895–922, 2017.

[27] A. Isidori, *Nonlinear control systems*. Springer Science & Business Media, 2013.

[28] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Trans. Automat. Contr.*, vol. 59, no. 4, pp. 876–891, 2014.

[29] G. M. Moatimid and T. S. Amer, "Dynamical system of a time-delayed ϕ 6-Van der Pol oscillator: a non-perturbative approach," *Sci. Rep.*, vol. 13, no. 1, p. 11942, 2023.

[30] B. Brogliato and B. Brogliato, *Nonsmooth mechanics*, vol. 3. Springer, 1999.

[31] E. R. Westervelt, G. Buche, and J. W. Grizzle, "Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds," *Int. J. Rob. Res.*, vol. 23, no. 6, pp. 559–582, 2004.

[32] J. W. Grizzle, F. Plestan, and G. Abba, "Poincare's method for systems with impulse effects: application to mechanical biped locomotion," in *Proceedings of the 38th*

IEEE Conference on Decision and Control (Cat. No. 99CH36304), 1999, vol. 4, pp. 3869–3876.

[33] C. Chevallereau, E. R. Westervelt, and J. W. Grizzle, "Asymptotic stabilization of a five-link, four-actuator, planar bipedal runner," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, 2004, vol. 1, pp. 303–310.

[34] B. G. Buss, K. A. Hamed, B. A. Griffin, and J. W. Grizzle, "Experimental results for 3D bipedal robot walking based on systematic optimization of virtual constraints," in *2016 American Control Conference (ACC)*, 2016, pp. 4785–4792.

[35] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, "Models, feedback control, and open problems of 3D bipedal robotic walking," *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.

[36] A. E. Martin, D. C. Post, and J. P. Schmiedeler, "Design and experimental implementation of a hybrid zero dynamics-based controller for planar bipeds with curved feet," *Int. J. Rob. Res.*, vol. 33, no. 7, pp. 988–1005, 2014.

[37] D. Kim, Y. Zhao, G. Thomas, B. R. Fernandez, and L. Sentis, "Stabilizing series-elastic point-foot bipeds using whole-body operational space control," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1362–1379, 2016.

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

[39] M. McGrath, D. Howard, and R. Baker, "A Lagrange-based generalised formulation for the equations of motion of simple walking models," *J. Biomech.*, vol. 55, pp. 139–143, 2017.

[40] S. Shield and A. Patel, "Balancing stability and maneuverability during rapid gait termination in fast biped robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4523–4530.

[41] S. Wang, W. Chaovalitwongse, and R. Babuska, "Machine learning algorithms in bipedal robot control," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 42, no. 5, pp. 728–743, 2012.

[42] H. Dallali, M. Mosadeghzad, G. Medrano-Cerda, V.-G. Loc, N. Tsagarakis, D. Caldwell, and M. Gesino, "Designing a high performance humanoid robot based on dynamic simulation," in *2013 European Modelling Symposium*, 2013, pp. 359–364.

[43] P. A. Lasota, T. Fong, and J. A. Shah, "A survey of methods for safe human-robot interaction," *Found. Trends® Robot.*, vol. 5, no. 4, pp. 261–349, 2017.

[44] J. Reher and A. D. Ames, "Dynamic walking: Toward agile and efficient bipedal robots," *Annu. Rev. Control. Robot. Auton. Syst.*, vol. 4, pp. 535–572, 2021.

[45] Y. Huang, Q. Luo, and Y. Jia, "Study on walk step gait of quadruped robot based on the support polygon," in *Journal of Physics: Conference Series*, 2018, vol. 1074, no.

1, p. 12011.

[46] B. Ding, A. Plummer, and P. Iravani, “Investigating balancing control of a standing bipedal robot with point foot contact,” *IFAC-PapersOnLine*, vol. 49, no. 21, pp. 403–408, 2016.

[47] 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 (Cat. No. 03CH37422)*, 2003, vol. 2, pp. 1620–1626.

[48] T. Geng, “Online regulation of the walking speed of a planar limit cycle walker via model predictive control,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2326–2333, 2013.

[49] C. Chevallereau, G. Bessonnet, G. Abba, and Y. Aoustin, *Bipedal robots: modeling, design and walking synthesis*. John Wiley & Sons, 2013.

[50] P. A. Bhounsule, A. Zamani, and J. Pusey, “Switching between limit cycles in a model of running using exponentially stabilizing discrete control lyapunov function,” in *2018 Annual American Control Conference (ACC)*, 2018, pp. 3714–3719.

[51] E. R. Westervelt, J. W. Grizzle, and C. C. De Wit, “Switching and PI control of walking motions of planar biped walkers,” *IEEE Trans. Automat. Contr.*, vol. 48, no. 2, pp. 308–312, 2003.

[52] V. Murali, A. D. Ames, and E. I. Verriest, “Optimal walking speed transitions for fully actuated bipedal robots,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 6295–6300.

[53] K. T. ShuujiKajita and A. Kobayashi, “Dynamic Walk Control of a Biped Robot along the Potential Energy Conserving Orbit,” in *International Workshop on Intelligent Robots and Systems (lROS)*, 1990.

[54] X. Da, O. Harib, R. Hartley, B. Griffin, and J. W. Grizzle, “From 2D design of underactuated bipedal gaits to 3D implementation: Walking with speed tracking,” *IEEE Access*, vol. 4, pp. 3469–3478, 2016.

[55] X. Xiong and A. Ames, “3-d underactuated bipedal walking via h-lip based gait synthesis and stepping stabilization,” *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2405–2425, 2022.

[56] S. Rezazadeh and J. W. Hurst, “Control of ATRIAS in three dimensions: Walking as a forced-oscillation problem,” *Int. J. Rob. Res.*, vol. 39, no. 7, pp. 774–796, 2020.

[57] S. Rezazadeh, C. Hubicki, M. Jones, A. Peekema, J. Van Why, A. Abate, and J. Hurst, “Spring-mass walking with atrias in 3d: Robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot,” in *Dynamic Systems and Control Conference*, 2015, vol. 57243, p. V001T04A003.

[58] D. G. E. Hobbelen and M. Wisse, “Controlling the walking speed in limit cycle

walking," *Int. J. Rob. Res.*, vol. 27, no. 9, pp. 989–1005, 2008.

[59] J.-H. Kim, "Multi-axis force-torque sensors for measuring zero-moment point in humanoid robots: A review," *IEEE Sens. J.*, vol. 20, no. 3, pp. 1126–1141, 2019.

[60] "Recent Progress on Atlas, the World's Most Dynamic Humanoid Robot," 2020. [Online]. Available: https://www.youtube.com/watch?v=EGABAx52GKI.

[61] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4730–4737.

[62] H. Almubarak, K. Stachowicz, N. Sadegh, and E. A. Theodorou, "Safety embedded differential dynamic programming using discrete barrier states," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2755–2762, 2022.

[63] M. Cho, Y. Lee, and K.-S. Kim, "Model predictive control of autonomous vehicles with integrated barriers using occupancy grid maps," *IEEE Robot. Autom. Lett.*, vol. 8, no. 4, pp. 2006–2013, 2023.

[64] M. B. Oguntola and R. J. Lorentzen, "Ensemble-based constrained optimization using an exterior penalty method," *J. Pet. Sci. Eng.*, vol. 207, p. 109165, 2021.

[65] A. Pavlov, I. Shames, and C. Manzie, "Interior point differential dynamic programming," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 6, pp. 2720–2727, 2021.

[66] B. Plancher, Z. Manchester, and S. Kuindersma, "Constrained unscented dynamic programming," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5674–5680.

[67] F. Ferraguti, C. T. Landi, A. Singletary, H.-C. Lin, A. Ames, C. Secchi, and M. Bonfè, "Safety and efficiency in robotics: The control barrier functions approach," *IEEE Robot. Autom. Mag.*, vol. 29, no. 3, pp. 139–151, 2022.

[68] W. S. Cortez, D. Oetomo, C. Manzie, and P. Choong, "Control barrier functions for mechanical systems: Theory and application to robotic grasping," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 530–545, 2019.

[69] Q. Nguyen and K. Sreenath, "Optimal robust control for constrained nonlinear hybrid systems with application to bipedal locomotion," in *2016 American Control Conference (ACC)*, 2016, pp. 4807–4813.

[70] X. Xiong and A. Ames, "Slip walking over rough terrain via h-lip stepping and backstepping-barrier function inspired quadratic program," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2122–2129, 2021.

[71] J. Breeden and D. Panagou, "Robust control barrier functions under high relative degree and input constraints for satellite trajectories," *Automatica*, vol. 155, p. 111109, 2023.

[72] Z. Xie, C. K. Liu, and K. Hauser, "Differential dynamic programming with nonlinear constraints," in *2017 IEEE International Conference on Robotics and*

Automation (ICRA), 2017, pp. 695–702.

[73] Y. Aoyama, G. Boutselis, A. Patel, and E. A. Theodorou, "Constrained differential dynamic programming revisited," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9738–9744.

[74] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1168–1175.

[75] X. Bao, S. Wang, L. Zheng, R. J. Housden, J. V Hajnal, and K. Rhode, "A novel ultrasound robot with force/torque measurement and control for safe and efficient scanning," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.

[76] Y. Tao, Y. Wu, B. Li, F. Cladera, A. Zhou, D. Thakur, and V. Kumar, "SEER: Safe efficient exploration for aerial robots using learning to predict information gain," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1235–1241.

[77] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[78] N. Scianca, D. De Simone, L. Lanari, and G. Oriolo, "MPC for humanoid gait generation: Stability and feasibility," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1171–1188, 2020.

[79] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4688–4695, 2021.

[80] T. Kamioka, H. Kaneko, T. Takenaka, and T. Yoshiike, "Simultaneous optimization of ZMP and footsteps based on the analytical solution of divergent component of motion," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1763–1770.

[81] X. Xiong, J. Reher, and A. D. Ames, "Global position control on underactuated bipedal robots: Step-to-step dynamics approximation for step planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2825–2831.

[82] E. Dantec, M. Taix, and N. Mansard, "First order approximation of model predictive control solutions for high frequency feedback," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4448–4455, 2022.

[83] S. Katayama, M. Murooka, and Y. Tazaki, "Model predictive control of legged and humanoid robots: models and algorithms," *Adv. Robot.*, vol. 37, no. 5, pp. 298–315, 2023.

[84] Z.-H. Guan, D. J. Hill, and X. Shen, "On hybrid impulsive and switching systems and application to nonlinear control," *IEEE Trans. Automat. Contr.*, vol. 50, no. 7, pp. 1058–1062, 2005.

[85] T. Sun, X.-M. Sun, X. Zhao, and H. Liu, “Attitude control of rigid bodies: An energy-optimal geometric switching control approach,” *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 2, pp. 1162–1173, 2021.

[86] B. Huang, B. Lu, Q. Li, and Y. Tong, “Average dwell time based smooth switching linear parameter-varying proportional-integral-derivative control for an F-16 aircraft,” *IEEE Access*, vol. 9, pp. 30979–30992, 2021.

[87] R. A. DeCarlo, M. S. Branicky, S. Pettersson, and B. Lennartson, “Perspectives and results on the stability and stabilizability of hybrid systems,” *Proc. IEEE*, vol. 88, no. 7, pp. 1069–1082, 2000.

[88] Z. Li, T. Shi, B. Shan, and S. Li, “Color-shift keying code design in visible light communications using finite-state machines,” *IEEE Commun. Lett.*, vol. 26, no. 6, pp. 1333–1337, 2022.

[89] D. Liberzon, *Switching in systems and control*. Springer Science & Business Media, 2003.

[90] J. Heikkonen and P. Koikkalainen, “Self-organization and autonomous robots,” in *Neural Systems for Robotics*, Elsevier, 1997, pp. 297–337.

[91] Q. Lin, R. Loxton, and K. L. Teo, “Optimal control of nonlinear switched systems: Computational methods and applications,” *J. Oper. Res. Soc. China*, vol. 1, no. 3, pp. 275–311, 2013.

[92] R. Goedel, R. G. Sanfelice, and A. R. Teel, “Hybrid dynamical systems: modeling stability, and robustness,” *Princeton, NJ, USA*, 2012.

[93] M. Margaliot, “Stability analysis of switched systems using variational principles: an introduction,” *Automatica*, vol. 42, no. 12, pp. 2059–2077, 2006.

[94] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel, “Stability analysis of switched systems with stable and unstable subsystems: an average dwell time approach,” *Int. J. Syst. Sci.*, vol. 32, no. 8, pp. 1055–1061, 2001.

[95] Q. Zhang, J. Zhang, and Y. Wang, “Sliding-mode control for singular Markovian jump systems with Brownian motion based on stochastic sliding mode surface,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 3, pp. 494–505, 2017.

[96] Q. Wang, H. Yu, M. Wang, and X. Qi, “An improved sliding mode control using disturbance torque observer for permanent magnet synchronous motor,” *IEEE Access*, vol. 7, pp. 36691–36701, 2019.

[97] C. Xiu and P. Guo, “Global terminal sliding mode control with the quick reaching law and its application,” *IEEE Access*, vol. 6, pp. 49793–49800, 2018.

[98] J. Zhai and Z. Li, “Fast-exponential sliding mode control of robotic manipulator with super-twisting method,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 69, no. 2, pp. 489–493, 2021.

[99] J. Cao, S. Q. Xie, and R. Das, “MIMO sliding mode controller for gait exoskeleton driven by pneumatic muscles,” *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1,

pp. 274–281, 2017.

[100] W. Li and Z. Zheng, “Robust gain-scheduling controller to LPV system using gap metric,” in *2008 International Conference on Information and Automation*, 2008, pp. 514–518.

[101] D. Xing, C. G. Atkeson, J. Su, and B. J. Stephens, “Gain scheduled control of perturbed standing balance,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4063–4068.

[102] E. Kljuno and R. L. Williams, “Humanoid walking robot: modeling, inverse dynamics, and gain scheduling control,” *J. Robot.*, vol. 2010, no. 1, p. 278597, 2010.

[103] Y. Liu and Q.-H. Wu, *Adaptive switching control of large-scale complex power systems: theory and applications*. Springer Nature, 2023.

[104] B. Xiao, Q. Guo, C. Tu, F. Xiao, P. Liu, L. Long, and F. Blaabjerg, “A novel switching strategy based on the driving voltage and switching sequence for Si/SiC hybrid switch,” *IEEE Trans. Ind. Electron.*, 2024.

[105] X. Guo, W.-P. Li, and Y. Pan, “A switching predictive control method of three phase VSCs based on switching affine model,” *IEEE Access*, vol. 11, pp. 30273–30281, 2023.

[106] J. Wang, Q. Wei, Q. Zhao, and Z.-E. Lou, “Adaptive PID switched tracking control of robot manipulator,” in *2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, 2021, pp. 380–384.

[107] I. F. Jasim and P. W. Plapper, “Enhanced decentralized robust adaptive control of robots with arbitrarily-switched unknown constraints,” in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 4027–4032.

[108] H.-W. Park, K. Sreenath, A. Ramezani, and J. W. Grizzle, “Switching control design for accommodating large step-down disturbances in bipedal robot walking,” in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 45–50.

[109] I. G. Castillo-Hernandez, C. A. Cruz-Villar, and J. Alvarez-Gallegos, “Switched force/impedance control for humanoid robot walking on irregular surfaces,” in *IFToMM World Congress on Mechanism and Machine Science*, 2023, pp. 630–640.

[110] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch, “Control and stabilization of nonholonomic dynamic systems,” *IEEE Trans. Automat. Contr.*, vol. 37, no. 11, pp. 1746–1757, 1992.

[111] A. K. Khalaji and S. A. A. Moosavian, “Stabilization of a tractor-trailer wheeled robot,” *J. Mech. Sci. Technol.*, vol. 30, pp. 421–428, 2016.

[112] J.-M. Yang and J.-H. Kim, “Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots,” *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 578–587, 1999.

[113] A. W. Divelbiss and J. T. Wen, “Trajectory tracking control of a car-trailer system,” *IEEE Trans. Control Syst. Technol.*, vol. 5, no. 3, pp. 269–278, 1997.

[114] C.-S. Chiu, K.-Y. Lian, and P. Liu, “Fuzzy gain scheduling for parallel parking a car-like robot,” *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 6, pp. 1084–1092, 2005.

[115] V. Sankaranarayanan and A. D. Mahindrakar, “Switched control of a nonholonomic mobile robot,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 14, no. 5, pp. 2319–2327, 2009.

[116] M. Rehan, M. Reyhanoglu, and J. van Steen, “Trajectory tracking of a knife-edge on a flat surface,” in *2019 18th European Control Conference (ECC)*, 2019, pp. 1648–1652.

[117] H. McClamroch, M. Reyhanoglu, and M. Rehan, “Knife-edge motion on a surface as a nonholonomic control problem,” *IEEE Control Syst. Lett.*, vol. 1, no. 1, pp. 26–31, 2017.

[118] M. Rehan and M. Reyhanoglu, “Control of rolling disk motion on an arbitrary smooth surface,” *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 357–362, 2018.

[119] A. M. Bloch and N. H. McClamroch, “Control of mechanical systems with classical nonholonomic constraints,” in *Proceedings of the 28th IEEE Conference on Decision and Control,* 1989, pp. 201–205.

[120] J. W. Grizzle, C. H. Moog, and C. Chevallereau, “Nonlinear control of mechanical systems with an unactuated cyclic variable,” *IEEE Trans. Automat. Contr.*, vol. 50, no. 5, pp. 559–576, 2005.

[121] X. Xin and M. Kaneda, “Swing-up control for a 3-DOF gymnastic robot with passive first joint: design and analysis,” *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1277–1285, 2007.

[122] J. I. Ne_mark and N. A. Fufaev, *Dynamics of nonholonomic systems*, vol. 33. American Mathematical Soc., 2004.

[123] J. Awrejcewicz, *Classical mechanics: Dynamics*. Springer New York, 2012.

[124] S. Roy, I. N. Kar, J. Lee, and M. Jin, “Adaptive-robust time-delay control for a class of uncertain Euler–Lagrange systems,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 7109–7119, 2017.

[125] R. Olfati-Saber, “Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles.” Massachusetts Institute of Technology, 2001.

[126] J. Ghommam, F. Mnif, A. Benali, and N. Derbel, “Asymptotic backstepping stabilization of an underactuated surface vessel,” *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 6, pp. 1150–1157, 2006.

[127] R. W. Brockett, “Control theory and singular Riemannian geometry,” *New Dir. Appl. Math. Pap. Present. April 25/26, 1980, Occas. Case Centen. Celebr.*, pp. 11–27, 1982.

[128] C.-S. Chiu and K.-Y. Lian, “Hybrid fuzzy model-based control of nonholonomic

systems: A unified viewpoint," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 85–96, 2008.

[129] J. C. Horn, A. Mohammadi, K. A. Hamed, and R. D. Gregg, "Hybrid zero dynamics of bipedal robots under nonholonomic virtual constraints," *IEEE Control Syst. Lett.*, vol. 3, no. 2, pp. 386–391, 2018.

[130] F. Jean, *Control of nonholonomic systems: from sub-Riemannian geometry to motion planning*. Springer, 2014.

[131] H. Goldstein, *Classical mechanics*. Pearson Education India, 2011.

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

[133] P.-B. Wieber, "Viability and predictive control for safe locomotion," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1103–1108.

[134] E. A. Coddington and N. Levinson, *Theory of ordinary differential equations*. Tata McGraw-Hill Education, 1955.

[135] M. W. Spong, "The swing up control problem for the acrobot," *IEEE Control Syst. Mag.*, vol. 15, no. 1, pp. 49–55, 1995.

[136] J. Awrejcewicz, *Classical mechanics: kinematics and statics*, vol. 28. Springer Science & Business Media, 2012.

[137] X. Yun and N. Sarkar, "Unified formulation of robotic systems with holonomic and nonholonomic constraints," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 640–650, 1998.

[138] X. Chen, J. Ma, H. Sun, S. Zhen, H. Zhao, A. Al Mamun, and T. H. Lee, "Fuzzy-Based Controller Synthesis and Optimization for Underactuated Mechanical Systems With Nonholonomic Servo Constraints," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 10, pp. 4152–4164, 2022.

[139] J. Wu, W. Ye, Y. Wang, and C.-Y. Su, "A general position control method for planar underactuated manipulators with second-order nonholonomic constraints," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4733–4742, 2019.

[140] D. E. Davison and S. A. Bortoff, "Stabilization using pseudolinearization and high-gain feedback," *Automatica*, vol. 33, no. 2, pp. 245–251, 1997.

[141] S. A. Bortoff, "Approximate state-feedback linearization using spline functions," *Automatica*, vol. 33, no. 8, pp. 1449–1458, 1997.

[142] E. D. Sontag, *Mathematical control theory: deterministic finite dimensional systems*, vol. 6. Springer Science & Business Media, 2013.

[143] J. Barraquand and J.-C. Latombe, "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles," *Algorithmica*, vol. 10, no. 2, pp. 121–155, 1993.

[144] C.-L. Shih and W. A. Gruver, “Control of a biped robot in the double-support phase,” *IEEE Trans. Syst. Man. Cybern.*, vol. 22, no. 4, pp. 729–735, 1992.

[145] D. Mayne, “A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems,” *Int. J. Control*, vol. 3, no. 1, pp. 85–95, 1966.

[146] R. B. Vinter and R. B. Vinter, *Optimal control*, vol. 2, no. 1. Springer, 2010.

[147] C. Mastalli, W. Merkt, J. Marti-Saumell, H. Ferrolho, J. Solà, N. Mansard, and S. Vijayakumar, “A feasibility-driven approach to control-limited DDP,” *Auton. Robots*, vol. 46, no. 8, pp. 985–1005, 2022.

[148] N. Wu, W. Huang, Z. Song, X. Wu, Q. Zhang, and S. Yao, “Adaptive dynamic preview control for autonomous vehicle trajectory following with DDP based path planner,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1012–1017.

[149] M. D. Houghton, A. B. Oshin, M. J. Acheson, E. A. Theodorou, and I. M. Gregory, “Path planning: Differential dynamic programming and model predictive path integral control on VTOL aircraft,” in *AIAA SCITECH 2022 Forum*, 2022, p. 624.

[150] B. Jeddi, Y. Mishra, and G. Ledwich, “Differential dynamic programming based home energy management scheduler,” *IEEE Trans. Sustain. Energy*, vol. 11, no. 3, pp. 1427–1437, 2019.

[151] F. Fioretto, E. Pontelli, and W. Yeoh, “Distributed constraint optimization problems and applications: A survey,” *J. Artif. Intell. Res.*, vol. 61, pp. 623–698, 2018.

[152] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet, “3D bipedal robotic walking: models, feedback control, and open problems,” *IFAC Proc. Vol.*, vol. 43, no. 14, pp. 505–532, 2010.

[153] T. Yoshioka, T. Adachi, T. Butsuen, H. Okazaki, and H. Mochizuki, “Application of sliding-mode theory to direct yaw-moment control,” *JSAE Rev.*, vol. 20, no. 4, pp. 523–529, 1999.

[154] X. Xin and M. Kaneda, “A robust control approach to the swing up control problem for the acrobot,” in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, 2001, vol. 3, pp. 1650–1655.

[155] E. N. Chukwu, “On the controllability of nonlinear economic systems with delay: The Italian example,” *Appl. Math. Comput.*, vol. 95, no. 2, pp. 245–274, 1998.

[156] A. Astolfi, “Discontinuous control of nonholonomic systems,” *Syst. Control Lett.*, vol. 27, no. 1, pp. 37–45, 1996.

[157] I. Kolmanovsky, M. Reyhanoglu, and N. H. McClamroch, “Switched mode feedback control laws for nonholonomic systems in extended power form,” *Syst. Control Lett.*, vol. 27, no. 1, pp. 29–36, 1996.

[158] F. Blanchini and S. Miani, *Set-theoretic methods in control*, vol. 78. Springer, 2008.

[159] J. Lu and L. J. Brown, “A multiple Lyapunov functions approach for stability of switched systems,” in *Proceedings of the 2010 American Control Conference*, 2010,

pp. 3253–3256.

[160] J. P. Hespanha and A. S. Morse, “Stability of switched systems with average dwell-time,” in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304)*, 1999, vol. 3, pp. 2655–2660.

[161] L. Long and J. Zhao, “An integral-type multiple Lyapunov functions approach for switched nonlinear systems,” *IEEE Trans. Automat. Contr.*, vol. 61, no. 7, pp. 1979–1986, 2015.

[162] C. Deng, C. Wen, J. Huang, X.-M. Zhang, and Y. Zou, “Distributed observer-based cooperative control approach for uncertain nonlinear MASs under event-triggered communication,” *IEEE Trans. Automat. Contr.*, vol. 67, no. 5, pp. 2669–2676, 2021.

[163] “Biped Switching Control Animation,” 2024. [Online]. Available: https://youtube.com/shorts/LWIHeXiQPXQ.

[164] J. Chai and R. G. Sanfelice, “Forward invariance of sets for hybrid dynamical systems (Part II),” *IEEE Trans. Automat. Contr.*, vol. 66, no. 1, pp. 89–104, 2021.

[165] T. Horibe and N. Sakamoto, “Nonlinear optimal control for swing up and stabilization of the acrobot via stable manifold approach: Theory and experiment,” *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 6, pp. 2374–2387, 2018.

Appendix

A. Algorithm 1 MATLAB Code

```
tic
clc
close all
clear all
warning off

syms q0 q1 q2 q0dt q1dt q2dt x v2 sig time w1 w2
assume([q0 q1 q2 q0dt q1dt q2dt w2 x v2 sig],'real');
fprintf('\n\n***Gait Termination With Continuous + iLQG
Method in Original Coordinates***\n\n');

% Initialization
T_sim = 10;
Cnt = cnt_GRF = cnt_FN = ti1 = 0;
dTA = 0.1; % Actuators demanded torque time
Actuator_Out = [0 0; 0 0];
a_Filter = b_Filter = 30;
fprintf('Actuator Response Time: %.2fs\n\n', 4/b_Filter);
dti = t_Switch = Disable = 0;
epsilon = 2.2;
Controller_Continuous = 1;
u_lim = [-200 200; -200 200];
res = 200; % u limits search resolution
T = 30;
full_DDP = false;
My_Method = 1;

iLQR_Cnt = iLQR_Check = iLQR_Horizon = 0;

% Constants
g0 = 9.8;
m0 = 6.4; m1 = 13.6; m2 = 12;
L0 = 0.4; L1 = 0.4; L2 = 0.3;
mu = 0.3;
B = [0 0; 1 0; 0 1];

% Dynamics and simulation
[D, H, G, X, Y, V_potential, ~, c0, c1, Right1_Hand,
Right2_Hand] = Lagrange_Model_Constraints(m, L, lc, I, g0,
mu, B);
q0ddt = q0ddt_Change(D, V_potential);
```

```
D_Function = matlabFunction(D);
H_Function = matlabFunction(H);
G_Function = matlabFunction(G);
c0_Function = matlabFunction(c0);
c1_Function = matlabFunction(c1);
Right1_Hand_Function = matlabFunction(Right1_Hand);
Right2_Hand_Function = matlabFunction(Right2_Hand);

% Initial Conditions
q0_star = deg2rad(90);
q1_star = q2_star = deg2rad(0);
q_star = [q0_star q1_star q2_star];
q0_Initial = deg2rad(80);
q1_Initial = deg2rad(30);
q2_Initial = deg2rad(-40);
q_Start = q0_Initial + q1_Initial + q2_Initial;
q0dt_Initial = 0.5;
q1dt_Initial = -0.8;
q2dt_Initial = -1.2;
v2_initial = 0;

% Controller parameters
K = 6;
C1 = poly([-6 -5 -4]);
C2 = poly([-6 -5 -4]);
K2 = [C1(2) 0; 0 C2(2)];
K1 = [C1(3) 0; 0 C2(3)];
K0 = [C1(4) 0; 0 C2(4)];

% Plot and simulation
PlotData = Posture_Plot(q0_Initial, q1_Initial, q2_Initial,
L, Same);
title('Posture')
pause(0.5)
t1 = 0;
cnt = 1;
omega = 2 * pi * 0.2;
y_ref = matlabFunction(0.4 * sin(omega * time));
ydt_ref = matlabFunction(diff(y_ref, time));
yddt_ref = matlabFunction(diff(ydt_ref, time));
ydddt_ref = matlabFunction(diff(yddt_ref, time));

% ODE simulation
[t, Q] = ode45(@f, [0 T_sim], [q0_Initial q1_Initial
q2_Initial    q0dt_Initial    q1dt_Initial    q2dt_Initial
v2_initial]);
```

```
toc

% Animation and Posture Plot
if Disable == 0
    run('Animate1.m')
else
    run('Animate1.m')
end
function [x, u] = Bipedal_OriginalCor_Fun(x_des_Op,
x_desired, full_DDP, My_Method, u0, lims, x0, B, mu,
D_Function, H_Function, G_Function, c0_Function, c1_Function,
Right1_Hand_Function, Right2_Hand_Function, FT_Fun, FN_Fun,
eqns_Fun, res)
% x = [q0 q1 q2 q0dt q1dt q2dt]'

% Determine operational mode for the desired state
if x_des_Op == 0
    x_desired = Biped_dynamics(x0, u0(:,1));
end

% Define the cost and dynamic model for optimization
DYNCST  = @(x, u, i) Biped_dyn_cst(x, u, full_DDP);
% Setup options for iLQG
Op.lims  = lims;  % Control limits
Op.plot = -1;  % plot the derivatives as well

% Run the iLQG optimization
[x, u] = iLQG_OriginalCor(DYNCST, x0, u0, Op, c0_Function,
c1_Function,  Right1_Hand_Function,  Right2_Hand_Function,
res, eqns_Fun, My_Method);

    function y = Biped_dynamics(x, u)
        % Integration step using Runge-Kutta 4th order
        dt = 2^-10;  % time step for integration
        y = zeros(size(x));
        k1 = dynamics_rate_of_change(x, u);
        k2 = dynamics_rate_of_change(x + 0.5*dt*k1, u);
        k3 = dynamics_rate_of_change(x + 0.5*dt*k2, u);
        k4 = dynamics_rate_of_change(x + dt*k3, u);
        y = x + (1/6)*dt*(k1 + 2*k2 + 2*k3 + k4);

        function dy = dynamics_rate_of_change(x, u)
            % Compute the rate of change using provided
dynamics functions
            q0 = double(x(1,:)); q1 = double(x(2,:)); q2 =
double(x(3,:));
```

```
            q0dt = double(x(4,:)); q1dt = double(x(5,:));
q2dt = double(x(6,:));
            dy = zeros(size(x));
            for i = 1:size(u,2)
                D = D_Function(q1(i), q2(i));
                H = H_Function(q1(i), q2(i), q0dt(i),
q1dt(i), q2dt(i));
                G = G_Function(q0(i), q1(i), q2(i));
                f_system = [q0dt(i); q1dt(i); q2dt(i); -D^-
1*(H+G)];
                g_system = [zeros(3,2); D^-1*B];
                dy(:, i) = f_system + g_system*u(:, i);
            end
        end
    end
end
function [D, H, G, X, Y, V, K, c0, c1, Right1_Hand,
Right2_Hand] = Lagrange_Model_Constraints(m, L, lc, I, g0,
mu, B)
    % Find Dynamic Equations of a 3-link series robot using
Lagrange method
    % Calculate D, H and G matrices in D*qddt + H(q,qdt) + G
= B*tau
    % m = Mass of each link vector
    % L = Link length vector
    % lc = Distance to the Center of Mass vector
    % I = Moment of Inertia for each link vector
    % g0 = Gravity constant
    % By Zahed Dastan 2022

    syms t q0ddt q1ddt q2ddt q0dt q1dt q2dt q0(t) q1(t) q2(t)
d00 d01 d02 d11 d12 d22 x0 x1 x2

    % Assign given parameters to individual variables
    m0 = m(1); m1 = m(2); m2 = m(3);
    L0 = L(1); L1 = L(2); L2 = L(3);
    lc0 = lc(1); lc1 = lc(2); lc2 = lc(3);
    I0 = I(1); I1 = I(2); I2 = I(3);

    % (X,Y) position of Center of Mass (COM)
    Xc0 = lc0 * cos(q0); Yc0 = lc0 * sin(q0);
    Xc1 = L0 * cos(q0) + lc1 * cos(q0 + q1); Yc1 = L0 *
sin(q0) + lc1 * sin(q0 + q1);
    Xc2 = Xc1 + lc2 * cos(q0 + q1 + q2); Yc2 = Yc1 + lc2 *
sin(q0 + q1 + q2);
    X = [Xc0; Xc1; Xc2];
```

```
    Y = [Yc0; Yc1; Yc2];

    % Linear Velocities
    V0 = sqrt(diff(Xc0, t)^2 + diff(Yc0, t)^2);
    V1 = sqrt(diff(Xc1, t)^2 + diff(Yc1, t)^2);
    V2 = sqrt(diff(Xc2, t)^2 + diff(Yc2, t)^2);

    % Kinetic and Potential energies
    K = 0.5 * I0 * diff(q0, t)^2 + 0.5 * I1 * (diff(q0, t)^2
+ diff(q1, t)^2) + 0.5 * I2 * (diff(q0, t)^2 + diff(q1, t)^2
+ diff(q2, t)^2) + 0.5 * (m0 * V0^2 + m1 * V1^2 + m2 * V2^2);
    V = g0 * (m0 * Yc0 + m1 * Yc1 + m2 * Yc2);

    % Lagrangian
    L = K - V;

    % Lagrange Equations
    E0 = diff(diff(L, diff(q0, t)), t) - diff(L, q0);
    E1 = diff(diff(L, diff(q1, t)), t) - diff(L, q1);
    E2 = diff(diff(L, diff(q2, t)), t) - diff(L, q2);
    HG = [E0; E1; E2];

    % Substitutions for simplification
    HG = subs(HG, [q0 q1 q2 diff(q0, t) diff(q1, t) diff(q2,
t) diff(q0, t, t) diff(q1, t, t) diff(q2, t, t)], [x0 x1 x2
q0dt q1dt q2dt q0ddt q1ddt q2ddt]);

    % Separate D matrix (mass matrix) and combined G+H matrix
    [D, HG] = equationsToMatrix(HG, [q0ddt q1ddt q2ddt]);

    % Decompose HG into H and G
    [G, H] = equationsToMatrix(HG, [q0dt q1dt q2dt]);

    % Constraint coefficients for control inputs
    q = [q0; q1; q2]; qdt = [q0dt; q1dt; q2dt];
    Temp_Y = mu * m * Y; Temp_X = m * X;
    Temp = Temp_Y - Temp_X;
    T_B = jacobian(Temp, q);
    S_B = hessian(Temp, q);
    T_B_Tilda = jacobian(m * Y, q);
    S_B_Tilda = hessian(m * Y, q);

    % Inequality equations
    D_inv = inv(D);
    c0 = T_B * D_inv * B;
```

```
    Right1_Hand = T_B * D_inv * (H + G) - mu * sum(m) * g0 -
transpose(qdt) * S_B * qdt;
    c1 = T_B_Tilda * D_inv * B;
    Right2_Hand = T_B_Tilda * D_inv * (H + G) - sum(m) * g0
- transpose(qdt) * S_B_Tilda * qdt;
end
```

Vita

Candidate's full name: Zahed Dastan

Universities attended:

Sahand University of Technology, B.Sc., 2012

K. N. Toosi University of Technology, M.Sc., 2015

Journal Publications:

1. Z. Dastan, and M. Tavakoli-Kakhki, "Suppression of high order disturbances and tracking for nonchaotic systems: a time-delayed state feedback approach", *Control Theory and Technology*, 2022, DOI: 10.1007/s11768-021-00073-9.

Conference Publications:

2. Z. Dastan and J. Sensinger, "Leveraging Control Inputs to Enforce Constraints in Differential Dynamic Programming for Nonlinear Optimization", in *63rd IEEE Conference on Decision and Control*, 2024

Conference Presentations:

3. F. Khalilzadeh and Z. Dastan, "Optically pumped semiconductor disk laser design, emitting at 1.55 μm, based on wafer bonding technology for efficient thermal management", *Canadian Photonics Online Meetup*, 2020.