Robust Predictive Control for Legged Locomotion

Abhishek Goud Pandala

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mechanical Engineering

Kaveh Akbari Hamed, Chair
Alexander Leonessa
Corina Sandu
Craig A. Woolsey

December 13, 2023
Blacksburg, Virginia

Keywords: Legged Locomotion, Robust Predictive Control, Model Predictive Control, Hybrid Zero Dynamics
Copyright 2024, Abhishek Goud Pandala

Robust Predictive Control for Legged Locomotion

Abhishek Goud Pandala

(ABSTRACT)

This dissertation aims to realize the goal of developing robust control solutions that can enable legged robots to navigate complex unknown environments. The idea of creating articulated-legged machines that can mimic animal locomotion has fueled the imagination of many researchers. These legged robots are designed to assist humans in their day-to-day tasks and challenging scenarios such as monitoring remote, inhospitable environments, disaster response, and other dangerous environments. Despite several decades of research, legged robots have yet to reach the dexterity or dynamic stability needed for real-world deployments. A fundamental gap exists in the understanding and development of reliable and scalable algorithms required for the real-time planning and control of legged robots. The overarching goal of this thesis is to formally develop computationally tractable, robust controllers based on nonlinear hybrid systems theory, model predictive control, and optimization for the real-time planning and control of agile locomotion in quadrupedal robots.

Toward this objective, this thesis first investigates layered control architectures. In particular, we propose a two-level hierarchical control architecture in which the higher level is based on a reduced-order model predictive control (MPC), and the lower level is based on a full-order quadratic programming (QP) based virtual constraints controller. Specifically, two MPC architectures are explored: 1) An event-based MPC scheme that generates the optimal center of mass (COM) trajectories using a reduced-order linear inverted pendulum (LIP) model, and 2) A time-based MPC scheme that computes the optimal COM and ground reaction forces (GRF) using the reduced-order single rigid body (SRB) dynamics model. The optimal COM trajectories in the event-based MPC and the optimal COM trajectories, along with the ground reaction forces in the time-based MPC, are then tracked by the low-level virtual constraints controller. The event-based MPC scheme is numerically validated on the Vision 60 platform in a physics-based simulation environment. It has significantly reduced the computational burden associated with real-time planning-based MPC schemes. However, owing to the quasi-static nature of the optimal trajectories generated by the LIP model, we explored a time-based MPC scheme using Single Rigid Body Dynamics. This time-based MPC scheme is also numerically validated using the mathematical model of the A1 quadrupedal robot.

Most MPC schemes use a reduced-order model to generate optimal trajectories. However, the abstraction and unmodeled dynamics in template models significantly increase the gap between reduced- and full-order models, limiting the robot's full scope and potential. In the second part of the thesis, we aim to develop a computationally tractable robust model predictive control (RMPC) scheme based on convex QPs to bridge this gap. The RMPC

framework considers the single rigid body model subject to a set of unmodeled dynamics and plans for the optimal reduced-order trajectory and GRFs. The generated optimal GRFs of the high-level RMPC are then mapped to the full-order model using a low-level nonlinear controller based on virtual constraints and QP. The key innovation of the proposed RMPC framework is that it allows the integration of the hierarchical controller with Reinforcement Learning (RL) techniques to train a neural network to compute the vertices of the uncertainty set numerically. The proposed hierarchical control algorithm is validated numerically and experimentally for robust and blind locomotion of the A1 quadrupedal robot on different indoor and outdoor terrains and at different speeds. The numerical analysis of the RMPC suggests significant improvement in the performance of the rough terrain locomotion compared to the nominal MPC. In particular, the proposed RMPC algorithm outperforms the nominal MPC by over 60% during rough terrain locomotion over 550 uneven terrains. Our experimental studies also indicate a significant reduction in the gap between the reduced full-order models by comparing the desired and actual GRFs.

Finally, the last part of the thesis presents a formal approach for synthesizing robust $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs to stabilize the periodic locomotion of legged robots. The proposed algorithm builds on the existing optimization-based control stack. We outline the set of conditions under which the closed-loop nonlinear dynamics around a periodic orbit can be transformed into a linear time-invariant (LTI) system using Floquet theory. We then outline an approach to systematically generate parameterized $\mathcal{H}_2$- and $\mathcal{H}_\infty$- robust controllers using linear matrix inequalities (LMIs). We subsequently established a set of conditions guaranteeing the existence of such robust optimal controllers. The proposed $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs are extensively validated both numerically and experimentally for the robust locomotion of the A1 quadrupedal robot subject to various external disturbances and uneven terrains. Our numerical analysis suggests a significant improvement in the performance of robust locomotion compared to the nominal MPC.

Robust Predictive Control for Legged Locomotion

Abhishek Goud Pandala

(GENERAL AUDIENCE ABSTRACT)

Legged robots have always been envisioned to work alongside humans, assisting them in mundane day-to-day tasks to challenging scenarios such as monitoring remote locations, planetary exploration, and supporting relief programs in disaster situations. Furthermore, research into legged locomotion can aid in designing and developing powered prosthetic limbs and exoskeletons. With these advantages in mind, several researchers have created sophisticated-legged robots and even more complicated algorithms to control them. Despite this, a significant gap exists between the agility, mobility, and dynamic stability shown by the existing legged robots and their biological counterparts. To work alongside humans, legged robots have to interact with complex environments and deal with uncertainties in the form of unplanned contacts and unknown terrains. Developing robust control solutions to accommodate disturbances explicitly marks the first step towards safe and reliable real-world deployment of legged robots.

Toward this objective, this thesis aims to establish a formal foundation to develop computationally tractable robust controllers for the real-time planning and control of legged robots. Initial investigations in this thesis report on the use of layered control architectures, specifically event-based and time-based Model Predictive Control(MPC) schemes. These layered control architectures consist of an MPC scheme built around a reduced-order model at the high level and a virtual constraints-based nonlinear controller at the low level. Using these layered control architectures, this thesis proposed two robust control solutions to improve the rough terrain locomotion of legged robots.

The first proposed robust control solution aims to mitigate one of the issues of layered control architecture. In particular, layered control architectures rely on a reduced order model at the high level to remain computationally tractable. However, the approximation of full-order models with reduced-order models limits the full scope and potential of the robot. The proposed algorithm aims to bridge the gap between reduced- and full-order models with the integration of model-free Reinforcement Learning (RL) techniques. The second algorithm proposes a formal approach to generate robust optimal control solutions that can explicitly accommodate the disturbances and stabilize periodic legged locomotion. Under some mild conditions, the MPC control solution is analyzed, and an auxiliary feedback control solution that can handle disturbances explicitly is proposed. The thesis also theoretically establishes the sufficient conditions for the existence of such controllers. Both the proposed control solutions are extensively validated using numerical simulations and experiments using an A1 quadrupedal robot as a representative example.

Dedication

To my parents, Ramulu and Anjamma, and my brother Narendra.

Acknowledgments

At the culmination of my graduate career, I would like to express my sincere gratitude to the people who have helped me succeed. First of all, I am highly indebted to Dr. Kaveh Hamed Akbari. His personal investment in my academic growth and his constant motivation have pushed me through tough times. Your deep understanding of control theory, intuitive explanations, and patient teaching style have helped me develop a solid theoretical foundation and a penchant for legged robots. Your approach to research and life will always be a massive inspiration to me. Thank you for serving as my advisor.

I would like to acknowledge my thesis committee, Dr. Alexander Leonessa, Dr. Corina Sandu, and Dr.Craig A. Woolsey, for their efforts in improving this thesis and the valuable advice they provided during my preliminary examination that directed the remainder of my dissertation.

I have had the great fortune to collaborate with the leading researchers in legged locomotion and controls theory. In particular, I would like to express my appreciation for the insightful discussion with Dr. Aaron D. Ames and Dr. Ugo Rosolia that helped shape a significant part of my thesis.

This work would not have been possible without the tremendous support of my colleagues at the Hybrid Dynamic Systems and Robot Locomotion Lab. I would like to thank Dr. Vinay R. Kamidi, Dr. Jeeseop Kim, and Dr. Randall Fawecett. Thank you for all the valuable discussions and for making my graduate life memorable.

There are many others whose mentorship has been invaluable in my career. Dr. Krishnan Balasubramaniam and Dr. C.V. Krishnamurthy have introduced me to scientific research, particularly in non-destructive testing, and I am forever grateful for the opportunity. I would also like to thank my other mentors, Dr. Martin Spies, Dr. Michael Lowe, and Dr. Peter Huthwaite, for hosting me at their prestigious institutions and teaching me valuable scientific research tools and techniques. I would also like to thank Dr. Shivaprasad Shridhara Bhat for guiding me through the initial phases of research. Changing scientific disciplines is a tough gamble, and I would like to extend my sincere thanks to Dr. Hae-Won Park for easing the transition. Your meticulous approach to legged locomotion will always motivate me to push through.

I would also like to thank my collaborators, Dr. Yanran Ding, Dr. Chad Li, and Dr JaeJun Park, who made my time at UIUC memorable and taught me the ways around the hardware of the PANTHER robot. I would also like to thank my friends Gnana Saurya and

Arulmurugan for being supportive throughout my student life.

Finally, crucial to my success and happiness are my parents. This journey would not have been possible without your constant love and support. Finally, I want to thank my brother Narendra for always being there for me.

Contents

List of Figures xi

List of Tables xiv

1 Introduction 1

1.1 Terrestrial Locomotion - A Case For Legs . . . 1

1.1.1 State-of-the-art . . . 2

1.2 Current Limitations and Motivation for Study . . . 4

1.3 Scope, Goals and Objectives . . . 7

1.4 Significance and Impact . . . 10

1.5 Background Work . . . 10

1.5.1 State-of-the-art Motion Planners . . . 10

1.5.2 State-of-the-art Control Algorithms . . . 14

1.6 Modeling Tools . . . 17

1.6.1 MultiBody Dynamics . . . 18

1.6.2 Physics Based Simulation Environments . . . 19

1.7 Relevant Publications . . . 20

1.8 Outline . . . 21

I Hierarchical Control Architectures based on Event- and Time-Based MPC for the Locomotion of Quadrupedal Robots 23

2 Reduced Order Models 24

2.1 Hierarchical Control Architectures - Motivation . . . 24

2.2 Event Based MPC Scheme . . . 24

2.2.1 Linear Inverted Pendulum Model . . . 24

2.2.2 Event-Based MPC Formulation . . . 27
2.3 Time-Based MPC . . . 29
2.3.1 Single Rigid Body Dynamics Model . . . 29
2.3.2 Time-Based MPC Formulation . . . 30
2.4 Low-Level Controllers . . . 32
2.4.1 Virtual Constrains Based Low-Level Controller . . . 32
2.5 Numerical Results . . . 34
2.5.1 Event-Based MPC scheme . . . 34
2.5.2 Time-Based MPC scheme . . . 38
2.6 Summary . . . 40

II Robust Model Predictive Control for Rough Terrain Locomotion 41

3 Robust Model Predictive Control 42

3.1 Introduction . . . 42
3.1.1 Motivation . . . 42
3.1.2 Background, Motivation, and Related Work . . . 43
3.1.3 Objectives and Contributions . . . 44
3.2 Robust MPC Formulation . . . 45
3.3 Low-Level Nonlinear Controller . . . 47
3.4 Learning to Close the Gap . . . 49
3.5 Numerical and Experimental Results . . . 50
3.5.1 Controller Synthesis . . . 50
3.5.2 Training of the MLP . . . 52
3.5.3 Numerical and Experimental Validation and Discussion . . . 53
3.6 Conclusion and Future Work . . . 56

III Robust Optimal Predictive Controller using Linear Matrix Inequailities **58**

4 $\mathcal{H}_2$- and $\mathcal{H}_\infty$-Optimal Predictive Controllers for Periodic Legged Locomotion: A Linear Matrix Inequality Approach **59**

4.1 INTRODUCTION . . . 59

4.1.1 Motivation . . . 59

4.1.2 Background and Related Work . . . 60

4.1.3 Objectives and Contributions . . . 61

4.2 PROBLEM FORMULATION . . . 61

4.3 MAIN RESULT and ALGORITHM . . . 63

4.4 APPLICATION TO QUADRUPEDAL LOCOMOTION . . . 67

4.4.1 Control Synthesis . . . 67

4.4.2 Numerical Simulations . . . 69

4.4.3 Experimental Evaluations . . . 69

4.5 Summary . . . 70

5 Conclusion and Future **71**

5.1 Summary of Contribution . . . 71

5.1.1 Development of Layered Control Architectures . . . 71

5.1.2 Developement of Robust Predictive Controllers . . . 71

5.1.3 $\mathcal{H}_2/\mathcal{H}_\infty$ Predictive Controllers for Periodic Legged Locomotion . . . 72

5.2 Possible Future Research Directions . . . 72

5.3 Final Thoughts . . . 74

Bibliography **75**

List of Figures

1.1 A brief history of legged robots starting from top right corner. (a) First patented design of Mechanical Horse (b) GE Walking Electric Truck (c) Phoney Pony robot (d) PV-II robot (e) Raibert Hopper (f) Boston Dynamics Big Dog (g) IIT HyQ (h) ETH-Zurich StarlETH (i) ETH-Zurich ANYMAL (j) MIT Cheetah -2 (k) ANYMAL - C (l) MIT mini cheetah (m) Boston Dynamics SPOT (n) Ghost Robotics Vision60 (o) Unitree A1. 2

1.2 Remarkable feats achieved by legged robots. High-speed locomotion by (a) Mabel (b) MIT Cheetah - 2 , agile motions by (c) Boston Dynamics Atlas (d) MIT Cheetah - 3 (e) MIT mini Cheetah, rough terrain navigation by (f) Boston Dynamics Big Dog (g) ANYMAL-C (h) ANYMAL - B (i) Ghost Robotics Vision 60 (j) Unitree A1, construction and industrial inspection by (k),(l) SPOT (m),(n) ANYMAL (o) Ghost Robotics Vision60. 3

1.3 Research Task R1 illustrating two hierarchical control architectures. 8

1.4 Research Task R2 illustrating Robust Model Predictive Controller. 9

2.1 Illustration of LIP model. 25

2.2 Event-Based MPC and Domains of Locomotion. 26

2.3 Illustration of Event-Based MPC. 28

2.4 Illustration of Single Rigid Body Dynamics. 29

2.5 Illustration of Time-Based MPC. 31

2.6 Plot of the COM and COP trajectories for the forward and diagonal trot gaits of the reduced-order system. Convergence to the target points $(0.82, 0)$ and $(0.58, 0.34)$ is clear. 36

2.7 Plot of the virtual constraints and torque inputs in RaiSim for (a) forward trot, (b) diagonal trot, and (c) forward trot with payloads. Here, y_x, y_y, y_z, y_{roll}, y_{pitch}, y_{yaw} denote the first six components of the virtual constraints that relate to the absolute position and orientation of the body. In addition, FH, FK, RH, and RK stand for the front hip, front knee, rear hip, and rear knee of the left-hand-side of the robot, respectively. The subscripts "roll" and "pitch" for the torque plots also denote the roll and pitch motions of the hip joints. (d) Plot of the virtual constraints and torques for trot gait subject to the LuGre contact model in MATLAB/Simulink when the control frequency is reduced to 500 Hz with a delay of 2 (ms) in solving QPs. 37

2.8 Plot of the virtual constraints and torque inputs of the full-order closed-loop system in RaiSim for forward trot with (a) and (b) external disturbances, (c) external push, (d) ground height variations. 37

2.9 Virtual Constraints and Phase Portraits of Body Pitch and Roll for the forward trot gait of A1 quadrupedal robot. 39

2.10 Virtual Constraints and Phase Portraits of Body Pitch and Roll for the lateral trot gait of A1 quadrupedal robot. 39

3.1 Overview of the proposed hierarchical control algorithm, based on RMPC, NN, and QP-based nonlinear controller. The real-time control loops and signals are shown by the solid blocks and arrows, respectively. The offline training procedure is shown by the dashed blocks and signals. 43

3.2 Indoor and outdoor experiments to validate the proposed hierarchical control algorithm on different terrains. (a) Blind locomotion of the A1 robot on uneven terrain with wooden blocks at the speed of 0.5 (m/s). (b) Locomotion on a compliant surface with unknown blocks. (c) Locomotion on a slippery surface with a payload of 4.54 (kg) (36% of the robot's weight). (d) Locomotion on blocks with a payload of 4.54 (kg). (e) Locomotion on uneven gravel terrain. (f) Locomotion on mulch. (g) Locomotion on a slope. (h) Locomotion on the grass at the speed of 1 (m/s). 49

3.3 The percentage of the success rate of RMPC with three and two vertices and the nominal MPC over 550 randomly generated uneven terrains. The dashed line represents the success rate across the length of the terrain. 52

3.4 Phase portrait of the unactuated DOFs (roll and pitch). The figure depicts robust stability in the phase portraits for two different experiments, including outdoor locomotion on inclined terrain (Fig. 4.3(g)) and indoor locomotion on a slippery surface with a payload of 4.54 (kg) (Fig. 4.3(c)). 53

3.5 Phase portrait of the unactuated DOFs (roll and pitch). The figure depicts robust stability in the phase portraits for two different experiments, including outdoor locomotion on inclined terrain (Fig. 4.3(g)) and indoor locomotion on a slippery surface with a payload of 4.54 (kg) (Fig. 4.3(c)). 54

3.6 Representative snapshots of the stable locomotion of A1 quadrupedal robot on rough terrain. 55

3.7 Comparison of the gaps between the GRFs generated based on the reduced- and full-order models. Plot of desired and actual GRFs (the z component) with the high-level RMPC algorithm and the low-level QP-based nonlinear controller on flat terrain (a) and on a slope (c). (b) Plot of the desired and actual GRFs with the hierarchical control algorithm in which the high-level RMPC is replaced by a nominal MPC for experiments on flat terrain. 55

3.8 Plot of the 2-norm for one of the vertices for the uncertainty set (i.e., $\|w_k^\ell\|$) versus time. The plot depicts the contribution of the trained MLP for locomotion on a flat ground and locomotion on an uneven terrain with two blocks. The gray portions indicate the stepping on blocks. 56

4.1 The percentage success rate of $\mathcal{H}_2$-optimal MPC, $\mathcal{H}_\infty$-optimal MPC, and normal MPC over 300 different trails. The dashed line represents the success rate over a time period of 30 (s). 68

4.2 The vertical component of the GRF, prescribed by the $\mathcal{H}_2$-optimal MPC, for the front right leg during the push experiment in Fig. 4.3 (a). 69

4.3 Snapshots illustrating the performance of $\mathcal{H}_2$-optimal MPC. Time increases from left to right. (a) Locomotion subject to external pushes at 0.5 (m/s). (b) Locomotion on uneven terrain with wooden blocks at 0.5 (m/s). The videos are available online []. 70

List of Tables

2.1 Numerical Parameters for the simulation of event-based MPC scheme. 35
2.2 Numerical Parameters for the simulation of time-based MPC scheme. 38

List of Abbreviations

ASLIP Assymetric Spring Loaded Inverted Pendulum

COM Center of Mass

COP Center of Pressure

DCM Divergent Component of Motion

DMPC Distributed Model Predictive Control

DOF Degrees of Freedom

DP Dynamic Programming

EE End Effector

EOM Equations of Motion

GRF Ground Reaction Forces

HZD Hybrid Zero Dynamics

IMU Inertial Measurement Unit

IP Interior Point Method

LIP Linear Inverted Pendulum

LIP Linear Matrix Inequalities

LLS Lateral Leg Spring

LP Linear Programming

MPC Model Predictive Control

MRS Multi Robot System

NLP Nonlinear Programming

OCP Optimal Control Problem

ODE Ordinary Differential Equation

QP Quadratic Programming

SLIP Spring Loaded Inverted Pendulum

SRB Single Rigid Body

TO Trajectory Optimization

VCs Virtual Constraints

WBC Whole Body Control

ZMP Zero Moment Point

Chapter 1

Introduction

1.1 Terrestrial Locomotion - A Case For Legs

Two types of locomotion dominate terrestrial movements: wheeled and legged. Wheeled locomotion via automobiles is the preferred method for modern ground transportation. They achieved this supremacy by continuously contacting the ground, and their design assumes good terrain conditions (paved surfaces). This artificial modification of terrain enables wheeled vehicles to achieve remarkable speed and efficiency and even travel faster than legged systems. On the other hand, billions of years of evolution of terrestrial animals have resulted in locomotion that predominantly uses legs (articulated morphology). The agility, dexterity, and dynamic stability shown by legged systems can easily dwarf wheeled locomotion. Legged systems (animals or machines) are typically characterized by articulated appendages called legs that can be used to make and break contact with the environment. This feature provides many advantages. First, the ability to create and configure intermittent contacts is beneficial for traversing discontinuous terrain and eliminates the need for continuous support, as is the case for wheeled locomotion. Articulated legged morphology can generate friction-limited forces in a broader range of directions. This enables omnidirectional motions without slippage and any constraints on lateral movements. Furthermore, the articulation of the limbs also offers an ability to dynamically reconfigure the center of mass for high-power manipulation. By using isolated footholds and varying leg configurations, legged systems can adapt to surface irregularities and easily traverse natural terrains. Another advantage of legged systems is the isolation of the torso from the movement of the feet. This isolation creates an active suspension that enables the torso (or even a payload) to navigate smoothly in terrain undulations and facilitates manipulation-like tasks.

With these advantages in mind, many researchers have envisioned the creation of legged robots to augment their natural environments. These legged robots are anticipated to help humans monitor remote locations such as volcanoes, space, and underwater, hostile and dangerous environments such as nuclear power plants, mining operations, and other adverse situations. In addition to disaster responses and assistance in hostile environments, research into legged locomotion opens avenues to understanding our own locomotion. Further, it informs the design and development of prosthesis arms and legs for differently-abled individuals. Despite decades of research, legged robots have not yet achieved the dynamic stability or dexterity required for real-world deployments. In particular, there

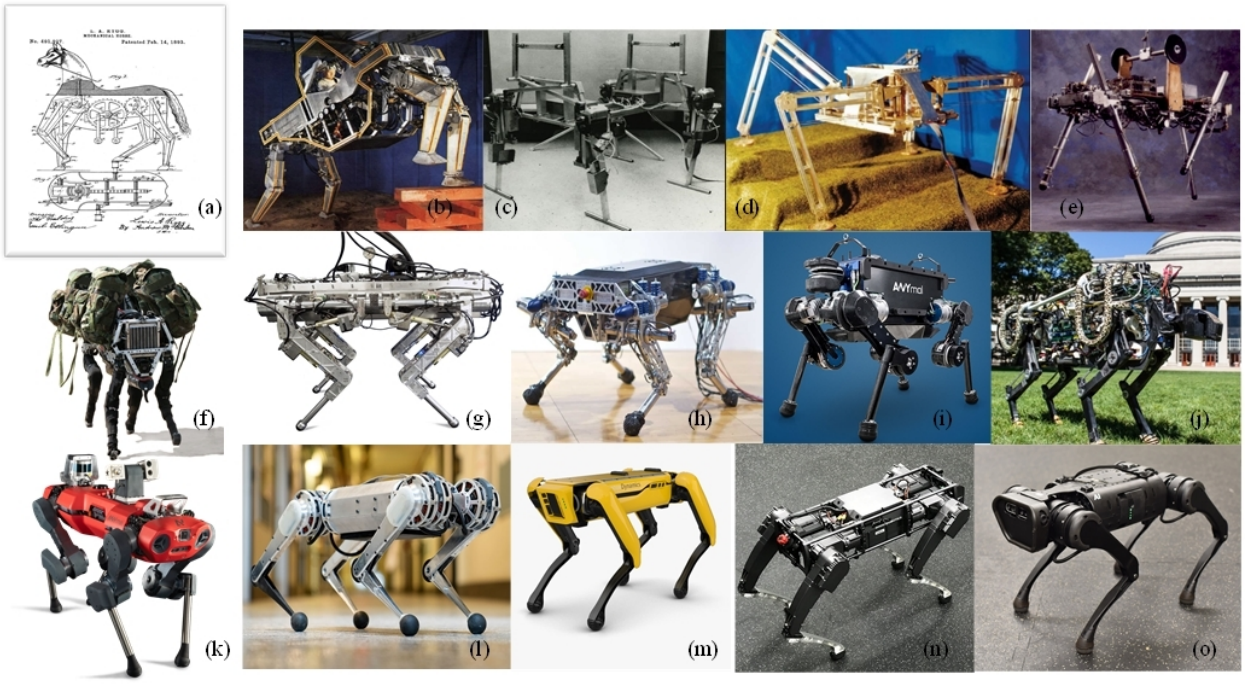


Figure 1.1: A brief history of legged robots starting from top right corner. (a) First patented design of Mechanical Horse (b) GE Walking Electric Truck (c) Phoney Pony robot (d) PV-II robot (e) Raibert Hopper (f) Boston Dynamics Big Dog (g) IIT HyQ (h) ETH-Zurich StarlETH (i) ETH-Zurich ANYMAL (j) MIT Cheetah -2 (k) ANYMAL - C (l) MIT mini cheetah (m) Boston Dynamics SPOT (n) Ghost Robotics Vision60 (o) Unitree A1.

is a ftextitfundamental gap in the knowledge of scalable algorithms for dynamic planning, safety, and robust control necessary for the mobility, agility, and autonomy of sophisticated legged robots in complex environments.

1.1.1 State-of-the-art

The dream of creating articulated machines that mimic animal locomotion has fueled the imagination of artists and researchers alike. Early attempts along these lines first started with the celebrated mathematician Chebychev [1], with the first patented designs by Rygg [2] in 1893. The first scientific studies into legged robots started in the second half of the 20^{th} century. A few milestones along these lines constitute the first teleoperated legged robot [3] to adapt to different gaits, the first autonomous legged robot with electrical actuation and computer control [4, 5], and the first legged robot to climb up and down stairs and handle obstacles [6]. Systematic study into dynamic balance and motion control of legged robots began with the introduction of hoppers by Marc Raibert [7, 8, 9]. For the history and evolution of legged robots, please refer to Chapter - I of [9] and the survey papers

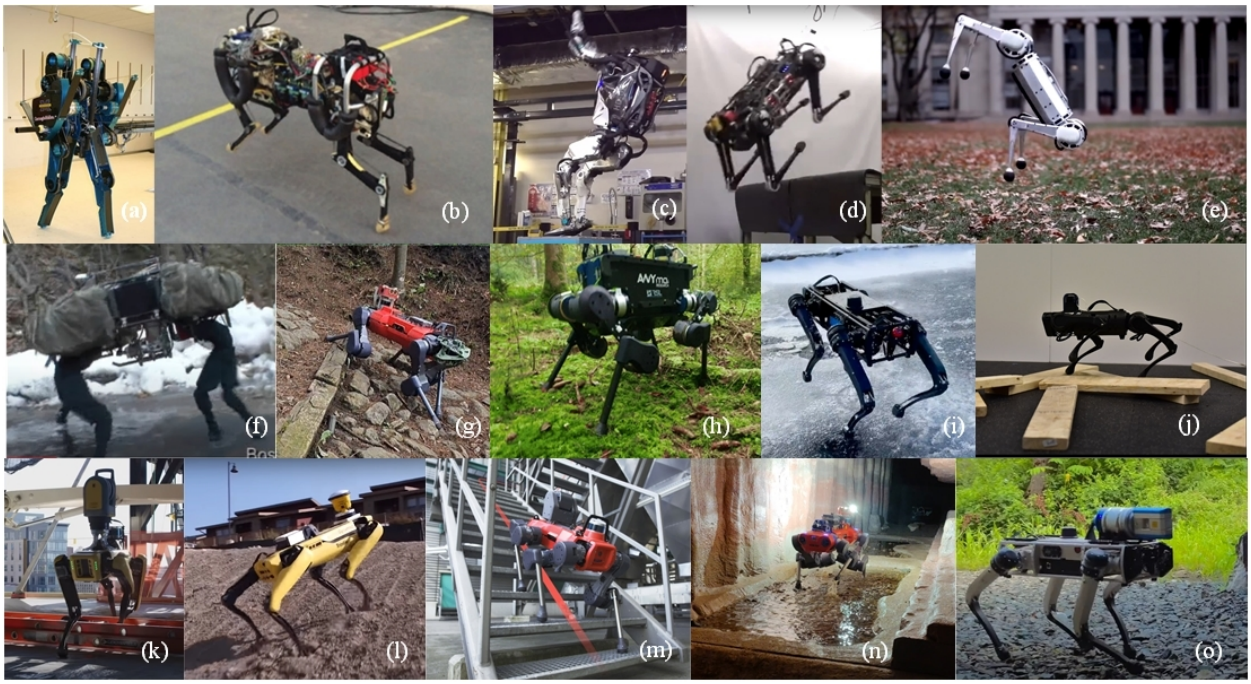


Figure 1.2: Remarkable feats achieved by legged robots. High-speed locomotion by (a) Mabel (b) MIT Cheetah - 2 , agile motions by (c) Boston Dynamics Atlas (d) MIT Cheetah - 3 (e) MIT mini Cheetah, rough terrain navigation by (f) Boston Dynamics Big Dog (g) ANYMAL-C (h) ANYMAL - B (i) Ghost Robotics Vision 60 (j) Unitree A1, construction and industrial inspection by (k),(l) SPOT (m),(n) ANYMAL (o) Ghost Robotics Vision60.

[10, 11, 12, 13, 14, 15]. As of the writing of this thesis, many sophisticated legged robots have come to exist, see Figure 1.1.

Thanks to seminal contributions from astute researchers, current legged robots have achieved good progress and feature remarkable feats such as high-speed locomotion, agility, manipulation capabilities, and robustness. High-speed locomotion has been achieved by Mabel [16], MIT Cheetah - 3 [17, 18] and Unitree A1 robot [19] with MIT Cheetah - 2 [20, 21] achieving speeds of the order of 6.4 m/s. Various agile maneuvers in the form of back flip [22] and discrete jumps [21, 23, 24, 25, 26] are also demonstrated. Using learning techniques and computer vision, several researchers developed legged robots that can navigate rough terrains [27, 28, 29] and monitor construction and industrial environments [30, 31, 32, 33]. Using a fixed-base manipulator arm, several researchers have increased the dexterity of quadrupedal robots to perform manipulation tasks [34, 35, 36, 37, 38] and cooperative locomotion [39, 40]. To avoid the increasing complexity of designing controllers for increasingly sophisticated legged robots, researchers have looked into scalable algorithms through distributed [41, 42] and decentralized controllers [43, 44].

1.2 Current Limitations and Motivation for Study

Toward the goal of designing machines that can assist humans under challenging scenarios such as disaster situations [45, 46], planetary exploration [47], and battlefields, as well as mundane day-to-day tasks, many sophisticated legged robots, along with even complicated algorithms to control them have been developed. Yet, a significant gap exists between the agility, mobility, and dynamic stability shown by the existing legged robots and their biological counterparts. The numerous falls observed during the DARPA Robotics Challenge [48] corroborates this observation. In particular, a *fundamental gap exists in the understanding and development of reliable and scalable algorithms* required for real-time planning and robust control for the safe locomotion of legged machines in complex environments. This hinders the real-world deployment of legged robots.

The necessity to create scalable and computationally tractable algorithms to synthesize robust controllers for agile locomotion: With advancements in design and manufacturing, legged robot hardware has become increasingly complex with sophisticated sensory and actuation capabilities. This increasing complexity puts stringent requirements on the algorithms needed for stable and robust locomotion. Algorithms have to process loads of sensory data, arrive at a reliably accurate estimation of the system's state, and perform complex mathematical operations using an appropriate dynamical model before prescribing a control action to the multiple links and joints of the robot. Further complexities are manifested with the increasing nonlinearity enabled by the high Degrees of Freedom (DoFs) in state-of-the-art designs. Although the dimensionality and complexity of algorithms seem acceptable for 2D planar robots [16, 49, 50, 51, 52, 53, 54] the same cannot be said in the case of 3D spatial robots [55, 56, 57, 58] navigating complex environments or cooperative locomotion [40, 59] involving multiple dynamical systems. This underlines the importance of developing scalable algorithms to cope with the curse of dimensionality of high DOF legged robots. On the other hand, as an inherently unstable nonlinear system, the prescription of stable control action on time is of utmost importance. Failure to do so results in the destabilization of the system. It can also be potentially detrimental to the legged robot and its surroundings, making hard real-time constraints a desideratum for the control algorithms of legged robots. This conflicting situation of developing algorithms that cope with the increasing complexity while achieving desired motions and without compromising real-time operational constraints motivates the *need to develop scalable and computationally tractable algorithms for synthesizing robust controllers.*

The necessity to accommodate planning and control using hierarchical control architectures: Existing algorithms for legged robots that are designed for the path planning and stabilization of dynamics gaits can be categorized into two sections based on the mathematical approximation of the system dynamics used. These algorithms either use the complete full-order dynamics of the legged robots or an approximation of the full-order dynamics called the reduced-order models (or template models) [60, 61, 62]. Algorithms that use the full-order dynamics of the legged robot come with many challenges. Full-order mod-

els of legged locomotion are inherently nonlinear and unstable with high degrees of freedom. This situation is further exacerbated by (1) the loss of control authority on some degrees of freedom (underactuation) in certain phases of locomotion, (2) the hybrid nature of system dynamics characterized by the continuous-time domains representing the Lagrangian dynamics and the discrete-time dynamics representing the initiation or termination of contacts with the environment. As a result, the algorithms geared toward real-time planning and control of legged robots using full-order models form a subset of general Nonlinear Programming Problems [63, 64, 65, 66] that lack computational tractability (NP-hard) [67, 68] and guarantees for safe real-time operation. These drawbacks render the *full-order model unsuitable as a safe real-time planning algorithm*. This problem is alleviated to some extent by restricting the usage of full-order models in a real-time fashion only for instantaneous and reactive controllers or offline for generating libraries of stable gaits.

Reduced-order models, on the other hand, provide a meaningful low-dimensional realization of the full-order dynamics targeted toward specific behaviors. Owing to the low-dimensional nature of reduced-order models, the planning and control algorithms hitherto developed are computationally tractable and also amenable for further analysis. For example, the popular Linear Inverted Pendulum (LIP) [69, 70, 71] model can be used in a Model Predictive Control framework [72, 73, 74, 75, 76] for the generation of stable locomotion [39, 77, 78, 79, 80, 81] and also develop footstep planning algorithms using capture point techniques [82, 83, 84, 85]. Furthermore, if the reduced-order models considered are linear or linearized (successively at every sample time), the planning algorithms can be transcribed as convex optimization algorithms with favorable properties such as the admission of polynomial-time [86, 87, 88] algorithms necessary for safe and guaranteed real-time operation on physical hardware. Some of the popular reduced-order models used for the generation and stabilization of locomotion in legged robots include the Linear Inverted Pendulum model with variants such as Spring Loaded Inverted Pendulum (SLIP) [89, 90, 91, 92], Asymmetric Spring Loaded Inverted Pendulum (ASLIP) [93], Lateral Leg Spring (LLS) [94] model, Angular Momentum Pendulum Model (AMPM) [95, 96], Reaction Mass Pendulum (RMP) [97, 98] and Single Rigid Body Dynamics [17, 99, 100, 101, 102, 103], Centrodial dynamics [104] and few other planar models [24, 105, 106, 107, 108]. However, to successfully transfer the motion onto the physical hardware, algorithms based on reduced-order models must be supplemented with another control approach to translate the optimal reduced-order trajectories into full-order joint-position and torque commands. These drawbacks render the algorithms developed based on ***reduced-order models unsuitable for a full-stack controller***.

The above two shortcomings of full-order and reduced-order models *motivates the development of hierarchical (layered) control architectures* with reduced-order models at the high level for planning and full-order nonlinear models at the low level for control. This has numerous advantages. First, using a full-order model at the lower level of the layered architecture facilitates the generation of an instantaneous control action that can easily handle minor disturbances effectively. This reactive control action can be supplemented by a planning approach that contemplates the system dynamics over a longer horizon, thereby

prescribing optimal control actions. Moreover, this decomposition of planning and control also provides ample execution times for MPC to plan control actions over a longer horizon while enabling low-level controllers to track the optimal trajectories from the path planner. However, proper care must be taken in choosing the appropriate reduced-order model that captures the significant dynamics required for robust locomotion. Furthermore, the control outputs of the high-level path planner, which become the input for the low-level controller, must be carefully selected.

The necessity to incorporate robustness explicitly into hierarchical control architectures: Traditional feedback controllers for dynamical systems rely on an explicit, well-defined mathematical model. For various practical reasons, approximations are made when devising a system model. How reasonable the mathematical model is with the actual reality is a subtle and complicated question, dictated by the available physical and computational resources and the performance requirements. Uncertainties are then defined as the difference in errors between the mathematical models and reality [109, 110, 111, 112]. Identifying the sources of uncertainty and designing the appropriate algorithms can only improve the controllers' robustness and scope of validity. In the case of a complicated system like a legged robot grappling with unknown environments, the sources of uncertainties abound. These uncertainties can be intrinsic to the system, such as noisy sensors, faulty/imperfect actuators, and communication delays. While these uncertainties can be minimized by investing in more precise and expensive hardware, this can only be done to a certain extent. Some uncertainties manifest themselves in the form of unknown and unmodeled dynamics. For instance, no practicable and comprehensible model of contact friction (usually observed as joint friction in the case of legged robots) is known to fully grasp all the experimentally observed aspects of friction [113, 114, 115] Moreover, the rigid body approximation of the individual links of the robot may be violated when large interaction forces are in play. There also exist extrinsic uncertainties in the form of unplanned contacts and unknown terrains.

This makes the explicit accommodation of uncertainty into control algorithms a first step towards the real-world deployment of legged robots. This *signifies the need to incorporate uncertainty and robustness explicitly*, especially into state-of-the-art layered control architectures. This has a couple of benefits. For instance, legged robots are often tasked to navigate unknown terrains, including hard and rocky surfaces like roads and indoor environments, compliant surfaces like grass and soil, and even discontinuous terrain like rubble and gravel. It may be impractical to know all the properties of the terrain ab-initio to inform the control algorithms, even in the presence of visual sensors. In these scenarios, robust control algorithms designed to accommodate unplanned environmental interactions can extend the variety of terrains a legged robot can negotiate. They can also be used to resolve the performance gaps introduced by the usage of reduced-order locomotion models. Finally, robust control algorithms can also expedite the process of transferring algorithms from simulation to reality, potentially circumventing the assumptions [116] made by typical physics-based simulation environments. Given the current motivation, the following section presents this thesis's key goals and objectives using a series of research tasks.

1.3 Scope, Goals and Objectives

The ***overarching goal*** of this thesis is to formally develop *computationally tractable robust controllers based on nonlinear hybrid systems theory, model predictive control, and optimization for the real-time planning and control of agile locomotion in quadrupedal robots.* This thesis is written with the following ***key objectives*** in mind

(1) Development of multiple hierarchical (layered) control architectures for the real-time planning and control of legged robots

(2) Creation of a computationally tractable Robust Model Predictive Control framework for bridging the gap between reduced and full-order models of locomotion arising in layered control architectures

(3) Development of theoretically stable robust controllers for the safe and robust locomotion of quadrupedal robots

(4) Deployment of the developed control algorithms on physical hardware and validation through extensive numerical and experimental studies on a quadrupedal robot.

To achieve these objectives, the following research tasks have been conceptualized.

Research Task (R1): Hierarchical Control Algorithms based on Model Predictive Control and Hybrid Zero Dynamics: This research task aims to develop hierarchical control algorithms for real-time planning and control of quadrupedal robots. Towards this objective, we propose a two-level hierarchical control architecture in which the higher level is based on a reduced-order Model Predictive Control, and the lower level is based on full-order QP-based virtual constraints nonlinear controller. Specifically, two Model Predictive Control architectures are explored, 1) An Event-based MPC scheme that generates the optimal center of mass (COM) trajectories using a reduced-order Linear Inverted Pendulum (LIP) model subject to the feasibility of the net ground reaction forces (GRF), and 2) A time-based MPC scheme that computes the optimal COM and ground reaction forces (GRF) using the reduced-order Single Rigid Body Dynamics model. The optimal COM trajectories in the event-based MPC and the optimal COM trajectories, along with the ground reaction forces in the time-based MPC, are then tracked by the low-level virtual constraints controller. The event-based MPC scheme is numerically validated on Ghost Robotics Vision 60 platform in a physics-based simulation environment. It has achieved a reduction in the computational burden associated with real-time planning-based MPC schemes. However, owing to the quasi-static nature of the optimal trajectories generated by the LIP model, we explored a time-based MPC scheme using Single Rigid Body Dynamics. This time-based MPC scheme is also numerically validated using the mathematical model of the Unitree A1 robot.

Research Task (R2): Robust Model Predictive Control for bridging the gap between reduced and full-order models and the generation of robust blind lo-

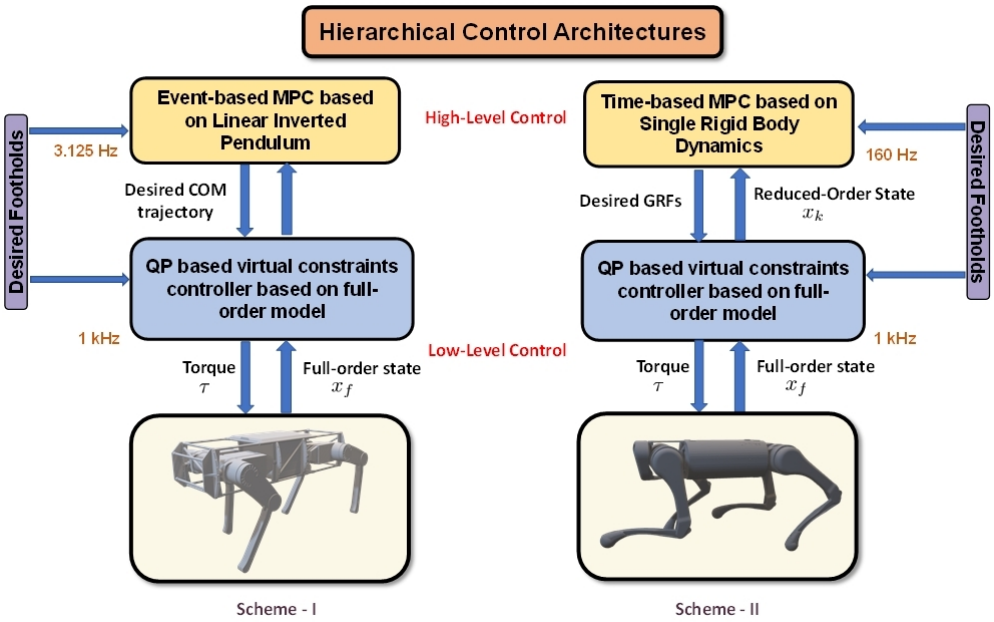


Figure 1.3: Research Task R1 illustrating two hierarchical control architectures.

comotion: The frameworks in **R1** assume a template-based reduced-order model at the higher level of the hierarchical control architecture to remain computationally tractable. However, the abstraction and unmodeled dynamics in template models significantly increase the gap between reduced- and full-order models limiting the full scope and potential of the robot. This research task aims to develop a computationally tractable robust model predictive control (RMPC) scheme based on convex quadratic programs (QP) to bridge this gap. The RMPC framework considers the single rigid body model subject to a set of unmodeled dynamics and plans for the optimal reduced-order trajectory and ground reaction forces (GRFs). The generated optimal GRFs of the high-level RMPC are then mapped to the full-order model using a low-level nonlinear controller based on virtual constraints and QP. The key innovation of the proposed RMPC framework is that it allows the integration of the hierarchical controller with Reinforcement Learning (RL) techniques to train an MLP to compute the vertices of the uncertainty set numerically. The proposed hierarchical control algorithm is validated numerically and experimentally for robust and blind locomotion of the A1 quadrupedal robot on different indoor and outdoor terrains and at different speeds. The numerical analysis of the RMPC suggests significant improvement in the performance of the rough terrain locomotion compared to the nominal MPC. In particular, during our numerical analysis comparing the percentage success rate of the RMPC algorithm with nominal MPC over 550 randomly generated rough terrains, we see an improvement of around 60%. Our experimental studies also indicate a significant reduction in the gap between the reduced- and full-order models by comparing the desired and actual GRFs computed by the proposed high- and low-level controllers.

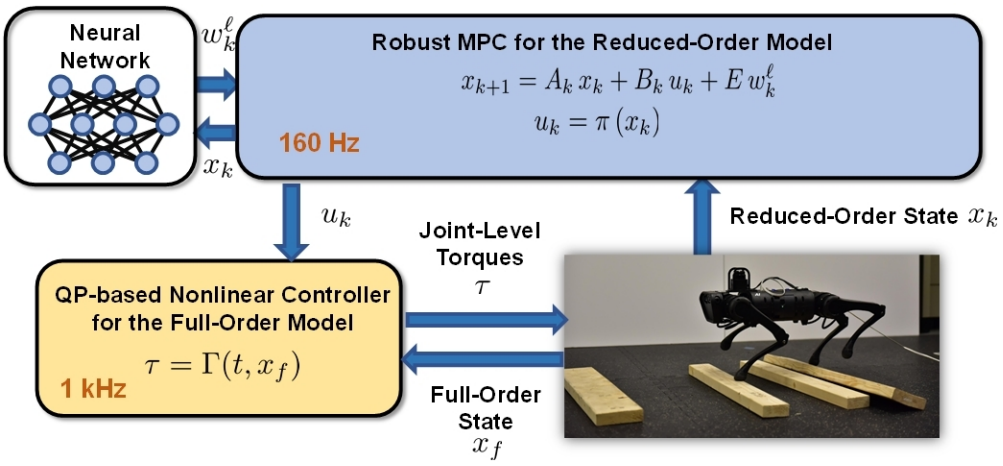


Figure 1.4: Research Task R2 illustrating Robust Model Predictive Controller.

Research Task (R3): Robust Control architecture for safe and robust locomotion of quadrupedal robot Within the frameworks of **R1** and **R2**, hierarchical control architectures are explored with the possibility of resolving the gaps between different locomotion models. The objective of this research task is to develop theoretically robust optimal MPCs. Toward this objective, we present a robust control approach that can explicitly accommodate disturbances. We propose an algorithm that builds on the conventional optimization-based control stack, especially convex quadratic programming (QP) based model predictive controllers (MPCs). Under some mild conditions, we linearize the resulting closed-loop nonlinear discrete-time system around the gait to obtain a linear time-varying (LTV) system. Using suitable coordinate transformation via discrete-time Floquet theory [117, 118], the resulting periodic LTV system is transformed into a linear time-invariant (LTI) system. This LTI system can now be analyzed using the classical $\mathcal{H}_2$- and $\mathcal{H}_\infty$-[119, 120] control methods and parameterized robust state feedback laws can be obtained as solutions to linear matrix inequalities (LMIs) [121]. In this work, we also rigorously establish the existence of $\mathcal{H}_2$- and $\mathcal{H}_\infty$- feedback control solutions. The proposed controllers are then utilized to generate and validate robust feedback control solutions for the periodic locomotion of the A1 quadrupedal robot both through simulation and experiments.

Thesis Structure We address the above-mentioned research tasks in three parts. Part I addresses the research task **R1** and focuses on developing two hierarchical control architectures. These layered architectures involve an event-based MPC and time-based MPC together with a QP-based nonlinear virtual constraints controller. Part II addresses the research task **R2** and focuses on developing a computationally tractable Robust Model Predictive Control scheme to resolve the performance gaps between full-order and reduced-order model approximations. Part III presents the development of a robust controller for the safe locomotion of a quadrupedal robot with theoretical guarantees. Concluding remarks and future work inline with research task **R3** are also discussed in Part III.

1.4 Significance and Impact

This thesis explores the development of stable controllers for the robust locomotion of legged robots. In particular, the innovative algorithms proposed in this thesis are theoretically significant as 1) they methodically synthesize stable, scalable, and computationally tractable hierarchical control architectures needed for robust locomotion, 2) they contemplate the most general mathematical models of locomotion involving underactuation, nonlinear and hybrid nature of system dynamics 3) they resolve the performance gaps between the reduced- and full-order models of locomotion, thereby reinforcing stability and robustness 4) they accommodate learning based techniques into the formal frameworks of hybrid systems theory and model predictive control within a layered architecture, exploiting the best of both the fields. Finally, the algorithms developed in this thesis can be easily transferred to physical robots using state-of-the-art software packages, bridging the gap between theory and implementation.

Apart from this, findings from this study will equip legged robots with versatile mobility in complex unstructured environments, enabling real-world uses for legged robots. Moreover, this work will add to the body of knowledge that is needed to create legged robots that can be sent into environments that are dangerous for humans, such as disaster scenes, battlefields, and planetary exploration. Finally, the proposed work can serve as a preliminary step toward integrating data-driven learning-based approaches with the traditional model-based control approaches.

1.5 Background Work

This section gives an overview of the techniques employed for the successful generation of locomotion in legged robots. As a system with complicated dynamics, successful locomotion in legged robots entails generating motion for almost all degrees of freedom and control. The following two subsections outline the general methods involved in the motion planning and control of legged robots. Although most of the techniques described in this section apply to a broad range of dynamical systems and robots (wheeled and legged), we restrict the following discussion to legged robots. Furthermore, care is taken to outline the subsections in a platform-agnostic manner (i.e., applicable to bipedal, quadrupedal, and other legged robots)

1.5.1 State-of-the-art Motion Planners

Motion planning for legged robots involves answering two questions. The first one entails the generation of continuous motions for each joint. The second is the combinatorial question of deciding which articulated appendages (hands or legs) should make and break contact

with the environment and in what particular order. This observation has led researchers to develop various motion planning approaches focusing on either one question at a time or both. The choice of pursuing each question is dictated by the computational resources available and the level of agile locomotion needed.

Footstep Planning

The purpose of a footstep planner is to generate a curated list of footstep locations that a robot can safely utilize to reach the desired location (navigation problem), generate desired motion (agile locomotion), or accomplish both. Given this definition, the following subsection details two approaches to footstep planning.

Graph-Based Approaches Graph-based approaches typically involve decision-making with a precomputed set of actions called the action set $\mathcal{A}$. These actions can range from a simple displacement of one footstep to another to a complex transition between either the floating base or complete configurations of the robot. Compounding each action together forms a tree in the search space of actions, which can be effectively explored with the help of a graph traversal algorithm. Algorithms in this domain include brute-force search techniques such as breadth-first, uniform-cost, depth-first, depth-first iterative deepening search, bidirectional search, and heuristic-guided searches such as A* [122] and its other variants such as D* [123], IDA [124], ARA* [125, 126, 127, 128] or R* [129]. For a survey of graph traversal algorithms and their implementations on legged robots, refer to [122, 130, 131, 132].

These algorithms have been successfully employed on a wide variety of legged such as Honda Asimov [133, 134, 135], Little Dog [136], Nao Humanoid [137], and various other robots [138]. The advantage of graph-based approaches is their simple but effective pruning of the tree of actions whenever an obstacle is encountered. However, these approaches fall short because they rely on an existing balance controller. Furthermore, these approaches require a trade-off between a small action set that limits the search space and a large action set that increases the complexity of the search. In addition, the efficiency of heuristic search algorithms is heavily dictated by the underlying heuristic, which may require more work to obtain ab initio.

Optimization-Based Approaches Contrary to the graph-based approaches, the search space of optimization-based approaches involves the continuous motion of the end-effectors' position. This alleviates the problem of dealing with a small and fixed action set of graph-based approaches and allows for more foothold locations. In this regard, several researchers have used Linear Model Predictive Control approaches to plan a sequence of footsteps in real-time [78]. Other approaches in this realm involved planning for both the footstep locations and their orientation using a sequence of optimization problems [77]. Finally, footstep planning in the context of obstacle avoidance is handled using Mixed Integer Convex programs. By discretizing the allowable workspace into a set of convex regions and appropriately choosing each convex region through binary decision variables, these algorithms achieved ag-

ile locomotion on 2D planar [23, 139] and 3D robots [140, 141, 142].

Whole Body Motions

Trajectory Optimization and Optimal Control Optimal control [143] - a control design process based on optimization is powerful tool for the synthesis of motions for highly dynamic systems like legged robots. This can be attributed to several reasons. First, it provides a natural language in which the motion plans of robots can be easily described. For example, optimization constraints enforce various physical limitations of the robot, such as actuator limits and collision avoidance, whereas the objective (cost function) is used to achieve desired motions while respecting system dynamics. Second, the optimal control framework is sufficiently general to accommodate a wide range of dynamical systems, whether it is linear or nonlinear, stochastic or full-determined, continuous or discrete-time. Furthermore, optimal control also lends itself well to numerical solutions. The following subsection briefly discusses the optimal control problem as it applies to robotics and discusses various numerical methods for the solution of OCP.

We start by considering a continuous-time control system

$$\dot{x} = f(t, x, u), \quad x(t_0) = x_0 \tag{1.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ represent the state and control in $n-$ and $m-$ dimensional Euclidean spaces respectively, x_0 represent the initial state at time t_0. We then define a cost functional $\mathcal{J}$ as

$$\mathcal{J}(u) := \int_{t_0}^{t_f} L(t, x(t), u(t))dt + K(t_f, x_f) \tag{1.2}$$

where, L and K are given functions known as running cost and terminal cost respectively, x_f is the final state at the final time t_f, both of which can either be free or fixed. The general optimal control problem involves finding the control $u(t)$ which minimizes the cost functional $\mathcal{J}(u)$ over all possible admissible controls. Formally this can be written as

$$\begin{aligned}
\underset{u}{min.} \quad & \mathcal{J}(u) = \int_{t_0}^{t_f} L(t, x(t), u(t))dt + K(t_f, x_f) && \textbf{(Cost-Functional)} \\
s.t. \quad & \dot{x} = f(t, x, u) && \textbf{(Dynamics)} \\
& x(t_0) = x_0 && \textbf{(Initial Condition)} \\
& C(x(t), u(t)) \leq 0 && \textbf{(Path Constraints)} \\
& x(t_f) \in \mathcal{X}_f, t_f \in \mathbb{R} && \textbf{(Final Conditions)}
\end{aligned} \tag{1.3}$$

where $\mathcal{X}_f$ represents the final state, and $C(x(t), u(t))$ represent the constraints on state and control inputs. This optimal control problem is said to be of Bolza form and two further

distinctions can be made depending on whether $L \equiv 0$ (Mayer form), and $K \equiv 0$ (Lagrange form). It is important to note that this is an infinite-dimensional optimization problem over the space of admissible controls.

Traditional optimal control problems trace their roots in the calculus of variations, where the problem is to find a curve amongst a family of curves (such as $\mathcal{C}^1$ curves with fixed endpoints) that minimizes a cost functional. The optimality conditions for such problems result in the famous Euler-Lagrange equations. In contrast, optimal control problems are concerned with generating control laws that minimize a cost functional while respecting the systems' dynamics. Necessary conditions for optimality for optimal control problems (without path constraints) can be obtained using Pontryagin maximum principle [143], which involves solving a boundary value ordinary differential equation (ODE). Using the principle of dynamic programming, both sufficient and necessary conditions for the optimal control problems can be obtained by solving Hamilton-Bellman-Jacobi [143] partial differential equations. While the maximum principle is computationally more tractable as it only involves ODE, HJB equations provide more information in the form of state-feedback control law, which can be extremely useful for practical applications.

With few minor exceptions when dealing with low-dimensional systems, analytical methods (maximum principle and HJB equations) remain intractable for solving general optimal control problems. With the advent of computers in the 20^{th} century, various numerical methods have lent themselves well to the practical solution of OCPs. These numerical methods, called trajectory optimization (TO) problems [144, 145, 146], involve the transcription of the infinite-dimensional OCP to finite-dimensional nonlinear programming problem (NLP) [63, 64, 65, 66] and then solving the corresponding NLP to optimality. A general NLP can be formulated as

$$\begin{aligned} \underset{z}{min.} \quad & f(z) && \textbf{(Cost Function)} \\ & c(z) = 0 && \textbf{(Equality Constraints)} \\ & g(z) \leq 0 && \textbf{(Inequality Constraints)} \end{aligned} \tag{1.4}$$

In contrast to the OCP, NLP is typically characterized by a finite number of decision variables, z and constraints. The process of converting the infinite-dimensional OCP into finite-dimensional NLP is called transcription, and a plethora of numerical methods exist to facilitate this. As mentioned by [144, 145, 146], the transcription methods fall roughly into two camps: direct and indirect methods. In the direct methods of TO, the OCP is first discretized with a finite number of decision variables and constraints, and the resulting NLP is solved to optimality. On the other hand, indirect methods of TO analytically construct the necessary and sufficient optimality conditions, which are then discretized and solved.

While the indirect method generally provides more accurate solutions, the lack of closed-form analytical expressions for optimality conditions for general OCP makes indirect methods impractical with few exceptions. One such exception is Differential Dynamic Programming

[102], which uses local approximations of OCP using linear and quadratic approximations of dynamics and cost functions and iteratively solves the HJB equations. These methods have been implemented on various bipedal [147, 148] and quadrupedal robots [102, 149, 150] , some of them even handling inequality constraints [147]. Direct Methods can be further classified into shooting and collocation methods depending on whether the trajectory is optimized using forward simulations or polynomials, respectively. Shooting methods do not accommodate path constraints and, as such, have little application in legged robots.

Other Notable Approaches

In contrast to the discrete footstep planning and continuous time trajectory optimization-based whole-body planners, several researchers have combined these problems using a single large optimization problem [151, 152, 153]. The drawback of this approach is that it involves the computation of very large nonlinear programs. The size and non-convexity of these optimization problems restrict these NLP's to local solutions and necessitate the use of proper initialization of the optimization problem. Recent times have seen a surge in the use of learning-based techniques, particularly deep Reinforcement Learning (RL) to generate optimal motion plans for legged robots [27, 28, 29, 154, 155, 156, 157, 158, 159]. The advancements in parallel computing technology and the availability of high-quality open-source software have led to this growth. Using a multi-layer perceptron network and gradient based optimization methods, RL techniques have shown significant robustness and agility in navigating various rough terrains [19, 154].

1.5.2 State-of-the-art Control Algorithms

The successful generation of trajectories with motion planning algorithms does not guarantee the desired motion, even in simulated environments. To accomplish the desired motion, the generated optimal trajectories have to be tracked in a stable manner. This subsection will briefly discuss various control algorithms as they apply to legged robots.

Hybrid Systems Theory

General Locomotion of legged robots, such as walking, running, and other gaits, display a rhythmic pattern of motions that can, under certain assumptions, be considered periodic. This periodic locomotion of legged robots can be treated as an interplay between the continuous-time dynamics, modeled using ODEs, and discrete-time dynamics, modeled using impact maps, and can thus be studied using tools from hybrid-systems theory [54, 160, 161].

Some of the very early works that considered the hybrid nature of locomotion can be seen in the field of *passive walking* [162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177]. Instead of focusing on creating feedback controllers for sophisticated mathematical models of locomotion, passive walking is primarily focused on designing a mechanical system

that can approximately realize stable locomotion. Research along these lines started with the work of McGeer [162, 163], who designed, created and, analyzed passive dynamic walkers that were able to walk down slopes under reasonable initial conditions. Following this, many researchers have looked into various aspects of passive dynamic walking, including the injection of small amounts of energy through electrical actuators into Cornell Walkers [164, 165], MIT biped [166, 167], and through a pneumatic actuator in the Delft biped [168, 169]. In terms of control, researchers have developed passivity-based control strategies [170, 171] to analyze passive dynamic walking. Although passive dynamic walkers arguably generate the most anthropomorphic locomotion, they are restricted to simple walking.

Control approaches that consider this hybrid nature of legged locomotion and develop stable controllers include hybrid reduction [178, 179, 180, 181, 182], controlled symmetries [171], geometric reduction [178, 179, 179, 180, 183], transverse linearization [184, 185] and hybrid zero dynamics [52, 54, 186, 187]. Hybrid reduction, controlled symmetries, and geometric reduction are developed on the premise of energy shaping and assume the availability of full actuation. However, during locomotion, legged robots display a wide variety of actuation phases, from underactuated to fully actuated to over-actuated. To handle various actuation phases (especially under-actuation), techniques such as transverse linearization and hybrid zero dynamics (HZD) are developed. While transverse linearization has yet to be successful in hardware implementation, hybrid zero dynamics has been successfully implemented in simulation and experiments on high-dimensional legged robots. These prospects proffer HZD as a reliable control approach and is the focus of the next subsection.

Hybrid Zero Dynamics

In the HZD approach, a set of kinematic constraints are defined in the configuration space of the mechanical system. These constraints, also called virtual constraints, capture the desired locomotion of the legged robot, which are then enforced (zeroed) with the help of a feedback control law, for example, I-O linearization, to induce an asymptotically stable periodic gait. When these virtual constraints are satisfied both during the continuous- and discrete-time dynamics, then the zero dynamics manifold (the space of virtual constraints) becomes invariant to the hybrid flow of the system, a phenomenon called the hybrid invariance. In order to investigate the stability of this periodic locomotion, researchers have used the method of Poincaré sections. In this method, the hybrid systems flow is replaced with a discrete-time map called the Poincaré return map, which maps the evolution of system's state from a point on hyperplane transversal to the periodic orbit back to the hyperplane. The stability of this discrete-time Poincaré map indeed determines the stability of the hybrid system. If the Poincare map depends on a configurable set of parameters, then these parameters can be tuned to induce desired stability properties. This idea has been exploited to design exponentially stable robust controllers for stable locomotion of robots using the Bilinear Matrix Inequality (BMI) algorithm [188, 189]. With this ideas, HZD has been seen widespread success with implementations on a wide variety of 2D and 3D monopods [93, 190, 191], bipedal [16, 50, 188, 189, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204], quadrupedal robots [41, 42, 59, 205, 206, 207, 208, 209, 210, 211, 212], multi-agent cooperative locomotion

[39, 40] as well 3D powered prosthetic legs [213, 214, 215] and exoskeletons [216, 217, 218].

Model Predictive Control

Model Predictive Control [72, 73, 74, 75, 76] with its numerous names such as receding-horizon control, dynamic matrix control, rolling-horizon control, and dynamic linear programming traces its origin to chemical process and industrial control where it was used to control plants with sufficiently 'slow' dynamics. With advancements in computing technology, MPC has gained widespread success due to its simplicity and ability to naturally encode hard constraints on state and control even for complex systems. The basic idea of MPC is to solve a discrete-time finite horizon optimal control problem at every sample time using the system's current state as the initial condition. This yields a sequence of control actions, and the first control action in the sequence is used to control the system. This process is repeated at every sample time. This contrasts with the OCP, where a finite horizon optimal control problem is usually solved offline to obtain a precomputed feedback control law. By trading arduous offline computations of control law with repeated online computations of a finite-dimensional optimization problem, MPC achieves desired performance, as evident in numerous applications in aerospace, chemical and process industry, and even robotics. From a controls perspective, various properties of MPC have been studied, such as closed-loop stability, recursive feasibility, and optimality. Depending on whether the system model used is linear, nonlinear, or hybrid, MPC algorithms can be classified into linear, nonlinear, and hybrid MPC. For an overview of the methods used in MPC, refer to [72, 73, 74, 75, 76]

In the legged robotics community, MPC has seen widespread acclaim with successful implementations on various bipedal and quadrupedal robots. Different MPC implementations on various legged robots can be classified based on the underlying robot model used. For example, using a linear system model such as Linear Inverted Pendulum (LIP), researchers have shown stable locomotion on both bipedal and quadrupedal robots [39, 77, 78, 79, 80, 81, 150, 219, 220]. The advantage of using LIP models is that the resulting MPC is a convex optimization problem, particularly QP which guarantees feasibility along with efficient real-time implementation using state-of-the-art QP solvers. Furthermore, LIP dynamics can also be used to develop push recovery algorithms using capture point techniques [82, 83, 84, 85], or develop stable controllers using Divergent Component of Motion (DCM) [79, 80, 221, 222] and ZMP [223, 224, 225, 226] stability criterion. Other variants of LIP such as SLIP, ASLIP, LLS, AMPM, and RMP have also seen successful implementation. However, due to the coarse, linear approximation of system dynamics, the gaits generated using LIP are mostly quasi-static. Ignoring the leg dynamics and considering the floating base of the robot results in the Single Rigid Body Dynamics Model. Recent times, many researchers have used SRBD [17, 99, 100, 101, 102, 103] either directly or using successive linearization to develop efficient real-time algorithms. Their success can be seen in the agile locomotion maneuvers performed by MIT Cheetah [18], ANYMAL [227] and A1 robot [228]. Finally, some researchers have combined the Single Rigid Body Dynamics Model with the kinematics of legs

to simultaneously optimize both the ground reaction forces and footsteps of the ANYMAL robot [150, 219, 220].

Whole Body Controllers

In robotics, a relatively straightforward method for computing control actions is via inverse dynamics. In this approach, given the desired motions of the robot in joint space, the required control action is obtained by inverting the multi-body dynamics. While the implementation of the inverse dynamics algorithm is straightforward for fixed-base manipulators with full actuation, this is not the case for floating-base underactuated legged robots. Most of the inverse dynamics methods for floating-base systems involve some form of pseudo-inversion. A commonly used control method in the family of inverse dynamics methods is called the Whole-Body Control (WBC) scheme, which entails the description of the dynamics of the entire system along with desired motion, often specified as tasks. These tasks can range from simple arm manipulation or rotation to complex tasks such as maintaining postural control or satisfying complex physical criteria such as ZMP, Capture Point, or Centroidal Momentum. Furthermore, WBC also features prioritization of tasks based on their importance in achieving the desired motion through a hierarchy. For instance, balancing the robot is often given a higher priority compared to secondary tasks such as manipulation.

Controllers developed within WBC architectures can be classified based on two criteria. First, the output produced by the WBC: 1) velocity-based WBC produces joint-level velocities, and 2)torque-based WBC produces joint torques. Another distinction can be made based on the approach used to obtain the solution 1) closed-form analytical expressions and 2) optimization-based methods. While closed-form analytical expressions are easier to solve, optimization-based methods can handle inequality constraints such as friction-cone constraints, necessary for generating suitable ground reaction forces and subsequent balance. Within the legged robotics community, several implementation of WBC controllers exist, see for example [141, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238]. Although, WBC controllers provide a powerful way of controlling the robot, the question of whether the aggregation of all task control policies results in properly damped behavior and convergence to desired trajectories within reasonable time frames still remains open. In other words, the guarantees for the cumulative stability of the entire system are lacking.

1.6 Modeling Tools

This section outlines some of the basic tools used for modeling legged robots. This includes methods for obtaining rigid body dynamics and the forward time simulation of rigid body dynamics with impacts through physics-based simulation environments.

1.6.1 MultiBody Dynamics

Legged robots are often classified as multi-body systems, which is a collection of rigid bodies held together by joints and actuated by force elements in order to produce a meaningful desired motion. The role of joints is to impose kinematic constraints between the two connecting rigid bodies. Various joints exist depending on the type of relative motions allowed. For example, the revolute joint allows rotation about a single axis between the two connecting bodies. It should be noted that every joint imposes kinematic constraints between connecting bodies by creating required constraint forces in the relevant directions.

From now on, we shall refer to every single rigid body in the system of rigid bodies as **links**. Rigid Body systems can be broadly classified into two categories, namely fixed-base and floating-base. If at least one of the links is firmly attached to the ground, then the system is called a fixed-base rigid body system and a floating-base rigid body system otherwise. Examples of fixed-base rigid body systems include manipulator arms, and examples of floating base rigid body systems include legged robots like bipeds, quadrupeds, etc. The structure of a rigid body system [239] can be defined in terms of a graph $\mathcal{G}$ with vertex set $\mathcal{V}$ representing the rigid bodies, the edge set $\mathcal{E}$ representing the connection (also the joints) between rigid bodies. In order to provide a reference frame, the ground is also added as one of the rigid bodies in the graph structure of the rigid body system. At this point, one can distinguish rigid body systems into open loop and closed loop kinematic chains depending on whether the corresponding graph contains cycles.

At this point, two approaches exist for computing the equations of motion for the system of rigid bodies. The simplest of these would be to consider the dynamics of each link as that of the single rigid body and then apply the constraints pertaining to each of the joints. This method is typical of simple dynamics algorithms and is the workhorse of physics engines like Bullet [240], ODE [241]. This method results in large sparse matrices, and algorithms must employ special sparse matrix techniques to achieve good computational efficiency. Other approaches include computing the equations of motion for the spanning tree and reapplying the rest of the constraints. For more details on algorithms for obtaining the rigid body dynamics, please refer to Section 3.2 of [239]. In this thesis, we use the second approach of obtaining the equations of motion for the spanning tree of the system of rigid bodies.

This approach starts by working with a minimal set of coordinates (equal to the number of Dof). For example, for fixed-base rigid body systems i, we can work with the relative angle between two links of manipulator arm (q_1, q_2, q_3, q_4). Consider an n-link rigid body system with generalized coordinates given by $q_1, q_2, ..., q_n$. The standard lagrangian ($L(t, q, \dot{q})$: $\mathbb{R} \times \mathcal{X} \rightarrow \mathbb{R}$) of the rigid body system taken as the difference of Kinetic and Potential energies turns out to be

$$L(t, q, \dot{q}) = \frac{1}{2}\dot{q}^T D(q)\dot{q} - V(q) \tag{1.5}$$

where $D(q)$ is the symmetric positive definite mass inertia matrix, $V(q)$ and $\frac{1}{2}\dot{q}^T D(q)\dot{q}$ is the

potential and kinetic energy of the rigid body system system respectively. The stationary points of this Lagrangian functional, also called Euler Lagrange equations [143] are the equations of motion for the system of rigid bodies. In the presence of external torques and constraints, the Euler-Lagrange equations take the form.

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \Gamma \tag{1.6}$$

where Γ is the vector of external constraints applied. This can either be the input toques provided by the actuators or the ground reaction forces from the contacting legs.

In the above derivation, the joint constraints (if any) are implicitly considered in the formulation of the Lagrangian function and the system is treated as a whole. With this set of new configuration space, we can rewrite the dynamics in the well known form of manipulator equation as

$$\begin{aligned} D(q)\ddot{q} + C(q,\dot{q}) + G(q) &= Bu + J^T F \\ J(q)\ddot{q} + \dot{J}q &= 0 \end{aligned} \tag{1.7}$$

Here $C(q,\dot{q})$ is the sum of centrifugal and Coriolis forces, $G(q)$ is the force due to gravity, B is the input distribution matrix (used to provide gear ratios for motor actuators), and u is the vector of actuation torques. Here $J(q)$ is the jacobian of the contact function $p(q)$. In other words $J(q) = \frac{\partial p}{\partial t}$. For a complete derivation, please refer to [242]. F is the vector of forces at the contacting legs, called the ground reaction forces. In the case of frictional contacts, further constraints on F in the form of friction cone must be enforced. This approach of obtaining the equations of motion by computing the derivatives of lagrangian is called the Lagrangian approach and has a computational complexity of $\mathcal{O}(n^4)$. Other efficient approaches like the Newton-Euler approach (with a computational complexity of $\mathcal{O}(n)$), Composite Rigid Body Algorithm (for computing the mass inertia matrix), and Articulated Rigid Body Algorithm for computing the forward dynamics are also used. As of the writing of this thesis, several packages are available to obtain rigid body dynamics from the model description such as RBDL [243], FROST [244], Pinocchio [245] etc.

1.6.2 Physics Based Simulation Environments

Numerical studies through physics-based simulation environments form an important aspect of validating control algorithms and a first step toward transferring control algorithms from conception to physical hardware. The simulation of rigid bodies with impact entails the solution of non-smooth equations of motion: the dynamics are non-smooth because of the discontinuous nature of non-interpenetration, collision, and adhesion constraints. In this regard, three major approaches tackle the forward time simulation of non-smooth dynamics. The first method is an integrate-detect-restart method [246, 247, 248] which involves simulating the continuous-time dynamics of rigid bodies until a contact event is detected. This contact event is handled in isolation appropriately, and the simulation is restarted

with a different set of initial conditions. Although this method is simple and effective, it is restricted to mechanical systems with fewer degrees of freedom and impacts. The second class of methods called the penalty or regularization methods [249, 250, 251, 252, 253] treats the contacts by means of compliant spring-dashpot linkages. The high stiffness required to simulate the contacts leads to stiff ODEs that are numerically intractable.

Alternative to the penalty methods is the modern and ubiquitous time-stepping schemes. These algorithms model the non-penetration constraint of rigid body interactions as a Linear Complementarity Problem (LCP) [254]. In the absence of friction, the existence and uniqueness of the LCP problem are guaranteed [254, 255], i.e., the LCP problem in the absence of friction is well-defined. In the presence of friction, especially when using the Coloumb model, the LCP problem is not forward time complete [256, 257]: no solutions are known to exist in certain configurations, and multiple solutions in certain configurations. By switching from a force acceleration model to a velocity impulse model, researchers are able to find solutions to the LCP at all configurations and for multiple impacts [258, 259]. However, solutions to LCP call for an NP-hard computation in its exact form. Modern physics engines relax the strict complementarity condition in the LCP problem and only enforce it approximately [260, 261]. Some of the popular physics engines as of the writing of this thesis include chrono [262], mujoco [263], RAISIM [264], bullet [240], open dynamics engine [241], PhysiX [265], dart [266], simbody [267], havoc [268]. As part of this thesis, we use RAISIM.

At this point, we would like to note that there exists numerous software packages such as FROST [244], CASADI [269] transcribe an OCP which are then solved using NLP solvers such SNOPT [270], IPOPT [271].

1.7 Relevant Publications

As part of this research, four research articles are written, two published by the IEEE Robotics and Automation Letters (RAL), one in the proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), and one in preparation to be submitted to the IEEE Control Systems Letters. Chapters 2, 3 and 4 comprise of the materials of the articles as listed below:

[**J1**] **A. Pandala**, Aaron. D. Ames, and K. Akbari Hamed, "$\mathcal{H}_2/\mathcal{H}_\infty$ Optimal Predictive Controller for Periodic Legged Locomotion: A Linear Matrix Inequality Approach," in preparation for submission to *IEEE Control Systems Letters*.

[**J2**] **A. Pandala**, R. T. Fawcett, U. Rosolia, Aaron D. Ames, and K. Akbari Hamed, "Robust Predictive Control for Quadrupedal Locomotion: Learning the Gap between Reduced- and Full-Order Models," *IEEE Robotics and Automation Letters*, Vol. 7, Issue 3, pp. 6622-6629, July 2022.

[**J3**] K. Akbari Hamed, J. Kim, and **A. Pandala**, "Quadrupedal locomotion via event-based predictive control and QP-based virtual constraints," *IEEE Robotics and Automation Letters*, Vol. 5, Issue 3, pp. 4463-4470, July 2020.

[**C1**] **A. Pandala**, V. R. Kamidi, and K. Akbari Hamed, "Decentralized control schemes for stable quadrupedal locomotion: A decomposition approach from centralized controllers," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3975-3981, Las Vegas, NV, October 2020

In addition, I have contributed to the following journal articles published by the ASME Journal of Dynamic Systems, Measurement, and Control, IEEE Controls System and Letters as well as the conferences, IEEE International Conference on Intelligent Robots and Systems (IROS) and American Control Conference (ACC) .

[**J3**] R. T. Fawcett, **A. Pandala**, A. D. Ames, and K. Akbari Hamed, "Robust stabilization of periodic gaits for quadrupedal locomotion via QP-based virtual constraint controllers," IEEE Control Systems Letters, pp. 1736-1741, December 2021

[**J4**] R. T. Fawcett, **A. Pandala**, J. Kim, and K. Akbari Hamed, "Real-time planning and nonlinear control for quadrupedal locomotion with articulated tails," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 143, Issue. 7, pp. 071004-1-071004-15, Jul, 2021.

[**C2**] K. Akbari Hamed, V. R. Kamidi, **A. Pandala**, W. Ma, and A. D. Ames, "Distributed feedback controllers for stable cooperative locomotion of quadrupedal robots: A virtual constraint approach," American Control Conference (ACC), Denver, CO, pp. 5314-5321, July 2020

[**C3**] J. Martin, V. R. Kamidi, **A. Pandala**, R. T. Fawcett, and K. Akbari Hamed, "Exponentially stabilizing and time-varying virtual constraint controllers for dynamic quadrupedal bounding," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3914-3921, Las Vegas, NV, October 2020

1.8 Outline

This preliminary thesis document is divided into three parts and four chapters. A brief outline of each of the successive chapters is provided below.

Chapter - 2 establishes and discusses the preliminaries and foundational framework required for developing layered control architectures. In particular, hierarchical control architectures are developed with reduced-order based MPC at the high-level and full-order virtual constraints controllers at the low level. Two hierarchical control architectures based on the

LIP model and SRBD model are developed and analyzed using numerical studies of two quadrupedal robots.

Chapter - 3 develops a computationally tractable Robust Model Predictive Control (RMPC) scheme based on convex quadratic programs (QP), to bridge the gap between reduced- and full-order models of locomotion. Following the work of Chapter - 2, the RMPC scheme considers the SRBD model subject to a set of unmodeled dynamics and plans for the optimal reduced-order trajectory and ground reaction forces (GRFs). The generated optimal GRFs of the high-level RMPC are then mapped to the low-level nonlinear controller. We leverage deep reinforcement learning to train a neural network to compute the set of unmodeled dynamics and validate the controller via extensive numerical simulations and experiments for robust and blind locomotion of the A1 quadrupedal robot on different terrains.

Chapter - 4 presents the development of robust stable optimal MPCs for the periodic locomotion of quadrupedal robot by expanding on the existing optimization control stack. Under some mild conditions, we linearize the resulting discrete-time, closed-loop nonlinear control system around the gait to obtain a linear time-invariant (LTI) system using discrete-time Floquet theory. The LTI system is then analyzed to accommodate parametric uncertainties and to synthesize robust optimal $\mathcal{H}_2$- and $\mathcal{H}_\infty$- feedback controllers via linear matrix inequalities (LMIs). Efforts are taken to theoretically establish the existence of such controllers and rigorous numerical and experimental validations are provided using A1 qaudrupedal robot.

Chapter - 5 presents concluding remarks and future directions to be conducted after the prelim exam. It briefly summarizes the work done as part of this thesis, which consists of the materials of Chapters 2, 3, and 4. It also provides future research directions that can be pursued using the ideas developed in this thesis.

Part I

Hierarchical Control Architectures based on Event- and Time-Based MPC for the Locomotion of Quadrupedal Robots

Chapter 2

Reduced Order Models

2.1 Hierarchical Control Architectures - Motivation

The objective of this chapter is to explore hierarchical control architectures for the real-time path planning and control of quadrupedal robots. In particular, two layered control architectures are developed based on Model Predictive Control and Quadratic Programming based virtual constraints controller. The first layered architecture employs an event-based MPC scheme using Linear Inverted Pendulum (LIP) model at the higher level and generates optimal center of mass (COM) trajectories subject to the feasibility of net ground reaction forces (GRF). Furthermore, the event-based MPC scheme is operated at the beginning of each continuous time domain to reduce the computational effort associated with the time-based MPC schemes. In order to overcome the shortcomings of the LIP model, the second layered architecture employs a time-based MPC scheme at the higher level using Single Rigid Body Dynamics. Both the layered architectures consists of a QP based virtual constraints controller at the lower level to track the optimal reduced order trajectories generated by each of the MPC schemes. Finally, the event-based and time-based MPC schemes are validated numerically using the dynamics full-order models of Ghost Robotics Vision60 and Unitree A1 robots respectively.

2.2 Event Based MPC Scheme

2.2.1 Linear Inverted Pendulum Model

One of the simplest templates in the class of reduced-order models is the Linear Inverted Pendulum (LIP) model. The concept of LIP model is closely related to the notion of Zero Moment Point and the identical Center of Pressure (CoP). Introduced through a series of observations on the stability of walkers by Vukubrativic [223, 272], ZMP is primarily defined as the *point on the ground where the reaction forces between the feet and the ground produce no horizontal moment*. This notion of ZMP is used to achieve walking by planning the optimal center of mass trajectories subject to the feasibility of net GRF's while satisfying the ZMP stability criterion. The ZMP stability criterion states that the *movement of a legged robot is stable as long as the ZMP remains within the convex hull of the stance feet*. In the LIP

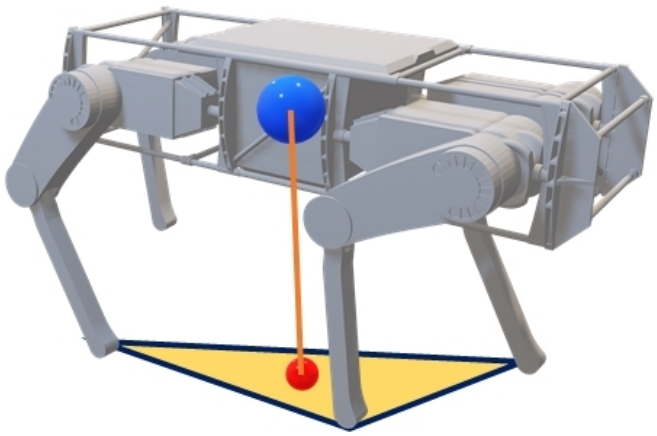


Figure 2.1: Illustration of LIP model.

model, the system dynamics of the legged robot are approximated as a point mass balancing on telescopic legs. This simplification results in a linear ordinary differential equations with the ZMP as a function of the center of mass. One of the other key assumptions behind the LIP model is that the motion of the COM of the robot is restricted to the horizontal plane and the height of the COM from the ground remains constant. Although these restrictions hinder the development of agile locomotion in legged robots, we show in the following sections the development of an event-based MPC scheme that results in stable locomotion with less computational burden.

The traditional LIP model can be described by the following linear ODEs [69, 70, 71]

$$\begin{bmatrix} \ddot{r}_x \\ \ddot{r}_y \end{bmatrix} = \frac{g}{r_z} \begin{bmatrix} r_x - u_x \\ r_y - u_y \end{bmatrix}, \tag{2.1}$$

where $r := \text{col}(r_x, r_y) \in \mathbb{R}^2$ denotes the cartesian coordinates of the COM with respect to the inertial world frame, projected onto the horizontal xy-plane, r_z represents the vertical component of the COM (a.k.a. height of the torso), g is the acceleration due to gravity, and $u := \text{col}(u_x, u_y) \in \mathbb{R}^2$ denotes the cartesian coordinates of the COP. We now define the following LIP state vector

$$x^{\text{COM}} := \text{col}(r_x, \dot{r}_x, r_y, \dot{r}_y) \in \mathbb{R}^4 \tag{2.2}$$

By transforming the linear ODE in equation 2.1 into state-space form and then applying a zero-order hold (ZOH) discretization scheme with sampling time T_d results in the following

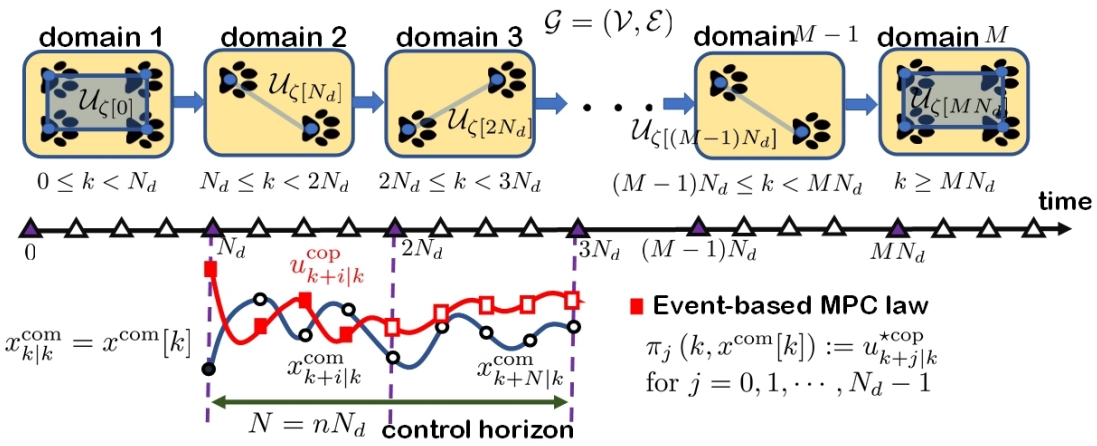


Figure 2.2: Event-Based MPC and Domains of Locomotion.

discrete time linear time invariant dynamics

$$x^{\text{COM}}[k+1] = A_d\, x^{\text{COM}}[k] + B_d\, u^{\text{COP}}[k], \tag{2.3}$$

where $k \in \mathbb{Z}_{\geq 0}$ represents the current sampling time with $A_d \in \mathbb{R}^{4\times 4}$ and $B_d \in \mathbb{R}^{4\times 2}$ being the state and input distribution matrices, respectively.

Steering Problem

Legged locomotion is typically characterized by both continuous time domains represented by Lagrangian dynamics and discrete-time dynamics represented by the change in the contacts of the ground. To address general locomotion patterns involving an arbitrary sequence of double-, triple- or even quadruple-contact domains, we consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V}$ represents the continuous time domains and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represents the transition between them. We are now interested in driving the discrete-time dynamics (2.3) from an initial state to a final state over M continuous-time domains, for some positive integer M. Futhermore, we assume that each continuous time domain $\mathcal{V}$ is sampled with N_d grid points. A *domain indicator function* is now defined as $\zeta : \mathbb{Z}_{\geq 0} \to \{1, 2, \cdots, M\}$ by $\zeta[k] := \lfloor \frac{k}{N_d} \rfloor + 1$ for $0 \leq k < MN_d$ and $\zeta[k] := M$ for $k \geq MN_d$ to assign the domain index for every time sample $k \in \mathbb{Z}_{\geq 0}$. Here, $\lfloor . \rfloor$ represents the floor function. In addition, we enforce two constraints on the evolution of system dynamics, one each for stability and feasibility. First, using the ZMP criterion, we restrict the center of pressure $u^{\text{COP}}[k]$ to remain within the convex hull of the stance feet (a.k.a support polygon) as

$$u^{\text{COP}}[k] \in \mathcal{U}_{\zeta[k]} \tag{2.4}$$

for all $k \in \mathbb{Z}_{\geq 0}$, where $\mathcal{U}_{\zeta[k]} \subset \mathbb{R}^2$ is the corresponding convex hull for the domain $\zeta[k]$ (see Fig. 2.2). Following the definition of a convex hull as the convex combination of all its vertices, we reformulate the ZMP criterion (2.4) as

$$\mathbf{0} \leq \lambda[k] \leq \mathbf{1}, \;\; \mathbf{1}^\top \lambda[k] = 1, \;\; u^{\mathrm{COP}}[k] = C_{\zeta[k]} \lambda[k]. \tag{2.5}$$

where $C_{\zeta[k]}$ represents the *contact coordinates matrix* for the domain $\zeta[k]$ whose columns encompass the cartesian coordinates of stance feet (a.k.a. vertices of support polygon). Furthermore, $\lambda[k]$ is a time-varying vector and $\mathbf{0}$ and $\mathbf{1}$ denote vectors whose elements are zero and one, respectively.

In addition, for the feasibility of the LIP model, the net force must lie in the friction cone, i.e., $F_{\mathrm{net}} \in \mathcal{FC}$ defined as

$$\mathcal{FC} := \{\mathrm{col}(F_x, F_y, F_z) \,|\, F_z \geq 0, \; \sqrt{F_x^2 + F_y^2} \leq \mu F_z\} \tag{2.6}$$

Here F_x, F_y, F_z represent the x-,y- and z- components of ground reaction force for some static friction coefficient μ

Remark 2.1. Throughout this thesis, we use a linearized version of the friction cone as $\mathcal{FC} := \{\mathrm{col}(F_x, F_y, F_z) \,|\, F_z \geq 0, \pm F_x < \frac{\mu}{\sqrt{2}} F_z, \pm F_y < \frac{\mu}{\sqrt{2}} F_z\}$. This is done to ensure that the control algorithms developed can be transcribed as a Quadratic Program with linear equality and inequality constraints, guaranteeing safe real time operations.

The linearized friction-cone condition together with (2.1) can be expressed as

$$\Phi \, x^{\mathrm{COM}}[k] + \Psi \, u^{\mathrm{COP}}[k] \leq \eta, \quad \forall k \in \mathbb{Z}_{\geq 0} \tag{2.7}$$

for some proper Φ and Ψ matrices and some proper η vector.

Problem 1 (Optimal Steering Problem)**.** *For a given locomotion graph $\mathcal{G}$, a domain indicator function ζ, a set of known contact coordinates matrices $\{C_{\zeta[k]}\}_{k \in \mathbb{Z}_{\geq 0}}$, an initial state x_0^{COM}, a final state x_f^{COM}, togehter with a steering time $T_f \in \mathbb{Z}_{\geq 0}$, the optimal steering problem consists of finding an optimal sequence of control (i.e., COP) inputs $u^{COP}[k]$ for $0 \leq k \leq T_f - 1$ that steer (2.3) from x_0^{COM} to x_f^{COM} subject to (2.5) and (2.7).*

2.2.2 Event-Based MPC Formulation

To address Problem 1, we set up an event-based MPC that is solved at the beginning of each continuous-time domain with some control horizon $N = nN_d$ and $n \geq 1$. In particular, for every time sample $k = mN_d$ with $m \in \mathbb{Z}_{\geq 0}$, we consider the following finite horizon optimal

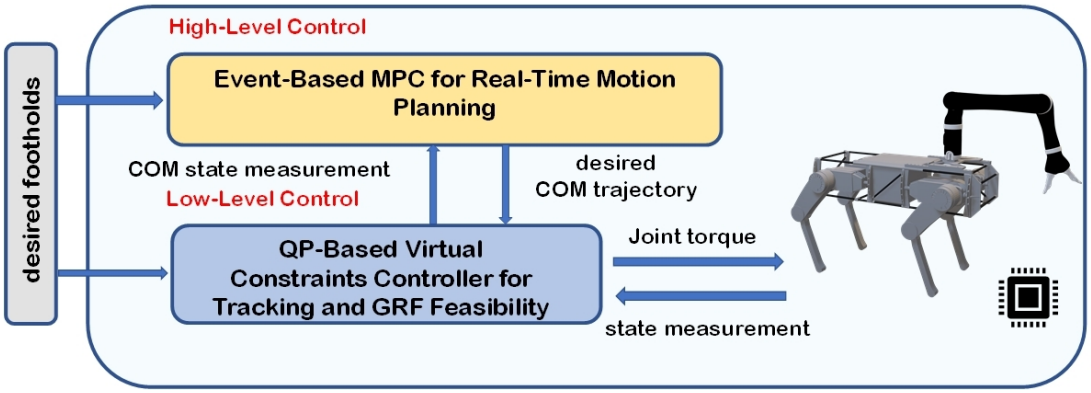


Figure 2.3: Illustration of Event-Based MPC.

control problem

$$\begin{aligned}\min_{U^{\mathrm{COP}}_{k\to k+N-1|k}} \ & \mathcal{J}_k\left(x^{\mathrm{COM}}[k], U^{\mathrm{COP}}_{k\to k+N-1|k}\right) = p\left(x^{\mathrm{COM}}_{k+N|k}\right) + \sum_{i=0}^{N-1} \mathcal{L}\left(x^{\mathrm{COM}}_{k+i|k}, u^{\mathrm{COP}}_{k+i|k}\right) \\ \text{s.t.} \quad & x^{\mathrm{COM}}_{k+i+1|k} = A_d\, x^{\mathrm{COM}}_{k+i|k} + B_d\, u^{\mathrm{COP}}_{k+i|k} \\ & \Phi\, x^{\mathrm{COM}}_{k+i|k} + \Psi\, u^{\mathrm{COP}}_{k+i|k} \leq \eta \\ & u^{\mathrm{COP}}_{k+i|i} \in \mathcal{U}_{\zeta[k+i]}, \qquad i = 0, 1, \cdots, N-1, \end{aligned} \tag{2.8}$$

where $U^{\mathrm{COP}}_{k\to k+N-1|k} := \mathrm{col}(u^{\mathrm{COP}}_{k|k}, \cdots, u^{\mathrm{COP}}_{k+N-1|k})$ and $x^{\mathrm{COM}}_{k+i|k}$ represents the estimated state vector at time $k+i$ predicted at time k according to the recursive law $x^{\mathrm{COM}}_{k+i+1|k} = A_d\, x^{\mathrm{COM}}_{k+i|k} + B_d\, u^{\mathrm{COP}}_{k+i|k}$ starting from the current state $x^{\mathrm{COM}}_{k|k} := x^{\mathrm{COM}}[k]$. In an analogous manner, $u^{\mathrm{COP}}_{k+i|k}$ denotes the COP input at time $k+i$ computed at time k. Furthermore, $p(x^{\mathrm{COM}}_{k+N|k})$ and $\mathcal{L}(x^{\mathrm{COM}}_{k+i|k}, u^{\mathrm{COP}}_{k+i|k})$ are the terminal and stage costs, respectively, defined as $p(x^{\mathrm{COM}}_{k+N|k}) := \|x^{\mathrm{COM}}_{k+N|k} - d^{\mathrm{COM}}_{k+N|k}\|^2_P$ and $\mathcal{L}(x^{\mathrm{COM}}_{k+i|k}, u^{\mathrm{COP}}_{k+i|k}) := \|x^{\mathrm{COM}}_{k+i|k} - d^{\mathrm{COM}}_{k+i|k}\|^2_Q + \|u^{\mathrm{COP}}_{k+i|k}\|^2_R$ for some positive definite matrices $P \in \mathbb{R}^{4\times4}$, $Q \in \mathbb{R}^{4\times4}$, and $R \in \mathbb{R}^{2\times2}$, in which $\|z\|^2_P := z^\top P z$. In our notation, $d^{\mathrm{COM}}_{k+i|k}$ represents a desired state trajectory for $x^{\mathrm{COM}}_{k+i|k}$ that is smooth in i (e.g., linear) while starting at the current state $x^{\mathrm{COM}}[k]$ and ending at the final state x^{COM}_f. Let $U^\star_{k\to k+N-1|k} := \mathrm{col}(u^{\star\mathrm{COP}}_{k|k}, \cdots, u^{\star\mathrm{COP}}_{k+N-1|k})$ be the optimal solution of the problem (2.8). Then in our proposed approach, the first N_d components of $U^\star_{k\to k+N-1|k}$, that correspond to the time samples of the current continuous-time domain, are employed to the system (2.3) (see Fig. 2.2), that is,

$$u^{\mathrm{COP}}[k+j] = u^{\star\mathrm{COP}}_{k+j|k}, \quad j = 0, 1, \cdots, N_d - 1. \tag{2.9}$$

Remark 2.2. We remark that the MPC formulation (2.8) together with (2.5) can be expressed as QP in terms of the decisions variables $\{x^{\mathrm{COM}}_{k+i|k}\}_{i=1}^{N}$, $\{u^{\mathrm{COP}}_{k+i|k}\}_{i=0}^{N-1}$, and $\{\lambda_{k+i|k}\}_{i=0}^{N-1}$ to retain the sparsity structure of [273]. To make the cost function of this QP positive

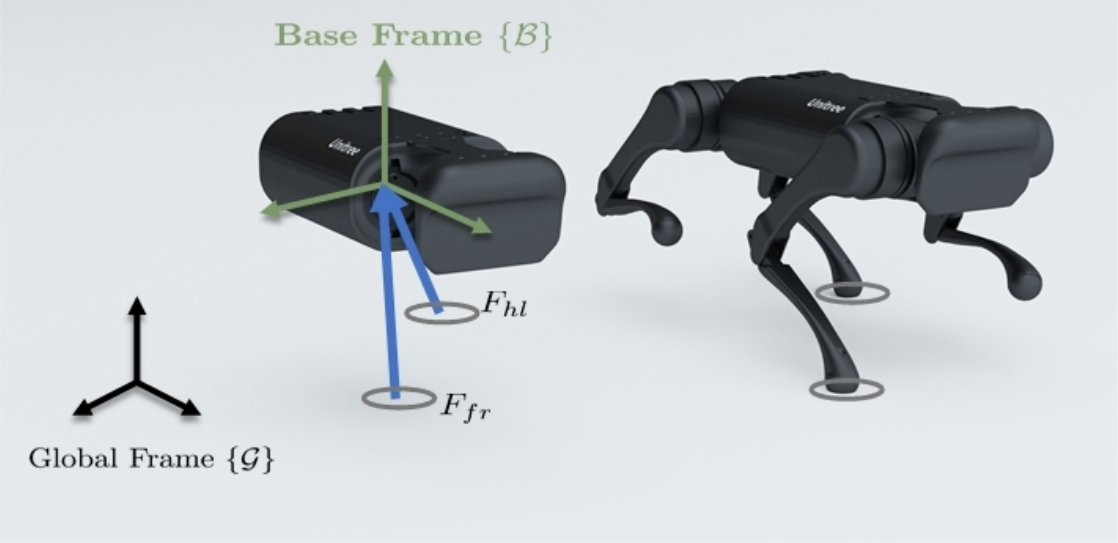


Figure 2.4: Illustration of Single Rigid Body Dynamics.

definite in terms of all decision variables, one can add a term corresponding to $\lambda_{k+i|k}$, i.e., $\mathcal{J}_k = p(x^{\text{COM}}_{k+N|k}) + \sum_{i=0}^{N-1} \mathcal{L}(x^{\text{COM}}_{k+i|k}, u^{\text{COP}}_{k+i|k}) + \mathcal{H}(\lambda_{k+i|k})$, where $\mathcal{H}(\lambda_{k+i|k}) := \|\lambda_{k+i|k} - \lambda^{\text{des}}_{k+i|k}\|^2_{\hat{R}}$ for some desired trajectory $\lambda^{\text{des}}_{k+i|k}$ and some positive definite matrix $\hat{R}$.

2.3 Time-Based MPC

2.3.1 Single Rigid Body Dynamics Model

Single Rigid Body Dynamics

The configuration or the Degrees of Freedom (**DoF**) of a single rigid body in three dimensional Euclidean space can be represented in terms of its position and orientation or attitude. The configuration space $SE(3) = \mathbb{R}^3 \times SO(3)$ (special euclidean group) of rigid bodies is six dimensional, three dimensions each for position and attitude. Here, we consider two coordinate frames, a non-inertial global frame $_{\mathcal{G}}$ and an inertial frame (moving frame) attached to the single rigid body called body frame $_{\mathcal{B}}$ as seen in Fig. 2.4. From now on, every variable is assumed to be in global frame $(\cdot)_{\mathcal{G}}$ unless explicity mentioned with a subscript $(\cdot)_{\mathcal{B}}$

Let $p \in \mathbb{R}^3$ be the position of the single rigid body in the global frame and let $R_{\mathcal{B}} \in SO(3)$ represent the rotation matrix a.k.a attitude of the rigid body in the body frame. Furthermore, $SO(3) = \{R \in \mathbb{R}^{3\times3} \mid RR^T = R^TR = I_{3\times3}\}$. The convention of using p in global frame and R in base frame is in accordance to the actual measurements available from the robot. Although rotation has several minimal representations in terms of Euler angles, quaternions,

Rodriguez parameters etc, we stick with the matrix notion for sake of clarity. Furthermore, the rate of change of position and the rate of change of $R_{\mathcal{B}}$ are given by linear velocity v and angular velocity $\omega_{\mathcal{B}}$ respectively. Together the tuple $\{p, v, R_{\mathcal{B}}, \omega_{\mathcal{B}}\}$ form the state of single rigid body system. The dynamics of single rigid body can be written as

$$\begin{aligned} \dot{p} &= v \\ m\dot{v} &= \sum_i f_i + mg \\ \dot{R}_{\mathcal{B}} &= R_{\mathcal{B}}\hat{\omega}_{\mathcal{B}} \\ I_{\mathcal{B}}\dot{\omega}_{\mathcal{B}} + \omega_{\mathcal{B}} \times I_{\mathcal{B}}\omega_{\mathcal{B}} &= R_{\mathcal{B}}^T \sum_j M_j \end{aligned} \tag{2.10}$$

where m is the mass of the robot, $I_{\mathcal{B}}$ is the moment of inertia, $f_i \forall\, i \in \{\text{set of external forces}\}$ and $M_j \forall\, j \in \{\text{set of external moments}\}$ are the external forces and moments acting on the rigid body and g is the acceleration due to gravity. Futhermore, we define the skew symmetric hat operator $\hat{(\cdot)}$ such that $\hat{a}b = a \times b$ where $\times$ defines the vector cross product. These dynamics can be obtained follows newtons second law which states that the rate of change of linear and angular momentum is equal to the total force and torque applied on the rigid body. Considering the linear and angular momentum of the rigid body to be the product of mass and velocity and inertia and angular velocity respectively, we have

Linear Momementum	Angular Momementum
$\dfrac{d(mv)}{dt} = \sum_i f_i + mg$	$\dfrac{d(I_{\mathcal{B}}\omega_{\mathcal{B}})}{dt} = R_{\mathcal{B}}^T \sum_j M_j$
$m\dot{v} = \sum_i f_i + mg$	$I_{\mathcal{B}}\dfrac{d(\omega_{\mathcal{B}})}{dt} = R_{\mathcal{B}}^T \sum_j M_j$
	$I_{\mathcal{B}}\dot{\omega}_{\mathcal{B}} + \omega_{\mathcal{B}} \times I_{\mathcal{B}}\omega_{\mathcal{B}} = R_{\mathcal{B}}^T \sum_j M_j$

(2.11)

The term $\omega_{\mathcal{B}} \times I_{\mathcal{B}}\omega_{\mathcal{B}}$ appears since the derivative is taken with respect to a moving frame (a.k.a base frame). Furthermore, since M_j is external torque in global frame, it is pre-multiplied with $R_{\mathcal{B}}$ to convert it back to base frame.

2.3.2 Time-Based MPC Formulation

Given the SRBD dynamics 2.10, we are now in a position to formulate a QP based MPC scheme. As the SRB dynamics involve nonlinear-terms in $R_{\mathcal{B}}$ and $\omega_{\mathcal{B}}$, we use a variation-based linearization followed by vectorization to obtain the discrete-time affine dynamics.

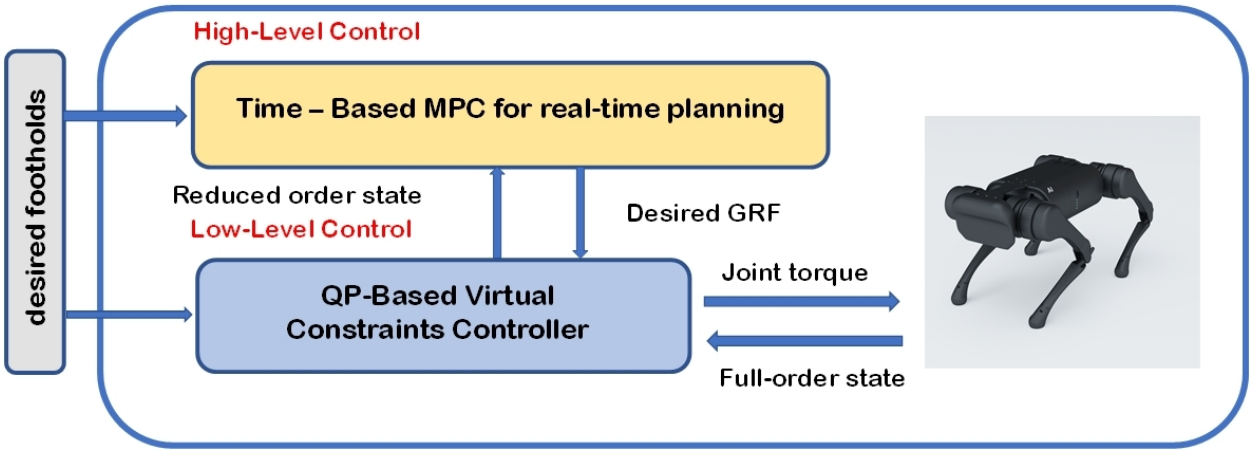


Figure 2.5: Illustration of Time-Based MPC.

Based on the procedure outlined in [100, 101], we choose the new set of state and control vectors as

$$\begin{aligned} x_k &:= [p_k^T \quad \dot{p}_k^T \quad \xi_k^T \quad {}^B\omega_k^T]^T \in \mathbb{R}^{12} \\ \delta u_k &:= [\delta u_{1,k}^T \quad \delta u_{2,k}^T \quad \delta u_{3,k}^T \quad \delta u_{4,k}^T]^T \in \mathbb{R}^{12}, \end{aligned} \tag{2.12}$$

where p_k and $\dot{p}_k$ represent the COM position and velocity, ξ_k^T is a three-dimensional vector in the tangent space of the operating point $R_{\mathcal{B}}$ and $\omega_k]$ represents the angular velocity in base frame. Furthermore δu_i represents the variation of the GRF from the operating point $u_{i.op}$ for the i^{th} contacting leg. With this defintion and state and control vector, the discrete-time affine dynamics could be expressed in the state-space form:

$$x_{t+k+1|t} = A|_{op} \cdot x_{t+k|t} + B|_{op} \cdot \delta u_{t+k|t} + d|_{op}, \tag{2.13}$$

where $A|_{op} \in \mathbb{R}^{n\times n}, B|_{op} \in \mathbb{R}^{n\times m}$, and $d|_{op} \in \mathbb{R}^n$ are matrices constructed by the measurements at the operating point. With this, the nonlinear SRB dynamics have been linearized about the operating point to result in a locally-valid linear time-varying (LTV) system. This system can now be stabilized to track reference trajectories.

We use the following Quadratic Program to generate optimal reduced-order SRB trajectories

$$\text{min.} \quad \ell_T(x_{t+N|t}) + \sum_{k=0}^{N-1} \ell(x_{t+k|t}, \delta u_{t+k|t}) \tag{2.14a}$$

$$\text{s.t.} \quad x_{t+k+1|t} = Ax_{t+k|t} + B\delta u_{t+k|t} + d \tag{2.14b}$$

$$\delta u_{t+k|t} \in \mathbb{U}_k, k = 0, \cdots, N-1 \tag{2.14c}$$

$$x_{t|t} = x(t) = x_{op}, \tag{2.14d}$$

Here $\mathbb{U}$ represents the linearized friction cone conditions and x_{op} is the measured operating point at the current sampling time. The cost function of the MPC consists of two terms, running cost ℓ and terminal cost ℓ_T. The running cost is a composition of five terms as

$$||x_k - x_{k,d}||_Q^2 = ||e_{p_k}||_{Q_p}^2 + ||e_{\dot{p}_k}||_{Q_{\dot{p}}}^2 + ||e_{R_k}||_{Q_R}^2 + ||e_{\omega_k}||_{Q_\omega}^2 + ||u_k - u_{d,k}||_{R_u}^2 \tag{2.15}$$

where $||x||_Q^2$ is a shorthand notation of the matrix norm $x^T Q x$ where Q is a positive definite matrix; $x_{d,k}$. The error terms e_{pk} and $e_{\dot{p}k}$ denote the deviation of the predicted state from the desired trajectories As

$$\begin{aligned} e_{pk} &:= p_k - p_k^{des} \\ e_{\dot{p}k} &:= \dot{p}_k - p_k^{des} \end{aligned} \tag{2.16}$$

For the orientation and angular velocity, we use the following error functions

$$\begin{aligned} e_{R_k} &= \log(R_{d,k}^T \cdot R_{op})^\vee + \xi_k \\ e_{\omega_k} &= \omega_k - R_k^T R_{d,k} \omega_{d,k}, \end{aligned} \tag{2.17}$$

where $\log(\cdot)$ represents the matrix logarithm. Similar error functions are used for terminal cost, albeit with different weighting factors. Using the following new optimization variable z defined as

$$z = [\delta u_0^T, x_1^T, \cdots, \delta u_{N-1}^T, x_N^T]^T \tag{2.18}$$

for some positive horizon $N \geq 0$, the linear MPC 2.14 can be cast into a standard Quadratic Program as

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} z^T P z + c^T z \\ \text{subject to} \quad & A_{ineq} \cdot z \leq b_{ineq} \\ & A_{eq} \cdot z = b_{eq}, \end{aligned} \tag{2.19}$$

2.4 Low-Level Controllers

2.4.1 Virtual Constrains Based Low-Level Controller

The objective of this section is to present a QP-based nonlinear controller that can asymptotically stabilize locomotion patterns for multi-domain hybrid dynamical models of locomotion while keeping the individual GRFs in the friction cone. Here, we make use of the concept of virtual constraints to design this nonlinear controller. Virtual constraints are defined as a set

of kinematic constraints that can be imposed by the action of feedback control laws to coordinate the motion of links within a stride [51, 54, 192]. In particular, virtual constraints are defined as output functions for continuous-time dynamics of locomotion models. The output functions are then regulated via the action of a feedback controller (e.g., I-O linearization [274]). In this work, we address holonomic virtual constraints with relative degree 2 for the purpose of position tracking. The QP-based nonlinear controller will be utilized at the low level of the proposed hierarchical control algorithm. The higher-level of the algorithm will be developed in Section 2.2 for the path planning purpose.

Virtual Constraints

In this work, we define a set of time-varying and holonomic virtual constraints as follows:

$$y(x,t) := h(q,t) := h_0(q) - h_d\left(s,\alpha\right), \tag{2.20}$$

in which $h_0(q)$ denotes a set of holonomic quantities to be controlled, referred to as controlled variables. In addition, $h_d(s,\alpha)$ represents the desired evolution of the controlled variables in terms of the phasing variable s. Here, the desired trajectory $h_d(s,\alpha)$ is taken as a Bézier polynomial with a coefficient matrix α. During the quadruple-contact domains, we choose $h_0(q) \in \mathbb{R}^6$ as the roll, pitch, and yaw angles of the torso together with the COM positions. The idea is to regulate the absolute orientation of the robot while imposing the actual COM coordinates to follow the optimal and desired COM trajectory generated by the MPC in Section 2.2. Here, the coefficient matrix α can be chosen via least squares at the beginning of each domain such that $h_d(s,\alpha)$ has the best fit to the optimal and desired COM trajectory over N_d samples. For double- and triple-contact domains, $h_0(q)$ is augmented with the Cartesian coordinates of the swing leg ends for foot placement. This makes the output function 12- and 9-dimensional for the double- and triple-contact domains, respectively. To control the configuration of the manipulator, we augment $h_0(q)$ and $h_d(s,\alpha)$ by the Cartesian coordinates of the end-effector (EE) and its desired trajectory in the workspace, respectively.

QP-Based I-O Linearization

Differentiating the output function (2.20) along (1.7) results in the following output dynamics

$$\begin{aligned}\ddot{y} &= \mathrm{L}_g\mathrm{L}_f y(x,t)\,\tau + \mathrm{L}_w\mathrm{L}_f y(x,t)\,F + \mathrm{L}_f^2 y(x,t) + \frac{\partial^2 y}{\partial t^2}(x,t)\\ &= -K_D\,\dot{y} - K_P\,y,\end{aligned} \tag{2.21}$$

in which K_P and K_D are positive definite matrices. In addition, $\mathrm{L}_g\mathrm{L}_f y$, $\mathrm{L}_w\mathrm{L}_f y$, and $\mathrm{L}_f^2 y$ are Lie derivatives To compute the required torques that drive the outputs to zero, one would need to solve for τ from (2.21). However, since the contact force measurements are

not available for the studied robot, one would need to estimate the contact forces $F = \text{col}\{F_\ell | \, \ell \in \mathcal{C}\}$. To address this problem, we assume a rigid contact model with the walking surface. The rigid contact assumption together with (1.7) then yields

$$\ddot{p}^{\text{st}} = \text{L}_g\text{L}_f p^{\text{st}}(x)\,\tau + \text{L}_w\text{L}_f p^{\text{st}}(x)\,F + \text{L}_f^2 p^{\text{st}}(x) = 0, \tag{2.22}$$

where $\text{L}_g\text{L}_f p^{\text{st}}(x), \text{L}_w\text{L}_f p^{\text{st}}(x), \text{L}_f^2 p^{\text{st}}(x)$ represent the corresponding Lie derivatives. We now set up the following real-time QP that computes the minimal 2-norm torque as

$$\begin{aligned} \min_{(\tau,F,\omega)} \quad & \frac{1}{2}\|\tau\|_2^2 + \frac{\gamma}{2}\|\omega\|_2^2 \\ \text{s.t.} \quad & \text{L}_g\text{L}_f y\,\tau + \text{L}_w\text{L}_f y\,F + \text{L}_f^2 y + \frac{\partial^2 y}{\partial t^2} + \omega = v_{\text{PD}}(y,\dot{y}) \\ & \text{L}_g\text{L}_f p^{\text{st}}\,\tau + \text{L}_w\text{L}_f p^{\text{st}}\,F + \text{L}_f^2 p^{\text{st}} = 0 \\ & F_\ell \in \mathcal{FC}, \quad \ell \in \mathcal{C}, \quad \tau_{\min} \le \tau \le \tau_{\max}, \end{aligned} \tag{2.23}$$

in which $v_{\text{PD}}(y,\dot{y}) := -K_P\,y - K_D\,\dot{y}$ is the PD action. We introduce a defect variable ω in the equality constraint of QP that corresponds to the output dynamics in case the decoupling matrix $\text{L}_g\text{L}_f y$ is singular. To minimize the effect of the defect variable, we then add a quadratic term $\frac{\gamma}{2}\|\omega\|_2^2$ to the cost function of the QP to ensure that the 2-norm of the defect variable is as small as possible. Here, $\gamma > 0$ is a weighting factor and $\tau_{\min}$ and $\tau_{\max}$ are the admissible bounds on the torques.

2.5 Numerical Results

2.5.1 Event-Based MPC scheme

The objective of this section is to numerically validate the efficacy of the event-based MPC scheme proposed in Section - 2.2. To this extent, we devise two numerical case-studies. The first case-study involves the simulation of the event-based MPC using only the reduced order LIP model. The second case study entails the complete simulation of the layered control architecture using the full-order models of quadrupedal robot.

Simulation Setup

TestBed: We consider the full-order dynamical model of Ghost Robotics [31] Vision 60 that is augmented by a Kinova robotic manipulator [275] for locomotion and manipulation purpose. The floating-base model of Vision 60 consists of 18 DOFs, 12 of which are actuated. Each of the four legs is composed of 3 actuated DOF's, two at the hip and one at the knee. The DOFS representing the absolute position and orientation of the floating base remain

unactuated. In addition, the Kinova arm is fully actuated with 4 DOF, making the composite robot a 22 DOF mechanical system.

Numerical Parameters: The following numerical parameters are chosen for the simulation of the event based MPC

Table 2.1: Numerical Parameters for the simulation of event-based MPC scheme.

MPC Parameters		MPC Cost Function Gains	
Attribute	Value	Attribute	Value
Sampling Time $[T_d]$	80 (ms)	State Terminal Cost $[P]$	$10^3\, I_{4\times4}$
Grid Points per domain $[N_d]$	4	State Running Cost $[Q]$	I_4
MPC horizon $[N]$	8	Control Terminal Cost $[R]$	$I_{2\times2}$
# of Continuous-Time domains [M]	20	Control Running Cost $[\hat{R}]$	$0.01I$

By selecting the height of the COM to be 0.5 (m) and the friction coefficient at $\mu = 0.4$, the higher-level MPC is solved in an event-based manner approximately every $N_d T_d = 0.32$ seconds.

Reduced Order Simulations

We consider five different directions (i.e., forward, backward, sideways, diagonal, and in-place) of trot gaits with start and stop conditions whose graphs $\mathcal{G}$ consists of $M = 20$ continuous-time domains (see Fig. 2.3). Figure 2.6 depicts the evolution of the COM and COP for the forward and diagonal trot gaits of the discrete-time dynamics (2.3) with the step lengths of $(10, 0)$ (cm) and $(7, 4)$ (cm) in $\mathbb{R}^2$, respectively. Here, we make use of ECOS QP [276] to solve the MPC in MATLAB. Convergence to the target (final) points $(0.82, 0)$ and $(0.58, 0.34)$ is clear. Target points are chosen as the geometric center of the contact points in the last domain. We remark that the proposed MPC problem of this chapter has less computational load compared to (2.3) with $\frac{1}{3}$ of the number of decision variables that are optimized at a slower rate.

Full Order Simulations

Next, we study the full-order model of the robot in RaiSim [264] with rigid contact models. The lower-level QP for I-O linearization has 37 decision variables for both double- and quadruple-contact domains, which is approximately 50% of the number of decisions variables used for the higher-level QP. The lower-level QP is solved with qpSWIFT [277] in RaiSim and $\gamma = 10^7$ at every 1 (ms). The computation time of the MPC on a laptop computer with an Intel(R) Core(TM) i7-5600U CPU 2.60GHz (2 cores) and 8GB of RAM is 0.2528 (ms).

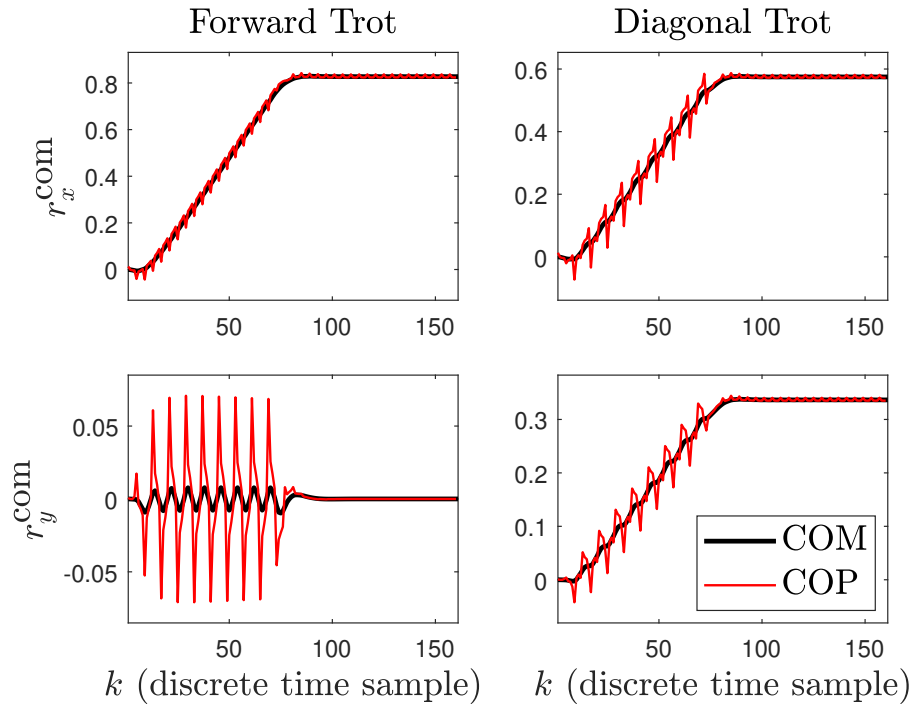


Figure 2.6: Plot of the COM and COP trajectories for the forward and diagonal trot gaits of the reduced-order system. Convergence to the target points $(0.82, 0)$ and $(0.58, 0.34)$ is clear.

The low-level QP problem for the double- and quadruple-contact domains also takes 0.2334 (ms) and 0.4735 (ms), respectively. All state components of the robot, except the absolute Cartesian coordinates of the torso (i.e., p_b), are measurable by an inertial measurement unit as well as encoders. Analogous to the approach of [278], we utilize a Kalman filter to estimate p_b. Figures 2.7(a) and 2.7(b) illustrate the evolution of the virtual constraints and torque inputs (i.e., before the gear ratio) for the full-order model of the forward and diagonal trot gaits with the maximum speed of 0.3 (m/s). To show the robustness of the control system against different contact models, the full-order model is also simulated in MATLAB/Simulink with LuGre contact models [253]. The robot still travels in a stable and robust manner towards the target. To demonstrate the robustness of the proposed control algorithm against the control frequency as well as time delays, we next assume that the control frequency is reduced from 1 kHz to 500 Hz while there is a latency of 2 (ms) in solving QPs. Figure 2.7(d) illustrates the virtual constraints profile and the GRFs in one of the stance legs for this case.

Robustness Analysis: To demonstrate the robustness of the closed-loop system against payloads, we assume that there are two payloads on the robot whose masses are not known for the controller: a payload of 1.4 (kg) in the robot's EE together with a payload of 12 (kg) on the torso (40% uncertainty in the total mass). Figure 2.7(c) illustrates the evolution of the virtual constraints and torques inputs versus time. It is observed that the robot is

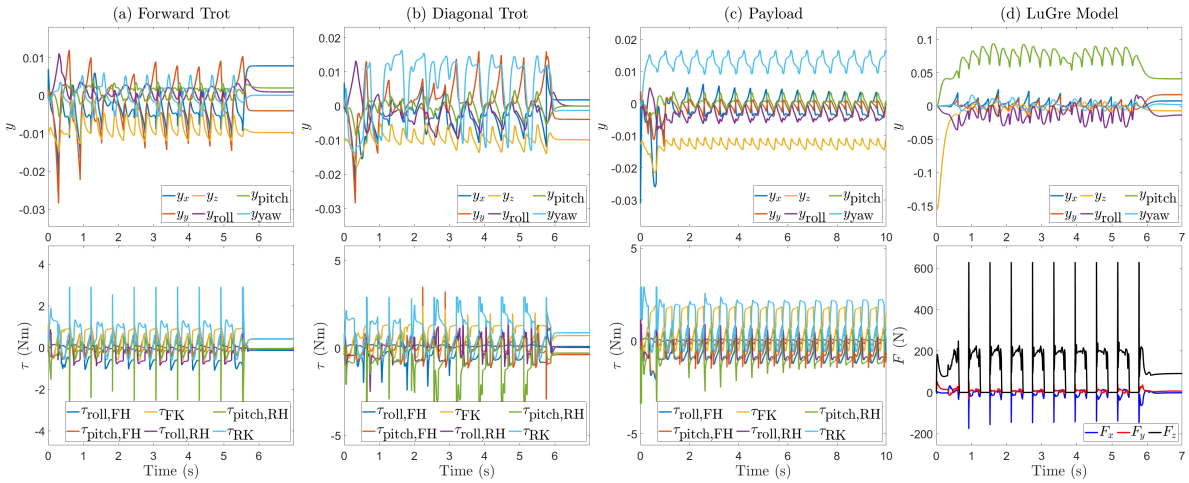


Figure 2.7: Plot of the virtual constraints and torque inputs in RaiSim for (a) forward trot, (b) diagonal trot, and (c) forward trot with payloads. Here, y_x, y_y, y_z, y_{roll}, y_{pitch}, y_{yaw} denote the first six components of the virtual constraints that relate to the absolute position and orientation of the body. In addition, FH, FK, RH, and RK stand for the front hip, front knee, rear hip, and rear knee of the left-hand-side of the robot, respectively. The subscripts "roll" and "pitch" for the torque plots also denote the roll and pitch motions of the hip joints. (d) Plot of the virtual constraints and torques for trot gait subject to the LuGre contact model in MATLAB/Simulink when the control frequency is reduced to 500 Hz with a delay of 2 (ms) in solving QPs.

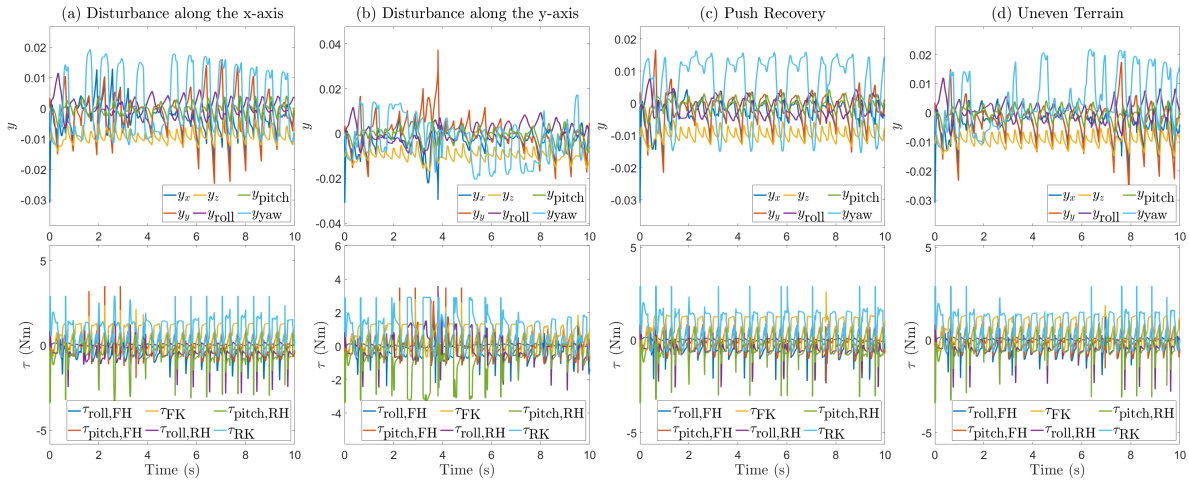


Figure 2.8: Plot of the virtual constraints and torque inputs of the full-order closed-loop system in RaiSim for forward trot with (a) and (b) external disturbances, (c) external push, (d) ground height variations.

capable of rejecting the effect of uncertainties in the total mass during locomotion. To show the robustness of the proposed controller against external forces acting on the robot, we study three different scenarios in RaiSim. In the first and second simulations, we consider persistent external forces along the x- (i.e., direction walking) and y-axes (i.e., lateral direc-

tion), respectively. The forces are taken as sinusoidal disturbance inputs with the period of $0.5\,\pi = 1.57$ (s), where the magnitudes along the x- and y-axes are chosen as 30 (N) and 8 (N), respectively. In the third simulation, we investigate the balance control of the robot while being pushed from the sides. More specifically, we make use of 4 pendulums with the mass of 3 (kg) and length of 2.5 (m) that hit the robot during locomotion. The robot is capable of stable locomotion subject to the above-mentioned external forces. Figures 2.8(a)-(c) depict the evolution of the virtual constraints and torque inputs versus time. To illustrate the robustness of the controller against ground height variations, we finally consider a random sequence of ground heights in the discrete set $\{\pm 1, \pm 2\}$ (cm) over 150 domains of blind locomotion. The time evolution of the outputs and inputs is depicted in Fig. 2.8(d).

2.5.2 Time-Based MPC scheme

The objective of this section is to numerically validate the efficacy of the time-based MPC scheme proposed in Section - 2.3. For this, we simulate forward trotting gait at 1 m/s and a lateral trotting gait at 0.5 m/s as case studies.

Simulation Setup

TestBed: We consider the full-order dynamical model of Unitree [228] A1 quadrupedal robot. The floating-base model of A1 robot consists of 18 DOFs, 12 of which are actuated. Each of the four legs is composed of 3 actuated DOF's, two at the hip and one at the knee. The DOFS representing the absolute position and orientation of the floating base remain unactuated.

Numerical Parameters: The following numerical parameters are chosen for the simulation of the time-based MPC

Table 2.2: Numerical Parameters for the simulation of time-based MPC scheme.

MPC Parameters		MPC Cost Function Gains	
Attribute	Value	Attribute	Value
Sampling Time $[T_d]$	6.25 (ms)	Running Cost $[Q_p]$	$\text{diag}([3e4, 3e4, 3e5])$
MPC horizon $[N]$	7	Running Cost $[Q_{\dot{p}}]$	$\text{diag}([1e5, 1e5, 1e5])$
		Running Cost $[Q_R]$	$\text{diag}([1e3, 1e3, 1e3])$
		Running Cost $[Q_\omega]$	$\text{diag}([5e4, 5e4, 5e4])$
		Running Control Cost $[R_u]$	$\text{diag}([0.1, 0.1, 0.1, 0.1])$

All the stage costs are multiplied by a factor 10 to obtain the corresponding terminal costs.

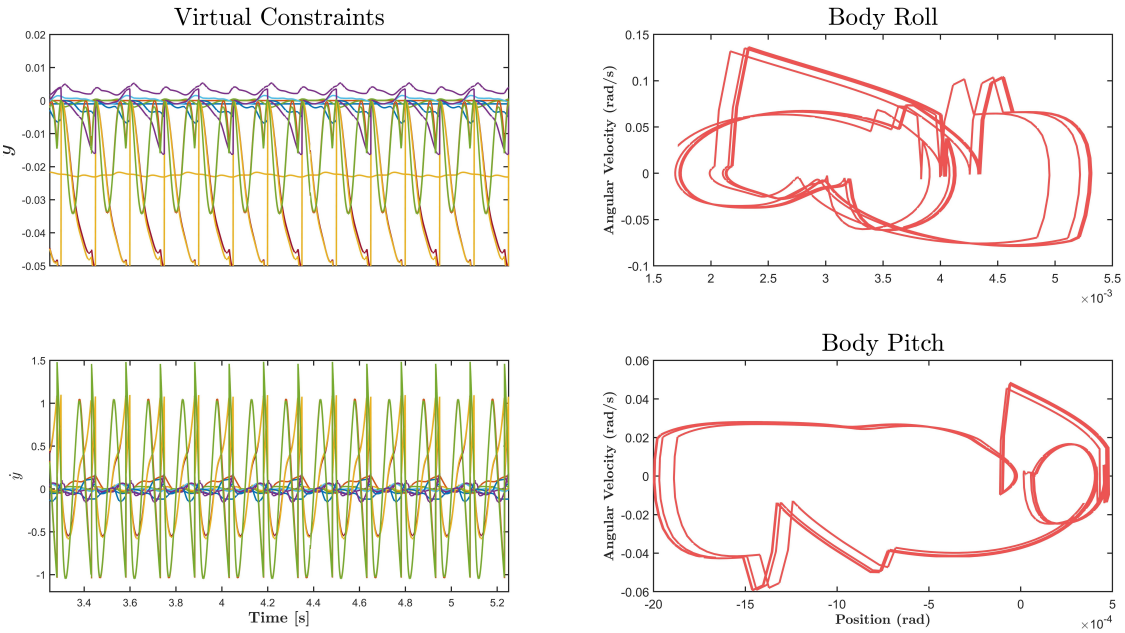


Figure 2.9: Virtual Constraints and Phase Portraits of Body Pitch and Roll for the forward trot gait of A1 quadrupedal robot.

Forward and Lateral Trotting

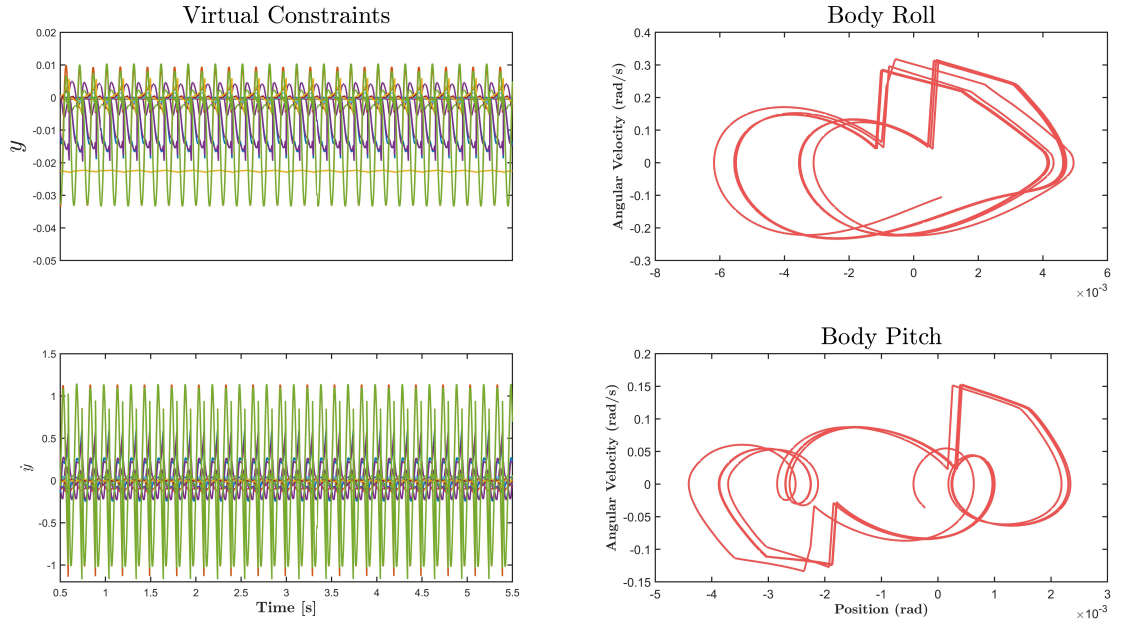


Figure 2.10: Virtual Constraints and Phase Portraits of Body Pitch and Roll for the lateral trot gait of A1 quadrupedal robot.

Figure 2.9 represents the virtual constraints and phase portraits of body pitch body roll of A1 quadrupedal robot trotting in the forward direction at a speed of 1.0m/s. It can be seen from Fig. 2.9 that the virtual constraints converge asymptotically to stable periodic gait. Similarly, Figure 2.10 represents the virtual constraints and phase portraits of body pitch body roll of A1 quadrupedal robot trotting in the lateral direction at a speed of 0.5m/s with asymptotic convergence to stable periodic orbit.

2.6 Summary

This chapter presented two path planning algorithms based on two different reduced order models, LIP dynamics and SRB dynamics. These reduced-order models are then used at the higher layer of a layered control architecture with a low-level QP-based virtual constraints controller. Implementation details of both the layered architectures are presented along with numerical simulations. The rest of the thesis will build upon the time-based MPC scheme presented in this chapter and develop robust predictive controllers.

Part II

Robust Model Predictive Control for Rough Terrain Locomotion

Chapter 3

Robust Model Predictive Control

3.1 Introduction

The objective of this chapter is to develop robust MPC (RMPC) framework that can accomoddate uncertainties explicitly and improve the rough terrain locomotion of quadrupedal robot. This chapter outlines the algorithm, implementation details along with rigorous numerical and experimental case studies.

3.1.1 Motivation

Real-time trajectory planning and control algorithms have been popular methods to realize versatile and dynamic motions in quadrupedal robots rivaling their biological counterparts. Approaches along these lines can be sectioned into two categories: the ones using full-order models and the others using reduced-order (i.e., template-based) models. Reduced-order models provide a low-dimensional realization of the nonlinear full-order dynamics [60]. This makes the planning algorithms computationally tractable and amenable for real-time implementation. Furthermore, if the template models are linear or linearized, the planning algorithms can be transcribed as convex optimization problems. Planning dynamic locomotion that accommodates the full-order models suffers from a computational bottleneck arising from high-dimensionality. Owing to the low-dimensional approximation of the full-order dynamics, reduced-order models inherently ignore some of the rich full-order and nonlinear dynamics that may limit the full potential of the robot. This motivates the development of a planning and control framework that bridges the gap between reduced- and full-order models.

Towards this goal, this chapter presents a layered control approach that learns to close the gap (see Fig. 3.1). At the higher layer, we propose a computationally tractable robust model predictive control (RMPC) framework that generates optimal reduced-order trajectories subject to a set of modeling uncertainties. This set of uncertainties may arise from abstraction and unmodeled dynamics, and is used to bridge the gap between reduced and full-order models. At the lower layer, a nonlinear controller, based on virtual constraints and quadratic programming (QP), is then developed to map the optimal reduced-order trajectories to the full-order locomotion model. The proposed layered control architecture can

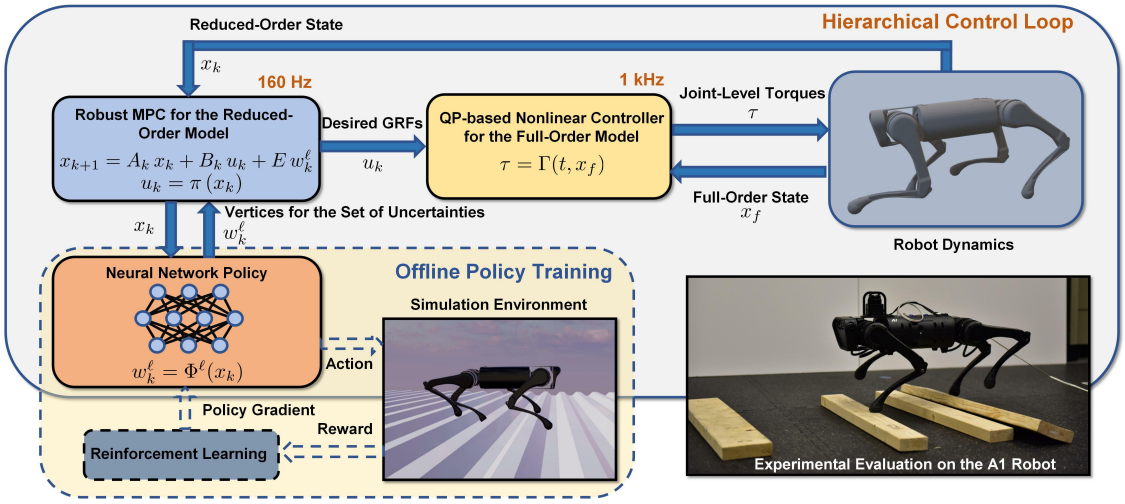


Figure 3.1: Overview of the proposed hierarchical control algorithm, based on RMPC, NN, and QP-based nonlinear controller. The real-time control loops and signals are shown by the solid blocks and arrows, respectively. The offline training procedure is shown by the dashed blocks and signals.

be integrated with deep reinforcement learning (RL) techniques to train a neural network (NN) that computes the set of uncertainties (a.k.a. the gap) for the RMPC algorithm. The robust control framework is finally validated via an extensive collection of numerical and experimental studies for the blind locomotion of the A1 quadrupedal robot on different terrains.

3.1.2 Background, Motivation, and Related Work

Reduced-order models can be easily integrated with the model predictive control (MPC) framework leading to more dynamic behaviors. Some of the popular reduced-order models include the linear inverted pendulum (LIP) model [69], centroidal dynamics [104], and single rigid body (SRB) dynamics [100, 101, 102, 278] that have been employed for the real-time trajectory planning and control of various bipedal [79, 82, 83, 237] and quadrupedal robots [17, 101, 102, 278]. Reduced-order models come with their own set of challenges. First, they ignore some of the full-order dynamics. For instance, the SRB model ignores the dynamics of legs. This is a valid assumption when the mass of the legs is insignificant compared to the total mass of the robot. The SRB dynamics have been integrated with the kinematics of the legs, but not the full-order dynamics in [150, 219, 220]. Second, when employing a reduced-order model on a legged robot, the optimal reduced-order trajectories have to be translated into full-order joint position or torque commands. In this regard, researchers have employed various techniques, most of them involving a hierarchical control structure with a high-level MPC and a low-level controller. For example, [100, 101, 102] have used Jacobian mapping, [39, 40] have used hybrid zero dynamics (HZD)-based controllers, [279] has used control barrier functions, and [150, 219] have used joint space whole-body controllers.

The problem of addressing uncertainties in legged locomotion has been predominantly approached at the low level. These approaches rely on the inherent robustness of the template dynamics integrated with a nominal MPC. Uncertainties are then addressed by making modifications to the existing low-level controller. For instance, the problem of addressing external disturbances such as forces is done by estimating them with conjugate momentum-based force observers [280, 281]. Input to state stability (ISS) has also been used to quantify uncertainty in walking robots [282]. Few other approaches have made use of frequency shaping [220, 283] to penalize high-frequency terms in the cost function to address locomotion on soft ground. However, these approaches do not directly accommodate disturbances at the high level.

In the MPC community, various popular approaches exist that consider uncertainties and robustness including closed-loop (feedback) min-max MPC [284, 285], open-loop min-max MPC [286, 287], and tube MPC [288, 289]. Open-loop MPC approaches determine control actions at every time sample by optimizing a performance criterion and addressing uncertainty in both cost minimization and constraint satisfaction. This leads to a small domain of feasibility and conservative control actions. Feedback MPC approaches address these issues by optimizing over a set of control policies rather than control actions. However, determining a suitable control policy beforehand is often prohibitively difficult [288]. Tube-based RMPC and stochastic MPC (SMPC) have been employed for legged locomotion using LIP models to accommodate uncertainties [290, 291]. Tube-based MPC involves the computation of robust invariant (RI) tubes and pre-stabilizing feedback controllers that can be computed offline for linear time-invariant systems. SMPC also leverages offline computation for the pre-stabilizing controller and chance constraints. For high-dimensional and nonlinear template models like SRB or even linear time-varying systems arising from their successive linearizations, the computation of RI tubes and pre-stabilizing linear feedback controllers should be done online, making these strategies computationally expensive. Alternative techniques [292] have used the Gaussian mixture model to learn the constraints to mitigate the gap.

In this work, we aim to answer the following *questions*: 1) How to develop a computationally tractable RMPC framework to steer template models subject to a set of unmodeled dynamics? 2) How to translate the optimal reduced-order trajectories to full-order models to bridge the gap between reduced- and full-order models? and 3) How to learn the set of unmodeled dynamics to be used in the RMPC framework?

3.1.3 Objectives and Contributions

The *objectives* and *contributions* of this chapter are as follows. The chapter presents a computationally tractable RMPC framework, based on convex QP, to steer template models subject to a set of modeling and abstraction uncertainties. The RMPC framework considers possible disturbances from the uncertainty set and optimizes the sum of performance criteria. We then present a hierarchical nonlinear control algorithm for real-time planning and control

of quadrupedal robots. At the higher level, the RMPC framework is applied to the SRB dynamics to generate optimal ground reaction forces (GRFs) for the reduced-order model subject to a convex set of uncertainties. The proposed robust control formulation allows the integration of the RMPC framework with a fully-connected multilayer perceptron (MLP) network, trained using RL, to numerically compute the uncertainty set (see Fig. 3.1). At the lower level, a nonlinear controller, developed based on virtual constraints and QP, maps the generated desired GRFs to the full-order dynamical model while regulating some output functions for whole-body motion control. The use of virtual constraints-based controller is motivated by the successful hardware implementation on various bipedal[16, 199, 203, 293] and quadrupedal [44, 211] robots and powered prosthetic legs [213, 215]. The proposed robust control strategy can bridge the gap between reduced- and full-order models while learning the uncertainty at the trajectory planning level. The hierarchical control algorithm is finally validated via a set of extensive numerical simulations and experiments for the blind and robust locomotion of the A1 quadrupedal robot on different terrains and at different speeds.

The current work differs from our previous work [211] in that it only considers low-level and QP-based nonlinear controllers for quadrupedal locomotion while addressing their continuous differentiability, but not the RMPC framework and the proposed hierarchical control algorithm. The work is also different from [39] in that it does not address the robust planning subject to modeling uncertainties.

3.2 Robust MPC Formulation

The objective of this section is to present the high-level RMPC framework for steering template models subject to a set of uncertainties. In this chapter, we consider the SRB model, but the results we present can be extended to other template-based models as well. The equations of motion for the SRB dynamics can be expressed as follows:

$$\Sigma : \begin{cases} \ddot{r} = \dfrac{f^{\text{net}}}{m} - g_0 \\ \dot{R} = R\, S(\omega) \\ I\, \dot{\omega} + S(\omega)\, I\, \omega = R^{\top} \tau^{\text{net}}, \end{cases} \tag{3.1}$$

where m is the total mass, g_0 is the constant gravitational vector, I is the body inertia, $r \in \mathbb{R}^3$ denotes the center of mass (COM) position of the robot in an inertial world frame, $R \in \mathrm{SO}(3)$ represents the rotation matrix of the body frame with respect to the world frame, and ω denotes the angular velocity in the body frame. In our notation, $S : \mathbb{R}^3 \to \mathfrak{so}(3)$ is the skew-symmetric operator. In addition, f^{net} and τ^{net} represent the net force and torque, generated by the legs, acting on the COM, respectively, that is, $f^{\text{net}} := \sum_{j \in \mathcal{C}} f_j$ and $\tau^{\text{net}} := \sum_{j \in \mathcal{C}} S(r_j)\, f_j$, where $j \in \mathcal{C}$ denotes the foot index, $\mathcal{C}$ is the set of contacting points with the ground, $f_j \in \mathbb{R}^3$ represents the GRF at the foot $j \in \mathcal{C}$, and $r_j \in \mathbb{R}^3$ denotes the

distance between the foot j and the COM. The model is valid if $f_j \in \mathcal{FC}$ for every contacting leg $j \in \mathcal{C}$, where $\mathcal{FC}$ represents the linearized friction cone.

Linearizing the SRB dynamics (3.1) around the current states at discrete time k via techniques such as the variational-based linearization [100] results in

$$x_{k+1} = A_k\, x_k + B_k\, u_k + E\, w_k, \quad k = 0, 1, \cdots, \tag{3.2}$$

where $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$, $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$, $w \in \mathcal{W} \subset \mathbb{R}^{n_w}$ denote the reduced-order states, control inputs (i.e., GRFs), and *unknown* uncertainties, respectively, for some positive integers n_x, n_u, and n_w. Here, w_k represents the uncertainties arising from abstraction and ignoring the full-order dynamics such as leg dynamics or compliant elements at the leg ends. The inclusion of w_k in (3.2) can help us to have a better mapping between the actual GRFs u_k and the states x_k in real world. In (3.2), (A_k, B_k) represents the linearized dynamics around the current state with E being the uncertainty distribution matrix. In our notation, $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$ denote the state space, admissible set of controls, and set of uncertainties, all taken as compact and convex sets containing the origin. Similar to [285], $\mathcal{W}$ is assumed to be chosen as the convex hull of some *known* vertices, that is,

$$\mathcal{W} := \text{co}\left\{w^\ell \,|\, \ell \in \mathcal{I}\right\} \tag{3.3}$$

for some *finite* set of vertices indexed by $\ell \in \mathcal{I}$. In particular, w_k in (3.2) is an unknown signal that can be represented as a convex combination of the known vertices w^ℓ, $\ell \in \mathcal{I}$. Section 3.4 will show how to numerically compute the uncertainty set $\mathcal{W}$ via a trained MLP.

The planning problem consists of designing an RMPC algorithm to steer (3.2) from an initial state to a final one in the presence of $w \in \mathcal{W}$ and subject to the feasibility conditions $x \in \mathcal{X}$ and $u \in \mathcal{U}$. To address this problem, we consider the following set of predictions at time k

$$x^\ell_{k+i+1|k} = A_k\, x^\ell_{k+i|k} + B_k\, u^\ell_{k+i|k} + E\, w^\ell\, \delta(i) \tag{3.4}$$

for all future times $i = 0, 1, ..., N-1$ and all realization indices $\ell \in \mathcal{I}$, where N represents the control horizon. Here, $w^\ell\, \delta(i)$ denotes a possible realization of the uncertainty with the index $\ell \in \mathcal{I}$, and $\delta(i)$ represents the discrete-time impulse (i.e., sample) function. Further, $x^\ell_{k+i|k}$ and $u^\ell_{k+i|k}$ represent the predicted state and control sequences associated with the realization $\ell \in \mathcal{I}$ and the initialization rule of $x^\ell_{k|k} = x_k$.

Remark 3.1. In the prediction model (3.4), the uncertainty is considered as $w^\ell\, \delta(i)$ which is nonzero at $i = 0$ and zero for all $i > 0$. In particular, at every time sample k, the RMPC algorithm is aware of possible uncertainties that can happen at the actual time k, but it does not consider further uncertainties for predicted future times $k+i$ with $i > 0$. This assumption will reduce the number of possible branches to realize the predicted future states and control inputs in the RMPC algorithm. Hence, it will reduce the associated computational burden. This notion will be clarified more in Section 3.5.1.

We next present the following real-time and convex QP

$$\begin{aligned}
\min_{\boldsymbol{u}} \quad & \sum_{\ell\in\mathcal{I}} \left\{ p\left(x^{\ell}_{k+N|k}\right) + \sum_{i=0}^{N-1} \mathcal{L}\left(x^{\ell}_{k+i|k}, u^{\ell}_{k+i|k}\right) \right\} \\
\text{s.t.} \quad & x^{\ell}_{k+i|k} \in \mathcal{X}, \qquad i = 1, \cdots, N-1, \; \forall \ell \in \mathcal{I} \\
& u^{\ell}_{k+i|k} \in \mathcal{U}, \qquad i = 0, \cdots, N-1, \; \forall \ell \in \mathcal{I} \\
& x^{\ell}_{k+N|k} \in \mathcal{X}_f, \qquad \forall \ell \in \mathcal{I} \\
& u^{\ell_1}_{k|k} = u^{\ell_2}_{k|k}, \qquad \forall \ell_1, \ell_2 \in \mathcal{I},
\end{aligned} \tag{3.5}$$

where $\boldsymbol{u} := \text{col}\{\boldsymbol{u}^{\ell} \,|\, \ell \in \mathcal{I}\}$ represents the sequence of optimal control inputs over all realizations $\ell \in \mathcal{I}$ and $\boldsymbol{u}^{\ell} := \text{col}\{u^{\ell}_{k+i|k} \,|\, i = 0, 1, \cdots, N-1\}$ denotes the sequence of optimal controls for one particular realization ℓ. In our notation, "col" represents the column operator. Here, $p(x^{\ell}_{k+N|k})$ and $\mathcal{L}(x^{\ell}_{k+i|k}, u^{\ell}_{k+i|k})$ are the terminal and stage costs, respectively, defined as $p(x) := \|x - x_{\text{des}}\|^2_P$ and $\mathcal{L}(x, u) := \|x - x_{\text{des}}\|^2_Q + \|u\|^2_{\hat{R}}$ for some positive definite matrices P, Q, and $\hat{R}$, and some desired trajectory $x_{\text{des}}(k)$. In addition, $\mathcal{X}_f \subset \mathcal{X}$ is a convex and compact terminal region containing the origin. The last constraint in (3.5) is referred to as the "causality constraint" to restrict the freedom on the control sequences $\boldsymbol{u}^{\ell}$ at the first time sample. In particular, the optimal value of the first control input to be applied to the real-world system (i.e., $u_k = u^{\ell}_{k|k} = \pi(x_k)$) will not depend on the realization index ℓ. We remark that the optimal control problem (3.5) optimizes over all possible sequences of control inputs for different realizations while satisfying the feasibility and causality conditions. Unlike [285] that uses a min-max optimization problem, we robustify against disturbances that happen only at the current time step. This will be clarified more in Section 3.5.1. In case the uncertainty set is taken as the trivial set of $\mathcal{W} = \{0\}$, the RMPC framework is reduced to the nominal MPC.

3.3 Low-Level Nonlinear Controller

The objective of this section is to present the low-level nonlinear control algorithm, based on virtual constraints and QPs. The low-level controller maps the desired GRFs, generated by the high-level RMPC, to the joint-level torques while considering the nonlinear full-order model for output regulation. The full-order and floating-based dynamical model of the robot can be described as follows:

$$D(q)\,\ddot{q} + H(q, \dot{q}) = \Upsilon\,\tau + \sum_{j\in\mathcal{C}} J_j^{\top}(q)\,f_j, \tag{3.6}$$

where $q \in \mathcal{Q} \subset \mathbb{R}^{n_q}$ represents the generalized coordinates for some n_q, $D(q) \in \mathbb{R}^{n_q \times n_q}$ denotes the mass-inertia matrix, $H(q, \dot{q}) \in \mathbb{R}^{n_q}$ represents the Coriolis, centrifugal, and gravitational terms, $\tau \in \mathbb{R}^{n_\tau}$ denotes the joint-level torques for some $n_\tau < n_q$, and $\Upsilon \in \mathbb{R}^{n_q \times n_\tau}$

represents the input distribution matrix. Further, $J_j(q)$ and f_j denote the contact Jacobian matrix and GRF at the contacting leg $j \in \mathcal{C}$, respectively. For future purposes, we define $x_f := \text{col}(q, \dot{q})$ as the full-order state vector of the system. In addition, $f := \text{col}\{f_j \,|\, j \in \mathcal{C}\}$ represents the GRF vector containing the individual GRFs at the contacting leg ends.

We next consider the following holonomic output functions, referred to as *virtual constraints* [39], to be regulated

$$y(t, q) := y_a(q) - y_{\text{des}}(t), \tag{3.7}$$

where $y_a(q)$ represents a set of holonomic controlled variables, and $y_{\text{des}}(t)$ denotes their desired evolution on the gait in terms of a time-based phase variable. In this chapter, the controlled variables are taken as the absolute position of the COM and the base frame orientation with respect to the inertial world frame together with the Cartesian coordinates of the swing leg ends for foot placement. The desired evolution for the Cartesian coordinates of the swing leg ends is parameterized as Bézier polynomials that connect the current footholds to the upcoming ones.

Differentiating the output (3.7) twice along the full-order dynamics of the robot results in

$$\ddot{y} = \Psi_1(x_f)\,\tau + \Psi_2(x_f)\,f + \zeta(x_f) = -K_P\,y - K_D\,\dot{y} + v, \tag{3.8}$$

where $\Psi_1(x_f)$, $\Psi_2(x_f)$, and $\zeta(x_f)$ are nonlinear matrices and vectors in x_f computed based on Lie derivatives and input-output (I-O) linearization [274]. Closed-form expressions for Ψ_1, Ψ_2, ζ can be extracted similar to [40, Appendix 8.2] and are not expressed here. Here, K_P and K_D are positive definite matrices, and v is an additional variable, referred to as the slack variable, to be used later. The objective is to solve for (τ, f, v) to meet (3.8). For this purpose, we consider an alternative set of algebraic equations that express zero accelerations for the stance leg ends. In particular, differentiating the Cartesian coordinates of the stance leg ends results in

$$\ddot{r}_{\text{st}} = \Omega_1(x_f)\,\tau + \Omega_2(x_f)\,f + \eta(x_f) = 0, \tag{3.9}$$

where r_{st} represents a vector containing the Cartesian coordinates of all stance legs, and $\Omega_1(x_f)$, $\Omega_2(x_f)$, and $\eta(x_f)$ are proper matrices and vectors.

We now aim to solve for (τ, f, v) to satisfy (3.8) and (3.9) while (τ, f) being feasible, that is, $f_j \in \mathcal{FC}$ for all $j \in \mathcal{C}$ and $\tau \in \mathcal{T}$, where $\mathcal{T}$ represents a convex and compact set of admissible torques. For this purpose, we set up the following real-time and strictly convex QP to be solved at 1kHz

$$\begin{aligned}
\min_{(\tau,f,v)} \quad & \frac{\gamma_1}{2}\|\tau\|^2 + \frac{\gamma_2}{2}\|f - f_{\text{des}}\|^2 + \frac{\gamma_3}{2}\|v\|^2 \\
\text{s.t.} \quad & \Psi_1(x_f)\,\tau + \Psi_2(x_f)\,f + \zeta(x_f) = -K_P\,y - K_D\,\dot{y} + v \\
& \Omega_1(x_f)\,\tau + \Omega_2(x_f)\,f + \eta(x_f) = 0 \\
& \tau \in \mathcal{T}, \quad f_j \in \mathcal{FC}, \qquad \forall j \in \mathcal{C}.
\end{aligned} \tag{3.10}$$

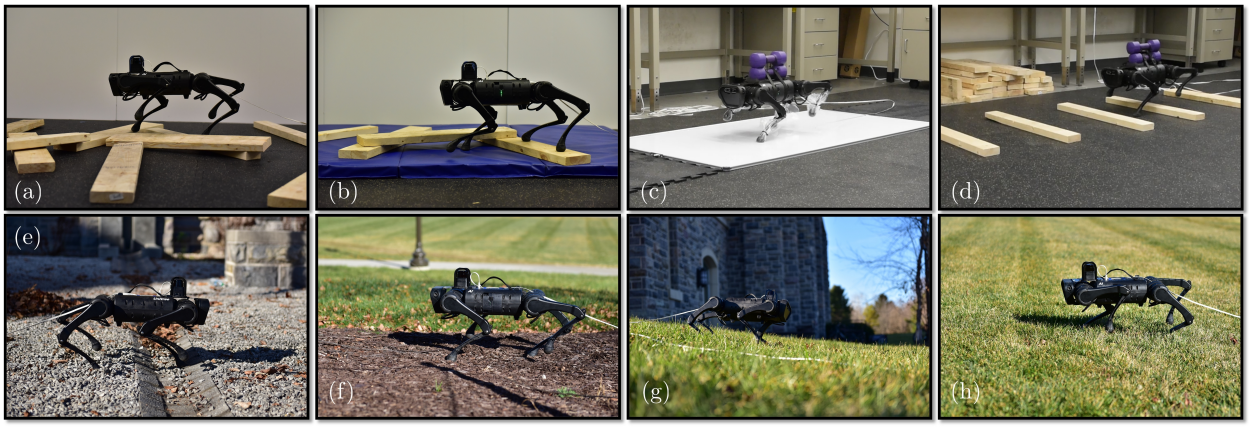


Figure 3.2: Indoor and outdoor experiments to validate the proposed hierarchical control algorithm on different terrains. (a) Blind locomotion of the A1 robot on uneven terrain with wooden blocks at the speed of 0.5 (m/s). (b) Locomotion on a compliant surface with unknown blocks. (c) Locomotion on a slippery surface with a payload of 4.54 (kg) (36% of the robot's weight). (d) Locomotion on blocks with a payload of 4.54 (kg). (e) Locomotion on uneven gravel terrain. (f) Locomotion on mulch. (g) Locomotion on a slope. (h) Locomotion on the grass at the speed of 1 (m/s).

Here, γ_1, γ_2, and γ_3 are positive weighting factors. The slack variable v is added to (3.8) to ensure the feasibility of the QP. To minimize the effect of the slack variable on the output dynamics (3.8), a penalty term $\frac{\gamma_3}{2}\|v\|^2$ is added to the cost function via a large weighting factor γ_3. The other two terms in the cost function try to minimize the 2-norm of the force tracking error $f - f_{\text{des}}$ while having minimum-power torques. This can help to bridge the gap between the planned (i.e., desired) GRFs (f_{des}), computed based on the template model and RMPC, and the actual ones (f). The optimal torques from this low-level nonlinear controller are applied to the full-order dynamics and are denoted by $\tau = \Gamma(t, x_f)$ (see Fig. 3.1).

3.4 Learning to Close the Gap

The objective of this section is to numerically compute the vertices of the uncertainty set $\mathcal{W}$ in (3.3) to be used in the RMPC framework (see Fig. 3.1 again). The problem of computing the uncertainty set $\mathcal{W}$ is posed as a sequential decision-making process that computes the mapping from the reduced-order states x_k to the vertices of the set $\mathcal{W} \subset \mathbb{R}^{n_w}$. This can be mathematically represented using a Markov Decision Process (MDP). We then use a model-free RL technique [294] to train a fully-connected MLP network that computes the vertices of the uncertainty set.

We implement our hierarchical control algorithm in-the-loop with the full-order simulation model of the quadrupedal robot to train the MLP in an offline manner. Here, we consider

the problem of locomotion over rough terrains and represent the sequential decision-making process in discrete time. At every time t, the MDP obtains an observation and performs an action while achieving a scalar reward $\mathcal{R}_t$. The actions of the MDP represent the vertices of uncertainty set $\mathcal{W}$, that is,

$$w_k^l = \Phi^\ell(x_k), \quad \ell \in \mathcal{I}, \tag{3.11}$$

where $\Phi^\ell(\cdot)$ denotes the trained NN, and x_k (reduced-order states) represents the observables for the MDP. Here, we consider one augmented MLP that computes all of the vertices, that is, $\Phi(\cdot) := \text{col}\{\Phi^\ell(\cdot) \,|\, \ell \in \mathcal{I}\}$. For the SRB dynamics expressed in (3.1), the observables for the MDP are taken as the z-component of the COM position (r_z), the base-frame orientation (R), the linear and angular velocities ($\dot{r}$ and ω), and the relative positions of feet from the COM (r_j). We aim to find an optimal policy $\Phi^\star$ that maximizes the following expected sum of rewards over T samples

$$\Phi^\star = \arg\max_{\Phi} \mathbb{E}\left\{\sum_{t=0}^{T-1} \lambda^t \mathcal{R}_t\right\}, \tag{3.12}$$

where $\lambda \in (0, 1)$ represents the discount factor.

In this chapter, we employ the Proximal Policy Optimization (PPO) algorithm [295] to solve (3.12) and use the A1 quadrupedal robot, developed by Unitree, as our test-bed. Similar to [29], the reward function at time t is chosen as

$$\mathcal{R}_t = \alpha_v \|\dot{r}_x\|^2 - \alpha_\tau \|\tau\|^2, \tag{3.13}$$

where $\dot{r}_x$ and τ denote the forward velocity of the COM (i.e., along the x-axis) and torques, respectively, and α_v and α_τ represent positive weighting factors. This work approximates both the policy and value functions using MLPs, each with two hidden layers of 128 neurons. We also choose the discounting factor as $\lambda = 0.998$ with $\alpha_v = 1$ and $\alpha_\tau = 4 \times 10^{-5}$. Details of the numerical simulations together with the controller and training parameters will be presented in Section 3.5.

3.5 Numerical and Experimental Results

This section aims to numerically and experimentally validate the proposed control algorithm for the robust and blind locomotion of the A1 quadrupedal robot on different terrains and at different speeds.

3.5.1 Controller Synthesis

The A1 robot is a torque-controlled robot with 18 Degrees of Freedom (DOFs) and 12 actuators, weighing 12.45 (kg) and standing up to 0.26 (m) (see Fig. 4.3). Six DOFs

describe the unactuated and absolute position and orientation (i.e., Euler angles) of the robot's body with respect to the inertial world frame. The remaining 12 DOFs describe the actuated joints in legs. In particular, each leg has a 2-DOF hip joint (hip roll and hip pitch) and an additional 1-DOF knee joint (knee pitch). We use RaiSim [264] and implement the RL framework outlined in [29] for training of the MLP. The proposed high-level RMPC and the low-level QP are solved online using qpSWIFT [277] at 160Hz and 1kHz, respectively. In the RMPC framework, we consider the linearized SRB dynamics (3.1) with 12 states for trotting and subject to a set of unmodeled dynamics (i.e., $E\,w$) that appear as additional forces/torques in the velocity dynamics (i.e., $\ddot{r}$ and $\dot{\omega}$) (see (3.2)). More specifically, the uncertainty vector w is taken as 6-dimensional arising from abstraction (e.g., ignoring the leg dynamics) to have a better mapping from the actual GRFs to the SRB dynamics. We then take $E = [\mathbb{I}_{6\times 6}; \mathbb{I}_{6\times 6}] \in \mathbb{R}^{12\times 6}$ as a selection matrix composed of zeros and ones to distribute the vertices w across the realization of the RMPC. We remark that one can alternatively consider the dynamics as $x_{k+1} = A_k\, x_k + B_k\, u_k + w_k$. However, our choice of the selection matrix E reduces the dimension of the vertices of $\mathcal{W}$ from 12 to 6, and hence, it reduces the hyperparameters for training the MLP in (3.11).

To reduce the computational burden associated with the RMPC algorithm (3.5), we consider the uncertainty set $\mathcal{W} \subset \mathbb{R}^6$ as a convex hull of two vertices, that is, $\mathcal{W} = \text{co}\{w^\ell \,|\, \ell \in \mathcal{I}\}$ and $\mathcal{I} = \{1, 2\}$ (See Section 3.5.3 for a comparative study on the effects of the number of vertices for $\mathcal{W}$). The control horizon for the RMPC problem is chosen as $N = 7$ discrete sample times. The resultant RMPC and QP-based low-level controller will be solved on an off-board laptop with i7-1185G7 running at 3.00 GHz in experiments of Section 3.5.3. Under nominal conditions, the computation time for the RMPC and low-level controller is approximately 4.2 (ms) and 0.22 (ms), respectively. As discussed in Remark 3.1, we only consider the branching in the state trajectories at the first time sample. More specifically, 2 different sets of state trajectories, referred to as realizations, are considered in the RMPC framework (3.5). The min-max technique of [285], however, considers all possible branches that can happen at every time sample within the control horizon and then minimizes the worst-case scenario. This approach would then result in $2^7 = 128$ different realizations, which is not computationally tractable for real-time planning. Unlike [285], the proposed RMPC approach does not guarantee the recursive feasibility. In the rare events in which the QP becomes infeasible, we terminate the QP solver at a maximum iteration number and use the solution. The feasibility set $\mathcal{X}$ and the terminal set $\mathcal{X}_f$ are chosen as hypercubes in $\mathbb{R}^{12}$ around the desired trajectory. The feasibility set $\mathcal{U}$ is then chosen as the linearized friction cone with the friction coefficient of $\mu = 0.6$.

For the low-level controller, the weighting factors are taken as $\gamma_1 = 1$, $\gamma_2 = 10^3$, and $\gamma_3 = 10^6$ with the same friction coefficient used for the high-level RMPC. In this work, we make use of a time-based switching approach for impact detection. In addition, we employ Raibert's heuristic [9, Eq. (2.4), pp. 46] for footstep planning.

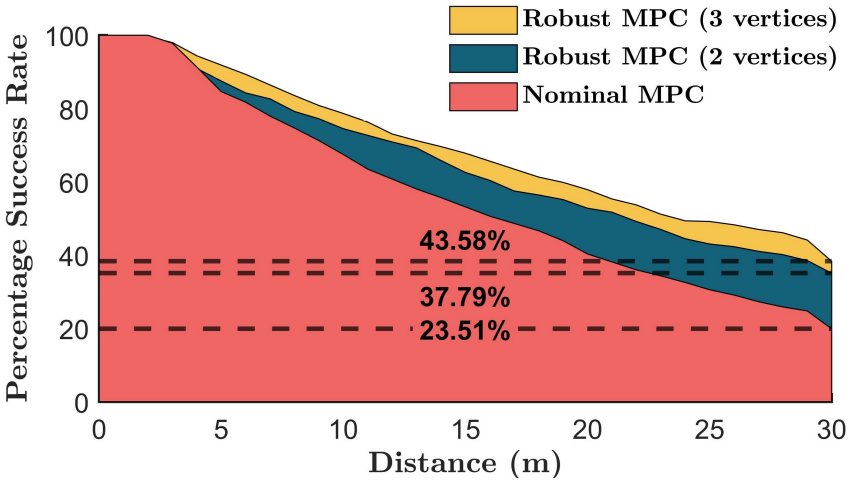


Figure 3.3: The percentage of the success rate of RMPC with three and two vertices and the nominal MPC over 550 randomly generated uneven terrains. The dashed line represents the success rate across the length of the terrain.

3.5.2 Training of the MLP

The training of the MLP network is performed offline using numerical simulations with the learning environment provided by RaiSim [264]. The objective of training the MLP is to learn the gap between reduced- and full-order models and also to aid in the locomotion of the quadrupedal robot on rough terrains by computing the appropriate uncertainty set $\mathcal{W}$. The simulation and training environments consist of a single fixed rough terrain composed of rigid blocks placed at regular intervals. The use of a single training environment is motivated by the following justification. The QP-based virtual constraints controller employed at the low level of the hierarchical control algorithm can result in stable locomotion patterns on flat terrains, as studied in our previous work [211]. The integration of the low-level controller with the high-level RMPC algorithm improves the robust stability of gaits over different sets of terrains. Unlike other *model-free* techniques that use NNs to learn the entire whole-body trajectory planning from scratch, our proposed MLP layer only computes the appropriate set of uncertainties to be used in the RMPC framework. Hence, the combination of the MLP layer with the *model-based* RMPC and the low-level nonlinear controller does not require training on a large set of randomly generated terrains as indicated by the experimental results in Section 3.5.3.

Each block in the fixed terrain is 5 (cm) (20% of the robot's nominal height) tall and spaced at regular intervals of 5 (cm). The robot is commanded by the MPC and low-level controller to perform a symmetrical forward trot gait at 0.5 (m/s). Each episode runs for four seconds or until it meets the termination criterion. Termination is encountered when the rigid body links connecting the knee with the toe hit the ground. The learning rate is set to 5×10^{-4},

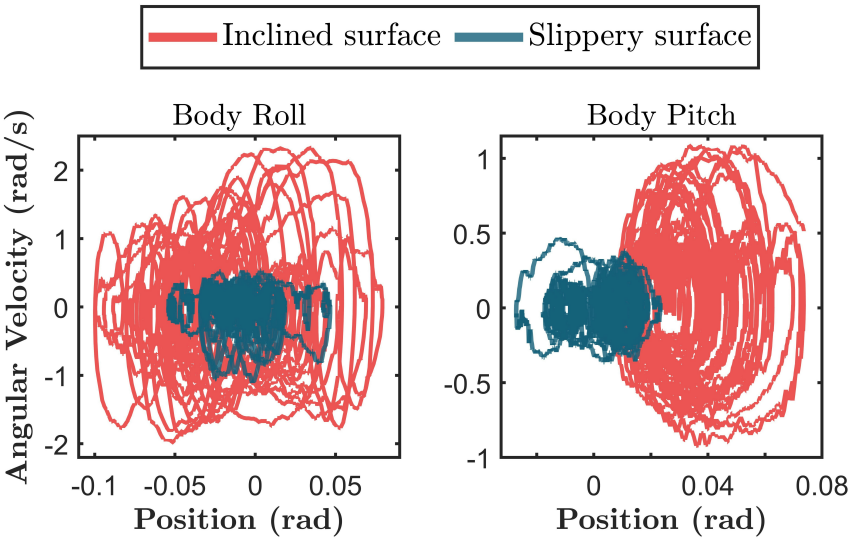


Figure 3.4: Phase portrait of the unactuated DOFs (roll and pitch). The figure depicts robust stability in the phase portraits for two different experiments, including outdoor locomotion on inclined terrain (Fig. 4.3(g)) and indoor locomotion on a slippery surface with a payload of 4.54 (kg) (Fig. 4.3(c)).

and the entire training roughly takes 12 hours on a Desktop PC equipped with an Intel Xeon W-2125 processor. We use two hidden layers with 128 neurons in each of the MLPs of the policy and value function approximations with a Rectified Linear Unit (ReLU) as the activation function.

3.5.3 Numerical and Experimental Validation and Discussion

The objective of this section is to numerically and experimentally validate the proposed control algorithm for the robust and blind locomotion of the A1 robot on different sets of terrains and at different speeds.

To study the effects of the number of vertices on the uncertainty set $\mathcal{W}$, we conduct two numerical experiments, each with a different set of vertices. In particular, we choose two and three vertices for the uncertainty set and train the corresponding MLP. Each of the RMPC algorithms takes 12 and 18 hours to learn the uncertainty set. We then quantify the performance of the two developed RMPC algorithms with the nominal MPC. More specifically, we randomly generate 550 heightmaps, each with a different distribution of blocks. The blocks, each 5 (cm) in height, are distributed asymmetrically and randomly over a terrain of 30 (m) length ($\approx$ 85 robot's body length). Figure 3.3 illustrates the percentage of the success rate of each of the RMPC algorithms and the nominal MPC on 550 randomly generated terrains over the traveled distance. Here, failure is determined based on the termination criterion outlined in Section 3.5.2. As depicted in Fig. 3.3, the success rate of

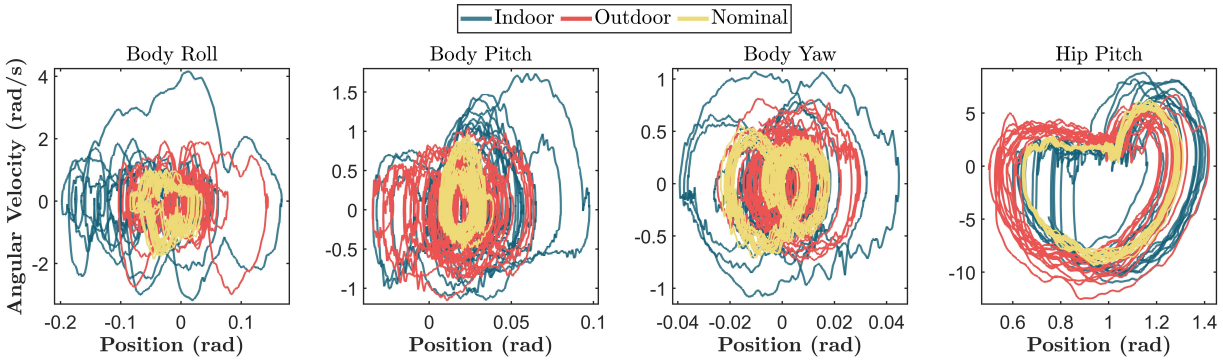


Figure 3.5: Phase portrait of the unactuated DOFs (roll and pitch). The figure depicts robust stability in the phase portraits for two different experiments, including outdoor locomotion on inclined terrain (Fig. 4.3(g)) and indoor locomotion on a slippery surface with a payload of 4.54 (kg) (Fig. 4.3(c)).

RMPC with three and two vertices and the nominal MPC across the entire length of the terrain is 43.58%, 37.79%, and 23.51%, respectively. We remark that for the experiments on the real robot, we choose the RMPC with two vertices as the difference in success rates between the two RMPC algorithms is not significant. Furthermore, the computational time taken by RMPC with three vertices (10.9 (ms)) is significantly higher than the RMPC with two vertices (4.2 (ms)) and the nominal MPC (1.24 (ms)), justifying our choice of $\mathcal{W}$ with two vertices.

As shown in Fig. 4.3, we consider four different subsets of indoor investigations, including locomotion a) on uneven terrain with a random configuration of wooden blocks, b) on a compliant surface covered with wooden blocks, c) with a payload of 4.54 (kg) (36% of the robot's weight) on a slippery surface, covered by a cooking spray, and d) with a payload of 4.54 (kg) on wooden blocks. The outdoor experiments then include locomotion on e) uneven gravel terrain, f) mulch, g) a slope, and h) the grass. For some of these experiments, we consider locomotion at different speeds of 0.1 (m/s), 0.5 (m/s), 0.75 (m/s), and 1 (m/s). We observe that the proposed hierarchical control algorithm can result in robustly stable gaits on these different sets of terrains. Figure 3.4 shows the phase portraits of the robot's roll and pitch motions during two experiments. The first experiment involves locomotion in an outdoor environment on an inclined lawn with an approximate inclination of $(10^\circ - 15^\circ)$ (Fig. 4.3(g)). The second experiment involves locomotion in an indoor environment on a slippery surface (Fig. 4.3(c)). The slippery environment is created with the use of a cooking spray. For this experiment, the coefficient of friction is reduced to 0.3 in both the MPC and the low-level controller. Figure 3.5 depicts the phase portraits for three of these experiments at the speed of 0.5 (m/s), including the nominal locomotion, indoor locomotion on uneven terrain with a random set of wooden blocks (Fig. 4.3(a)), and outdoor locomotion on uneven gravel terrain (Fig. 4.3(e)). Figure 3.6 shows the representative simulation snapshots of the

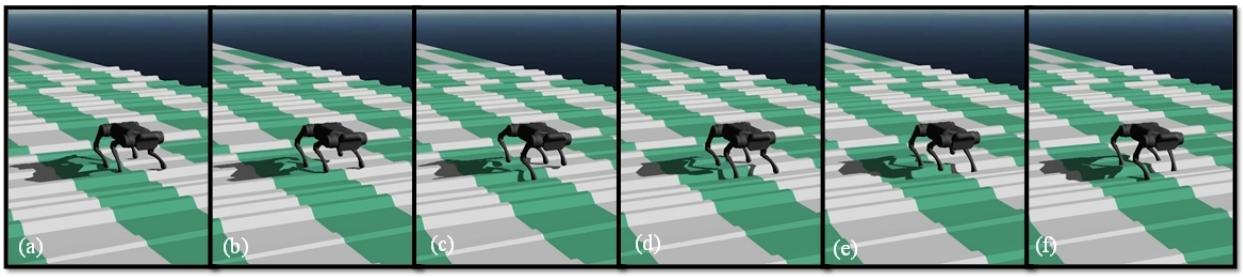


Figure 3.6: Representative snapshots of the stable locomotion of A1 quadrupedal robot on rough terrain.

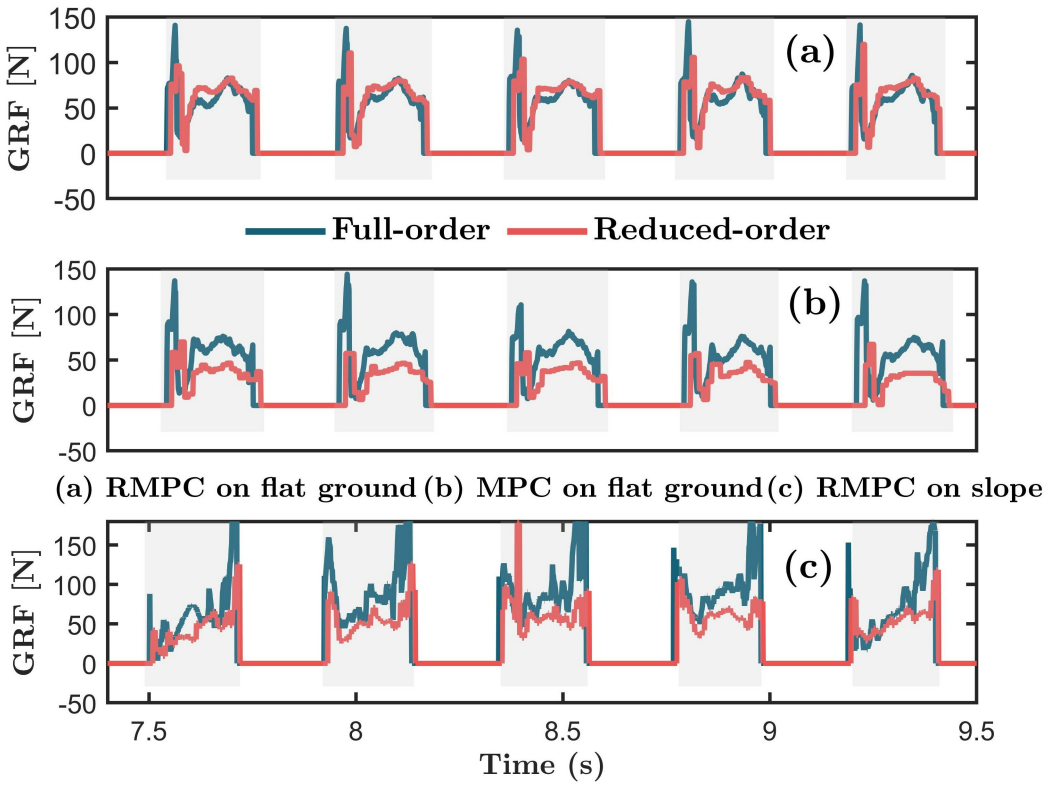


Figure 3.7: Comparison of the gaps between the GRFs generated based on the reduced- and full-order models. Plot of desired and actual GRFs (the z component) with the high-level RMPC algorithm and the low-level QP-based nonlinear controller on flat terrain (a) and on a slope (c). (b) Plot of the desired and actual GRFs with the hierarchical control algorithm in which the high-level RMPC is replaced by a nominal MPC for experiments on flat terrain.

stable locomotion of RMPC framework on rough terrain.

Figure 3.7 studies the gap between the reduced- and full-order models of locomotion with and without the proposed hierarchical control algorithm. Here, we plot the time evolution of

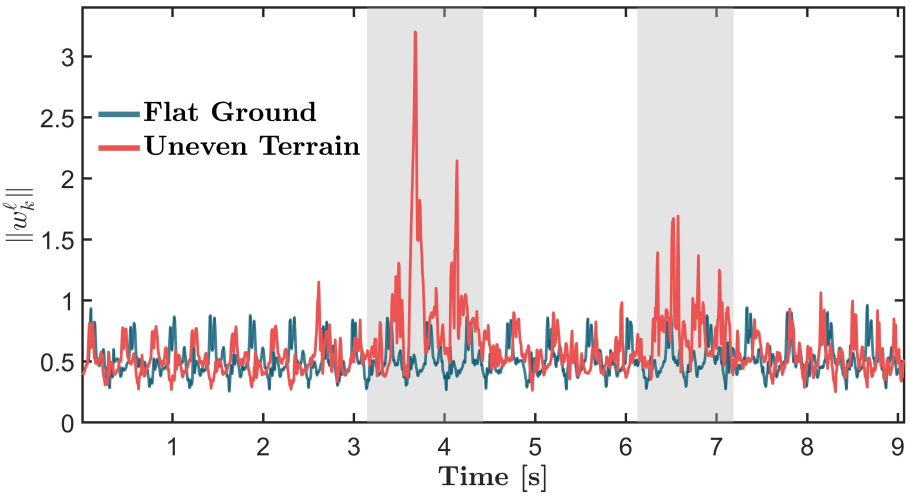


Figure 3.8: Plot of the 2-norm for one of the vertices for the uncertainty set (i.e., $\|w_k^\ell\|$) versus time. The plot depicts the contribution of the trained MLP for locomotion on a flat ground and locomotion on an uneven terrain with two blocks. The gray portions indicate the stepping on blocks.

the desired and actual GRFs at the front left leg for experiments on flat ground and trotting backward on a downhill slope. The desired GRF is the optimal control input computed for the SRB dynamics via the MPC algorithm. The actual GRF is then computed according to the full-order dynamics via the QP-based nonlinear controller in (3.10). Figures 3.7(a) and 3.7(c) compare the vertical components of the desired and actual GRFs with the proposed RMPC framework. In Fig. 3.7(b), the RMPC framework at the higher level of the control algorithm is replaced with a nominal MPC. It is clear that there is a significant reduction between the GRFs computed based on the reduced- and full-order models compared to Fig. 3.7(b). In addition, it can be seen from Fig. 3.7(c), that bridging the gap occurs for experiments involving locomotion on a slope.

To study the contribution of the trained MLP in the RMPC framework, we plot the 2-norm of the upper-bound vertex of the uncertainty set versus time in Fig. 3.8. Here, we consider two indoor experiments, including the nominal locomotion on flat ground and the locomotion on two wooden blocks. From Fig. 3.8, it can be observed that $\|w_k^\ell\|$ spikes during stepping on blocks and decreases as the robot continues to step forward. During the steady-state, $\|w_k^\ell\|$ reaches periodic behavior that is smaller than the spikes for uneven terrains.

3.6 Conclusion and Future Work

This chapter presented a hierarchical control algorithm for the real-time motion planning and control of quadrupedal robots while bridging the gap between reduced- and full-order models. At the higher level, a computationally tractable RMPC framework is developed based on

QPs to steer reduced-order locomotion models subject to a convex set of uncertainties arising from abstraction and unmodeled dynamics. In particular, the RMPC algorithm is applied to the SRB dynamics to generate optimal GRFs. At the lower level, a nonlinear controller, based on I-O linearization and QPs, is developed to map the optimal reduced-order GRFs to the full-order model while imposing virtual constraints for whole-body motion control. The proposed RMPC framework allows the integration of the hierarchical controller with RL techniques to train an MLP to compute the vertices of the uncertainty set numerically. The proposed hierarchical control algorithm is finally validated numerically and experimentally for robust and blind locomotion of the A1 quadrupedal robot on different indoor and outdoor terrains and at different speeds. The numerical analysis of the RMPC suggests significant improvement in the performance of the rough terrain locomotion compared to the nominal MPC. Our experimental studies indicate a significant reduction in the gap between the reduced- and full-order models by comparing the desired and actual GRFs computed by the proposed high- and low-level controllers.

The current work considered uncertainty in the template model dynamics. For future research, we will investigate the incorporation of uncertainties in kinematic constraints. We will also explore alternative reward functions to mitigate the gap between reduced- and full-order models.

Part III

Robust Optimal Predictive Controller using Linear Matrix Inequailities

Chapter 4

$\mathcal{H}_2$- and $\mathcal{H}_\infty$-Optimal Predictive Controllers for Periodic Legged Locomotion: A Linear Matrix Inequality Approach

4.1 INTRODUCTION

The objective of this chapter is to outline a formal approach to generate robust optimal predictive control solutions that can accommodate disturbances and stabilize periodic legged locomotion. We expand on the existing optimization-based controllers, particularly quadratic programming (QP)-based model predictive controllers (MPCs). Under some mild conditions, we linearize the resulting discrete-time, closed-loop nonlinear control system around the gait to obtain a linear time-varying (LTV) system. This periodic LTV system is further transformed into a linear time-invariant (LTI) system using suitable coordinate transformation via discrete-time Floquet theory. The LTI system is then analyzed to accommodate parametric uncertainties and to synthesize robust optimal $\mathcal{H}_2$- and $\mathcal{H}_\infty$- feedback controllers via linear matrix inequalities (LMIs). The chapter also presents sufficient conditions for the existence and synthesis of such robust optimal controllers. The proposed optimal controllers are numerically and experimentally validated for the robust locomotion of the A1 quadrupedal robot subject to various disturbances and uneven terrains. Our numerical results suggest that the performance of the $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPC controllers is notably robust compared to the baseline MPC.

4.1.1 Motivation

Complex and high-dimensional dynamical systems like legged robots interacting with unknown environments grapple with various uncertainties in the form of unplanned contacts and unknown terrains. Developing robust control solutions that accommodate uncertainties explicitly initiates the first step toward safe and reliable real-world deployment of legged robots. This has a couple of benefits. For instance, it may be impractical to know all the properties of the terrain a legged robot is navigating ab initio to inform the control algo-

rithms, even in the presence of visual sensors. In these scenarios, robust control algorithms designed to accommodate unplanned environmental interactions can extend the variety of terrains a legged robot can negotiate. Furthermore, robust control algorithms can expedite the transfer of algorithms from simulation to reality, potentially circumventing the assumptions made by typical physics-based simulation environments [116].

Toward this objective, this chapter presents a robust control approach that can explicitly accommodate disturbances. We propose an algorithm that builds on the conventional optimization-based control stack, especially convex quadratic programming (QP)-based model predictive controllers (MPCs). Under mild conditions, we linearize the resulting closed-loop nonlinear discrete-time system around the gait to obtain a linear time-varying (LTV) system. Using suitable coordinate transformation via discrete-time Floquet theory [117, 296], the resulting periodic LTV system is transformed into a linear system with a constant state-transition matrix. This system can then be analyzed using the classical $\mathcal{H}_2$- and $\mathcal{H}_\infty$-control methods subject to parametric uncertainty and robust state feedback laws can be obtained as solutions to linear matrix inequalities (LMIs) [121]. In this work, we also rigorously establish the existence of $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPC solutions. The proposed controllers are then extensively validated for robust locomotion of the A1 quadrupedal robot both through simulations and experiments.

4.1.2 Background and Related Work

Existing control algorithms for the real-time planning and control of legged robots predominantly use optimization-based techniques, particularly MPCs. These MPCs typically use a reduced-order (i.e., template) model such as linear inverted pendulum (LIP) [297], single rigid body (SRB) [100, 101, 102, 298] dynamics, and centroidal dynamics [104, 299]. Designing robust optimal MPC solutions for periodic gaits of these template systems is challenging, particularly because state-of-the-art $\mathcal{H}_2$ and $\mathcal{H}_\infty$ approaches pertain to the robust stabilization of equilibrium points for ordinary differential equations (see, e.g., [119, 120]) but not periodic solutions (i.e., gaits) of template-based MPCs. The situation is exacerbated with template models such as SRB, where the control inputs are switched depending on the robot's gait. Other approaches considering uncertainty directly at the MPC level can be sectioned into two categories: robust MPC (RMPC) and stochastic MPC (SMPC). Popular RMPC approaches include closed-loop (feedback) min-max MPC [284, 285], open-loop min-max MPC [286, 287], and tube MPC [288, 289]. While open-loop MPC often produces conservative control actions, feedback MPC mitigates this issue by optimizing over a set of control policies rather than control actions. However, determining a suitable control policy beforehand is prohibitively expensive, especially in the context of real-time systems [288]. On the other hand, tube-based MPC and SMPC rely on the offline computation of robust invariant (RI) tubes and pre-stabilizing feedback control laws for chance constraints, respectively. Recently, SMPC and tube-based MPC have been utilized to address uncertainty in legged locomotion using LIP [290, 291] and centroidal models [300]. Alternatively, our pre-

vious work [212] considered integrating model-free Reinforcement Learning (RL) techniques with an RMPC algorithm to improve the rough terrain locomotion of quadrupedal robots.

Within legged locomotion and nonlinear control theory, several robust optimal control solutions are proposed. For instance, to account for uncertainties in the ground height variations, researchers have developed feedback control solutions using either a finite state machine [301] or through robust Lyapunov functions [302] by considering ground height variations as disturbances in the context of input-to-state stability (ISS). Control Lyapunov Functions (CLFs) have also been used to generate input-to-state stable control laws for the stabilization of bipedal [303] and quadrupedal robots [304]. Linear and bilinear matrix inequalities have also been used to generate robust centralized [305] and decentralized controllers [306] for legged robots. The current work differs from [305, 306] in that the proposed controller operates in a real-time MPC fashion instead of slow event-based controllers.

4.1.3 Objectives and Contributions

The *overarching* goal of this chapter is to present a systematic approach to developing robust $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal predictive controllers for periodic legged locomotion. In this regard, the *objectives* and *key contributions* of the chapter are as follows. The chapter develops a robust state feedback control law that builds on the existing MPC architectures, thereby acting as a complementary control solution. The chapter analyzes the conditions under which the discrete-time closed-loop template dynamics with MPC can be transformed into a linear system with a constant state-transition matrix using Floquet theory. The chapter theoretically establishes a set of conditions guaranteeing the existence of robust $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal controllers. The proposed robust MPC laws are then rigorously validated using extensive numerical and experimental studies for the robust locomotion of the A1 quadrupedal robot at various speeds subject to disturbances and rough terrains. Our numerical results indicate an improvement of around 50% for $\mathcal{H}_\infty$-optimal MPC compared to the normal MPC for locomotion subject to randomly generated external forces and another 15% improvement with the $\mathcal{H}_2$-optimal MPC over the $\mathcal{H}_\infty$-optimal MPC.

4.2 PROBLEM FORMULATION

This section aims to present the problem statement to develop robust optimal predictive controllers for periodic gaits of legged robots. Without loss of generality, we consider the

template-based SRB dynamics as follows:

$$\Sigma : \begin{cases} \ddot{r} = \dfrac{f^{\text{net}}}{m} - g_0 \\ \dot{R} = R\,\hat{\omega} \\ I\,\dot{\omega} + \hat{\omega}\,I\,\omega = R^\top \tau^{\text{net}}, \end{cases} \tag{4.1}$$

where $r \in \mathbb{R}^3$ denotes the position of the center of mass (COM) of the robot in an inertial world frame, $R \in \mathrm{SO}(3)$ represents the rotation matrix of the body frame with respect to the world frame, $\omega \in \mathbb{R}^3$ denotes the angular velocity in the body frame, m is the total mass, g_0 is the acceleration due to gravity, and I is the inertia of the robot in body frame. Furthermore, $\hat{(\cdot)} : \mathbb{R}^3 \to \mathfrak{so}(3)$ is the skew-symmetric operator. In our notation, f^{net} and τ^{net} denote the total net force and torque generated by the legs, respectively. In other words, $f^{\text{net}} := \sum_{j\in\mathcal{C}} f_j$ and $\tau^{\text{net}} := \sum_{j\in\mathcal{C}} \hat{r}_j\, f_j$, where $j \in \mathcal{C}$ represents the stance foot index and $\mathcal{C}$ is the set of contacting points with the ground. In addition, $f_j \in \mathbb{R}^3$ represents the ground reaction force (GRF), taken as control inputs, at each stance foot $j \in \mathcal{C}$, and $r_j \in \mathbb{R}^3$ denotes the distance between the stance foot j and the COM.

Most template models, including SRB and centroidal dynamics, can be discretized and written in a general nonlinear form as

$$x_{k+1} = f^{\text{ol}}(x_k, u_k), \quad k = 0, 1, ..., \tag{4.2}$$

where $x_k \in \mathcal{X} \subset \mathbb{R}^{n_x}$ and $u_k \in \mathcal{U} \subset \mathbb{R}^{n_u}$ denote the state variables and control inputs, respectively, for some positive integers n_x and n_u, and $k \in \mathbb{Z}_{\geq 0} := \{0, 1, \cdots\}$ is the discrete-time sample. Furthermore, the superscript "ol" stands for the open-loop system, and $\mathcal{X}$ and $\mathcal{U}$ are polyhedra representing the feasible states and admissible control inputs. For the SRB dynamics (4.1), $\mathcal{U}$ can be taken as the linearized friction cone. These physics-based template dynamics (i.e., f^{ol}) are usually smooth (i.e., $\mathcal{C}^\infty$) with respect to (x, u). We are interested in MPC-based feedback control solutions for template locomotion models that typically take the form of a finite-dimensional and convex QP as follows:

$$\begin{aligned} \min_{(u(\cdot),x(\cdot))} \quad & p\left(x_{k+N|k}\right) + \sum_{i=0}^{N-1} \mathcal{L}\left(x_{k+i|k}, u_{k+i|k}\right) \\ \text{s.t.} \quad & x_{k+i|k} \in \mathcal{X}, \qquad i = 1, \cdots, N \\ & u_{k+i|k} \in \mathcal{U}, \qquad i = 0, \cdots, N-1, \end{aligned} \tag{4.3}$$

where N is the finite control horizon, and $p(x)$ and $\mathcal{L}(x, u)$ represent the terminal and stage costs, respectively, defined as $p(x_{k+N|k}) := \|x_{k+N|k} - x^{\text{des}}_{k+N|k}\|^2_P$ and $\mathcal{L}(x, u) := \|x_{k+i|k} - x^{\text{des}}_{k+i|k}\|^2_Q + \|u_{k+i|k} - u^{\text{des}}_{k+i|k}\|^2_R$ for some positive definite matrices P, Q, and R and some desired state and control trajectories $x^{\text{des}}(\cdot)$ and $u^{\text{des}}(\cdot)$. In our notation, $x(\cdot)$ and $u(\cdot)$ represent the predicted state and control trajectories over the control horizon, i.e., $x(\cdot) := \text{col}\{x_{k+i|k} \,|\, i =$

$1, \cdots, N\}$ and $u(\cdot) := \text{col}\{u_{k+i|k} \,|\, i = 0, \cdots, N-1\}$, where "col" denotes the column operator. The MPC feedback solution is taken as the first optimal input denoted by $u_k = \pi(k, x_k)$. By applying this time-varying control law, the evolution of the template model is described by the nonlinear time-varying closed-loop dynamics $x_{k+1} = f^{\text{cl}}(k, x_k) := f^{\text{ol}}(x_k, \pi(k, x_k))$. We now make the following assumptions.

Assumption 4.1 (Periodic Orbit)**.** There exists a *periodic orbit* $\mathcal{O}$, referred to as the *gait*, for the nonlinear time-varying closed-loop system $x_{k+1} = f^{\text{cl}}(k, x_k)$ with the fundamental period of M. For future purposes, the evolution of the state variables and the MPC feedback law on the orbit $\mathcal{O}$ is given by the periodic functions $x_k^\star$ and $u_k^\star$, respectively.

Assumption 4.2 ($\mathcal{C}^1$ Differentiability)**.** We further suppose that the closed-loop system $x_{k+1} = f^{\text{cl}}(k, x_k)$ is continuously differentiable (i.e., $\mathcal{C}^1$) with respect to the state variables x on an open neighborhood of the periodic orbit, denoted by $\mathcal{N}(\mathcal{O}) \subset \mathbb{R}^{n_x}$, for all $k \in \mathbb{Z}_{\geq 0}$.

Assumption 4.2 is not a restrictive assumption. In particular, if for every initial condition $x_k^\star$ on the orbit $\mathcal{O}$, the predicted state and control trajectories $(x(\cdot), u(\cdot))$ strictly belong to the interior of the feasibility sets $\mathcal{X}$ and $\mathcal{U}$, then the inequality constraints of the QP (4.3) become inactive, and the optimal control problem is reduced to a QP with equality constraints. Consequently, the solutions of this QP become $\mathcal{C}^1$ with respect to the initial condition of $x_k^\star$ (see Fiacco's Theorem [307, Theorem 2.1]).

With the above assumptions in hand, we are interested in robust stabilization of the periodic orbit $\mathcal{O}$ in the presence of some unknown external disturbances (e.g., external force disturbances in the SRB dynamics (4.1)) denoted by $w_k \in \mathbb{R}^{n_w}$ for some positive integer n_w. We alter the control input to $u_k = \pi(k, x_k) + v_k$ and study the following closed-loop system subject to disturbances

$$\Sigma^{\text{cl}} : \begin{cases} x_{k+1} = f^{\text{cl}}(k, x_k, v_k, w_k), & k = 0, 1, \cdots, \\ z_k = c(x_k), \end{cases} \tag{4.4}$$

where v_k is the auxiliary input to be designed and $z_k := c(x_k) \in \mathbb{R}^{n_z}$ represents a smooth, measured performance in terms of state variables for a positive integer n_z.

Problem 2 (Robust Stabilization)**.** The robust stabilization problem of the periodic orbit $\mathcal{O}$ consists of designing the auxiliary control input v_k such that the effect of the unknown disturbance input w_k on the measured performance $z_k = c(x_k)$ is attenuated.

4.3 MAIN RESULT and ALGORITHM

This section presents the main results of this chapter. In particular, it presents a systematic approach to synthesize $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPC laws that address the robust stabilization

of the periodic orbits in Problem 2. From Assumption 4.1, $\mathcal{O}$ is a periodic orbit for the *unperturbed system* $x_{k+1} = f^{\mathrm{cl}}(k, x_k, 0, 0)$ with $v_k \equiv 0$ and $w_k \equiv 0$. From Assumption 4.2, one can differentiate the closed-loop dynamics (4.4) around the orbit $\mathcal{O}$ to obtain a periodic LTV system as follows:

$$\begin{aligned} \delta x_{k+1} &= A_k^{\mathrm{cl}}\, \delta x_k + B_k\, v_k + E_k\, w_k, \quad k = 0, 1, \cdots \\ \delta z_k &= C_k\, \delta x_k \end{aligned} \tag{4.5}$$

with the property $A_{k+M}^{\mathrm{cl}} = A_k^{\mathrm{cl}}$, $B_{k+M} = B_k$, $E_{k+M} = E_k$, and $C_{k+M} = C_k$, where

$$A_k^{\mathrm{cl}} := \frac{\partial f^{\mathrm{cl}}}{\partial x}(k, x_k^\star, 0, 0) = \left(\frac{\partial f^{\mathrm{ol}}}{\partial x} + \frac{\partial f^{\mathrm{ol}}}{\partial u}\, \frac{\partial \pi}{\partial x} \right) \Big|_{x = x_k^\star, u = u_k^\star}$$

and $B_k := \frac{\partial f^{\mathrm{cl}}}{\partial v}(k, x_k^\star, 0, 0) = \frac{\partial f^{\mathrm{ol}}}{\partial u}(x_k^\star, u_k^\star)$, $E_k := \frac{\partial f^{\mathrm{cl}}}{\partial w}(k, x_k^\star, 0, 0)$ and $C_k := \frac{\partial c}{\partial x}(x_k^\star)$ for all $0 \le k < M$. In addition, δx_k represents the change in the state variables with respect to the periodic orbit, i.e., $\delta x_k := x_k - x_k^\star$. Next, we make the following assumption.

Assumption 4.3 (Rank Conditions)**.** We suppose that the state-transition matrices A_k^{cl} satisfy the following rank conditions

$$\operatorname{rank}\left(A_{j+i-1}^{\mathrm{cl}} \cdots A_{j+1}^{\mathrm{cl}}\, A_j^{\mathrm{cl}} \right) = \rho_i, \tag{4.6}$$

where ρ_i is independent of the running index j for all $1 \le i \le n_x$ and $1 \le j \le M$.

This assumption is not restrictive and will be utilized in the following theorem to generate a suitable change of coordinates that converts the periodic LTV system (4.5) into a linear one with a time-invariant state-transition matrix.

Theorem 4.4 (Floquet Transform)**.** *Under Assumptions 4.1-4.3, there exits a time-varying, periodic, and invertible matrix Λ_k, referred to as the discrete-time Floquet transform, such that the changes of coordinates $\overline{\delta x}_k = \Lambda_k^{-1}\, \delta x_k$ transforms the closed-loop system (4.5) into a linear one with a constant state-transition matrix, that is,*

$$\begin{aligned} \overline{\delta x}_{k+1} &= \overline{A}^{cl}\, \overline{\delta x}_k + \Lambda_{k+1}^{-1} B_k\, v_k + \Lambda_{k+1}^{-1} E_k\, w_k, \;\; k = 0, 1, \cdots \\ \delta z_k &= C_k\, \Lambda_k\, \overline{\delta x}_k, \end{aligned} \tag{4.7}$$

where $\overline{A}^{cl} := \Lambda_{k+1}^{-1}\, A_k^{cl}\, \Lambda_k$ is independent of time k.

Proof. We are looking for a time-varying, periodic, and invertible change of coordinates Λ_k, which renders the matrix $\overline{A}^{\mathrm{cl}}$ constant. In other words, we need to satisfy the set of matrix equations $\Lambda_{k+1}\, \overline{A}^{\mathrm{cl}} = A_k^{\mathrm{cl}}\, \Lambda_k$ for all $k = 1, \cdots, M$. Without loss of generality, we assume

$\Lambda_1 = \Lambda_{M+1} = I$ and expand the recursive equation to get

$$\left(\overline{A}^{\mathrm{cl}}\right)^M = A_M^{\mathrm{cl}} \cdots A_2^{\mathrm{cl}} A_1^{\mathrm{cl}}. \tag{4.8}$$

Assumption 4.3 together with [117, Theorem 2] guarantees the existence of the Λ_k matrix. In addition, the time-invariant state-transition matrix $\overline{A}^{\mathrm{cl}}$ can be obtained as the M^{th} root of the matrix product in (4.8). □

Remark 4.5. We remark that for periodic and continuous-time LTV systems with bounded and piecewise continuous state-transition matrices, the Floquet transform always exists [117, Theorem 1]. However, this is not true for periodic and discrete-time LTV systems [117]. Assumption 4.3 allows us to have the Floquet transform for translating the system (4.5) into the linear one with a time-invariant state-transition matrix in Theorem 4.4. From (4.8), the asymptotic stabilization of the periodic orbit $\mathcal{O}$ for the unperturbed system (i.e., $v_k \equiv 0$ and $w_k \equiv 0$) is equivalent to having the eigenvalues of $(\overline{A}^{\mathrm{cl}})^M$, or equivalently, the eigenvalues of $\overline{A}^{\mathrm{cl}}$, inside the unit circle, which coincidences with the Floquet stability theory of periodic and nonlinear discrete-time systems [296].

To address the synthesis of the auxiliary control input v_k for robust optimal stability, we next assume that the leftover time-varying input and disturbance distribution matrices together with the output matrix in (4.7) take values in a convex and bounded set. More specifically, we make the following assumption.

Assumption 4.6 (Paramteric Uncertainty)**.** The time-varying input and disturbance distribution matrices $\overline{B}_k := \Lambda_{k+1}^{-1} B_k$ and $\overline{E}_k := \Lambda_{k+1}^{-1} E_k$ together with the output matrix $\overline{C}_k := C_k \Lambda_k$ take values in a convex and bounded set. In particular, the matrix

$$H_k := \begin{bmatrix} 0_{n_x \times n_x} & \overline{B}_k & \overline{E}_k \\ \overline{C}_k & 0_{n_z \times n_u} & 0_{n_z \times n_w} \end{bmatrix} \tag{4.9}$$

takes values in a convex and bounded polyhedron (i.e., polytope) $\mathcal{H}$ for all time samples $0 \leq k < M$ as follows:

$$\mathcal{H} := \left\{ H(\theta) = \sum_{\ell=1}^{n_{\mathcal{H}}} H^\ell \, \theta^\ell \,|\, \theta^\ell \geq 0, \sum_{\ell=1}^{n_{\mathcal{H}}} \theta^\ell = 1 \right\}, \tag{4.10}$$

where each matrix in this domain can be written as an *unknown* convex combination of $n_{\mathcal{H}}$ given and known vertex matrices H^ℓ for $1 \leq \ell \leq n_{\mathcal{H}}$.

This assumption is not restrictive, and our numerical simulation results in Section 4.4 for the linearized SRB dynamics (4.1) show that the time-varying matrices $(\overline{B}_k, \overline{E}_k, \overline{C}_k)$ are indeed quite constant for $0 \leq k < M$, which is a practical consideration. Hence, one can utilize

Assumption 4.6 together with Theorem 4.4 to study a suboptimal robustness problem that translates the synthesis of $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPC for the periodic orbit into LMIs for an LTI system via parametric uncertainties. More specifically, the vertices of the convex polytope $\mathcal{H}$ are given by

$$H^\ell = \begin{bmatrix} 0_{n_x \times n_x} & \overline{B}^\ell & \overline{E}^\ell \\ \overline{C}^\ell & 0_{n_z \times n_u} & 0_{n_z \times n_w} \end{bmatrix}, \quad \ell = 1, \cdots, n_{\mathcal{H}}. \tag{4.11}$$

which results in the following uncertain LTI system

$$\begin{aligned} \overline{\delta x}_{k+1} &= \overline{A}^{\mathrm{cl}}\, \overline{\delta x}_k + \left(\sum_{\ell=1}^{n_{\mathcal{H}}} \overline{B}^\ell\, \theta^\ell \right) v_k + \left(\sum_{\ell=1}^{n_{\mathcal{H}}} \overline{E}^\ell\, \theta^\ell \right) w_k \\ \delta z_k &= \left(\sum_{\ell=1}^{n_{\mathcal{H}}} \overline{C}^\ell\, \theta^\ell \right) \overline{\delta x}_k. \end{aligned} \tag{4.12}$$

Here, the parameters vector $\theta := \mathrm{col}\{\theta^\ell \,|\, \ell = 1, \cdots, n_{\mathcal{H}}\}$ is assumed to be *unknown* with the properties $\theta^\ell \geq 0$ and $\sum_{\ell=1}^{n_{\mathcal{H}}} \theta^\ell = 1$. Next, we are interested in designing state feedback laws for the suboptimal problem that minimizes the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of the transfer matrix T_{wz} relating w_k to δz_k in the uncertain system (4.12), in which

$$\begin{aligned} \|T_{wz}\|_{\mathcal{H}_2}^2 &:= \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathrm{trace}\left(T_{wz}^{\mathsf{H}}\left(e^{j\omega}\right)\, T_{wz}\left(e^{j\omega}\right) \right) \mathrm{d}\omega \\ \|T_{wz}\|_{\mathcal{H}_\infty} &:= \max_{\omega \in [-\pi, \pi]} \sigma_{\max}\left(T_{wz}\left(e^{j\omega}\right) \right), \end{aligned} \tag{4.13}$$

and "trace", "H", and "$\sigma_{\max}$" denote the trace operator, conjugate transpose, and largest singular value of a matrix, respectively. We are now in a position to present the following theorem to address the suboptimal problem.

Theorem 4.7 (Robust Stabilizing Controllers)**.** *Under Assumptions 4.1-4.6, the following statements are correct.*

(1) *($\mathcal{H}_2$-Optimal Controller): There exists a linear state feedback controller of the form $v_k = K\, \overline{\delta x}_k$ such that the inequality $\|T_{wz}\|_{\mathcal{H}_2}^2 < \mu$ holds if, and only if, the following LMIs hold*

$$\begin{aligned} & \mathit{trace}(W^\ell) < \mu, \quad \begin{bmatrix} W^\ell & \overline{C}^\ell X \\ (\cdot)^\top & X + X^\top - P^\ell \end{bmatrix} > 0, \\ & \begin{bmatrix} P^\ell & \overline{A}^{cl} X + \overline{B}^\ell\, L & \overline{E}^\ell \\ (\cdot)^\top & X + X^\top - P^\ell & 0 \\ (\cdot)^\top & (\cdot)^\top & I \end{bmatrix} > 0 \end{aligned} \tag{4.14}$$

for all $\ell = 1, \cdots, n_{\mathcal{H}}$, where X and L, and the sequence $P^\ell = (P^\ell)^\top$ and $W^\ell = (W^\ell)^\top$ are decision variables. The gain is also calculated as $K = L\, X^{-1}$.

(2) *($\mathcal{H}_\infty$-Optimal Controller): There exists a linear state feedback controller of the form $v_k = K\,\overline{\delta x}_k$ such that the inequality $\|T_{wz}\|^2_{\mathcal{H}_\infty} < \mu$ holds if, and only if, the following LMI holds*

$$\begin{bmatrix} P^\ell & \overline{A}^{cl}X + \overline{B}^\ell L & \overline{E}^\ell & 0 \\ (\cdot)^\top & X + X^\top - P^\ell & 0 & X^\top \overline{C}^{\ell\top} \\ (\cdot)^\top & (\cdot)^\top & I & 0 \\ (\cdot)^\top & (\cdot)^\top & (\cdot)^\top & \mu\, I \end{bmatrix} > 0 \tag{4.15}$$

for all $\ell = 1, \cdots, n_{\mathcal{H}}$, where the matrices X, L, and $P^\ell = (P^\ell)^\top$ are decision variables. In addition, the gain matrix can be calculated as $K = L\,X^{-1}$.

Proof. The theorem's proof can be established by combining Theorems 3 and 4 with 5 and 6 of [120]. □

Finally, the robust stabilizing feedback controller in the original coordinates can take a time-varying form as follows:

$$u_k = \pi(k, x_k) + K\,\Lambda_k^{-1}(x_k - x_k^\star), \tag{4.16}$$

where the time-varying feedback term $K\,\Lambda_k^{-1}(x_k - x_k^\star)$ is added to the original MPC law $\pi(k, x_k)$ for robust optimality.

4.4 APPLICATION TO QUADRUPEDAL LOCOMOTION

This section aims to validate the proposed $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs using numerical case studies and experiments for the robust locomotion of the A1 quadrupedal robot subject to various external disturbances and uneven terrains.

4.4.1 Control Synthesis

A1 is a fully torque-controlled robot with 18 degrees of freedom (DOFs), 6 representing the unactuated position and orientation of the COM, while the remaining 12 DOFs correspond to the actuated joints of each leg. Specially, each leg has a 2 DOF hip joint in addition to a 1 DOF knee joint. The robot weighs 12.45 (kg), with motors at each actuated joint. We consider the full-order nonlinear model of the A1 quadrupedal robot for the numerical and experimental studies. We synthesize both the $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs at various speeds. The auxiliary feedback controllers are used alongside the MPC in a hierarchical control structure. In particular, at the higher level, we propose an MPC scheme along with the newly synthesized $\mathcal{H}_2$ and $\mathcal{H}_\infty$ controllers. At the lower level, the nonlinear virtual

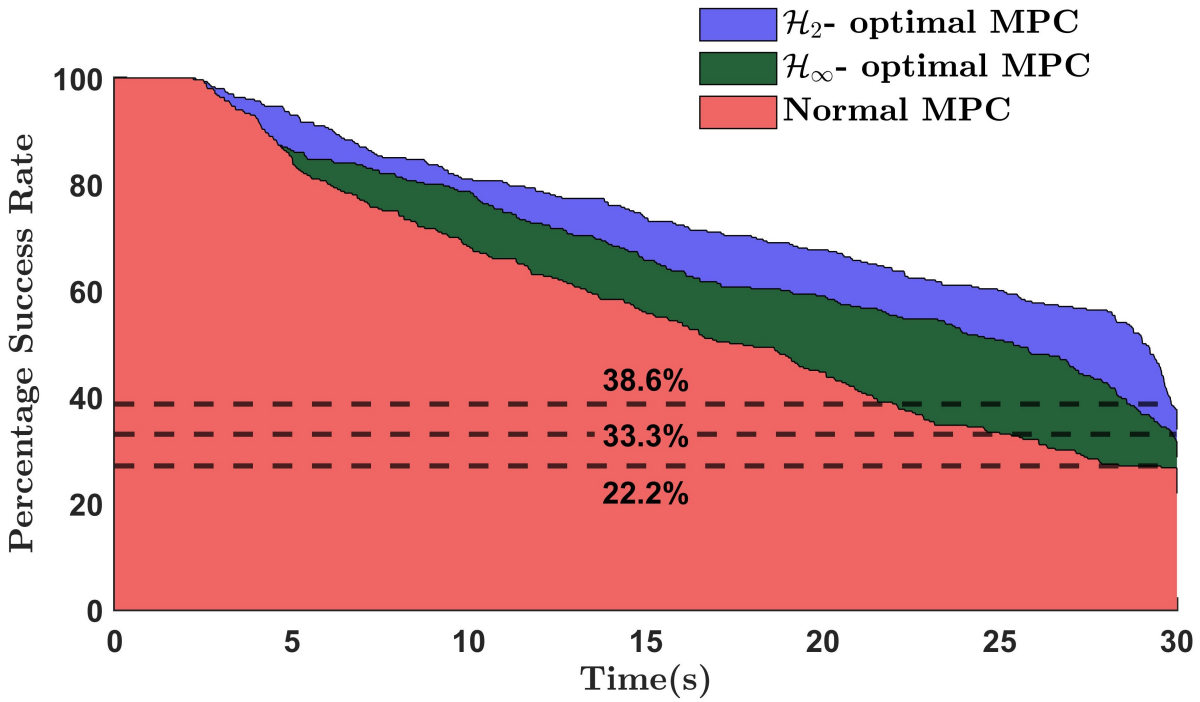


Figure 4.1: The percentage success rate of $\mathcal{H}_2$-optimal MPC, $\mathcal{H}_\infty$-optimal MPC, and normal MPC over 300 different trails. The dashed line represents the success rate over a time period of 30 (s).

constraints controller of [212, 304], which is designed based on QP, is used to map the optimal reduced-order trajectories (particularly GRFs) to the full-order locomotion model. Both the QP-based high-level MPC and the low-level virtual constraints controller are solved online using qpSWIFT [277] at 160 Hz and 1kHz, respectively.

We linearize the SRB dynamics (4.1) with the state variables taken as $x := \text{col}(r, \dot{r}, \alpha, \omega)$, where α represents the body's Euler angles similar to [298]. We next consider the MPC solution for trotting at the speed of 0.5 (m/s) with the control horizon of $N = 7$. The MPC parameters in (4.3) are chosen as $Q = \text{diag}\{Q_r, Q_{\dot{r}}, Q_\alpha, Q_\omega\}$ with $Q_r = \text{diag}\{3e5, 3e5, 3e6\}$, $Q_{\dot{r}} = \text{diag}\{1e4, 1e4, 1e4\}$, $Q_\alpha = \{1e5, 1e5, 1e5\}$, and $Q_\omega = \text{diag}\{5e3, 5e3, 5e3\}$, $P = 10\,Q$, and R as the identity matrix. We get a two-step periodic gait with the fundamental period of $M = 60$ samples (or 0.375 (s)). The following performance and disturbances are then chosen for the SRB dynamics. We assume that the external force disturbances are applied along the x-, y-, and z-axes to the robot's COM, and the performance index is chosen as the position of the COM. Assumptions 4.1-4.3 are met for the gait with the MPC. We next choose $n_{\mathcal{H}} = 4$ vertices to parameterize the time-varying matrices in Assumption 4.6, as they are almost quite constant. More specifically, the four vertices H^ℓ, $\ell = 1, \cdots, 4$ in (4.11) are chosen based on the time-varying values of the matrix H_k in (4.9) at time samples of $k = 0, 15, 45, 59$. The LMIs in Theorem 4.7 are synthesized using the MATLAB Robust Control Toolbox with the upper bound of the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms taken as $\mu = 2.89$ and

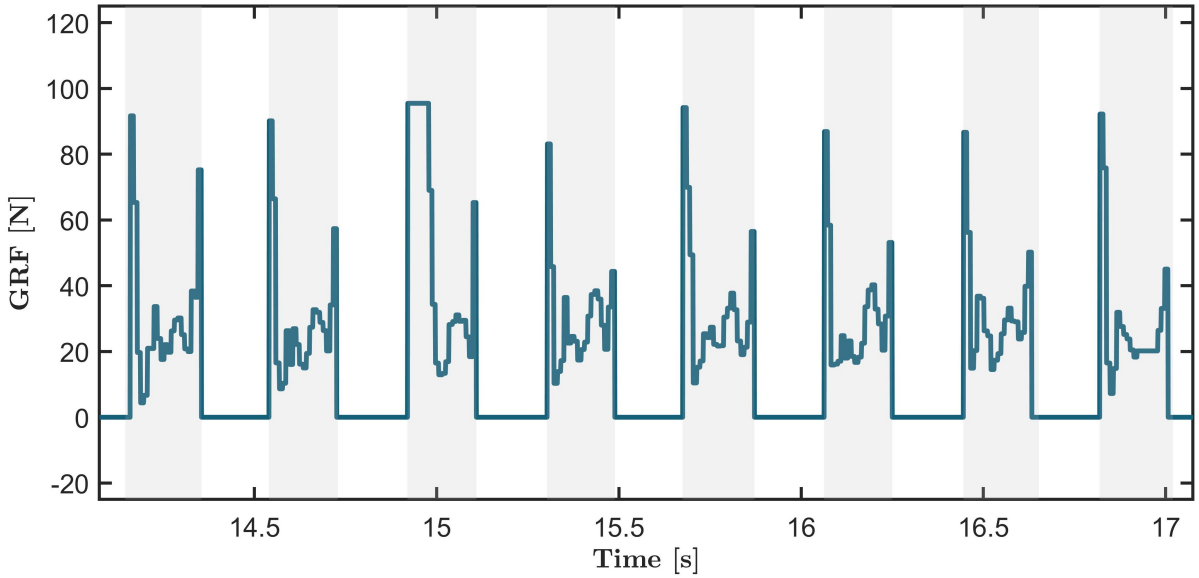


Figure 4.2: The vertical component of the GRF, prescribed by the $\mathcal{H}_2$-optimal MPC, for the front right leg during the push experiment in Fig. 4.3 (a).

$\mu = 1.96$, respectively.

4.4.2 Numerical Simulations

To study the efficacy of the proposed $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs, we conduct a numerical case study in the physics-based simulation environment RaiSim [308]. We compare the performance of $\mathcal{H}_2$-optimal MPC, $\mathcal{H}_\infty$-optimal MPC, and normal MPC for a trotting gait at speed of 0.5 (m/s) subject to randomly generated external forces in the x- and y- directions. In particular, the robot is subject to 300 different randomly generated external forces of maximum absolute magnitude of 70 (N) for a time period of 30 (s). Figure 4.1 illustrates the percentage success rate of $\mathcal{H}_2$-optimal MPC, $\mathcal{H}_\infty$-optimal MPC, and normal MPC on 300 different trails. The figure indicates that the robust optimal MPCs outperform the normal MPC. In particular, the success rate of $\mathcal{H}_2$-optimal MPC, $\mathcal{H}_\infty$-optimal MPC, and normal MPC is 38.6%, 33.3%, and 22.2%, respectively. Here, the termination criterion or failure is determined when the rigid body links connecting the knee with the toe hit the ground.

4.4.3 Experimental Evaluations

For the experimental validations, we synthesize both $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs at 0.1, 0.3, and 0.5 (m/s). We consider various indoor experimental investigations, particularly a) forward trot gait with $\mathcal{H}_2$-optimal MPC at 0.1, 0.3, and 0.5 (m/s), b) forward trot gait with $\mathcal{H}_\infty$-optimal MPC at 0.1, 0.3, and 0.5 (m/s), c) $\mathcal{H}_2$-optimal MPC trotting on even

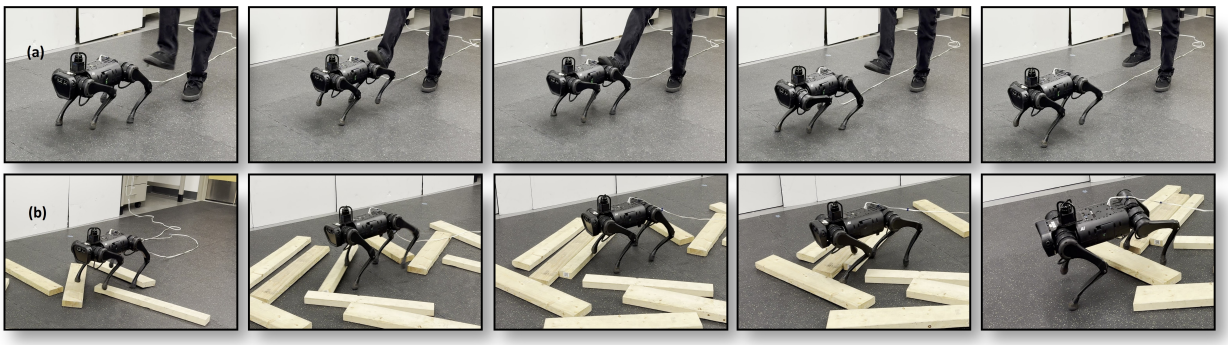


Figure 4.3: Snapshots illustrating the performance of $\mathcal{H}_2$-optimal MPC. Time increases from left to right. (a) Locomotion subject to external pushes at 0.5 (m/s). (b) Locomotion on uneven terrain with wooden blocks at 0.5 (m/s). The videos are available online [].

terrains at 0.5 (m/s), and d) push disturbances to $\mathcal{H}_2$-optimal MPC. Figure 4.3 (a) indicates the snapshots of stable locomotion of the A1 quadrupedal robot in the presence of external disturbances, and Fig. 4.3 (b) shows locomotion on uneven terrain with wooden blocks. Both experiments are performed using $\mathcal{H}_2$-optimal MPC with a forward speed of 0.5 (m/s). In addition, Fig. 4.2 shows the vertical component of the GRF, prescribed by the high-level $\mathcal{H}_2$-optimal MPC in (4.16), for the front right leg when the robot is subject to external disturbances, as indicated in Fig. 4.3 (a). In all of the experiments, the $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs successfully stabilized the locomotion of the quadrupedal robot. Simulation and experimental videos are available online [].

4.5 Summary

This chapter presented a formal approach for synthesizing robust $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs to stabilize the periodic locomotion of legged robots. The proposed algorithm builds on the existing optimization-based control stack. The chapter presented the conditions under which the discrete-time, closed-loop template dynamics with MPC around a periodic gait can be transformed into a linear system with a constant state-transition matrix using Floquet's theory. The chapter then presented an approach to systematically generate $\mathcal{H}_2$- and $\mathcal{H}_\infty$- robust controllers subject to parametric uncertainty using LMIs. We subsequently established a set of conditions guaranteeing the existence of such robust optimal controllers. The proposed $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs were extensively validated both numerically and experimentally for the robust locomotion of the A1 quadrupedal robot subject to various external disturbances and uneven terrains. Our numerical analysis suggests a significant improvement in the performance of robust locomotion compared to the normal MPC. Future work will investigate the incorporation of different kinds of uncertainties with robust adaptive predictive controllers.

Chapter 5

Conclusion and Future

The objective of this chapter is to present the concluding remarks and future research directions to be pursued. In Section 5.1, we will briefly summarize the work done during this thesis, which consists of the materials outlined in Chapters 2, 3, and 4. The chapter also provides possible future research directions in Section - 5.2 and concludes with final thoughts in Section - 5.3.

5.1 Summary of Contribution

5.1.1 Development of Layered Control Architectures

Chapter 2 proposed and developed two hierarchical control architectures for the real-time motion planning and control of legged robots. In particular, the first layered control architecture consists of an event-based MPC scheme at the higher level that computes the optimal COM trajectories of the LIP model. A low-level QP-based virtual constraints controller then tracks these optimal COM trajectories. The event-based MPC scheme is employed at the beginning of each continuous-time domain, significantly reducing the computational burden associated with real-time planning MPC schemes. The proposed event-based MPC scheme is then numerically evaluated using the full-order mathematical model of Ghost Robotics Vision60. To alleviate the problems associated with the LIP model and generate agile locomotion, the second layered control architecture uses a time-based MPC scheme based on the SRBD model. This time-based MPC scheme generates optimal GRFs using the SRBD model at every sample time, which are then tracked using a low-level virtual constraints controller. The proposed time-based MPC scheme is then numerically evaluated for the generation of agile locomotion using the full-order mathematical model of the Unitree A1 robot.

5.1.2 Developement of Robust Predictive Controllers

Following the work in Chapter 2, Chapter 3 also presents a hierarchical control algorithm for the real-time motion planning and control of quadrupedal robots while simultaneously bridging the gap between reduced- and full-order models. At the higher level, a computationally tractable Robust Model Predictive Control framework is developed to steer reduced-order

locomotion models subject to a convex set of uncertainties arising from abstraction and unmodeled dynamics. In particular, the RMPC algorithm is applied to the SRBD model to generate optimal GRFs, which are then mapped to the full-order model while imposing virtual constraints for whole-body motion control using a low-level QP-based nonlinear controller. The proposed RMPC framework allows the integration of the hierarchical controller with Reinforcement Learning techniques to train a neural network to compute the vertices of the uncertainty set numerically. The proposed hierarchical control algorithm is validated numerically and experimentally for robust and blind locomotion of the A1 quadrupedal robot on different indoor and outdoor terrains and at different speeds. The numerical analysis of the RMPC suggests significant improvement in the performance of the rough terrain locomotion compared to the nominal MPC. In particular, there is a percentage improvement in the success rate of about 60.7% for rough terrain locomotion over 550 randomly generated terrains. Finally, our experimental studies indicate a significant reduction in the gap between the reduced- and full-order models by comparing the desired and actual GRFs computed by the proposed high- and low-level controllers.

5.1.3 $\mathcal{H}_2/\mathcal{H}_\infty$ Predictive Controllers for Periodic Legged Locomotion

Chapter 4 presented a formal approach for synthesizing robust $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs to stabilize the periodic locomotion of legged robots. We proposed an algorithm that builds on the existing optimization-based control stack. The first contribution of this chapter is to outline the set of conditions under which the closed-loop nonlinear dynamics around a periodic orbit can be transformed into an LTI system using Floquet theory. We then outlined an approach to systematically generate parameterized $\mathcal{H}_2$- and $\mathcal{H}_\infty$- robust controllers using LMIs. We subsequently established a set of conditions guaranteeing the existence of such robust optimal controllers. The proposed $\mathcal{H}_2$- and $\mathcal{H}_\infty$-optimal MPCs are extensively validated both numerically and experimentally for the robust locomotion of the A1 quadrupedal robot subject to various external disturbances and uneven terrains. Our numerical analysis suggests a significant improvement in the performance of robust locomotion compared to the nominal MPC. In particular, we notice an improvement of around 50% for $\mathcal{H}_\infty$-optimal MPC compared to the nominal MPC and another 15% improvement with the $\mathcal{H}_2$-optimal MPC over $\mathcal{H}_\infty$-optimal MPC. Experimental case studies also show significant robustness.

5.2 Possible Future Research Directions

This dissertation builds robust layered control architectures for the real-time planning and control of legged robots. However, we believe there are certain limitations that can be addressed in future work.

For instance, the work in Chapter - 2 develops event-based MPC scheme with a shorter planning horizon. While this may be sufficient for simple gaits like trotting and walking, it may not be the case for complicated and more agile gaits like bounding. In such cases, the planning horizon must include multiple domains in order to ensure stability. This may cause other problems, such as larger optimization problems, which can be computationally expensive to solve. One potential solution can be offline trajectory optimization, where a dynamically stable gait is generated and provided as a well-designed heuristic to the real-time MPC.

The work in Chapter - 3 builds a computationally tractable RMPC framework using SRB dynamics. The fundamental assumption behind SRB dynamics is that the mass of the legs is insignificant compared to the mass of the torso. This is only valid for small quadrupedal robots. In order to accommodate the legs, more sophisticated template models have to be devised. Using data-driven or model-free techniques such as Reinforcement Learning could potentially offer a solution.

In order to retain computational tractability, the work in this thesis relies heavily on convex QPs. While QPs provide good control solutions in a real-time manner, they significantly restrict the dynamic models one can use. In particular, convex QP-based MPCs require discrete-time linear dynamics or linearized nonlinear dynamics. This restricts the full potential and limits it to certain gaits. In order to realize the full scope of the robot, general nonlinear programming problems (NLP) have to be considered. The disadvantages of NLP such as no theoretical guarantees and computational intractability can be mitigated by approximating them using either a single convex program or a sequence of convex programs. These convex programs can either be QPs or more complicated algorithms such as second-order cone programs or semi-definite programs.

One other future direction could be to explore the possibility of using tube-based MPC or stochastic MPC. The limiting factor in both cases is their heavy reliance on offline computation, which restricts them to linear template models. To circumvent this issue, we can use the idea developed in Chapter -4. Using a suitable coordinate transformation, the general nonlinear discrete-time dynamics can be converted into linear time-invariant dynamics under some mild assumptions. This new LTI system can be analyzed within the frameworks of tube MPC and stochastic MPC to generate more robust solutions.

Finally, the last but interesting research avenue to persuade would be the development of disturbance observers. The work in this thesis developed controllers that can accommodate uncertain explicitly. By incorporating disturbance observers, such as conjugate momentum-based observers, which directly measure the external force, one can inform the robust controllers more effectively. Combining disturbance observers with robust controllers can provide a solid control solution and capacitate legged robots the power to navigate complex environments.

5.3 Final Thoughts

This dissertation has contributed to the development of computationally tractable control architectures for the robust locomotion of quadrupedal robots. We have developed two innovative algorithms to improve the robustness of existing MPC schemes. We have also established theoretical guarantees for the proposed algorithms. We hope that this work will sufficiently highlight the significance of using robust controllers and will motivate other researchers to explore the possibility of extending these controllers to more complex legged systems such as bipedal robots.

Bibliography

[1] E Lucas. Huitieme recreation-la machine a marcher. *Recreations Mathematiques*, 4:198–204, 1894.

[2] L. A. Rygg. Mechanical horse, US Patent number 491927, 1893.

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

[4] Robert B McGhee. Some finite state aspects of legged locomotion. *Mathematical Biosciences*, 2(1-2):67–84, 1968.

[5] Robert B McGhee and Andrew A Frank. On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, 3:331–351, 1968.

[6] Shigeo Hirose. A study of design and control of a quadruped walking vehicle. *The International Journal of Robotics Research*, 3(2):113–133, 1984.

[7] Marc H Raibert, H Benjamin Brown Jr, and Michael Chepponis. Experiments in balance with a 3d one-legged hopping machine. *The International Journal of Robotics Research*, 3(2):75–92, 1984.

[8] Marc Raibert, Michael Chepponis, and HBJR Brown. Running on four legs as though they were one. *IEEE Journal on Robotics and Automation*, 2(2):70–82, 1986.

[9] Marc H Raibert. *Legged robots that balance*. MIT press, 1986.

[10] Sangbae Kim and Patrick M Wensing. Design of dynamic legged robots. *Foundations and Trends in Robotics*, 5(2):117–190, 2017.

[11] Manuel F Silva and JA Tenreiro Machado. A historical perspective of legged robots. *Journal of Vibration and Control*, 13(9-10):1447–1486, 2007.

[12] JA Tenreiro Machado and Manuel F Silva. An overview of legged robots. In *International symposium on mathematical methods in engineering*, pages 1–40. MME Press Ankara Turkey, 2006.

[13] Priyaranjan Biswal and Prases K Mohanty. Development of quadruped walking robots: A review. *Ain Shams Engineering Journal*, 12(2):2017–2031, 2021.

[14] Hui Chai, Yibin Li, Rui Song, Guoteng Zhang, Qin Zhang, Song Liu, Jinmian Hou, Yaxian Xin, Ming Yuan, Guoxuan Zhang, et al. A survey of the development of quadruped robots: Joint configuration, dynamic locomotion control method and mobile manipulation approach. *Biomimetic Intelligence and Robotics*, 2(1):100029, 2022.

[15] Shigeo Hirose and Keisuke Kato. Study on quadruped walking robot in tokyo institute of technology-past, present and future. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 414–419. IEEE, 2000.

[16] Koushil Sreenath, Hae-Won Park, Ioannis Poulakakis, and Jessy W Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *The International Journal of Robotics Research*, 30(9):1170–1193, 2011.

[17] Jared Di Carlo, Patrick M Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1–9. IEEE, 2018.

[18] Gerardo Bledt, Matthew J Powell, Benjamin Katz, Jared Di Carlo, Patrick M Wensing, and Sangbae Kim. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2245–2252. IEEE, 2018.

[19] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *arXiv preprint arXiv:2205.02824*, 2022.

[20] Hae-Won Park, Sangin Park, and Sangbae Kim. Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3d running of mit cheetah 2. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5163–5170. IEEE, 2015.

[21] Hae-Won Park, Patrick M Wensing, and Sangbae Kim. High-speed bounding with the mit cheetah 2: Control design and experiments. *The International Journal of Robotics Research*, 36(2):167–192, 2017.

[22] Boston Dynamics Atlas Humanoid. What's new, atlas?[Online]. Available: https://www.youtube.com/watch?v=fRj34o4hN4I.

[23] Yanran Ding, Chuanzheng Li, and Hae-Won Park. Single leg dynamic motion planning with mixed-integer convex optimization. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6. IEEE, 2018.

[24] Quan Nguyen, Matthew J Powell, Benjamin Katz, Jared Di Carlo, and Sangbae Kim. Optimized jumping on the mit cheetah 3 robot. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7448–7454. IEEE, 2019.

[25] Quan Nguyen, Ayush Agrawal, William Martin, Hartmut Geyer, and Koushil Sreenath. Dynamic bipedal locomotion over stochastic discrete terrain. *The International Journal of Robotics Research*, 37(13-14):1537–1553, 2018.

[26] Alexander W Winkler, C Dario Bellicoso, Marco Hutter, and Jonas Buchli. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3):1560–1567, 2018.

[27] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. 2021.

[28] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[29] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[30] Marco Tranzatto, Takahiro Miki, Mihir Dharmadhikari, Lukas Bernreiter, Mihir Kulkarni, Frank Mascarich, Olov Andersson, Shehryar Khattak, Marco Hutter, Roland Siegwart, et al. Cerberus in the darpa subterranean challenge. *Science Robotics*, 7(66):eabp9742, 2022.

[31] Ghost Robotics, https://www.ghostrobotics.io/.

[32] Spot for Construction,https://www.bostondynamics.com/solutions/construction.

[33] ANYbotics Robotic Industrial inspection, https://www.anybotics.com/robotic-industrial-inspection/.

[34] Michael P Murphy, Benjamin Stephens, Yeuhi Abe, and Alfred A Rizzi. High degree-of-freedom dynamic manipulation. In *Unmanned Systems Technology XIV*, volume 8387, pages 339–348. SPIE, 2012.

[35] Yuntao Ma, Farbod Farshidian, Takahiro Miki, Joonho Lee, and Marco Hutter. Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators. *IEEE Robotics and Automation Letters*, 2022.

[36] Simon Zimmermann, Roi Poranne, and Stelian Coros. Go fetch!-dynamic grasps using boston dynamics spot with external robotic arm. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4488–4494. IEEE, 2021.

[37] Jean-Pierre Sleiman, Farbod Farshidian, Maria Vittoria Minniti, and Marco Hutter. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6(3):4688–4695, 2021.

[38] C Dario Bellicoso, Koen Krämer, Markus Stäuble, Dhionis Sako, Fabian Jenelten, Marko Bjelonic, and Marco Hutter. Alma-articulated locomotion and manipulation for a torque-controllable robot. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8477–8483. IEEE, 2019.

[39] Kaveh Akbari Hamed, Jeeseop Kim, and Abhishek Pandala. Quadrupedal locomotion via event-based predictive control and qp-based virtual constraints. *IEEE Robotics and Automation Letters*, 5(3):4463–4470, 2020.

[40] Jeeseop Kim and Kaveh Akbari Hamed. Cooperative locomotion via supervisory predictive control and distributed nonlinear controllers. *Journal of Dynamic Systems, Measurement, and Control*, 144(3), 2022.

[41] Vinay R Kamidi, Jeeseop Kim, Randall T Fawcett, Aaron D Ames, and Kaveh Akbari Hamed. Distributed quadratic programming-based nonlinear controllers for periodic gaits on legged robots. *IEEE Control Systems Letters*, 6:2509–2514, 2022.

[42] Abhishek Pandala, Vinay R Kamidi, and Kaveh Akbari Hamed. Decentralized control schemes for stable quadrupedal locomotion: A decomposition approach from centralized controllers. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3975–3981. IEEE, 2020.

[43] Wen-Loong Ma, Noel Csomay-Shanklin, and Aaron D. Ames. Coupled control systems: Periodic orbit generation with application to quadrupedal locomotion. *IEEE Control Systems Letters*, 5(3):935–940, 2021.

[44] Wen-Loong Ma, Noel Csomay-Shanklin, Shishir Kolathaya, Kaveh Akbari Hamed, and Aaron D. Ames. Coupled control lyapunov functions for interconnected systems, with application to quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 6(2):3761–3768, 2021.

[45] Matthew Johnson, Brandon Shrewsbury, Sylvain Bertrand, Tingfan Wu, Daniel Duran, Marshall Floyd, Peter Abeles, Douglas Stephen, Nathan Mertins, Alex Lesman, et al. Team ihmc's lessons learned from the darpa robotics challenge trials. *Journal of Field Robotics*, 32(2):192–208, 2015.

[46] Gill Pratt and Justin Manzo. The darpa robotics challenge [competitions]. *IEEE Robotics & Automation Magazine*, 20(2):10–12, 2013.

[47] Joel Burdick and Paolo Fiorini. Minimalist jumping robots for celestial exploration. *The International Journal of Robotics Research*, 22(7-8):653–674, 2003.

[48] DARPA Tactical Technology Office. DARPA Robotics challenge, https://www.darpa.mil/program/darpa-robotics-challenge, 2012.

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

[50] Aaron D Ames. Human-inspired control of bipedal walking robots. *IEEE Transactions on Automatic Control*, 59(5):1115–1130, 2014.

[51] Christine Chevallereau, Gabriel Abba, Yannick Aoustin, Franck Plestan, Eric Westervelt, Carlos Canudas De Wit, and Jessy Grizzle. Rabbit: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, 2003.

[52] Eric R Westervelt, Jessy W Grizzle, and Daniel E Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE transactions on automatic control*, 48(1):42–56, 2003.

[53] Wen-Loong Ma, Shishir Kolathaya, Eric R Ambrose, Christian M Hubicki, and Aaron D Ames. Bipedal robotic running with durus-2d: Bridging the gap between theory and experiment. In *Proceedings of the 20th international conference on hybrid systems: computation and control*, pages 265–274, 2017.

[54] E.R. Westervelt, J.W. Grizzle, C. Chevallereau, J.H. Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. Taylor & Francis/CRC, 2007.

[55] Alireza Ramezani, Jonathan W Hurst, Kaveh Akbari Hamed, and Jessy W Grizzle. Performance analysis and feedback control of atrias, a three-dimensional bipedal robot. *Journal of Dynamic Systems, Measurement, and Control*, 136(2):021012, 2014.

[56] Christine Chevallereau, Jessy W Grizzle, and Ching-Long Shih. Asymptotically stable walking of a five-link underactuated 3-d bipedal robot. *IEEE transactions on robotics*, 25(1):37–50, 2009.

[57] Ayonga Hereid, Eric A Cousineau, Christian M Hubicki, and Aaron D Ames. 3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1447–1454. IEEE, 2016.

[58] Nicolaus A Radford, Philip Strawser, Kimberly Hambuchen, Joshua S Mehling, William K Verdeyen, A Stuart Donnan, James Holley, Jairo Sanchez, Vienny Nguyen, Lyndon Bridgwater, et al. Valkyrie: Nasa's first bipedal humanoid robot. *Journal of Field Robotics*, 32(3):397–419, 2015.

[59] Kaveh Akbari Hamed, Vinay R Kamidi, Wen-Loong Ma, Alexander Leonessa, and Aaron D Ames. Hierarchical and safe motion control for cooperative locomotion of robotic guide dogs and humans: A hybrid systems approach. *IEEE Robotics and Automation Letters*, 5(1):56–63, 2019.

[60] R. J. Full and D. E. Koditschek. Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202:3325–3332, December 1999.

[61] Philip Holmes, Robert J Full, Dan Koditschek, and John Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM review*, 48(2):207–304, 2006.

[62] Jessy W Grizzle, Christine Chevallereau, Ryan W Sinnet, and Aaron D Ames. Models, feedback control, and open problems of 3d bipedal robotic walking. *Automatica*, 50(8):1955–1988, 2014.

[63] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[64] Dimitri P Bertsekas. *Nonlinear programming*. Taylor & Francis, 1997.

[65] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

[66] David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[67] David S Johnson and Michael R Garey. *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman, 1979.

[68] Katta G Murty and Santosh N Kabadi. Some np-complete problems in quadratic and nonlinear programming. Technical report, 1985.

[69] Shuuji Kajita and Kazuo Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1405–1406. IEEE Computer Society, 1991.

[70] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 1620–1626. IEEE, 2003.

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

[72] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[73] Manfred Morari and Jay H Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, 1999.

[74] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[75] James B Rawlings. Tutorial overview of model predictive control. *IEEE control systems magazine*, 20(3):38–52, 2000.

[76] David Q Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[77] Andrei Herdt, Nicolas Perrin, and Pierre-Brice Wieber. Walking without thinking about it. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 190–195. IEEE, 2010.

[78] Andrei Herdt, Holger Diedam, Pierre-Brice Wieber, Dimitar Dimitrov, Katja Mombaur, and Moritz Diehl. Online walking motion generation with automatic footstep placement. *Advanced Robotics*, 24(5-6):719–737, 2010.

[79] Robert J Griffin, Georg Wiedebach, Sylvain Bertrand, Alexander Leonessa, and Jerry Pratt. Walking stabilization using step timing and location adjustment on the humanoid robot, atlas. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 667–673. IEEE, 2017.

[80] Johannes Englsberger, Christian Ott, and Alin Albu-Schäffer. Three-dimensional bipedal walking control based on divergent component of motion. *Ieee transactions on robotics*, 31(2):355–368, 2015.

[81] Alexander W Winkler, Farbod Farshidian, Diego Pardo, Michael Neunert, and Jonas Buchli. Fast trajectory optimization for legged robots using vertex-based zmp constraints. *IEEE Robotics and Automation Letters*, 2(4):2201–2208, 2017.

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

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

[84] Twan Koolen, Tomas De Boer, John Rebula, Ambarish Goswami, and Jerry Pratt. Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models. *The international journal of robotics research*, 31(9):1094–1113, 2012.

[85] Jerry Pratt, Twan Koolen, Tomas De Boer, John Rebula, Sebastien Cotton, John Carff, Matthew Johnson, and Peter Neuhaus. Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid. *The international journal of robotics research*, 31(10):1117–1133, 2012.

[86] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[87] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[88] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984.

[89] William John Schwind. *Spring loaded inverted pendulum running: A plant model*. PhD thesis, 1998.

[90] RMcN Alexander et al. Three uses for springs in legged locomotion. *International Journal of Robotics Research*, 9(2):53–61, 1990.

[91] Richard Altendorfer, Uluc Saranli, Haldun Komsuoglu, Daniel Koditschek, H Benjamin Brown, Martin Buehler, Ned Moore, Dave McMordie, and Robert Full. Evidence for spring loaded inverted pendulum running in a hexapod robot. In *Experimental Robotics VII*, pages 291–302. Springer, 2001.

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

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

[94] Ioannis Poulakakis, Evangelos Papadopoulos, and Martin Buehler. On the stability of the passive dynamics of quadrupedal running with a bounding gait. *The International Journal of Robotics Research*, 25(7):669–687, 2006.

[95] Taku Komura, Howard Leung, Shunsuke Kudoh, and James Kuffner. A feedback controller for biped humanoids that can counteract large perturbations during gait. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1989–1995. IEEE, 2005.

[96] Taku Komura, Akinori Nagano, Howard Leung, and Yoshihisa Shinagawa. Simulating pathological gait using the enhanced linear inverted pendulum model. *IEEE Transactions on biomedical engineering*, 52(9):1502–1513, 2005.

[97] Sung-Hee Lee and Ambarish Goswami. Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 4667–4672. IEEE, 2007.

[98] Avinash Siravuru, Sasi P Viswanathan, Koushil Sreenath, and Amit K Sanyal. The reaction mass biped: Geometric mechanics and control. *Journal of Intelligent & Robotic Systems*, 89(1):155–173, 2018.

[99] Gerardo Bledt, Patrick M Wensing, and Sangbae Kim. Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4102–4109. IEEE, 2017.

[100] Yanran Ding, Abhishek Pandala, Chuanzheng Li, Young-Ha Shin, and Hae-Won Park. Representation-free model predictive control for dynamic motions in quadrupeds. *IEEE Transactions on Robotics*, 37(4):1154–1171, 2021.

[101] Yanran Ding, Abhishek Pandala, and Hae-Won Park. Real-time model predictive control for versatile dynamic motions in quadrupedal robots. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8484–8490. IEEE, 2019.

[102] Matthew Chignoli and Patrick M Wensing. Variational-based optimal control of underactuated balancing for dynamic quadrupeds. *IEEE Access*, 8:49785–49797, 2020.

[103] Seungwoo Hong, Joon-Ha Kim, and Hae-Won Park. Real-time constrained nonlinear model predictive control on so (3) for dynamic legged locomotion. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3982–3989. IEEE, 2020.

[104] David E Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous robots*, 35(2):161–176, 2013.

[105] Hae-Won Park, Patrick Wensing, and Sangbae Kim. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. 07 2015.

[106] Yanran Ding, Chuanzheng Li, and Hae-Won Park. Single leg dynamic motion planning with mixed-integer convex optimization. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6. IEEE, 2018.

[107] Avik De and Daniel E Koditschek. Parallel composition of templates for tail-energized planar hopping. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4562–4569. IEEE, 2015.

[108] Avik De, Samuel A Burden, and Daniel E Koditschek. A hybrid dynamical extension of averaging and its application to the analysis of legged gait stability. *The International Journal of Robotics Research*, 37(2-3):266–286, 2018.

[109] Kemin Zhou and John Comstock Doyle. *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ, 1998.

[110] IS Khalil, JC Doyle, and K Glover. *Robust and optimal control*. Prentice hall, 1996.

[111] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton university press, 2009.

[112] Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization–methodology and applications. *Mathematical programming*, 92(3):453–480, 2002.

[113] Brian Armstrong-Hélouvry, Pierre Dupont, and Carlos Canudas De Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138, 1994.

[114] EJ Berger. Friction modeling for dynamic system simulation. *Appl. Mech. Rev.*, 55(6):535–577, 2002.

[115] Farid Al-Bender. Fundamentals of friction modeling. In *Proceedings, ASPE Spring Topical Meeting on Control of Precision Systems, MIT, April 11-13, 2010*, volume 48, pages 117–122. ASPE-The American Society of precision Engineering; 301 Glenwood Avenue ..., 2010.

[116] Anindya Chatterjee. On the realism of complementarity conditions in rigid body collisions. *Nonlinear Dynamics*, 20(2):159–168, 1999.

[117] Paul Van Dooren and J Sreedhar. When is a periodic discrete-time system equivalent to a time-invariant one? *Linear Algebra and its Applications*, 212:131–151, 1994.

[118] Carmen Chicone. *Ordinary differential equations with applications*, volume 34. Springer Science & Business Media, 2006.

[119] Maurício C De Oliveira, Jacques Bernussou, and José C Geromel. A new discrete-time robust stability condition. *Systems & control letters*, 37(4):261–265, 1999.

[120] Maurıcio C De Oliveira, José C Geromel, and Jacques Bernussou. Extended H_2 and H_∞ norm characterizations and controller parametrizations for discrete-time systems. *International journal of control*, 75(9):666–679, 2002.

[121] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[122] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[123] Anthony Stentz et al. The focussed d^* algorithm for real-time replanning. In *IJCAI*, volume 95, pages 1652–1659, 1995.

[124] Richard E Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial intelligence*, 27(1):97–109, 1985.

[125] Maxim Likhachev, Geoffrey J Gordon, and Sebastian Thrun. Ara*: Anytime a* with provable bounds on sub-optimality. *Advances in neural information processing systems*, 16, 2003.

[126] Mrinal Kalakrishnan, Jonas Buchli, Peter Pastor, Michael Mistry, and Stefan Schaal. Learning, planning, and control for quadruped locomotion over challenging terrain. *The International Journal of Robotics Research*, 30(2):236–258, 2011.

[127] Matt Zucker, Nathan Ratliff, Martin Stolle, Joel Chestnutt, J Andrew Bagnell, Christopher G Atkeson, and James Kuffner. Optimization and learning for rough terrain legged locomotion. *The International Journal of Robotics Research*, 30(2):175–191, 2011.

[128] Alexander W Winkler, Carlos Mastalli, Ioannis Havoutis, Michele Focchi, Darwin G Caldwell, and Claudio Semini. Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5148–5154. IEEE, 2015.

[129] Paul Vernaza, Maxim Likhachev, Subhrajit Bhattacharya, Sachin Chitta, Aleksandr Kushleyev, and Daniel D Lee. Search-based planning for a legged robot over rough terrain. In *2009 IEEE International Conference on Robotics and Automation*, pages 2380–2387. IEEE, 2009.

[130] Joel Chestnutt. *Navigation planning for legged robots*. PhD thesis, 2007.

[131] Richard E Korf. Artificial intelligence search algorithms. In *Algorithms and theory of computation handbook: special topics and techniques*, pages 22–22. 2010.

[132] Dave Ferguson, Maxim Likhachev, and Anthony Stentz. A guide to heuristic-based path planning. In *Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS)*, pages 9–18, 2005.

[133] Joel Chestnutt, Manfred Lau, German Cheung, James Kuffner, Jessica Hodgins, and Takeo Kanade. Footstep planning for the honda asimo humanoid. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 629–634. IEEE, 2005.

[134] Joel Chestnutt, James Kuffner, Koichi Nishiwaki, and Satoshi Kagami. Planning biped navigation strategies in complex environments. In *IEEE Int. Conf. Hum. Rob., Munich, Germany*, 2003.

[135] Joel Chestnutt, Koichi Nishiwaki, James Kuffner, and Satoshi Kagami. An adaptive action model for legged navigation planning. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 196–202. IEEE, 2007.

[136] Peter D Neuhaus, Jerry E Pratt, and Matthew J Johnson. Comprehensive summary of the institute for human and machine cognition's experience with littledog. *The International Journal of Robotics Research*, 30(2):216–235, 2011.

[137] Daniel Maier, Armin Hornung, and Maren Bennewitz. Real-time navigation in 3d environments based on depth camera data. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 692–697. IEEE, 2012.

[138] Tim Bretl, Sanjay Lall, Jean-Claude Latombe, and Stephen Rock. Multi-step motion planning for free-climbing robots. In *Algorithmic Foundations of Robotics VI*, pages 59–74. Springer, 2004.

[139] Yanran Ding, Chuanzheng Li, and Hae-Won Park. Kinodynamic motion planning for multi-legged robot jumping via mixed-integer convex program. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3998–4005. IEEE, 2020.

[140] Robin Deits and Russ Tedrake. Footstep planning on uneven terrain with mixed-integer convex optimization. In *2014 IEEE-RAS international conference on humanoid robots*, pages 279–286. IEEE, 2014.

[141] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40(3):429–455, 2016.

[142] Andrés Klee Valenzuela. *Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain*. PhD thesis, Massachusetts Institute of Technology, 2016.

[143] Daniel Liberzon. Calculus of variations and optimal control theory. In *Calculus of Variations and Optimal Control Theory*. Princeton university press, 2011.

[144] John T Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.

[145] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[146] Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, 2017.

[147] Yuval Tassa, Nicolas Mansard, and Emo Todorov. Control-limited differential dynamic programming. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1168–1175. IEEE, 2014.

[148] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012.

[149] Ruben Grandia, Farbod Farshidian, René Ranftl, and Marco Hutter. Feedback mpc for torque-controlled legged robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4730–4737. IEEE, 2019.

[150] Farbod Farshidian, Michael Neunert, Alexander W Winkler, Gonzalo Rey, and Jonas Buchli. An efficient optimal planning and control framework for quadrupedal locomotion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 93–100. IEEE, 2017.

[151] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4):1–8, 2012.

[152] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.

[153] Michael Neunert, Markus Stäuble, Markus Giftthaler, Carmine D Bellicoso, Jan Carius, Christian Gehring, Marco Hutter, and Jonas Buchli. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robotics and Automation Letters*, 3(3):1458–1465, 2018.

[154] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[155] Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. 2021.

[156] Gabriel Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. In *Robotics: Science and Systems*, 2022.

[157] Lokesh Krishna, Utkarsh A Mishra, Guillermo A Castillo, Ayonga Hereid, and Shishir Kolathaya. Learning linear policies for robust bipedal locomotion on terrains with varying slopes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5159–5164. IEEE, 2021.

[158] Lokesh Krishna, Guillermo A Castillo, Utkarsh A Mishra, Ayonga Hereid, and Shishir Kolathaya. Linear policies are sufficient to realize robust bipedal walking on challenging terrains. *IEEE Robotics and Automation Letters*, 7(2):2047–2054, 2022.

[159] Guillermo A Castillo, Bowen Weng, Wei Zhang, and Ayonga Hereid. Reinforcement learning-based cascade motion policy design for robust 3d bipedal locomotion. *IEEE Access*, 10:20135–20148, 2022.

[160] Sebastien Corner, Corina Sandu, and Adrian Sandu. Modeling and sensitivity analysis methodology for hybrid dynamical system. *Nonlinear Analysis: Hybrid Systems*, 31:19–40, 2019.

[161] Sebastien Corner, Adrian Sandu, and Corina Sandu. Adjoint sensitivity analysis of hybrid multibody dynamical systems. *Multibody System Dynamics*, 49(4):395–420, 2020.

[162] Tad McGeer et al. Passive dynamic walking. *Int. J. Robotics Res.*, 9(2):62–82, 1990.

[163] Tad McGeer. Passive walking with knees. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 1640–1645. IEEE, 1990.

[164] Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

[165] Steven H Collins and Andy Ruina. A bipedal walking robot with efficient and human-like gait. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 1983–1988. IEEE, 2005.

[166] Russ Tedrake, Teresa Weirui Zhang, Ming-fai Fong, and H Sebastian Seung. Actuating a simple 3d passive dynamic walker. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pages 4656–4661. IEEE, 2004.

[167] Russ Tedrake, Teresa Weirui Zhang, and H Sebastian Seung. Stochastic policy gradient reinforcement learning on a simple 3d biped. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2849–2854. IEEE, 2004.

[168] Martjin Wisse and Richard Q Van der Linde. *Delft pneumatic bipeds*, volume 34. Springer Science & Business Media, 2007.

[169] Stuart O Anderson, Martijn Wisse, Christopher G Atkeson, Jessica K Hodgins, GJ Zeglin, and Brian Moyer. Powered bipeds based on passive dynamic principles. In *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pages 110–116. IEEE, 2005.

[170] Mark W Spong. Passivity based control of the compass gait biped. *IFAC Proceedings Volumes*, 32(2):506–510, 1999.

[171] Mark W Spong and Francesco Bullo. Controlled symmetries and passive walking. *IEEE Transactions on Automatic Control*, 50(7):1025–1031, 2005.

[172] Elena Borzova and Yildirim Hurmuzlu. Passively walking five-link robot. *Automatica*, 40(4):621–629, 2004.

[173] B Espiau and A Goswami. Compass gait revisited. *IFAC Proceedings Volumes*, 27(14):839–846, 1994.

[174] Mariano Garcia, Anindya Chatterjee, Andy Ruina, and Michael Coleman. The Simplest Walking Model: Stability, Complexity, and Scaling. *Journal of Biomechanical Engineering*, 120(2):281–288, 04 1998.

[175] Stuart O Anderson, Martijn Wisse, Christopher G Atkeson, Jessica K Hodgins, GJ Zeglin, and Brian Moyer. Powered bipeds based on passive dynamic principles. In *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pages 110–116. IEEE, 2005.

[176] Arthur D Kuo. Energetics of actively powered locomotion using the simplest walking model. *J. Biomech. Eng.*, 124(1):113–120, 2002.

[177] Arthur D Kuo. Stabilization of lateral motion in passive dynamic walking. *The International journal of robotics research*, 18(9):917–930, 1999.

[178] Aaron D Ames, Robert D Gregg, Eric DB Wendel, and Shankar Sastry. On the geometric reduction of controlled three-dimensional bipedal robotic walkers. In *Lagrangian and Hamiltonian Methods for Nonlinear Control 2006*, pages 183–196. Springer, 2007.

[179] Aaron D. Ames and Shankar Sastry. Hybrid geometric reduction of hybrid systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 923–929, 2006.

[180] Aaron D Ames, Ryan W Sinnet, and Eric DB Wendel. Three-dimensional kneed bipedal walking: A hybrid geometric approach. In *International Workshop on Hybrid Systems: Computation and Control*, pages 16–30. Springer, 2009.

[181] Robert D Gregg and Mark W Spong. Reduction-based control of three-dimensional bipedal walking robots. *The International Journal of Robotics Research*, 29(6):680–702, 2010.

[182] Robert D Gregg and Ludovic Righetti. Controlled reduction with unactuated cyclic variables: Application to 3d bipedal walking with passive yaw rotation. *IEEE transactions on automatic control*, 58(10):2679–2685, 2013.

[183] Ryan W Sinnet and Aaron D Ames. Extending two-dimensional human-inspired bipedal robotic walking to three dimensions through geometric reduction. In *2012 American Control Conference (ACC)*, pages 4831–4836. IEEE, 2012.

[184] Ian R Manchester, Uwe Mettin, Fumiya Iida, and Russ Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, 2011.

[185] Anton S Shiriaev, Leonid B Freidovich, and Sergei V Gusev. Transverse linearization for controlled mechanical systems with several passive degrees of freedom. *IEEE Transactions on Automatic Control*, 55(4):893–906, 2010.

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

[187] Aaron D Ames, Kevin Galloway, Koushil Sreenath, and Jessy W Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, 2014.

[188] Kaveh Akbari Hamed and Jessy W Grizzle. Event-based stabilization of periodic orbits for underactuated 3-d bipedal robots with left-right symmetry. *IEEE Transactions on Robotics*, 30(2):365–381, 2013.

[189] Kaveh Akbari Hamed, Brian G Buss, and Jessy W Grizzle. Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations. *The International Journal of Robotics Research*, 35(8):977–999, 2016.

[190] K Akbari Hamed, N Sadati, WA Gruver, and GA Dumont. Exponential stabilisation of periodic orbits for running of a three-dimensional monopedal robot. *IET control theory & applications*, 5(11):1304–1320, 2011.

[191] N Sadati, GA Dumont, K Akbari Hamed, and WA Gruver. Two-level control scheme for stabilisation of periodic orbits for planar monopedal running. *IET control theory & applications*, 5(13):1528–1543, 2011.

[192] Jessy W Grizzle, Gabriel Abba, and Franck Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE Transactions on automatic control*, 46(1):51–64, 2001.

[193] Christine Chevallereau, Gabriel Abba, Franck Plestan, Eric Westervelt, Carlos Canudas de Wit, Jessy Grizzle, et al. Rabbit: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, 2003.

[194] ER Westervelt, B Morris, and KD Farrell. Analysis results and tools for the control of planar bipedal gaits using hybrid zero dynamics. *Autonomous Robots*, 23(2):131–145, 2007.

[195] Christine Chevallereau, Jessy W Grizzle, and Ching-Long Shih. Asymptotically stable walking of a five-link underactuated 3-d bipedal robot. *IEEE transactions on robotics*, 25(1):37–50, 2009.

[196] Hae-Won Park, Alireza Ramezani, and Jessy W Grizzle. A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking. *IEEE Transactions on Robotics*, 29(2):331–345, 2012.

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

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

[199] Anne E Martin, David C Post, and James P Schmiedeler. The effects of foot geometric properties on the gait of planar bipeds walking under hzd-based control. *The International Journal of Robotics Research*, 33(12):1530–1543, 2014.

[200] Alireza Ramezani, Jonathan W Hurst, Kaveh Akbari Hamed, and Jessy W Grizzle. Performance analysis and feedback control of atrias, a three-dimensional bipedal robot. *Journal of Dynamic Systems, Measurement, and Control*, 136(2):021012, 2014.

[201] Cenk Oguz Saglam and Katie Byl. Meshing hybrid zero dynamics for rough terrain walking. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5718–5725. IEEE, 2015.

[202] Eric Ambrose, Wen-Loong Ma, Christian Hubicki, and Aaron D Ames. Toward benchmarking locomotion economy across design configurations on the modular robot: Amber-3m. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1270–1276. IEEE, 2017.

[203] Ayonga Hereid, Christian M Hubicki, Eric A Cousineau, and Aaron D Ames. Dynamic humanoid locomotion: A scalable formulation for hzd gait optimization. *IEEE Transactions on Robotics*, 34(2):370–387, 2018.

[204] Kaveh Akbari Hamed and Aaron D Ames. Nonholonomic hybrid zero dynamics for the stabilization of periodic orbits: Application to underactuated robotic walking. *IEEE Transactions on Control Systems Technology*, 28(6):2689–2696, 2019.

[205] Qu Cao and Ioannis Poulakakis. Quadrupedal running with a flexible torso: control and speed transitions with sums-of-squares verification. *Artificial Life and Robotics*, 21(4):384–392, 2016.

[206] Kaveh Akbari Hamed, Wen-Loong Ma, and Aaron D Ames. Dynamically stable 3d quadrupedal walking with multi-domain hybrid system models and virtual constraint controllers. In *2019 American Control Conference (ACC)*, pages 4588–4595. IEEE, 2019.

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

[208] Wen-Loong Ma, Noel Csomay-Shanklin, and Aaron D Ames. Quadrupedal robotic walking on sloped terrains via exact decomposition into coupled bipedal robots. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4006–4011. IEEE, 2020.

[209] Wen-Loong Ma, Noel Csomay-Shanklin, Shishir Kolathaya, Kaveh Akbari Hamed, and Aaron D Ames. Coupled control lyapunov functions for interconnected systems, with application to quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 6(2):3761–3768, 2021.

[210] Wen-Loong Ma. *From Bipedal to Quadrupedal Locomotion, Experimental Realization of Lyapunov Approaches*. PhD thesis, California Institute of Technology, 2021.

[211] Randall T Fawcett, Abhishek Pandala, Aaron D Ames, and Kaveh Akbari Hamed. Robust stabilization of periodic gaits for quadrupedal locomotion via qp-based virtual constraint controllers. *IEEE Control Systems Letters*, 6:1736–1741, 2021.

[212] Abhishek Pandala, Randall T Fawcett, Ugo Rosolia, Aaron D Ames, and Kaveh Akbari Hamed. Robust predictive control for quadrupedal locomotion: Learning to close the gap between reduced-and full-order models. *IEEE Robotics and Automation Letters*, 7(3):6622–6629, 2022.

[213] Robert D Gregg and Jonathon W Sensinger. Towards biomimetic virtual constraint control of a powered prosthetic leg. *IEEE Transactions on control systems technology*, 22(1):246–254, 2013.

[214] Robert D Gregg, Tommaso Lenzi, Levi J Hargrove, and Jonathon W Sensinger. Virtual constraint control of a powered prosthetic leg: From simulation to experiments with transfemoral amputees. *IEEE Transactions on Robotics*, 30(6):1455–1471, 2014.

[215] Huihua Zhao, Jonathan Horn, Jacob Reher, Victor Paredes, and Aaron D Ames. Multicontact locomotion on transfemoral prostheses via hybrid system models and optimization-based control. *IEEE Transactions on Automation Science and Engineering*, 13(2):502–513, 2016.

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

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

[218] Omar Harib, Ayonga Hereid, Ayush Agrawal, Thomas Gurriet, Sylvain Finet, Guilhem Boeris, Alexis Duburcq, M Eva Mungai, Mattieu Masselin, Aaron D Ames, et al. Feedback control of an exoskeleton for paraplegics: Toward robustly stable, hands-free dynamic walking. *IEEE Control Systems Magazine*, 38(6):61–87, 2018.

[219] C Dario Bellicoso, Christian Gehring, Jemin Hwangbo, Péter Fankhauser, and Marco Hutter. Perception-less terrain adaptation through whole body control and hierarchical optimization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 558–564. IEEE, 2016.

[220] Ruben Grandia, Farbod Farshidian, Alexey Dosovitskiy, René Ranftl, and Marco Hutter. Frequency-aware model predictive control. *IEEE Robotics and Automation Letters*, 4(2):1517–1524, 2019.

[221] Michael A Hopkins, Dennis W Hong, and Alexander Leonessa. Humanoid locomotion on uneven terrain using the time-varying divergent component of motion. In *2014 ieee-ras international conference on humanoid robots*, pages 266–272. IEEE, 2014.

[222] Michael A Hopkins, Dennis W Hong, and Alexander Leonessa. Compliant locomotion using whole-body control and divergent component of motion tracking. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5726–5733. IEEE, 2015.

[223] Miomir Vukobratović and Branislav Borovac. Zero-moment point—thirty five years of its life. *International journal of humanoid robotics*, 1(01):157–173, 2004.

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

[225] Jonghoon Park and Youngil Youm. General zmp preview control for bipedal walking. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 2682–2687. IEEE, 2007.

[226] Scott Kuindersma, Frank Permenter, and Russ Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2589–2594. IEEE, 2014.

[227] ANYbotics, https://www.anybotics.com/.

[228] Unitree, http://www.unitree.cc/.

[229] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[230] Luis Sentis and Oussama Khatib. Control of free-floating humanoid robots through task prioritization. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1718–1723. IEEE, 2005.

[231] Luis Sentis and Oussama Khatib. A whole-body control framework for humanoids operating in human environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2641–2648. IEEE, 2006.

[232] Luis Sentis. *Synthesis and control of whole-body behaviors in humanoid systems*. PhD thesis, 2007.

[233] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Fast resolution of hierarchized inverse kinematics with inequality constraints. In *2010 IEEE International Conference on Robotics and Automation*, pages 3733–3738. IEEE, 2010.

[234] Marco Hutter, Hannes Sommer, Christian Gehring, Mark Hoepflinger, Michael Bloesch, and Roland Siegwart. Quadrupedal locomotion using hierarchical operational space control. *The International Journal of Robotics Research*, 33(8):1047–1062, 2014.

[235] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.

[236] Scott Kuindersma, Frank Permenter, and Russ Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2589–2594. IEEE, 2014.

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

[238] Federico L Moro and Luis Sentis. Whole-body control of humanoid robots. *Humanoid Robotics: A reference, Springer, Dordrecht*, 2019.

[239] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

[240] Erwin Coumans. Bullet physics engine https://pybullet.org/wordpress/.

[241] Smith Russel. Open dynamics engine http://ode.org/.

[242] Mark W Spong, Seth Hutchinson, Mathukumalli Vidyasagar, et al. *Robot modeling and control*, volume 3. Wiley New York, 2006.

[243] Martin L Felis. Rbdl: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots*, 41(2):495–511, 2017.

[244] Ayonga Hereid and Aaron D Ames. Frost: Fast robot optimization and simulation toolkit. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 719–726. IEEE, 2017.

[245] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. The pinocchio c++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, pages 614–619. IEEE, 2019.

[246] Edward J Haug. *Computer aided kinematics and dynamics of mechanical systems. Vol. 1: basic methods*. Allyn & Bacon, Inc., 1989.

[247] Edward J Haug. Dynamic mechanical systems with coulomb friction, stiction, impact and constraint addition-detection. i. *Mechanisms and Machine Theory*, 21:407–416, 1986.

[248] Mihai Anitescu and Alessandro Tasora. An iterative approach for cone complementarity problems for nonsmooth dynamics. *Computational Optimization and Applications*, 47(2):207–235, 2010.

[249] Bruce R Donald and Dinesh K Pai. On the motion of compliantly-connected rigid bodies in contact. ii. a system for analyzing designs for assembly. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 1756–1762. IEEE, 1990.

[250] Jong-Shi Pang, Vijay Kumar, and Peng Song. Convergence of time-stepping method for initial and boundary-value frictional compliant contact problems. *SIAM journal on numerical analysis*, 43(5):2200–2226, 2005.

[251] JS Pang, V Kumar, and J Trinkle. On a continuous-time quasistatic frictional contact model with local compliance. *International Journal for Numerical Methods in Engineering*, 2007.

[252] Peng Song, Jong-Shi Pang, and Vijay Kumar. A semi-implicit time-stepping model for frictional compliant contact problems. *International journal for numerical methods in engineering*, 60(13):2231–2261, 2004.

[253] C.C. De Wit, H. Olsson, K.J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, Mar 1995.

[254] Per Lötstedt. Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal on Applied Mathematics*, 42(2):281–296, 1982.

[255] David Baraff. *Dynamic simulation of nonpenetrating rigid bodies*. PhD thesis, 1992.

[256] David Baraff. Issues in computing contact forces for non-penetrating rigid bodies. *Algorithmica*, 10(2):292–352, 1993.

[257] Painlevé Paul. Sur le lois frottement de glissemment. *Comptes Rendu des Séances de lÍAcadémie desSciences*, 121(3):112–115, 1895.

[258] David E Stewart and Jeffrey C Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15):2673–2691, 1996.

[259] Mihai Anitescu and Florian A Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14(3):231–247, 1997.

[260] Emanuel Todorov. A convex, smooth and invertible contact model for trajectory optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 1071–1076. IEEE, 2011.

[261] Evan Drumwright and Dylan A Shell. Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation. In *Algorithmic foundations of robotics IX*, pages 249–266. Springer, 2010.

[262] Hammad Mazhar, Toby Heyn, Arman Pazouki, Daniel Melanz, Andrew Seidl, Aaron Bartholomew, Alessandro Tasora, and Dan Negrut. Chrono: a parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics. *Mechanical Sciences*, 4(1):49–64, 2013.

[263] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[264] Jemin Hwangbo, Joonho Lee, and Marco Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018.

[265] Physx physics engine https://developer.nvidia.com/physx-sdk.

[266] Dart physics engine http://dartsim.github.io/.

[267] Michael A Sherman, Ajay Seth, and Scott L Delp. Simbody: multibody dynamics for biomedical research. *Procedia Iutam*, 2:241–261, 2011.

[268] Havok physics engine https://www.havok.com/.

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

[270] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[271] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[272] Miomir Vukobratović and Juri Stepanenko. On the stability of anthropomorphic systems. *Mathematical biosciences*, 15(1-2):1–37, 1972.

[273] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. *IEEE Transactions on control systems technology*, 18(2):267–278, 2009.

[274] A. Isidori. *Nonlinear Control Systems*. Springer; 3rd edition, 1995.

[275] Kinova Robotic arm, https://assistive.kinovarobotics.com/product/jaco-robotic-arm.

[276] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *2013 European Control Conference (ECC)*, pages 3071–3076, July 2013.

[277] A. G. Pandala, Y. Ding, and H. Park. qpSWIFT: A real-time sparse quadratic program solver for robotic applications. *IEEE Robotics and Automation Letters*, 4(4):3355–3362, Oct 2019.

[278] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim. MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2245–2252, Oct 2018.

[279] Ruben Grandia, Andrew J Taylor, Aaron D Ames, and Marco Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8352–8358. IEEE, 2021.

[280] Alessandro De Luca, Alin Albu-Schaffer, Sami Haddadin, and Gerd Hirzinger. Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1623–1630. IEEE, 2006.

[281] Gerardo Bledt, Patrick M Wensing, Sam Ingersoll, and Sangbae Kim. Contact model fusion for event-based locomotion in unstructured terrains. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4399–4406. IEEE, 2018.

[282] Shishir Kolathaya, Jacob Reher, Ayonga Hereid, and Aaron D Ames. Input to state stabilizing control lyapunov functions for robust bipedal robotic locomotion. In *2018 Annual American Control Conference (ACC)*, pages 2224–2230. IEEE, 2018.

[283] Jean-Hubert Hours, Melanie N Zeilinger, Ravi Gondhalekar, and Colin N Jones. Constrained spectrum control. *IEEE Transactions on Automatic Control*, 60(7):1969–1974, 2014.

[284] Mayuresh V Kothare, Venkataramanan Balakrishnan, and Manfred Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

[285] P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

[286] Peter J Campo and Manfred Morari. Robust model predictive control. In *1987 American control conference*, pages 1021–1026, 1987.

[287] Zhi Q Zheng and Manfred Morari. Robust stability of constrained model predictive control. In *1993 American Control Conference*, pages 379–383, 1993.

[288] Wilbur Langson, Ioannis Chryssochoos, SV Raković, and David Q Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

[289] David Q Mayne, María M Seron, and SV Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[290] Nahuel A Villa and Pierre-Brice Wieber. Model predictive control of biped walking with bounded uncertainties. In *IEEE-RAS International Conference on Humanoid Robotics*, pages 836–841, 2017.

[291] Ahmad Gazar, Majid Khadiv, Andrea Del Prete, and Ludovic Righetti. Stochastic and robust MPC for bipedal locomotion: A comparative study on robustness and performance. In *IEEE-RAS International Conference on Humanoid Robots*, pages 61–68, 2021.

[292] Justin Carpentier and Nicolas Mansard. Multicontact locomotion of legged robots. *IEEE Transactions on Robotics*, 34(6):1441–1460, 2018.

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

[294] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[295] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

[296] T.P. Chagas, P.-A. Bliman, and K.H. Kienitz. Stabilization of periodic orbits of discrete-time dynamical systems using the prediction-based control: New control law and practical aspects. *Journal of the Franklin Institute*, 355(12):4771–4793, 2018.

[297] Shuuji Kajita and Kazuo Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1405–1406. IEEE Computer Society, 1991.

[298] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–9, Oct 2018.

[299] Ziyi Zhou, Bruce Wingo, Nathan Boyd, Seth Hutchinson, and Ye Zhao. Momentum-aware trajectory optimization and control for agile quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 7(3):7755–7762, 2022.

[300] Ahmad Gazar, Majid Khadiv, Andrea Del Prete, and Ludovic Righetti. Multi-contact stochastic predictive control for legged robots with contact locations uncertainty. *arXiv:2309.04469*, 2023.

[301] Hae-Won Park, Alireza Ramezani, and Jessy W Grizzle. A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking. *IEEE Transactions on Robotics*, 29(2):331–345, 2012.

[302] Maegan Tucker and Aaron D Ames. An input-to-state stability perspective on robust locomotion. *IEEE Control Systems Letters*, 2023.

[303] Shishir Kolathaya, Jacob Reher, Ayonga Hereid, and Aaron D. Ames. Input to state stabilizing control Lyapunov functions for robust bipedal robotic locomotion. In *American Control Conference*, pages 2224–2230, 2018.

[304] Randall T Fawcett, Abhishek Pandala, Aaron D Ames, and Kaveh Akbari Hamed. Robust stabilization of periodic gaits for quadrupedal locomotion via qp-based virtual constraint controllers. *IEEE Control Systems Letters*, 6:1736–1741, 2021.

[305] Kaveh Akbari Hamed and Jessy W Grizzle. Robust event-based stabilization of periodic orbits for hybrid systems: Application to an underactuated 3d bipedal robot. In *2013 American Control Conference*, pages 6206–6212. IEEE, 2013.

[306] Kaveh Akbari Hamed and Robert D Gregg. Decentralized event-based controllers for robust stabilization of hybrid periodic orbits: Application to underactuated 3-d bipedal walking. *IEEE Transactions on Automatic Control*, 64(6):2266–2281, 2018.

[307] Anthony V Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical programming*, 10(1):287–311, 1976.

[308] J. Hwangbo, J. Lee, and M. Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, April 2018.