

Optimization for Control and Planning of Multi-contact Dynamic Motion

by

Michael Posa

B.S. Mechanical Engineering, Stanford University (2007)

M.S. Mechanical Engineering, Stanford University (2008)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

© Massachusetts Institute of Technology 2017. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
May 17, 2017

Certified by.....
Russ Tedrake
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by
Leslie A. Kolodziejcki
Chair, Department Committee on Graduate Students

Optimization for Control and Planning of Multi-contact Dynamic Motion

by

Michael Posa

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The fundamental promise of robotics centers on the ability to productively interact with a complex and changing world. Yet, current robots are largely limited to basic tasks in structured environments and act slowly and cautiously, afraid of incidental contact. In this thesis, we consider a class of control and planning problems for robots dynamically interacting with their environment. We address challenges that arise from non-smooth motions induced by contact, where discontinuities result from impact events and frictional forces. First, we examine the problem of trajectory optimization in contact-rich environments, and present two algorithms for synthesizing motions which make and break contact. The novel contact-implicit trajectory optimization algorithm lifts the problem and reasons over the set of possible contact forces. In doing so, we eliminate the requirement for an *a priori* sequencing of the active contacts, and avoid explicit combinatorial complexity. We also introduce a direct collocation algorithm for optimizing high-accuracy trajectories, given an arbitrary contact schedule. This approach eliminates drift in the numerical integration of contact constraints, even when constraints result in closed kinematic chains and require non-minimal coordinates.

Second, this thesis concerns questions of control synthesis and provable stability verification of a robot making and breaking contact. To verify stability, we introduce an algorithm for discovering polynomial Lyapunov functions, where the system dynamics include impacts and friction. We leverage the measure differential inclusion representation of non-smooth contact mechanics to efficiently optimize over Lyapunov functions in multi-contact settings. Since avoiding hazardous falls is a primary necessity for bipedal walking robots, we use similar tools to characterize the capabilities of multiple simple models used for balancing and push recovery. Using the notions of barrier functions and occupation measures, we explicitly bound the set of disturbances from which a robot can recover by balancing or stepping.

The primary contributions of this thesis are computational in nature, and we heavily leverage modern approaches to both general nonlinear programming and convex optimization. Sums-of-squares, an approach to polynomial optimization utilizing semidefinite programming, plays a central role in our methods for formal stability analysis.

Thesis Supervisor: Russ Tedrake

Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

Thanks to the generosity of many, my time at MIT has been marked by incredible growth and learning. I am extremely grateful to Russ Tedrake, for his support, mentorship, and vision throughout the course of my Ph.D. Russ' blend of guidance and advice, along with the freedom to explore and develop my own ideas, has played an irreplaceable role in my development as a researcher. I am far better for having known Russ and for my time in his lab. His energy, enthusiasm, and creativity built the Robot Locomotion Group into the brilliant, collaborative place it is today—and will forever serve as inspiration as I move forward in my career.

I would also like to thank the readers on my thesis committee, Professors Tomás Lozano-Perez, Sertac Karaman, and Andy Ruina. Over the course of my stay at MIT, all three have provided fantastic guidance and support. I would especially like to thank Andy for constantly questioning me and for never been satisfied with an answer that was not crystal clear. These insightful questions and comments have greatly helped to guide the work presented in this thesis. During my tenure as a member of the MIT DRC team, I was incredibly fortunate to benefit from the leadership and guidance of Professor Seth Teller. The joy with which he imagined, and then created, the impossible will always inspire me to dream big.

I have been privileged to collaborate with Mark Tobenkin, Scott Kuindersma, and Twan Koolen. Mark introduced me to the depths of sums-of-squares programming, and working with him gave me a far greater appreciation for the mathematical beauty of robotics research. Mark's aid in writing my first SOS programs and published proofs was critical to the results presented in Chapter 6. From sitting next to Scott during the DARPA Robotics Challenge, to later collaborating on a paper, I have becoming a better experimental roboticist. Twan and I first started working together on computational approaches to capturability. While our paths ultimately led to separate ideas, this collaboration was critical in sharpening my thoughts regarding simple models of walking. It has also been my great pleasure to work with three MIT undergraduate researchers. Cecilia Cantu, Sarah Hensley, and Julia Wu have all been fantastic to work with, and have great futures in research, should they so choose.

For the past six years, the Robot Locomotion Group has been my second home. I

joined alongside (and sat next to!) Frank Permenter and Anirudha Majumdar, who, through their brilliance and camaraderie made my stay at MIT what it was. Beyond our technical discussions, I we filled countless hours discussing the philosophical nature of all topics, from politics to legged locomotion and machine learning. Both professionally and personally, working and sharing laughs with Mark and Scott were some of the highlights of my time at MIT. I have also been incredibly privileged to work alongside, and share conversation with John Roberts, Hongkai Dai, Andy Barry, Joseph Moore, Andres Valenzuela, Robin Deits, Lucas Manuelli, Shen Shen, Pete Florence, Greg Izatt, Pat Marion, Tao Pang, Benoit Landry, Vincent Tjeng, Geronimo Mirano, John Carter, and Steve Proulx. I have been fortunate to learn from the brilliant postdocs and research scientists who have passed through the lab: soaking up knowledge from Ian Manchester and Zico Kolter in my early days, and my many chats with Scott, Ram Vasudevan, Amir Ali Ahmadi, Maurice Fallon, Aykut Satici, and Kanako Miura. The strength of the Robot Locomotion Group lies in the collective greatness of its members.

Many people throughout the community have provided invaluable career advice to me, particularly throughout the end of my Ph.D. and my job search. In particular, I would like to acknowledge Russ, Scott, Ani, Tomás, Ram, Silvio Micali, Jessy Grizzle, and Sam Burden. Furthermore, without the support of Kathy Bates, Mieke Moran, and Bryt Bradley, nothing in this thesis would have been possible. I fondly recall Kathy's bright and cheerful postings, Mieke's fabulous parties, and the way Bryt deftly managed the chaos that was the DRC.

It was at Stanford that I first discovered my love of engineering and robotics, with special thanks to my mentors and teachers Chris Gerdes and Paul Mitiguy. They also gave me the opportunity to teach, without which I would have been apprehensive about turning to a path in academia. I owe a debt of gratitude to friends and classmates in mechanical engineering who made my classes fun and collaborative—especially my mechatronics groupmmates Paul Yu, RK MacLean, and Candice Hsu. That love turned into a career at Vecna, where I am particularly grateful to Andreas Hofmann, without whom my path would never have led me to MIT.

I am incredibly grateful to the friends that I have made at and through MIT, many of whom I met on that eventful (for the wrong reasons) orientation hiking trip. From the

time spent and apartments shared with Leo Banchik, Rob Hemsley, Jacob Izraelevitz, Leah Mendelson, Alice Nasto, Tom O'Grady, Charles Richter, Faina Rozental, and Shu Dar Yao, we have formed lifelong bonds.

I owe everything to the unwavering love and support of my family. I would like to thank my parents for all that they have done for me over the years and my little sisters for putting up with me all this time. Most importantly, a thank you to Molly for making the last four years my Ph.D. the best of my life.

Contents

1	Introduction	15
1.1	Contributions	16
1.2	Outline	17
2	Related Work	19
2.1	Motion Planning with Contact	19
2.1.1	Hybrid optimization with mode schedules	19
2.1.2	Related work in mode invariant optimization	21
2.2	Reachability and Stability Verification	22
2.3	Balancing and Step Recovery	23
3	Background	25
3.1	Rigid-Body Dynamics with Contact	25
3.2	Collision Modeling	27
3.2.1	Inelastic Collisions with Friction	28
3.2.2	Simultaneous Impacts	29
3.2.3	Measure Differential Inclusions	31
3.2.4	Contact Dynamics as a Complementarity Problem	32
3.3	Trajectory Optimization	34
3.3.1	Direct Collocation	34
3.4	Stability and Reachability	36
3.4.1	Lyapunov and Barrier Functions	36
3.4.2	Sums-of-squares and the S-procedure	38

3.4.3	Bilinear Alternations	39
4	Contact-implicit Trajectory Optimization	41
4.1	Contact-implicit Approach	43
4.1.1	Optimization Constraints	43
4.1.2	Solving the Optimization Problem	45
4.1.3	Extension to Three Dimensions	46
4.1.4	Time Discretization	47
4.2	Example Applications	48
4.2.1	Finger Contact	48
4.2.2	Simple Manipulation	49
4.2.3	Spring Flamingo Walking Gait	51
4.2.4	FastRunner Gait	54
4.3	Discussion	57
5	Contact-constrained Optimization and Tracking	59
5.1	Constrained Direct Collocation	61
5.1.1	Friction Limits	65
5.1.2	Hybrid Collocation	65
5.2	Equality-Constrained LQR	66
5.2.1	Example: Kinematic Constraint	67
5.3	QP Feedback Controller	68
5.4	Example Applications	69
5.4.1	Underactuated planar biped	69
5.4.2	Multi-contact climbing	71
5.4.3	3D biped	72
5.5	Discussion	73
6	Stability Verification and Control Design	75
6.1	Definitions	75
6.2	Conditions for Stability	76

6.2.1	Lyapunov Conditions for MDIs	77
6.2.2	Conditions For Complementarity Systems	80
6.2.3	Separability of Contacts	82
6.2.4	Extension to Three Dimensions	84
6.2.5	Semialgebraic Representation	84
6.3	Verification Algorithms	85
6.3.1	Global Verification	86
6.3.2	Regional Verification	86
6.3.3	Verifying Safety	89
6.3.4	Control Design	89
6.3.5	Complexity	92
6.4	Example Applications	92
6.4.1	Bean Bag Toss	92
6.4.2	Rimless Wheel	93
6.4.3	Perching Glider	96
6.4.4	Balancing Robot Control Design	100
6.5	Discussion	102
7	Balancing and Push Recovery	105
7.1	Preliminaries	106
7.1.1	Model Classes	106
7.1.2	Capturability	107
7.2	Reachability via occupation measures	108
7.3	Approach	109
7.3.1	Bilinear inner approximation	110
7.3.1.1	Strict Feasibility for Alternations	111
7.3.1.2	Balancing	112
7.3.1.3	N -step	113
7.3.2	Outer approximations	114
7.3.2.1	Balancing	114

7.3.2.2	<i>N</i> -Step	115
7.4	Example Applications	116
7.4.1	LIPM	117
7.4.2	Variable Height	117
7.4.3	Incorporating Impact Dynamics	119
7.4.4	Variable Height and Rotational Inertia	121
7.5	Discussion	124
7.A	Appendix	124
8	Conclusion	127
8.1	Challenges and Open Questions	128
8.1.1	Computational Challenges	128
8.1.2	Reliance Upon Models	129
8.1.3	Real-time Control	130

List of Figures

2-1	Hybrid trajectory optimization cartoon.	20
3-1	Example impact resolutions via Routh's method.	30
3-2	The direct collocation algorithm spline and defect.	35
4-1	The bipedal FastRunner robot.	42
4-2	Two link finger.	49
4-3	Simple planar gripper.	50
4-4	Spring Flamingo walking.	51
4-5	Optimized mode sequence for the Spring Flamingo.	52
4-6	Periodic FastRunner gait.	54
4-7	Optimized mode sequence for FastRunner.	55
4-8	FastRunner ascending and descending steps.	56
4-9	FastRunner gait with foot placement constraints.	56
5-1	An Atlas robot climbing a step with aid of a hand.	60
5-2	Block diagram of trajectory optimization and tracking components.	60
5-3	DIRCON constraint projection	62
5-4	Tracking error for underactuated walking.	70
5-5	A sequence of states as Atlas uses its arm to help climb up onto a step.	71
5-6	A sequence of states from a rapid halting maneuver.	72
6-1	The rimless wheel.	94
6-2	Verified regions for a slice of state space where all velocities are zero.	95
6-3	A perching aircraft model.	97

6-4	Verified regions for two different slices of the glider state space.	98
6-5	A simple balancing robot.	99
6-6	Verified regions for two slices of state space.	101
7-1	A cartoon of the viable-capture basins.	109
7-2	The planar LIPM with a foot.	116
7-3	The LIPM viable-capture basins.	118
7-4	The variable height capture basins.	119
7-5	The LIPM with impact dynamics.	120
7-6	A slice of the viable-capture basins for a variable height model with impact dynamics.	120
7-7	Variable height and rotational inertia model.	121
7-8	The viable-ions for a model including variable height and inertia.	122
7-9	Viable-capture basin comparisons across models.	123

Chapter 1

Introduction

The field of robotics has grown tremendously in recent years, and robots are poised to have a transformative impact on everyday life. We hope to soon see an abundance of autonomous or semi-autonomous devices in the home, e.g. providing assistance to older adults and people with disabilities, and of robots in the workplace performing advanced manufacturing and logistical tasks. Whether a robot is assisting a stroke victim to move about his or her home, or packing shipping containers in a warehouse, the fundamental promise of robotics centers on the ability to productively interact with a complex and changing environment in a safe and controlled fashion. Yet, for the most part, current robots are limited to basic tasks in highly structured environments; they typically operate slowly and cautiously, afraid that any incidental contact with the outside world, however minor, will result in catastrophic failure. There is little room for dynamic adaptation since new interactions, like initiating contact to gain sensory information or bracing against a nearby surface, must be planned in advance.

Safely and reliably achieving dynamic interaction, encompassing both locomotion and manipulation, in multi-contact settings continues to pose significant challenges to the field. These challenges are characterized by the complex structure of the dynamical descriptions of interaction. Many computationally efficient techniques for modeling contact use impulsive impact models along with the dry Coulomb friction assumption. However, these models introduce discontinuities that make standard tools for planning and control especially poorly suited in scenarios with complex or uncertain contacts between the robot and the environment. Impacts and friction additionally lead to extremely high sensitivity in the governing

equations of motion, which, in some scenarios, is expressed by rigid-body models as the existence of non-unique solutions. When applied to smooth systems, both classical and optimization-based existing methods heavily leverage the uniqueness and differentiability of trajectories of such systems; therefore, new paradigms and new algorithms are required to address contact-rich tasks. To address these issues, this thesis will introduce computationally tractable algorithmic techniques for planning, control, and formal analysis of such systems.

1.1 Contributions

In this thesis, we consider a series of control and planning problems for a robot dynamically interacting with its environment. While the primary examples studied relate to questions of robot locomotion, the methods introduced are general to many multi-contact applications, like dexterous manipulation. First, we investigate the optimal control problem for such systems, with the goal of producing locally optimal trajectories in contact-rich environments where the robot must initiate and break contact to achieve some specified goal. This problem is especially difficult when the sequencing of such contacts is unknown *a priori*, as explicitly reasoning over the set of such possible sequences is computationally intractable. This thesis presents an algorithm for contact-implicit optimization, published in [98, 100], where a single nonlinear program determines both the smooth trajectory and implicitly defines the mode sequence. With this method, we are now able to automatically generate new motions for systems with dozens of potential contacts.

The contact-implicit optimization algorithm is based in a first-order time-stepping method, and so would require a large number of discretization points to produce high-accuracy trajectories. To compensate for this shortcoming, we will also introduce an extension of the direct collocation algorithm of Hargraves and Paris [45], which we published in [101], to properly handle any constraint introduced by sustained contact. Since this algorithm does require a mode schedule, it can be used to improve upon trajectories first generated by the contact-implicit approach. These algorithms are evaluated on high-dimensional walking and running robots in simulation.

This thesis also explores algorithms for formal, numerical analysis of systems contacting

the environment. Methods based on sums-of-squares (SOS) for numerical computation of certificates have proven to be powerful tools for analyzing the stability of continuous nonlinear systems, and can additionally be used to automatically synthesize stabilizing feedback controllers. However, to address systems in contact, these methods also have traditionally used hybrid models and hybrid certificates, which must explicitly enumerate and verify every hybrid mode and transition. In this thesis, we will avoid this exponential enumeration, and instead generate Lyapunov certificates for stability, positive invariance, and safety over admissible (non-penetrating) states and contact forces [99, 102]. The approach is demonstrated on multiple robotics examples, including simple models of a walking robot, a perching aircraft, and control design of a balancing robot.

Tools for formal analysis, like those mentioned above, typically only scale to systems of moderate dimensionality, while humanoid robots have dozens of joints. At the same time, low-dimensional models like the linear inverted pendulum model (LIPM) have proven incredibly powerful in the realm of robotic walking and push recovery (e.g. [54, 58]). Despite their widespread use, it is not fully understood what limitations these simple models impose when used to design balancing and recovery controllers. Here, we utilize convex optimization and provide concrete answers to this question by formally analyzing and bounding the capabilities of policies that exploit angular momentum and the impact dynamics of stepping to prevent falling [103]. Furthermore, this approach will generate explicit push recovery policies and compare their performance against the provable upper bound and against policies based on the LIPM.

1.2 Outline

In Chapter 2, we discuss prior and related work on control and planning with contact, placing the work of this thesis into context. Chapter 3 reviews necessary background material: a description of rigid-body dynamics with impacts and friction, relevant classical trajectory optimization techniques, and an introduction to sums-of-squares techniques for Lyapunov and reachability analysis. In Chapter 4, we introduce the contact-implicit trajectory optimization for simultaneously synthesizing trajectories and contact sequences. Chapter 5 extends

the classical direct collocation trajectory optimization algorithm and the linear quadratic regular (LQR) controller to address systems with arbitrary contact constraints. Chapter 6 presents a scalable algorithm, utilizing sums-of-squares optimization, to design provably stable local control policies, through contact, and associated Lyapunov functions. Chapter 7 presents additional SOS-based algorithms for the formal analysis of a series of bipedal models, explicitly bounding the balancing and push recovery capability gained by deviating from the classical linear inverted pendulum approach. Chapter 8 concludes the thesis with a discussion of the contributions and limitations of the presented methods, and the potential for further extensions and new research directions.

Chapter 2

Related Work

2.1 Motion Planning with Contact

Contact between the robot and environment has long played a pivotal role in questions of motion planning, with a great deal of work devoted to attempting to *avoid* contact via collision-free planning and optimization (e.g [70, 111, 116] and many others). However, the discontinuities due to impact and friction greatly complicate the task of intentionally planning for contact. Optimization-based methods are heavily reliant upon smooth, local approximations of the optimization landscape—using gradients of the objective and constraints with respect to the control input and state parameters. As such, these local approximations are only valid when constrained to a particular contact sequence, and classical trajectory optimization methods (see [10] for an overview) typically cannot utilize new contacts. Stochastic approaches, such as the genetic algorithms in [72, 38], are capable of bypassing these discontinuities via chance, but these methods still contain no guiding information about unexplored potential contact.

2.1.1 Hybrid optimization with mode schedules

The vast majority of existing techniques for optimal control subject to contact make use of hybrid models, where the optimization is performed with a hybrid mode schedule, defined *a priori* or selected via an outer loop optimization [117, 123, 142, 21, 85, 112, 145, 48]. In

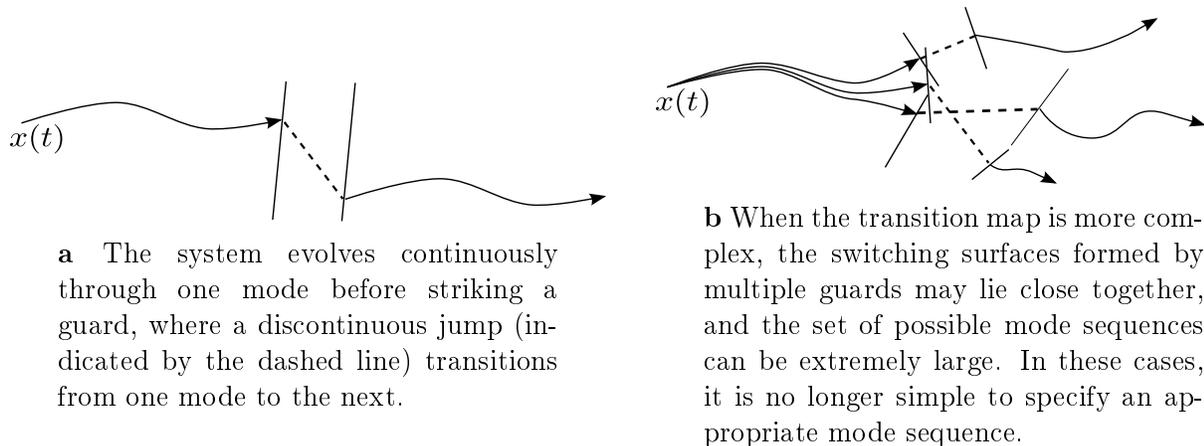


Figure 2-1: Hybrid trajectories can be found by optimizing over the continuous dynamics defined by a specified mode sequence.

this setting, a hybrid mode defines a contact mode (the set of active contacts and whether the contact is sticking or sliding). The system dynamics, restricted to the set of states corresponding to a particular mode, are differentiable. Discontinuities, in both trajectories and vector fields, are confined to the moments in which guard conditions are met (e.g., the robot’s foot hits the ground), where a discontinuous jump in state-space occurs, as cartooned in Figure 2-1a. For a fixed mode schedule, direct methods for hybrid trajectory optimization proceed by optimizing each segment independently, with additional constraints ensuring that the segments connect to each other through the hybrid events.

However, in contact-rich settings, the geometric constraints imposed by the hybrid system become more daunting. Hybrid events occur every time contact is initiated or broken, during slip-stick transitions, whenever one of the robot joints strikes or leaves a joint-limit. The number of possible hybrid modes of the system therefore grows exponentially with the number of possible constraints. The geometry of the hybrid guards becomes very complex, as cartooned in Figure 2-1b. In these models, small changes to the control input can result in a very different schedule of hybrid modes. Restricting the trajectory optimization search to the initial mode schedule can result in a very limited search and in failure to find high-quality, feasible trajectories.

Despite the obvious limitation of requiring a mode schedule, it has proven surprisingly difficult to remove this assumption in the direct methods. Some variations from the original

sequence are possible if the formulation allows the time duration of individual modes to vanish, as in the work of Srinivasan and Ruina [123]. For problems with fewer possible modes, Wampler and Popović [142] used outer optimization loops to determine the hybrid mode schedule. In some cases, the combinatorial problem of solving for a mode schedule has been addressed by combinatorial planners. For example, a variant of the Rapidly-Exploring Random Tree (RRT) algorithm was used by Shkolnik et al. [119] to produce quadrupedal trajectories over terrain. Methods for optimal control which approximate the global optimum, such as brute force methods based on dynamic programming, have also been applied [16], but are computationally limited to low dimensional problems.

2.1.2 Related work in mode invariant optimization

A number of researchers are currently pursuing mode invariant trajectory optimization algorithms, as contrasted with the more traditional hybrid systems-based approach. Berard et al. [8] used the linear complementarity problem (LCP) formulation of contact to design trajectories of single body on a vibrating plate. Here, they optimize over a small set of parameters describing the oscillating behavior of the plate; this approach can best be described as a shooting method. For systems with only a few potential contacts, Yunt and Glocker [146] used a nonlinear programming penalty method to jointly optimize over trajectories and contacts. Tassa and Todorov [131] have also explored the use of stochastic complementarity for optimal control using differential dynamic programming (DDP). Since this, and similar approaches make local, gradient-based improvements to a nominal trajectory, they will not naturally discover new contact sequences not already present—therefore relying on the natural dynamics of the system to make contact. Mordatch et al. [87] demonstrated contact invariant optimization of complex, lifelike behaviors of humanoid figures using relaxations of the contact complementarity constraints governing physically realistic forces. To pose a simpler problem, Mordatch et al. additionally assume that the limbs of the figures are massless. The work of [31] optimized a human running gait by smoothing the contact dynamics to make use of a custom inverse dynamics formulation. Mordatch includes the feasibility of the contact forces as a penalty term in the optimization cost function, rather than as a direct constraint. Broadly speaking, the work in [31, 87, 86] make varying relaxations of

physical or dynamic constraints to pose more tractable (and unconstrained) optimization problems. In particular, these works relax constraints to allow contact forces to act at a distance; while this may produce dynamically infeasible trajectories, it has a smoothing effect and provides gradient information that would otherwise be unavailable in typical shooting methods. The weighting parameters that define these relaxations present a trade-off between physical realism and the stiffness (and, correspondingly, numerical performance) of the resulting optimization problems. Additionally, all three works make heavy use of inverse dynamics. For highly underactuated systems, arbitrary state or end effector trajectories are not dynamically feasible, posing difficulties for methods reliant upon inverse dynamics.

2.2 Reachability and Stability Verification

For smooth nonlinear dynamical systems, techniques for reachability analysis stability verification have been shown to play a pivotal role in incremental motion planning and control design strategies and in direct optimization over feedback laws. To accommodate potential contact, the standard approach here has again been to use the formalisms of hybrid systems. In this space, a number of researchers have built computational approaches based on the Hamilton-Jacobi-Bellman equation, as in the level set methods of [27], with hybrid formulations and variations found in [84, 77, 40]. Recent research has utilized sums-of-squares (SOS) programming, described in detail in 3.4, to compute Lyapunov and barrier functions, along with inner approximations to the associated regions of attraction and feedback control policies (e.g. [105, 104, 49, 134, 135, 78]). Hybrid barrier functions, utilizing SOS, were introduced in [93], though examples were limited to only two hybrid modes. Manchester additionally computed regions of stability of transverse dynamics about hybrid limit cycles with a fixed mode sequence [81]. Related approaches, utilizing the notions of occupation measures and moment relaxations, have been introduced to compute outer approximations to the reachable set [46], with extensions to control synthesis [60, 80] and hybrid systems [118]. Any approach to stability and reachability analysis must consider all potential contact events and modes in a given region of state space. Therefore, in this setting, there can be no notion of restricting the analysis to a desirable contact sequence, and hybrid formula-

tions will inherently be combinatorial in nature. However, for computational reasons, the hybrid approaches discussed above only address systems with an extremely small number of potential modes. This limitation leaves systems with even two potential contacts, and associated sticking and sliding modes, outside the reach of these methods. Also of interest, the recent work of Várkonyi and Or [140] has specifically explored stability criteria for a rigid body with two contacts, although this analytically rigorous approach is limited to a local approximation called zero order dynamics.

2.3 Balancing and Step Recovery

Push recovery is a fundamental skill for any bipedal robot—necessary to prevent damaging falls in the presence of unanticipated disturbances. Broadly speaking, typical approaches for push recovery blend together techniques for active balancing and for stepping. To be successful, control policies must be able to rapidly decide when stepping is necessary, and to make such decisions while considering physical limitations on potential motion. The study of push recovery strategies has heavily focused upon simple models, many of which admit closed-form solutions to the critical questions of control synthesis and reachability analysis. In particular, the Linear Inverted Pendulum Model (LIPM) [53, 54] has been broadly influential on the study of walking robots. The LIPM is appealing because it captures the critical centroidal dynamics of the robot while remaining relatively simple to analyze and control. Stephens [126] characterized and studied the LIPM in the context of push recovery. Utilizing the notions of the capture point and capturability, recent work has fully determined the set of states in which can be stabilized within N steps [108, 58]. This notion of capturability is critical to safe execution of bipedal motion, as an accurate understanding of the stabilizable states enables effective balancing and step recovery.

The LIPM makes a number of key assumptions to simplify the dynamical equations of motion, resulting in a two dimensional linear model that is easy to control and analyze. Recent robotics research has examined some of these assumptions, both in terms of balancing and step recovery. Goswami and Kallem [42] studied the role of angular momentum on balance, and Koolen et al. [59] characterized the role of height variation. Capturability

analysis was also used to analyze reaction wheel models [58] and inverted pendulum models [147]. Additionally, Mummolo et al. [91] used discretization and nonlinear optimization to perform exhaustively calculation on low-dimensional jointed models. Biomechanical studies have additionally investigated balancing from the perspective of optimal control [64] and the role of arm swing as a mechanism for modifying angular momentum [20]. In this thesis, we present a more unified approach to this analysis, based on recent developments from convex optimization.

Chapter 3

Background

This chapter serves as a survey of necessary material for this thesis and introduces the principal notation used throughout. Section 3.1 discusses the continuous dynamics of rigid-bodies, both in free space and when constrained by active contacts. Section 3.2 presents models used to represent the discontinuous impact dynamics. Section 3.3 presents an introduction to trajectory optimization algorithms, most relevant to Chapters 4 and 5. Lastly, Section 3.4 covers background material on sums-of-squares optimization and applications to stability and reachability analysis, relevant to Chapters 6 and 7.

3.1 Rigid-Body Dynamics with Contact

Many robotic systems are appropriately modeled as a set of rigid links connected through some combination of joints [120, 32]. The continuous dynamics of these rigid-body systems subject to frictional contact forces, derived via the constrained Lagrangian, can be modeled by the manipulator equations

$$H(q(t))\dot{v}(t) + C(q(t), v(t)) = Bu(t) + \tau(t),$$

where q and v are the generalized positions and velocities, respectively. For simplicity, we will assume that $q, v \in \mathbb{R}^n$ and $v(t) = \dot{q}(t)$, though this may not always be the case (e.g. with the quaternion formulation of three dimensional rotations). τ represents the net external forces

acting on the system. Here, $H(q)$ is the inertia matrix, $C(q, v)$ is the combined Coriolis and gravitational terms, and B maps the control inputs u into joint coordinates. Where appropriate, we will also write the entire state vector $x = \begin{bmatrix} q \\ v \end{bmatrix}$.

We will focus on the dynamics of rigid bodies interacting at a finite number of contact points, though these points may not necessarily be fixed on their respective bodies. When the external forces are generated at a set of m such point contacts, we will write them in the local contact frame and split the forces into the components normal to the contacts, $\lambda_N \in \mathbb{R}^m$, and the frictional forces tangential to the contact surface, λ_T . Note that we assume contacts do not “pull”, and so $\lambda_N \geq 0$. This and all other vector inequalities will be taken elementwise. For planar systems, $\lambda_T \in \mathbb{R}^m$, while $\lambda_T \in \mathbb{R}^{2m}$ in the full three dimensional case. We will also write λ to be the stacked vector $\begin{bmatrix} \lambda_N \\ \lambda_T \end{bmatrix}$. The matrices $J_N(q) \in \mathbb{R}^{m \times n}$ and $J_T(q) \in \mathbb{R}^{m \times n}$ (in the planar case) project the normal and frictional contact forces into joint coordinates, with J similarly representing the stacked matrix $\begin{bmatrix} J_N \\ J_T \end{bmatrix}$. We will also refer to $J_{N,i}(q)$ and $J_{T,i}(q)$ as the i th row of J_N and J_T , associated with the particular contact forces $\lambda_{N,i}$ and $\lambda_{T,i}$. With this notation, we write the manipulator equations

$$H(q)\dot{v} + C(q, v) = Bu + J_N(q)^T \lambda_N + J_T(q)^T \lambda_T, \quad (3.1)$$

where the dependence of q, v, λ_N , and λ_T on time has been suppressed for clarity. When ϕ represents the distances between pairs of bodies, we can take J_N to be the Jacobian $\frac{\partial \phi}{\partial q}$ and J_T defines the frictional plane. Hence, the velocity of the i th contact point has components $J_{N,i}v(t)$ and $J_{T,i}v(t)$ normal to and tangential to the contact surface. We use a simple Coulomb friction model to represent our contact forces:

$$\begin{aligned} J_{T,i}v &= 0 \Rightarrow \|\lambda_{T,i}\| \leq \mu \lambda_{N,i}, \\ J_{T,i}v &\neq 0 \Rightarrow \lambda_{T,i} = -\frac{J_{T,i}v}{\|J_{T,i}v\|} \mu \lambda_{N,i}. \end{aligned}$$

When the tangential velocity vanishes, $\lambda_{T,i}$ can take on any value within the friction cone. If the contact point is sliding, then $\lambda_{T,i}$ directly opposes the direction of slip. Note that

Coulomb friction results in a differential equation (3.1) with a discontinuous right-hand side. To address this issue, we make use of the notion of a *differential inclusion* of the form $\dot{x} \in \mathcal{F}(t, x)$, where \mathcal{F} is a set valued function. For instance, we can rewrite the formula for Coulomb friction above as

$$\lambda_{T,i} \in -\mu \lambda_{N,i} \operatorname{sgn} J_{T,i} v,$$

where $\operatorname{sgn} x$ is a set-valued signum function, with $\operatorname{sgn} 0 = [-1, 1]$. See [121, 5] for an overview of differential inclusions.

Contact exists when the gap function vanishes: $\phi(q(t)) = 0$. Therefore, for contact to be sustained over some interval, we must also have $\dot{\phi}(q(t)) = \ddot{\phi}(q(t)) = 0$. Similarly, if the contacts are sticking, the tangential velocity and accelerations must also vanish. As a result, if the contact mode is known, the set of feasible forces can be described by a set of convex equations and inequalities. For example, in the planar, sticking case, the forces must satisfy

$$\begin{aligned} J_N v + J_N \dot{v} &= 0 \\ J_T v + J_T \dot{v} &= 0 \\ \lambda_N &\geq 0 \\ \mu \lambda_N - |\lambda_T| &\geq 0, \end{aligned}$$

where \dot{v} and λ are related by (3.1). If no such forces exist, then either contact separation or a stick-slip transition must occur to produce a solution.

3.2 Collision Modeling

When two relatively rigid bodies collide, such as when a stiff robot impacts its environment, these collisions are events characterized by relatively large forces applied over short durations. Broadly speaking, this leads to two approaches to modeling these events. In one, material properties are used to attempt to capture the force profile and material deformation throughout an impact event. Naturally, for rigid robots, this approach results in stiff differential equations which can be problematic in both simulation and control algorithms.

In the second approach, we consider the limiting case where the event is instantaneous and the forces are approximated by Dirac delta functions. As a result, velocity profiles are discontinuous in time and differential equations are only used to describe motion between impact events. Following an impact, contact between multiple bodies may be sustained over some duration, resulting in continuous contact forces. While instantaneous collisions are only an approximation of the complex interactions between bodies, if the impulsive forces can be determined, this approach offers significant computational advantages. In this thesis, we will treat impacts as instantaneous events with different pre- and post-impact velocities, $v_-(t)$ and $v_+(t)$, using standard notation for the left and right limits. An overview of the modeling techniques used in this thesis is provided below, but a more detailed discussion of the discontinuous dynamics can be found in [127], [71], or [13].

3.2.1 Inelastic Collisions with Friction

Using the above notation, impacts occur at some time t^* when there is contact, $\phi_i(q(t^*)) = 0$, and when consistency requires an instantaneous change in velocity to prevent penetration. Note that this consistency requirement often results from $\left. \frac{d\phi_i(q(t))}{dt} \right|_{t^*} < 0$, but impacts may also necessarily occur when the first derivative is exactly zero, as in the case of Painlevé's Paradox [127]. As with the continuous case, we let Λ_N, Λ_T be the normal and frictional components of the net impulse. Derived from the manipulator equations, the pre- and post-impact velocities for a collision at the i th contact point are related by $v_+(t) = v_-(t) + H^{-1}(J_{N,i}^T \Lambda_{N,i} + J_{T,i}^T \Lambda_{T,i})$. In the special case of a single, frictionless inelastic collision, we observe that the inelastic condition is $J_{N,i} v_+(t) = 0$ and so we can explicitly solve for the normal impulse and post-impact state:

$$\Lambda_{N,i} = -(J_{N,i} H^{-1} J_{N,i}^T)^{-1} J_{N,i} v_-(t), \quad (3.2)$$

$$v_+(t) = (I - H^{-1} J_{N,i}^T (J_{N,i} H^{-1} J_{N,i}^T)^{-1} J_{N,i}) v_-(t). \quad (3.3)$$

However, when considering Coulomb friction, we have no explicit formula. Multiple approaches have been proposed for modeling frictional impacts, see [18] for an overview and more detailed discussion of some common models. In particular, these approaches may

differ in how they resolve stick-slip transitions and slip reversal (where the direction of slip is changed by an impact). Here, we first adopt an impact law first proposed by Routh [114] and described in detail in [143, 11]. Originally a graphical approach, this method constructs a path in impulse space. To briefly summarize Routh’s technique for computing the net impulses and the post-impact state:

1. Monotonically increase the normal impulse $\Lambda_{N,i}$ with some slope $\Lambda'_{N,i}$.
2. Increment the tangential impulse $\Lambda_{T,i}$ with slope $\Lambda'_{T,i}$, according to the friction law:

$$J_{T,i}\bar{v} = 0 \Rightarrow \|\Lambda'_{T,i}\| \leq \mu\Lambda'_{N,i},$$

$$J_{T,i}\bar{v} \neq 0 \Rightarrow \Lambda'_{T,i} = -\frac{J_{T,i}\bar{v}}{\|J_{T,i}\bar{v}\|}\mu\Lambda'_{N,i},$$

where $\bar{v} = v_-(t) + H^{-1}(J_{N,i}^T\Lambda_{N,i} + J_{T,i}^T\Lambda_{T,i})$ is the velocity after impulses $\Lambda_{N,i}$ and $\Lambda_{T,i}$.

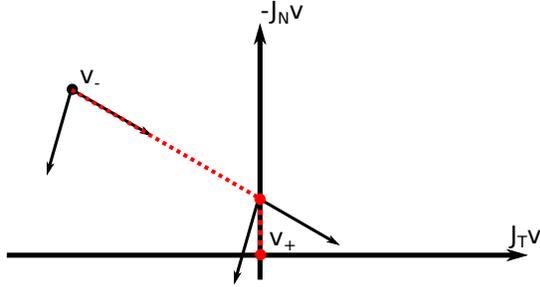
3. Terminate when the normal contact velocity vanishes¹, $J_{N,i}\bar{v} = 0$, and take $v_+(t) = \bar{v}$.

This method amounts to following continuous path in the impulse space where the slopes of the impulses are $\Lambda'_{N,i}$ and $\Lambda'_{T,i}$. This process could equivalently be modeled as a using a differential inclusion where $\Lambda_{N,i}$ takes the role of “time” in the inclusion. Diagrams depicting the resolution of two potential planar impacts are shown in Figure 3-1. Along each linear section, these slopes must satisfy the Coulomb friction constraints. Solutions may transition from sliding to sticking and vice versa and the direction of slip may even reverse as a result of each impact. While the generated path is piecewise linear in the planar case, this is not true in three dimensions.

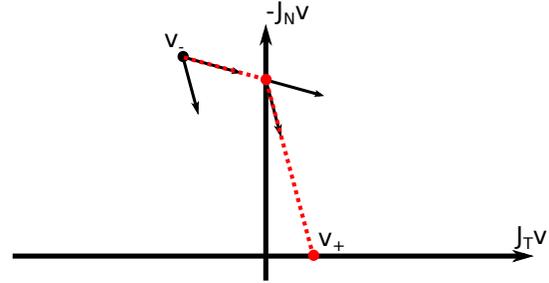
3.2.2 Simultaneous Impacts

The question of simultaneous impacts has been well studied in both simulation and analysis (e.g. [18, 113]), but remains a difficult problem. General, impulsive models to simultaneous impacts do not necessarily produce unique solutions, where the primary difficulty lies in determining an order in which to resolve the impacts (think, for instance, of a pool break

¹To permit resolutions to Painlevé’s Paradox, terminate only when consistency no longer requires an instantaneous change in velocity.



a The contact velocity, in the contact frame, is plotted throughout an impact resolution by Routh’s method. At the initial state, v_- , the extreme rays of the friction cone are shown as solid arrows. Since the contact is sliding, $J_T v < 0$, the impulse increments along the appropriate ray. When $v(s)$, shown in the dotted line, intersects the $J_T v = 0$ axis, the contact transitions to sticking and the impact terminates when $J_N v(s) = 0$.



b In this case, the impulse required to maintain sticking is not within the friction cone. Therefore, when $v(s)$ crosses the vertical axis, the contact does not stick, but the direction of slip reverses, and the impulse increments along the other extreme ray until termination.

Figure 3-1: Example impact resolutions via Routh’s method.

in a game of billiards). In Routh’s method, multiple impacts might be resolved sequentially or via a blended approach. The simulation and computer graphics communities typically resolve the question of uniqueness by modifying the model to select a single outcome, for instance by optimizing some user-specified desirable criteria, as in Kaufman et al. [55]. In the context of control and planning, we will typically adopt a similar approach, with the acknowledgement that this remains a potential shortcoming to be addressed in the future.

In this thesis, we will also formally investigate the question of stability of such systems. Here, we will take a permissive view of simultaneous impacts by explicitly *not* attempting to resolve the issue of uniqueness. If the set $\mathcal{J} \subset \{1, \dots, m\}$ is the set of active impacts ($\phi_i = 0$, $J_{N,i} \bar{v} \leq 0$), then we proceed with Routh’s method where the incremental impulses are any arbitrary convex combination of the incremental impulses for each impact. That is,

where e_i are the standard basis vectors,

$$\begin{aligned}\Lambda'_N &= \sum_{i \in \mathcal{J}} e_i \Lambda'_{N,i} \\ \Lambda'_T &= \sum_{i \in \mathcal{J}} e_i \Lambda'_{T,i} \\ 1 &= \sum_{i \in \mathcal{J}} \Lambda'_{N,i}, \\ \Lambda'_{N,i} &= 0 \text{ for } i \notin \mathcal{J}.\end{aligned}$$

With this approach, for a system to be considered stable, it must be stable even if the collisions are resolved simultaneously or in an arbitrary order.

3.2.3 Measure Differential Inclusions

To capture both impulsive and continuous contact forces, the dynamics community has developed models built using *measure differential inclusions* (MDIs), first introduced by Moreau [89], with additional details and precise definitions in [127, 71, 13]. By taking the time-derivative of state from a set-valued (and not necessarily bounded) function, MDIs address both the discontinuities and non-smoothness of the system evolution that arise from impacts and standard friction force laws. We provide a high-level overview of MDIs here, focused on autonomous Lagrangian mechanical systems.

A solution of a measure differential inclusion will be taken to be a pair of functions, $q(t)$ and $v(t)$, such that $q(t)$ is absolutely continuous and $v(t)$ is of locally bounded variation, allowing for countably many discontinuities. The left and right limits of $v(t)$, denoted $v_-(t)$ and $v_+(t)$, are guaranteed to exist and we require that solutions satisfy:

$$q(t) - q(t_0) = \int_{t_0}^t v(\tau) d\tau, \quad (3.4)$$

$$v_+(t) - v_-(t_0) = \int_{t_0}^t \dot{v}(\tau) d\tau + \int_{t_0}^t v_+(\tau) - v_-(\tau) d\eta(\tau). \quad (3.5)$$

Here $\dot{v}(t)$ is an integrable function and η is a sum of Dirac measures centered at times $\{t_k\}_{k=1}^\infty$, which model the continuous evolution and jumps in the velocity respectively. By

assumption, $v(t)$ has no singular part (see [71], Ch. 3). One advantage of this framework is that it naturally permits Zeno executions: where an infinite number of impact events occur in a finite period of time, as might result from a bouncing ball or rocking block. Within the language of hybrid systems, addressing Zeno Phenomena requires special care and attention [50, 92].

Rules for specifying legal values of $\dot{v}(t)$, the locations $\{t_k\}_{k=1}^{\infty}$, and the jumps $v_+(t_k) - v_-(t_k)$ are described in Sec. 3.2.1. Furthermore, as with a standard differential inclusion, we require $\dot{v}(t) \in \mathfrak{F}(q(t), v(t))$ for almost all t , where $\mathfrak{F}(q, v)$ is a set valued function. Similarly, the value of jumps, $v_+(t) - v_-(t)$, will be drawn from a set which generally depends on $q(t)$ and $v_-(t)$. The locations of impacts will be defined implicitly by the locations where $v_+(t)$ and $v_-(t)$ disagree. Finally, we take $v(t)$ to be undefined at points of discontinuity. Our problems center around systems where solutions must lie in an *admissible set*, \mathcal{A} , defined as the set of non-penetrating states:

$$\mathcal{A} = \{(q, v) \in \mathbb{R}^{2n} \mid \phi_i(q) \geq 0 \forall i \in \{1, \dots, m\}\}. \quad (3.6)$$

3.2.4 Contact Dynamics as a Complementarity Problem

We can formulate the dynamics of rigid bodies as an MDI where the contact forces, both continuous and impulsive, must satisfy a set of *complementarity constraints*. For instance, one such constraint relates the contact normal force to the gap function:

$$\begin{aligned} \lambda_N &\geq 0 \\ \phi(q) &\geq 0 \\ \phi(q)^T \lambda_N &= 0 \end{aligned}$$

where the inequalities are taken element-wise and the third constraint implies that there can be no force at distance. These three constraints, taken together, form a complementarity constraint and can be written succinctly as $0 \leq \lambda_N \perp \phi(q) \geq 0$. With this notation, we can

write the frictionless, continuous dynamics as:

$$\begin{aligned} H(q)\dot{v}(t) + C(q, v) &= Bu + J_N(q)^T \lambda_N \\ 0 &\leq \lambda_N \perp \phi(q) \geq 0 \end{aligned}$$

This format can also be extended to include Coulomb friction, though it requires additional notation from convex analysis which would otherwise be unnecessary in this thesis. We refer interested readers to Stewart [127] and Trinkle et al. [138] for the details. The framework has been further adapted for use within time-stepping methods where it has been widely successful in simulation, starting with [128] and [4]. Given the current state q_k, v_k and control input u_k at time t_k , and a timestep h , these approaches pose the problem of simulation as a search for the state at time $t_{k+1} = t_k + h$ along with dynamically consistent contact forces. Typically, these methods will linearize the relevant dynamics and kinematics functions, posing a Linear Complementarity Problem (LCP) [25]. For example, the case for a planar simulation can be written:

$$\text{find } q_{k+1}, v_{k+1} \in \mathbb{R}^n \text{ and } \lambda_N, \lambda_{T+}, \lambda_{T-}, \gamma \in \mathbb{R}^m \quad (3.7)$$

$$\text{s.t. } h^{-1}H(\bar{q})(v_{k+1} - v_k) + C(\bar{q}, \bar{v}) = Bu_k + J(\bar{q})^T \lambda, \quad (3.8)$$

$$h^{-1}(q_{k+1} - q_k) = v_{k+1} \quad (3.9)$$

$$0 \leq \phi(q_k) + hJ_N(\bar{q})v_{k+1} \perp \lambda_N \geq 0 \quad (3.10)$$

$$0 \leq \mu\lambda_N - \lambda_{T+} - \lambda_{T-} \perp \gamma \geq 0 \quad (3.11)$$

$$0 \leq \gamma - J_T(\bar{q})v_{k+1} \perp \lambda_{T+} \geq 0 \quad (3.12)$$

$$0 \leq \gamma + J_T(\bar{q})v_{k+1} \perp \lambda_{T-} \geq 0, \quad (3.13)$$

where $\lambda = \begin{bmatrix} \lambda_N^T & (\lambda_{T+} - \lambda_{T-})^T \end{bmatrix}^T$ and $\gamma \in \mathbb{R}^m$ is a slack variable. The state (\bar{q}, \bar{v}) is used for computation of the manipulator equations, and is chosen to be some explicit function of (q_k, v_k) , such as $\bar{q} = q_k + hv_k$ and $\bar{v} = v_k$. Taken together, these constraints enforce an exact representation of the friction cone for planar simulation. In three dimensions, a linear approximation of the friction cone is typically used.

3.3 Trajectory Optimization

There is a rich literature on both control and planning of nonlinear systems as applied to mobile robotics. Trajectory optimization has been particularly successful in synthesizing highly dynamic motions in high-dimensional state spaces. See Betts [10] for both an overview and a description of the variety of existing algorithms. Broadly speaking, trajectory optimization aims to find dynamically consistent state and control trajectories $x(t), u(t)$ that minimize a cost functional (specified here via ℓ and ℓ_f) subject to an additional set of specified constraints given by \bar{g} .

$$\begin{aligned} \underset{x(t), u(t), T}{\text{minimize}} \quad & \ell_f(x(T)) + \int \ell(x(t), u(t)) dt \\ \text{s.t.} \quad & \dot{x}(t) = f(x(t), u(t)) \\ & \bar{g}(x(t), u(t)) \geq 0 \end{aligned}$$

A popular class of techniques, commonly known as transcription methods, forms a finite dimensional optimization by discretizing the trajectories in time as $x_1, \dots, x_N, u_1, \dots, u_N$ and, between these knot points, enforcing the integral of the dynamics as a constraint. Within this class are multiple-shooting methods, which typically use variable step numerical integrators from x_k to x_{k+1} and have been successfully applied to robotic locomotion tasks [85, 117, 48]. Other common methods avoid this costly integration by using implicit, single step schemes between these knot points (e.g. via Euler integration schemes or by approximating trajectories as Hermitian splines [10, 45]). The direct collocation algorithm of Hargraves and Paris [45], and similar techniques, have been widely used within the robotics community to optimize walking and running gaits (e.g [15, 112, 48]). In this thesis, we will make use of an Euler-based scheme in Chapter 4 and a direct collocation approach in Chapter 5.

3.3.1 Direct Collocation

The direct collocation algorithm, introduced by Hargraves and Paris, uses cubic Hermite splines to interpolate between a sequence of knot points [45]. The algorithm optimizes over the decision parameters $z = (x_1, \dots, x_N, u_1, \dots, u_N)$ defined at the times t_1, \dots, t_N with the

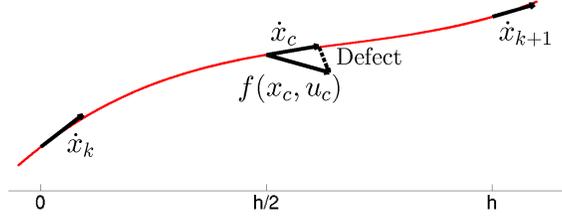


Figure 3-2: The Hermite spline of the direct collocation algorithm, shown in red, is constructed between two knot points. The defect is the discrepancy between the collocation dynamics $f(x_c, u_c)$ and the slope of the spline \dot{x}_c .

uniform timestep h . We briefly describe the approach here, as applied to a second-order system. The plant dynamics are evaluated at each knot point $\dot{v}_k = f(x_k, u_k)$ and, for every sequential pair of knot points x_k, x_{k+1} and inputs u_k, u_{k+1} , a cubic spline $x_s : [t_k, t_k+h] \rightarrow \mathbb{R}^{2n}$ is generated that matches the state and its first derivative at the knot points:

$$x_s(t_k) = x_k, \quad x_s(t_{k+1}) = x_{k+1}, \quad (3.14)$$

$$\dot{x}_s(t_k) = \begin{bmatrix} v_k \\ f(x_k, u_k) \end{bmatrix}, \quad \dot{x}_s(t_{k+1}) = \begin{bmatrix} v_{k+1} \\ f(x_{k+1}, u_{k+1}) \end{bmatrix}. \quad (3.15)$$

The state at the midpoint of this spline, $x_c = x_s(t_k + \frac{1}{2}h)$, called the collocation point, is simply a linear combination of the state and its derivative at the adjacent knot points. Similarly, $\dot{x}_s(t_k + \frac{1}{2}h)$ can be easily calculated:

$$\begin{aligned} x_s(t_k + \frac{1}{2}h) &= \frac{1}{2}(x_k + x_{k+1}) + \frac{h}{8}(f(x_k, u_k) - f(x_{k+1}, u_{k+1})), \\ \dot{x}_s(t_k + \frac{1}{2}h) &= \frac{3}{2h}(-x_k + x_{k+1}) - \frac{1}{4}(f(x_k, u_k) + f(x_{k+1}, u_{k+1})). \end{aligned}$$

The collocation constraint function, g , matches the slope of the spline to the plant dynamics at the collocation point, where the control input is typically taken to be the result of a first-order hold, $u_c = \frac{u_k + u_{k+1}}{2}$. Therefore, we define:

$$g(x_k, u_k, x_{k+1}, u_{k+1}) = \dot{x}_s(t_k + \frac{1}{2}h) - \begin{bmatrix} v_c \\ f(x_c, u_c) \end{bmatrix}. \quad (3.16)$$

These collocation constraints, illustrated in Figure 3-2, are an efficient and accurate representative of the plant dynamics. The integration error over a single timestep is $\mathcal{O}(h^4)$ and so the error over a fixed time interval is then $\mathcal{O}(h^3)$ [44]. This third-order accuracy compares favorably with Euler-integration based methods, which have only $\mathcal{O}(h)$ accuracy over similar intervals. With higher-order methods, accurate trajectories can be achieved with fewer knot points and, therefore, with smaller optimization problems. The resulting trajectory optimization problem, with running cost $\ell(x_k, u_k)$ and a final cost $\ell_f(x_N)$ can then be expressed as:

$$\begin{aligned}
 & \underset{z}{\text{minimize}} && \ell_f(x_N) + h \sum_{k=1}^N \ell(x_k, u_k) \\
 & \text{subject to} && 0 = g(x_k, u_k, x_{k+1}, u_{k+1}) \text{ for } k = 1, \dots, N - 1 \\
 & && 0 \geq \bar{g}(z),
 \end{aligned} \tag{3.17}$$

where $\bar{g}(z)$ represents arbitrary additional constraints on state and input, such as variable bounds or boundary values.

3.4 Stability and Reachability

Recent advances in polynomial optimization, particularly sums-of-squares (SOS) optimization, have proven to be powerful numerical tools for formal analysis of dynamical systems [95, 66]. In this section, we review convex optimization based numerical methods for computing stability and reachability analysis of nonlinear dynamical systems. We first discuss the necessary concepts from nonlinear control, and then relate these concepts to SOS computational approaches. SOS optimization will be used throughout Chapters 6 and 7.

3.4.1 Lyapunov and Barrier Functions

Lyapunov functions capture the stability properties of nonlinear dynamical systems (see [56] for a thorough treatment). For example, given an autonomous system $\dot{x} = f(x)$ with

$f(0) = 0$ and some class \mathcal{K} function α , if $V : \mathbb{R}^n \rightarrow \mathbb{R}$ can be found such that

$$V(0) = 0 \tag{3.18}$$

$$V(x) \geq \alpha(\|x\|) \tag{3.19}$$

$$\frac{dV(x(t))}{dt} \leq 0, \tag{3.20}$$

then $V(x)$ is a Lyapunov function and the origin is locally stable in the sense of Lyapunov. Furthermore, the conditions above prove that every sublevel set of V is positively invariant.

Remark 3.1. *Note that if the inequality (3.20) holds strictly away from the origin (bounded by another class \mathcal{K} function), then the origin is globally asymptotically stable. However, for systems with friction, equilibria are rarely isolated and asymptotically stable, and so we will primarily focus on local stability and invariance.*

Since typical robotic systems will not be globally stable, we are primarily concerned with regional statements. If the criteria (3.20) holds for all x in the ρ -sublevel set of V ,

$$\{x : V(x) < \rho\},$$

then the origin is stable in the sense of Lyapunov, the ρ -sublevel set is positively invariant, and all contained sublevel sets are also invariant.

Barrier functions, similar in principle to Lyapunov functions, are a popular technique for verifying reachable sets for nonlinear systems. Here, we briefly discuss the role of time-varying barrier functions [135]. The time-invariant case, also used in this thesis, is merely a simple modification. Barrier functions can be used to partition the state space by capturing relevant dynamical properties. Under mild conditions, if functions $V : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$ and $\rho : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ can be found, such that

$$V(t, x) = \rho(t) \text{ and } t \in [0, T] \Rightarrow \frac{d\rho(t)}{dt} - \frac{dV(t, x(t))}{dt} > 0 \tag{3.21}$$

then the ρ -sublevel set of V ,

$$\{(t, x) : t \in [0, T], V(t, x) < \rho\},$$

is positively invariant over the given time interval. In other words, if, on the boundary of the sublevel set, V is decreasing in time faster than ρ is, then trajectories that start within the sublevel set cannot leave it. Note that if (3.21) were to hold for all $V(t, x) \leq \rho(t)$, then V would be a Lyapunov function—thus the barrier condition is generally less restrictive.

3.4.2 Sums-of-squares and the S-procedure

Observe that if V , ρ , and the system dynamics are known polynomial functions, then both the Lyapunov and barrier criteria above are fundamentally questions of positivity of polynomials on basic semialgebraic sets. We will first utilize a technique referred to as the S-procedure [95], deriving from the Positivstellensatz [125, 109], to construct a sufficient condition. For example, to demonstrate that $g(x) \geq 0$ and $h(x) = 0$, together, imply $f(x) \geq 0$, we introduce multiplier polynomials $\sigma_i(x)$ and $q(x)$, requiring

$$\sigma_1(x)f(x) - \sigma_2(x)g(x) - q(x)h(x) \geq 0, \tag{3.22}$$

$$\sigma_1(x) - 1 \geq 0, \tag{3.23}$$

$$\sigma_2(x) \geq 0. \tag{3.24}$$

While some theoretical guarantees do exist, for multipliers of limited degree, this sufficient condition is generally not necessary. See Laurent [69] for an overview of the topic. With the S-procedure, the Lyapunov and barrier criteria can now be cast as questions of global positivity. Sum-of-squares (SOS) methods enable optimization over linearly parameterized polynomials that are guaranteed to be non-negative [94, 95].. If constraints like (3.22) are linear in the coefficients of the unknown polynomials, then this sufficient condition can be efficiently represented and solved via Semidefinite Programming (SDP), a form of convex optimization (see [12] for an overview). Given some polynomial $p(x) \in \mathbb{R}[x]$, a sufficient condition for $p(x)$ to be non-negative for all x is the existence of a positive semidefinite

matrix Q and the decomposition

$$p(x) = m(x)^T Q m(x) \tag{3.25}$$

$$Q = Q^T \succeq 0, \tag{3.26}$$

where $m(x)$ is a polynomial basis vector, e.g. the vector of all monomials up to some degree. Ultimately, by restricting the search for Lyapunov and barrier functions to a particular (but expressive) subclass of functions, SOS-based approaches allow synthesis of answers to the fundamentally hard questions of stability and reachability.

3.4.3 Bilinear Alternations

In this thesis, we will often be interested in situations where both the original polynomials and multipliers in (3.22) are unknown, and so bilinear approaches will be used. Bilinear formulations are also required when the control policy is unknown, as (3.20) and (3.21) will include the product of V and u . Fixing one set of free coefficients (e.g. those of f, g and h), we can search over the free coefficients of the others, and vice versa. These allows for a coordinate-descent strategy, related to the techniques of DK-iteration [73], where each step consists of a single convex optimization problem and is guaranteed to improve the objective value. Similar approaches have been applied to determine region-of-attraction estimates for smooth dynamical systems [137]. Note that bilinear techniques require a feasible initialization, a condition that may be difficult to satisfy for some problems.

In practice, bilinear alternations have the possibility of failing due to numerical conditioning. Many of the SDPs generated from SOS programs have no feasible interior, due to an overparameterization of the the sums-of-squares basis. This issue is referred to as the problem of facial reduction, and recent work has been done to enhance the typical pre-processing steps to reduce the dimensionality of the SDP [97, 96]. However, no tractable general solution exists for solving the facial reduction problem.

Chapter 4

Contact-implicit Trajectory

Optimization

As described in 3.3, trajectory optimization is a powerful framework for planning. In this chapter, we consider the problem of trajectory optimization for rigid-body systems in contact-rich environments [98, 100]. In particular, we will plan new motions, interacting with the environment via inelastic collisions and friction, where the sequencing of such contacts is entirely unknown *a priori*. A popular method for control of such systems uses the language of hybrid dynamical systems (refer back to 2.1.1). While there are many impressive success stories for trajectory optimization of these hybrid models, they are plagued with the major short-coming that the optimization is constrained to operate within pre-specified ordering of the hybrid modes (contacts). For some motions with relatively few or intuitive mode transitions, this may be acceptable. It is much more difficult to imagine a mode specification for a multi-fingered hand manipulating a complex object that is frequently making and breaking contact with different links on the hand. Perhaps as a result, there is an apparent lack of planning solutions for robotic manipulation which plan *through* contact - most planners plan up to pre-grasp then activate a separate, heuristic based, grasping controller.

To address this short-coming, we present here a contact-implicit trajectory optimization algorithm. We demonstrate that it is natural to fold the complementarity constraints used to implicitly represent contact forces (see 3.2.4) directly into nonlinear optimization for trajectory design, resulting in a mathematical Program with complementarity constraints

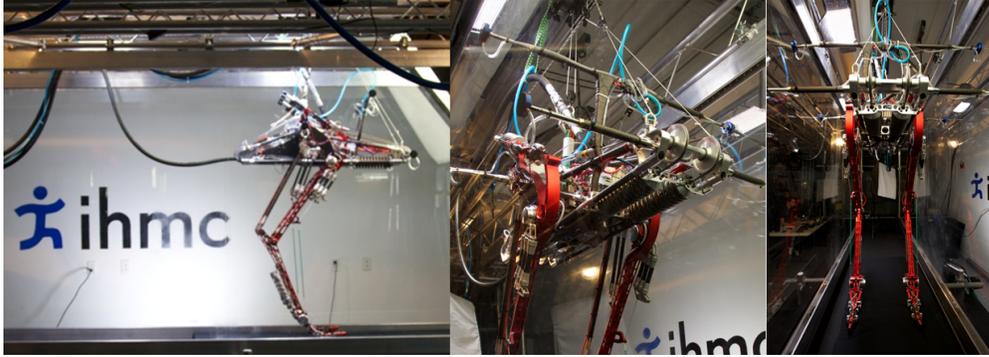


Figure 4-1: The bipedal FastRunner robot is designed to run at speeds of over 20 mph. Each leg has 5 degrees of freedom and multiple passive springs and tendons. The legs are driven at the hip to keep the leg mass as low as possible.

(MPCC), or, equivalently, a mathematical program with equilibrium constraints (MPEC) [76]. While these are generally difficult to solve, significant research has been done in this area, and we leverage sequential quadratic programming (SQP) techniques - a particular class of algorithms for solving general nonlinear programs that have been shown to be effective [3, 35]. Broadly speaking, SQP solves a sequence of quadratic programs which each approximate the original nonlinear program. The key to this formulation is in resolving the contact forces, the mode-dependent component of the dynamics in the traditional formulation, as additional decision variables in the optimization. We demonstrate that this is an effective and numerically robust way to solve complex trajectories without the need for a mode schedule.

This work was motivated by the challenge of optimizing trajectories for a novel running robot called “FastRunner”, designed and built by researchers at the Florida Institute for Human and Machine Cognition (IHMC) [26]. FastRunner, shown in Figure 4-1 is a bipedal robot concept designed to run at speeds over 20 mph and up to 50 mph. Most notably, FastRunner has a clever, but also complex, leg design with four-bar linkages, springs, clutches, hard joint stops, tendons and flexible toes. The planar FastRunner model has 13 degrees of freedom, 6 contact points, 16 additional constraint forces, and only 2 actuators, and was beyond the scope of our previously existing trajectory optimization tools.

4.1 Contact-implicit Approach

The dynamics of robots in contact, formulated using the complementarity conditions, fit naturally into the direct formulation of trajectory optimization. Rather than solving the LCP for the contact forces λ at each step, we directly optimize over the space of feasible states, control inputs, constraint forces, and trajectory durations. Treating the contact forces as optimization parameters is similar to how direct methods treat the state evolution implicitly. The number of parameters and constraints increases, but the problem is often better conditioned and more tractable to state-of-the-art solvers. Over this expanded parameter set, the optimization problem can be written as

$$\underset{\{h, x_0, \dots, x_N, u_1, \dots, u_N, \lambda_1, \dots, \lambda_N\}}{\text{minimize}} \quad \ell_f(x_N) + h \sum_{k=1}^N \ell(x_{k-1}, u_k). \quad (4.1)$$

4.1.1 Optimization Constraints

This optimization problem is subject to constraints imposed by the manipulator dynamics and by rigid body contacts. To integrate the dynamics, both forward and backward Euler methods are equally applicable. Time-stepping simulation methods commonly use semi-implicit methods, but the dynamics constraints in our optimization problem are already fully implicit, and so we chose backwards integration for added numerical stability. For ease of notation, we will write $H_k = H(q_k)$ and likewise for other matrix functions in the manipulator equations. For $k = 1, \dots, N-1$, the dynamics from (3.1) imply the constraints:

$$\begin{aligned} q_k - q_{k+1} + h v_{k+1} &= 0 \\ H_{k+1}(v_{k+1} - v_k) + h (C_{k+1} - B_{k+1} u_{k+1} - J_{k+1}^T \lambda_{k+1}) &= 0. \end{aligned} \quad (4.2)$$

For notational simplicity, we first consider the case where the (frictional) contact dynamics are planar and later discuss the extension to 3D contacts. For a given contact point, write the contact force $\lambda = [\lambda_{T+} - \lambda_{T-} \quad \lambda_N]^T$. Following the formulation of [128], we have split the tangential force into its positive and negative components. We also introduce the additional slack variable γ , which can be thought of as the magnitude of the relative tan-

gential velocity at a contact. Similar to the simulation description in (3.8)-(3.13), we have the set of complementarity constraints

$$0 \leq \phi(q_k) \perp \lambda_{k,N} \geq 0, \quad (4.3)$$

$$0 \leq \mu \lambda_{k,N} - \lambda_{k,T+} - \lambda_{k,T-} \perp \gamma_k \geq 0, \quad (4.4)$$

$$0 \leq \gamma_k - J_T(q_k)v_k \perp \lambda_{k,T+} \geq 0, \quad (4.5)$$

$$0 \leq \gamma_k + J_T(q_k)v_k \perp \lambda_{k,T-} \geq 0. \quad (4.6)$$

Taken together, (4.3)-(4.6) describe inelastic impacts and a Coulomb coefficient of friction μ . Unlike with the task of pure simulation, as in 3.2.4, where these constraints and the dynamical constraints in (4.2) can be linearized about the initial state, here we must consider the higher-order behavior along complex trajectories, and so we use the true nonlinear formulation.

In addition to expressing frictional contacts, we can also describe simple position constraints such as hard joint limits or kinematic loops in a similar manner. Here, λ is an internal torque or force acting directly on a joint. For example, if there is a physical stop enforcing the requirement that $q \geq q_{min}$, write

$$0 \leq \underbrace{q_k - q_{min}}_{\phi(q_k)} \perp \lambda_k \geq 0 \quad (4.7)$$

It is important to note the relative indexing of the complementarity and dynamical constraints. Over the interval $[t_k, t_{k+1}]$, the contact impulse can be non-zero if and only if $\phi(q_{k+1}) = 0$; that is, the bodies must be in contact at the end of the given interval. This allows the time-stepping integration scheme to approximate inelastic collisions where the interacting bodies stick together. This is not necessarily an appropriate approximation for bodies that may rapidly rebound off one another, since any compliance must be modeled through a linkage in one of the bodies and the time step must be appropriately small.

Remark 4.1. *At a high level, our approach lifts up the problem via overparameterization and embeds the discontinuous and combinatoric planning problem into the nonlinear complementarity constraints. At pre-terminal stages of the optimization solver, whether by a SQP or other method, some or all of these constraints will likely be violated. This, in a sense,*

permits force to be exerted at a distance in these intermediate stages. Intuitively, the gradients of these violated constraints represent information about the set of contacts that might be made—allowing the optimization to simultaneously search over all possible sequences.

4.1.2 Solving the Optimization Problem

The optimization problem (4.1), subject to the constraints in Section 4.1.1, forms an MPCC: a class of nonlinear programs that is generally difficult to solve due to the ill-posed nature of the constraints [76]. However, it is an area of optimization research that has garnered significant attention in recent years. There are a number of theoretical and practical results which we leverage here to ensure that our trajectory optimization problem is solvable with current techniques, particularly those used by the nonlinear solver SNOPT [39]. For vector-valued functions $G(z)$ and $H(z)$, many of our constraints are of the form

$$G(z) \geq 0, \tag{4.8}$$

$$H(z) \geq 0, \tag{4.9}$$

$$G(z)^T H(z) = 0. \tag{4.10}$$

To improve the convergence properties of the optimization routines, we can consider equivalent formulations of these complementarity conditions. Fukushima et al. [37] propose an iterative method that sequentially tightens relaxations of the complementarity constraints. In our work, we primarily adopt the scheme of Anitescu [3] who proposed leveraging the elastic mode of SQP solvers like SNOPT to solve a set of similar, and equivalent constraints,

$$G(z) \geq 0, \tag{4.11}$$

$$H(z) \geq 0, \tag{4.12}$$

$$G_i(z)H_i(z) \leq 0, \tag{4.13}$$

where the last inequality is evaluated element-wise. Additionally, it was observed by Fletcher et al. in [35] that, since SQP iterations always satisfy linear constraints, the introduction of

slack variables α and β can help avoid infeasible QP iterations:

$$\alpha, \beta \geq 0, \tag{4.14}$$

$$\alpha = G(z), \tag{4.15}$$

$$\beta = H(z), \tag{4.16}$$

$$\alpha_i \beta_i \leq 0. \tag{4.17}$$

In practice, these seemingly innocuous substitutions have greatly improved the speed and robustness of our optimization routines relative to our initial formulation, described in [98]. For more complex examples, we have also found it to be practically useful to temporarily relax the final constraint to $\alpha_i \beta_i \leq \epsilon$ and solve a sequence of problems (typically no more than three or four), starting with some $\epsilon > 0$ and finishing with $\epsilon = 0$ to achieve strict feasibility. This has the effect of allowing intermediate iterations to exert contact force at a small distance, and experimentally has improved the conditioning of the optimization problem and the quality of our solutions. This is similar in principle to existing approaches, like that in [37].

4.1.3 Extension to Three Dimensions

To handle three dimensional contacts, note that only the variables and constraints in 4.1.1 related to the frictional force λ_T and tangential velocity $J_T(q)v$ are specific to the 2D case. One straightforward approach to extending to 3D would be to treat both λ_T and $J_T(q)v$ as two-vectors, and write down a set of nonlinear constraints for Coulomb friction in the tangent plane, such as in [1]. However, to preserve the MPCC structure of our problem, we instead use a polyhedral approximation of the friction cone, as in [128]. Let D_i for $i = 1, \dots, d$ be unit vectors in \mathbb{R}^2 whose convex hull is the polyhedral approximation. Then, let $\lambda_T = \sum_i^d D_i \lambda_{T_i}$ be the net frictional force where each λ_{T_i} is a scalar. We replace the friction cone constraints

(4.4)-(4.6) with

$$0 \leq \mu \lambda_{k,N} - \sum_i^d \lambda_{k,T_i} \perp \gamma_k \geq 0, \quad (4.18)$$

$$0 \leq \gamma_k + D_i^T J_T(q_k) v_k \perp \lambda_{k,T_i} \geq 0, \quad (4.19)$$

where (4.19) is repeated for all i . By increasing d and growing the size of the MPCC, the approximation can be made arbitrarily tight to the true friction cone.

4.1.4 Time Discretization

We also note here the role of the discrete time steps when resolving contacts. Since we use a time-stepping model, our approach makes no effort to determine the exact time that contact between bodies is made or broken. Impulsive and continuous forces are not treated independently, and so we avoid the difficult and potentially combinatorial task of hybrid mode resolution. Instead, the constraint forces over the time step directly before a collision are precisely those required for the two bodies to be in contact. Additionally, since no force is permitted during the period when contact is being broken, there is the implicit requirement that take-off exactly coincide with one of the discrete time intervals. While it is common for numerical implementations of trajectory optimization to allow the overall duration of the trajectory to change, they typically do not adjust the individual time steps. Here, this would result in an overly restrictive optimization problem that may exclude desirable trajectories. Overly simple parameterizations which use each time step duration as a parameter can have trivial or undesirable solutions (e.g, with many time steps having zero duration). One feasible approach is to create decision variables that divide each time step h into two periods. This can alternatively be expressed as having individual time steps h_k with pairwise constraints:

$$h_{2j-1} + h_{2j} = h_{2j+1} + h_{2j+2}, \quad j = 1, \dots, \text{floor} \left(\frac{N-3}{2} \right), \quad (4.20)$$

If the total duration of the motion is fixed, this is equivalent to constraining $h_{2j-1} + h_{2j}$ to be constant. In practice, these additional free parameters are useful in expanding the space

of feasible solutions while still allowing for relatively large time steps. Since both state and constraint forces are solved implicitly, this program has a relatively large number of decision variables and constraints. However, as is typical in direct methods, this resulting program is generally sparse and so is suitable for implementation with sparse solvers.

4.2 Example Applications

In this section, we apply our contact-implicit algorithm to four increasingly complex examples. These problems were solved on a standard desktop computer. While the solve time varied from problem to problem, the simpler examples completed in a few minutes or less in about one hundred major iterations of the solver, and the FastRunner trajectories took up to an hour to converge. As discussed above, the MPCCs are formulated to take advantage of the elastic mode in the SQP solver, although, in all cases, we achieve final convergence to a strictly feasible solution.

4.2.1 Finger Contact

Recent research by Tassa and Todorov [131] used a DDP based approach to find a trajectory for the sample problem of a two link manipulator that must spin an ellipse. This is a simple example with three degrees of freedom and only one contact point, so there are only two possible modes (excluding sliding). However, it provided an early test for our methods. We constrained the system to start from rest, $v = 0$, and optimized for a quadratic cost on control input and velocity of the free ellipse:

$$g(x, u) = \sum_{k=1}^N v_3^T Q v_3 + u^T R u. \quad (4.21)$$

The parameters for size and mass and for the cost function were chosen to directly parallel the previous work. Our approach succeeded in quickly finding a locally optimal trajectory. As we increased the overall duration of the trajectory, the optimization process found an increasing number of flicking motions where, after making contact, it drew the finger back up to make another pass. Additionally, Tassa and Todorov note that the effect of gravity was

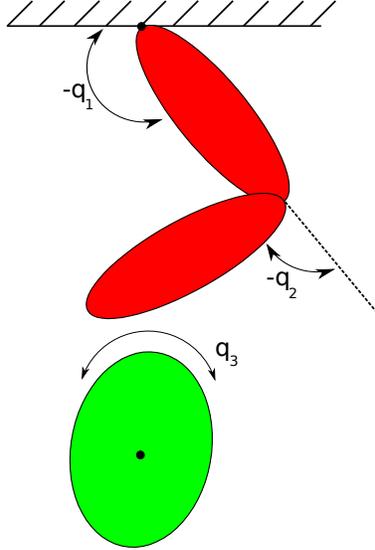


Figure 4-2: The two link finger, shown above, is fully actuated and makes contact with the unactuated third ellipse to drive it about its axis. Here, $\phi(q)$ is the shortest distance between the distal finger and the free ellipse.

required to pull the manipulator into contact with the ellipse in order for the optimization process to discover the possibility of contact. Our approach does not have this limitation. If we eliminate gravity from the system, even given an initial trajectory that starts at rest with $u(t) = 0$, our methods successfully initiate contact between the manipulator and ellipse. Both of these results speak to the capability of our algorithm to actively identify a mode schedule that is not forced by the initial trajectory or the system’s dynamics.

4.2.2 Simple Manipulation

Given the complexity of manipulator problems, trajectory optimization usually involves dividing the planner into two parts: planning the motion to the object through unobstructed space and then subsequently planning the grasp. During grasp planning, specifying a mode schedule would require determining the order in which the manipulator fingers should interact with the object. However, it is clear that in many situations, the precise order is not important so long as a proper grasp is ultimately achieved. Thus, an optimization technique that does not require *a priori* specification of contact order is far more appropriate for these types of problems. Furthermore, some grasp planners neglect the dynamics of manipulated objects, essentially treating them as fixed to the manipulator. The method presented here

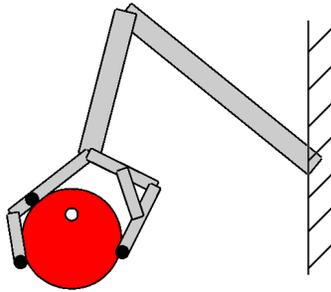
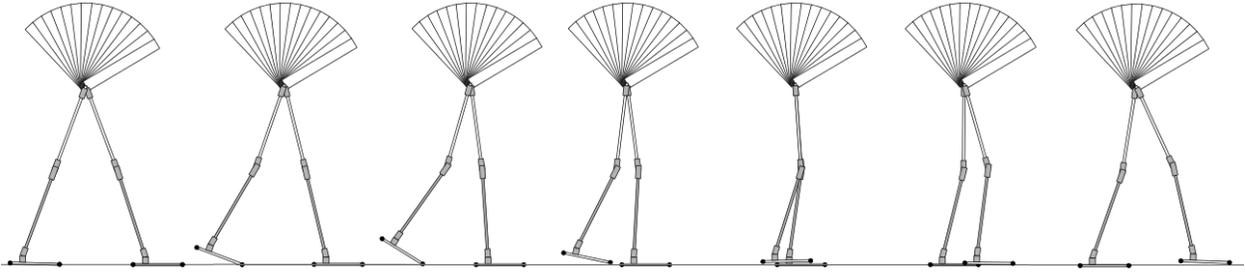


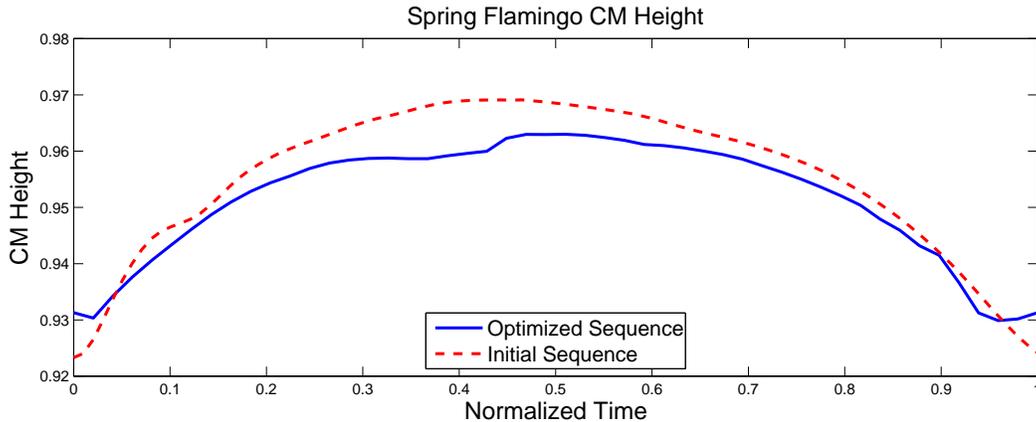
Figure 4-3: A simple planar gripper was modeled with five actuated joints and three contact points, shown as black dots. Both the ball and the three contact points could also interact with the ground, where the ball is initially resting, resulting in seven possible contacts.

fully accounts for the dynamical properties of the manipulated object throughout optimization of the entire trajectory. The following example consists of a planar manipulator tasked with grasping a circular object, originally on the ground, and lifting it into the air. We model the gripper with three contact points and five actuated joints, shown in Figure 4-3. Here, we desire to minimize the overall effort while moving the ball to a strictly specified goal location.

There are, of course, a number of different ways in which to precisely specify this optimization problem. Since we use a local method, the problem definition and initial trajectories can have a significant impact on the final result. We choose an approach similar to standard grasp planning by specifying an intermediate state where the manipulator fully grasps the ball before attempting to raise it; and we additionally specify the final state in terms of both ball and manipulator. For an initial trajectory, we construct a simple, three point linear interpolation between the initial, intermediate, and final states where the intermediate state is the grasp chosen by the user. Additionally, the optimization problem is initialized with all zeros for both control inputs and contact forces. Furthermore, we are able to require that the trajectory achieve a force closure on the ball, before attempting to raise it. With this simple problem formulation, our algorithm quickly converges to a trajectory that grasps and raises the ball. If, instead, we were to only specify a fixed initial state with the manipulator held above the ball, the algorithm could produce interesting motions that, while locally optimal, may lack robustness. For instance, using a single finger to hit the ball into the air satisfies



a A walking gait for Spring Flamingo that minimizes mechanical cost of transport. To generate this trajectory, the height of the swing foot was not considered, so the solution is a minimalist trajectory with very little ground clearance.



b The height of the center of mass over the optimized and initial trajectories is plotted over the sequence. The optimal trajectory minimizes unnecessary vertical motion of the robot.

Figure 4-4: Spring Flamingo walking.

this problem statement.

Since they are included as optimization parameters, the cost function and constraints can be easily modified to explicitly include the contact forces. For example, to handle the object gently, we could minimize the total contact forces between fingers and ball or even prohibit these forces from crossing a specified threshold.

4.2.3 Spring Flamingo Walking Gait

To analyze a more realistic system, we tested our methods on a planar simulation of the Spring Flamingo robot [107]. On Spring Flamingo, each leg has three actuated joints (hip, knee, and ankle) and there are contact points at the toe and heel of each foot. Many hybrid walking models use a constrained form of the dynamics, where a foot in contact with the

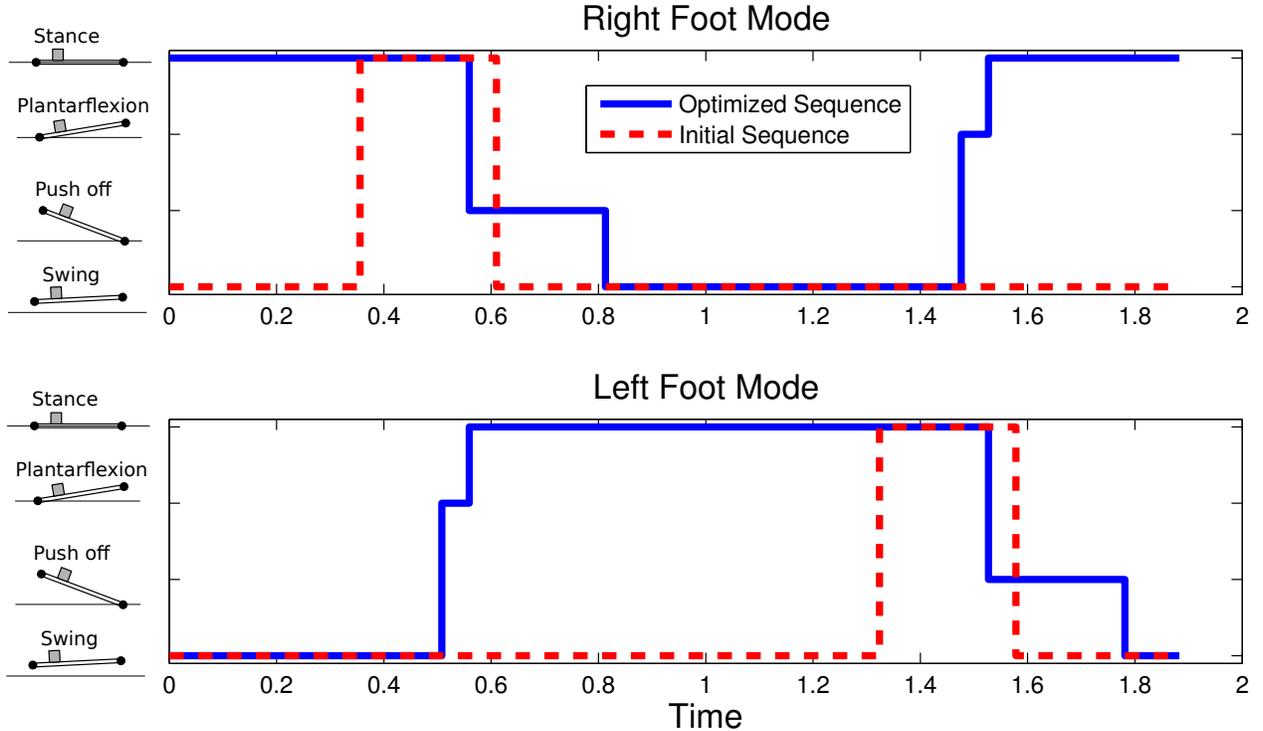


Figure 4-5: The optimized mode sequence of the left and right feet is plotted against time and the mode transitions are labeled. The SQP was initialized with a significantly different sequence, demonstrating the ability of the algorithm to independently plan through contact discontinuities. Note that, in this case, the locally optimal trajectory has distinct heel strike and heel off events.

ground is treated as a pin joint. Here, however, we deal with the full constrained dynamics where the body of the robot is modeled as a floating base parameterized by the variables (x, y, θ) , which represent the planar position and pitch of the robot. Periodic constraints were used to generate a cyclic walking gait and the trajectory was optimized for mechanical cost of transport. Cost of transport is a common, unitless indication of the energy consumption required for locomotion, and the “mechanical” cost of transport is computed using the total positive work done on the system independent of losses in the actuators or costs due to onboard electronics [23]. Where d is the total distance traveled, we write the cost as:

$$g(x, u) = \frac{1}{mgd} \sum_{k=1}^N \sum_i |v_{k,i} u_{k,i}|. \quad (4.22)$$

Note that negative work, which could potentially be stored in an elastic element or harvested by regenerative breaking, is simply treated here with an equivalent cost to positive work.

Since the solutions to the MPCC are local, our methods discovered a wide variety of feasible gaits that satisfied the general constraints dependent on the initial condition set. For instance, given the task of finding a periodic gait that travels a specific distance, hopping motions and gaits with relatively short or long strides are possible local solutions. In particular, the input $\lambda(t)$ and $x(t)$ sequences implicitly identify the nominal mode sequence of the initial guess. However, the solution is not restricted to the given ordering. Figure 4-5 shows the initial and optimized mode sequences of a particular Spring Flamingo gait. Here, the initial trajectory leads to a solution with a right-left walking gait but details such as independent heel strike and heel off were identified in the optimization process. To optimize for a cyclic gait, it is natural to write the periodicity constraint $x_N = x_1$.

To shrink the optimization problem, we search over the smaller space of a half gait. We add an additional constraint that the robot walk a minimum distance, we formulate a periodicity requirement to account for symmetry. Where q_l and q_r are the left and right joint vectors, respectively, d_{min} is the minimum stride length, and (x_{CM}, y_{CM}, θ) represents the position and orientation of the center of mass, we have:

$$\begin{bmatrix} y_{N,CM} \\ \theta_N \\ \dot{x}_{N,CM} \\ \dot{y}_{N,CM} \\ \dot{\theta}_N \\ q_{N,l} \\ q_{N,r} \\ v_{N,l} \\ v_{N,r} \end{bmatrix} = \begin{bmatrix} y_{1,CM} \\ \theta_1 \\ \dot{x}_{1,CM} \\ \dot{y}_{1,CM} \\ \dot{\theta}_1 \\ q_{1,r} \\ q_{1,l} \\ v_{1,r} \\ v_{1,l} \end{bmatrix}, \quad (4.23)$$

$$x_{N,CM} \geq x_{1,CM} + d_{min}. \quad (4.24)$$

With these linear constraints and given a nominal trajectory from Pratt's original work on the robot where the mechanical cost of transport was 0.18 [106], our methods identified a periodic walking gait which reduced the cost to 0.04. It is important to note that this is

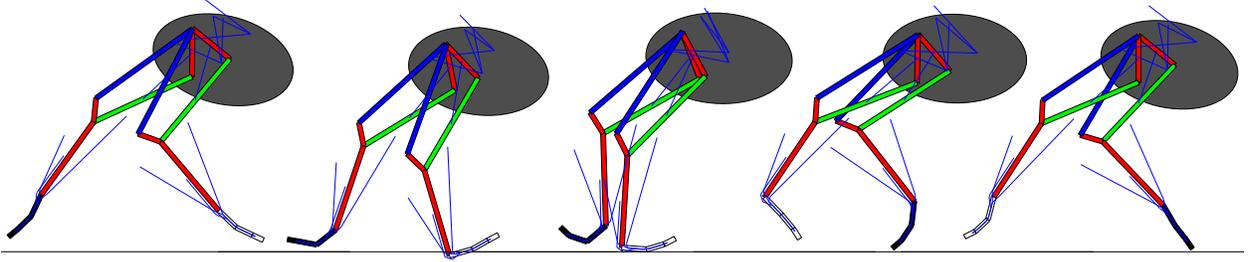


Figure 4-6: A generated trajectory for the FastRunner robot running at over 20 mph. The solid elements show the leg linkages and the thin lines indicate springs and tendons. Only the hip joints of the robot are actuated.

merely the cost of the nominal gait as calculated from (4.22), and that stabilizing the gait in the presence of any disturbances or model error will result in a higher closed loop cost, even in simulation. Using native C++ code for rapid computation of rigid body dynamics, interfaced with a general purpose MATLAB framework, we are able to converge to solutions for the Spring Flamingo in under ninety seconds. The other examples discussed in this work are analyzed primarily in unoptimized MATLAB, and so are less relevant for comparison. This is a significant reduction in cost and corresponds to an impressive level of walking efficiency for a system with no passive elements to store and release energy. Figure 4-4a shows the optimized walking gait and the height of the center of mass (CM) throughout the trajectory, compared with that of the nominal gait. The optimal trajectory minimizes wasteful up and down motion of the CM. Note also that the foot swing height is very low to minimize any velocity at impact.

4.2.4 FastRunner Gait

This research was originally motivated by the challenges posed by the FastRunner platform, shown in Figure 4-1. For the previous examples, it is certainly possible to identify a desired mode sequence. This is a difficult task, however, for a system like FastRunner. A planar model of the robot has 13 degrees of freedom, including three articulated toe segments on each foot that can make or break contact with the ground. Additionally, there are a total of 16 unilateral joint limits, many of which play a critical role in bearing load during the high-speed running gait cycle. Scheduling the order of these contacts and joint limits is not practical.

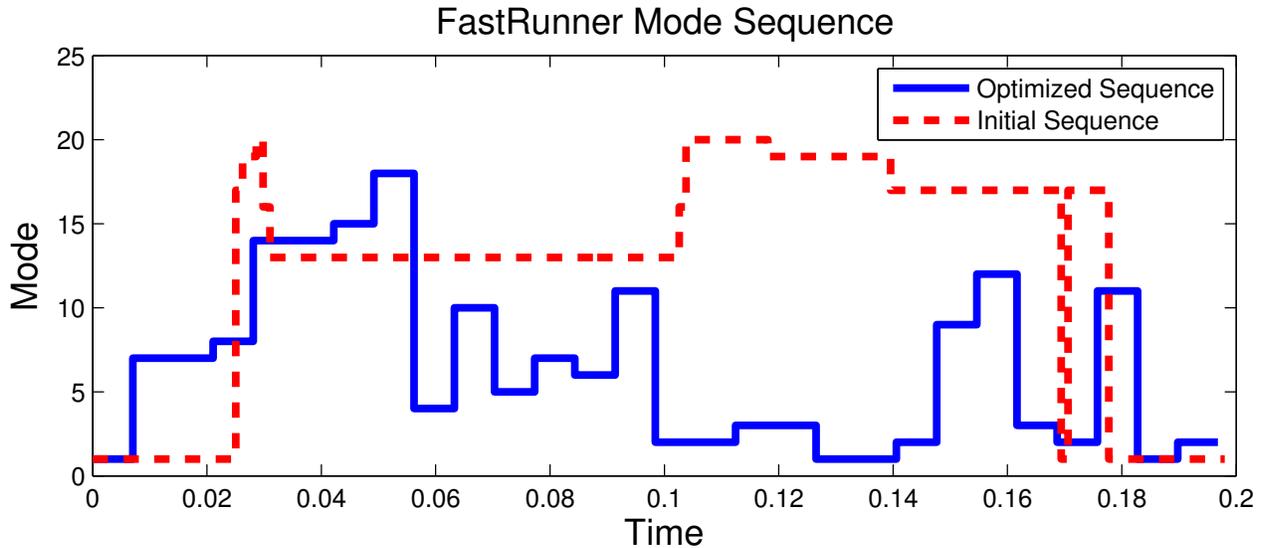


Figure 4-7: Out of more than 4 million possible discrete modes, the sequence for one locally optimal cyclic trajectory is shown. This sequence, which passes through 15 different modes, is compressed and plotted above against the largely distinct mode sequence of the initial trajectory.

Figure 4-6 shows a motion sequence of an optimized periodic running gait, averaging over 20 mph. As with Spring Flamingo, constraints (4.23) and (4.24) restricted the search space and this trajectory was optimized for mechanical cost of transport. Both the leg linkages and passive elements like springs and tendons are shown in the figure. For our model, we treat the system as a linkage of rigid bodies, where the passive elements are treated as massless. The complexity of the system and the stiffness of some of the springs posed additional problems for the optimization. In this case, additional linear constraints were useful in guiding the solver away from poorly conditioned or infeasible regions. This is typical for SQP methods, where the program can be difficult to solve if the local QP is a poor estimate of the true nonlinear program.

With 22 discrete variables, there are over 4 million possible discrete modes for the FastRunner robot. One possible mode sequence discovered by the optimization process is illustrated in Figure 4-7. The individual mode transitions shown occur when the contact state of one of the toes changes or when a joint limit becomes active or inactive. While the states of individual discrete variables, such as toe contacts, may overlap between the initial and optimal trajectories, the aggregate discrete states show almost no agreement. This speaks to the combinatorial complexity of planning a mode schedule for a system like FastRunner.

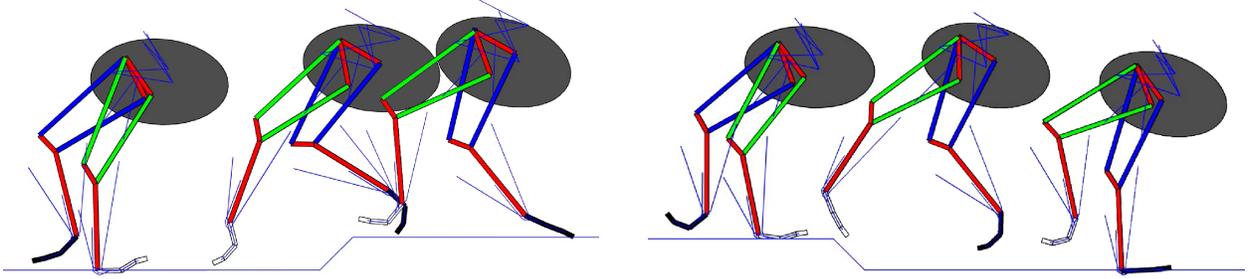


Figure 4-8: Constrained trajectory optimization can be used to generate gaits that deviate from the nominal motion. The images above show the robot ascending and descending 20 cm steps, more than double the height that has been achieved through passive stabilization alone.

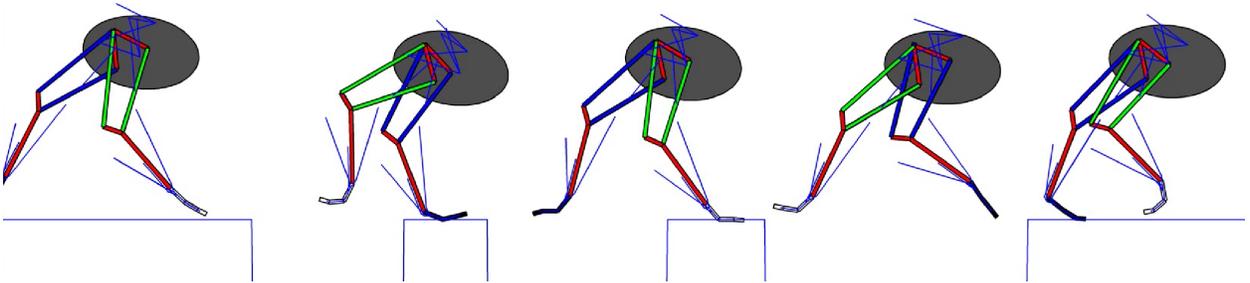


Figure 4-9: The nominal gait can also be modified with explicit foot placement constraints. Here, the robot must significantly alter its stride length to bridge the gaps shown above.

Despite the complexity of the system, by taking advantage of the complementarity constraint formulation, our methods are now able to generate a locally optimal gait for FastRunner.

Our collaborators on the FastRunner project [26] designed the robot to be open loop stable while tracking a simple sinusoidal gait. The design of the robot features an active clutch that connects a large suspensory spring to the knee joint during stance and then disconnects the spring during swing. It is critical that the clutch only be activated and deactivated when the spring is in the neutral state and it has proven extremely difficult to design such a trajectory by hand. Such a constraint, however, fits naturally into our optimization algorithm. If c_k corresponds to the clutch activation at time t_k and $l(q_k)$ is the length of the spring, we then encode the two constraints:

$$c_k^2 - c_k = 0 \quad (4.25)$$

$$l(q_{k+1})(c_{k+1} - c_k) = 0 \quad (4.26)$$

The first equation ensures that c_k is binary, restricted to $\{0,1\}$. The second describes the requirement that the clutch activation can only change if the spring is in the neutral position. With this new set of conditions, our optimization algorithm generates the complex trajectories required for FastRunner.

The trajectory optimization algorithm presented here allows us to synthesize efficient gaits for a wide range of different tasks. Of equal importance to generating a nominal gait is the ability to generate additional motions to handle atypical situations. For example, by modifying the model of the environment, our algorithm was successful at finding trajectories where the FastRunner robot must take 20 cm steps up or down, simulating running over rough (but known) terrain, all while running at high speeds. Figure 4-8 shows the robot mid-flight as it must step up and down. For these tasks, the initial and final states of the trajectory were constrained to precisely match the nominal running gait so that these motions can be smoothly strung together. We also applied our method to the task of explicitly modifying stride length, for situations where we must more tightly control foot placement. Figure 4-9 shows a stop-motion of the results of this optimization, where the small ledges force large deviations from the nominal stride length.

4.3 Discussion

In our approach, we write a complex MPCC that, in practice, has been tractable for modern solvers. Since the problem is non-convex, we are limited to locally optimal solutions. As is typical for non-convex problems, applying linear constraints on the decision variables to steer the solver away from singularities or other poorly conditioned regions can be critical to finding a desirable solution. In the examples above, this is typically done by eliminating obviously undesirable or infeasible regions of the joint space. We have also found that intermediate relaxations of the complementarity conditions can greatly improve the rate of convergence and reduce the likelihood of a poor, local solution. These relaxations are then tightened so the algorithm results in a strictly feasible trajectory. Other smoothing functions for nonlinear complementarity problems exist and have been used to directly solve these problems, such as the Fischer-Burmeister function [34] or the class of functions suggested by Chen and

Mangasarian [19], and these functions may be applicable here as well.

As is mentioned above, throughout this work we have solely dealt with inelastic collisions where the effective coefficient of restitution vanishes. While this is an appropriate assumption for the locomotion and manipulation examples explored here, there are other potential applications where the impacts are better modeled as partially elastic events. The work in [4] has developed LCP-based simulation tools for multi-body contact and elastic collisions, and we believe our methods can be extended to these areas as well.

To control highly-nonlinear robotic systems through real-world environments, it is critical that we be able to generate feasible, high quality trajectories. Prior techniques struggle when presented with complex systems where the hybrid sequence is difficult to intuit. Here, we have presented a method for trajectory optimization through the discontinuities of contact that does not rely on *a priori* specification of a mode schedule. Our approach combines traditional, direct local control approaches with an complementarity based contact model into a single nonlinear program. By writing the dynamics and constraints without explicit reference to hybrid modes, we are now able to easily plan through the discontinuous dynamics of contact. Additionally, unlike with other methods, we do not require arbitrary or hand-tuned parameters nor do we rely on the passive system dynamics to generate a mode schedule. Once convergence is reached, the solution strictly satisfies all contact and dynamics constraints.

Chapter 5

Contact-constrained Optimization and Tracking

In this chapter, we present further progress toward executing and stabilizing motions in contact-rich settings. In Chapter 4, we demonstrated an approach to trajectory optimization with an unknown contact schedule. The contact-implicit method is based on a first-order Euler integration scheme, which, when combined with the relatively large step sizes that are computationally required, results in significant integration error. Particularly for underactuated systems, it can be incredibly difficult or impossible to track and stabilize nominal trajectories that are dynamically inconsistent (as a result of integration error). Even with a known contact schedule (resulting from a contact-implicit optimization or user-specified), standard approaches to trajectory optimization often fail to address complex contact configurations. Contact, when sustained over time, is represented through kinematic constraints on the evolution of the dynamical system. In simple cases, the constrained system can be described by a set of minimal coordinates. However, these constraints frequently create closed kinematic chains, such as in four-bar linkages, or when a walking robot is in double support with both legs contacting the ground. When minimal coordinates do not exist, we must consider the robot's state to be on a manifold embedded in a higher dimensional space [82, 14]. The standard approach, used in [48] and elsewhere, considers only the dynamics induced by the manifold. However, by not directly constraining motions to remain on the manifold, numerical integration error will necessarily result in drift (e.g. a foot leaving the

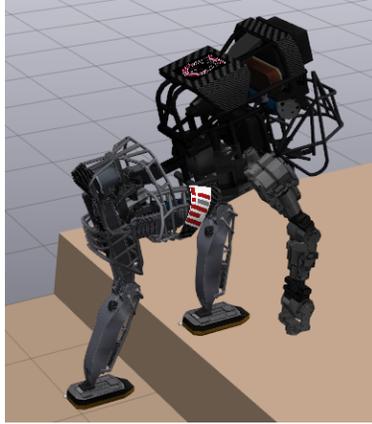


Figure 5-1: An Atlas robot climbing a step with aid of a hand. The algorithms presented here synthesize and stabilize dynamic, multi-contact trajectories.

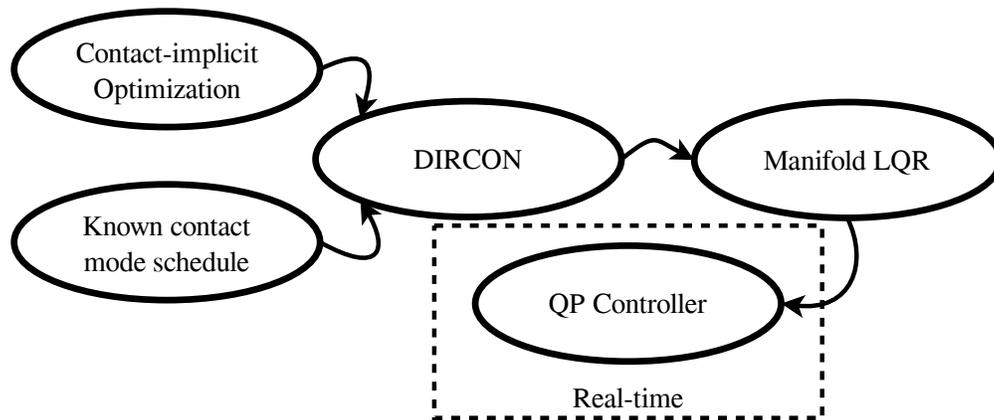


Figure 5-2: A block diagram cartoons the interaction between components. The DIRCON algorithm, given an objective and constraints, produces a nominal trajectory. The constrained version of LQR solves for a quadratic cost-to-go function. This cost-to-go is used to synthesize a QP, which is solved in real-time as a feedback control policy.

ground).

To address this issue, we introduce Constrained Direct Collocation (DIRCON), a planning algorithm for constrained dynamical systems with third-order integration accuracy [101]. DIRCON maintains the advantages of direct collocation [45], thereby enabling more reliable stabilization as compared with existing methods. We additionally introduce a modest extension of the classical linear quadratic regulator (LQR) stabilization technique which addresses the challenges posed by working on these manifolds.

To fully demonstrate these results, we then combine the optimized trajectories and LQR policies with recent advances in humanoid control. By leveraging quadratic programming, we

incorporate constraints such as input saturations and friction limits into the LQR feedback policy as in Kuindersma et al. [61, 62]. These elements, illustrated in Figure 5-2, provide an end-to-end recipe for generating and stabilizing optimal trajectories that exhibit complex and varying contact configurations. We demonstrate the approach on three different locomotion examples in simulation: walking in three dimensions, underactuated planar (2D) walking, and planar climbing utilizing contact between the hand and the environment.

5.1 Constrained Direct Collocation

To plan trajectories with arbitrary contact and other kinematic constraints, we wish to ensure that motions always satisfy the manifold constraints and that the dynamic flow of the trajectories respects the dynamics on the manifold. For ease of notation, we introduce here the functions ψ and α which implicitly define the constraint forces λ :

$$\psi(q, v) \equiv \frac{d\phi}{dt} = J(q)v = 0, \quad (5.1)$$

$$\alpha(q, v, u, \lambda) \equiv \frac{d^2\phi}{dt^2} = \frac{dJ(q)}{dt}v + J(q)\bar{f}(x, u, \lambda) = 0, \quad (5.2)$$

where we have written the dynamics in an implicit form $\dot{v}(t) = \bar{f}(x(t), u(t), \lambda(t))$. For instance, deriving \bar{f} from the manipulator equations (3.1), we would have

$$\bar{f}(x, u, \lambda) = H(q)^{-1}(-C(q, v) + Bu + J(q)^T\lambda).$$

With a slight abuse of notation, in this chapter we will let $\phi(q) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ represent the m active kinematic constraints (potentially corresponding to the normal and tangential directions of a sticking contact constraint).

Remark 5.1. *Observe that simply adding the manifold constraints $\phi(q_k) = \psi(q_k, v_k) = 0$ to the standard direct collocation optimization (3.17) results in an over-constrained problem. This can be easily seen by formulating the optimization as a single-step forward prediction: fix x_0, u_0 , and u_1 and solve for x_1 . Assuming that x_0 lies on the manifold, we would still have a total of $(2n + 2m)$ equality conditions, greater than the dimensionality of the unknown x_1 .*

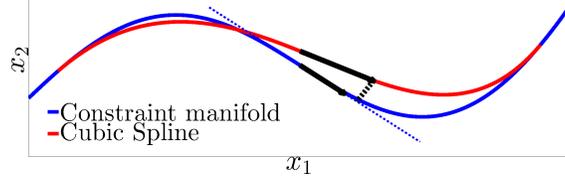


Figure 5-3: A one dimensional constraint manifold, embedded in a two dimensional space, is cartooned in blue. A Hermite spline, in red, between two points will not overlap the manifold, and its slope will not lie within the tangent plane at the collocation point. DIRCON implicitly projects the spline slope onto the manifold to form the proper constraint defect.

Alternatively, if the position and velocity level manifold constraints are ignored and only the acceleration level constraint (5.2) is enforced, then integration error will cause the trajectories to drift off the manifold.

The DIRCON algorithm extends classical direct collocation to naturally handle the difficulties presented by an implicit constraint manifold. This algorithm has two main contributions:

1. we achieve $\mathcal{O}(h^3)$ accuracy for constrained Lagrangian systems,
2. by *explicitly* representing the forces λ , constraints like friction limits are easily expressed.

As in Chapter 4, we incorporate the constraint forces at the knot points, $\lambda_1, \dots, \lambda_N$, as explicit decision variables in the optimization. The key insight in our approach is to reduce the effective dimensionality of the additional collocation constraint (recalling (3.16)) by restricting it to the tangent plane of the constraint manifold through the use of additional slack variables $\bar{\lambda}_1, \dots, \bar{\lambda}_{N-1}, \bar{\gamma}_1, \dots, \bar{\gamma}_{N-1}$. These variables represent forces and a velocity correction, respectively, applied at the collocation point. The resulting projected collocation constraint, cartooned in Figure 5-3, is:

$$\bar{g}(x_k, u_k, \lambda_k, x_{k+1}, u_{k+1}, \lambda_{k+1}, \bar{\lambda}_k, \bar{\gamma}_k) = \dot{x}_s(t_k + \frac{1}{2}h) - \begin{bmatrix} v_c + J(q_c)^T \bar{\gamma}_k \\ \bar{f}(x_c, u_c, \bar{\lambda}_k) \end{bmatrix}. \quad (5.3)$$

Defining the set of optimization parameters as

$$z = (x_1, \dots, x_N, u_1, \dots, u_N, \lambda_1, \dots, \lambda_N, \bar{\lambda}_1, \dots, \bar{\lambda}_{N-1}, \bar{\gamma}_1, \dots, \bar{\gamma}_{N-1}),$$

we have the trajectory optimization problem:

$$\begin{aligned}
& \underset{z}{\text{minimize}} && \ell_f(x_N) + h \sum_{k=1}^N \ell(x_k, u_k) \\
& \text{subject to} && 0 = \bar{g}(x_k, u_k, \lambda_k, x_{k+1}, u_{k+1}, \lambda_{k+1}, \bar{\lambda}_k, \bar{\gamma}_k) \quad \text{for } k = 1, \dots, N-1 \\
& && 0 = \phi(q_k) = \psi(q_k, v_k) = \alpha(q_k, v_k, u_k, \lambda_k) \quad \text{for } k = 1, \dots, N \\
& && 0 \leq \bar{g}(z).
\end{aligned} \tag{5.4}$$

As with standard collocation, $\bar{g}(z)$ represents additional constraints on the state, input, and constraint forces.

Theorem 5.2. *If the dynamics and kinematics functions $(\bar{f}, \phi, \phi', \phi'', \psi, \alpha)$ are analytic and Lipschitz continuous, the algorithm above has $\mathcal{O}(h^3)$ accuracy over a fixed time-interval. More specifically, take $(x_0, u_0, \lambda_0, x_1, u_1, \lambda_1, \bar{\lambda}_0, \bar{\gamma}_0)$ to be bounded and satisfy (5.3). Let $x(t)$ for $t \in [0, h]$ be the true solution to $\dot{x}(t) = f(x(t), u(t))$ with $x(0) = x_0$ and $u(t)$ a first-order hold between u_0 and u_1 . Then, we have that the error $\|x(h) - x_1\| < Ch^4$ for some constant C .*

Proof. Let $q_s(t)$ and $v_s(t)$ correspond to the joint position and joint velocity cubic splines of the collocation algorithm. For simplicity, we will write $q_c = q_s(\frac{1}{2}h)$, $\dot{q}_c = \dot{q}_s(\frac{1}{2}h)$, $v_c = v_s(\frac{1}{2}h)$, and $\dot{v}_c = \dot{v}_s(\frac{1}{2}h)$. Note, because the parameters z are bounded, $q_s(t)$ and $v_s(t)$ are also bounded, and so $\phi(q_s(t))$ will be both bounded and analytic. First, we demonstrate that $\phi(q_c) = \mathcal{O}(h^4)$. Since $\phi(q_s(0))$ and $\dot{\phi}(q_s(0))$ both vanish, the Taylor expansion of $\phi(q_s(t))$ is

$$\frac{t^2}{2} \frac{d^2\phi}{dt^2} \Big|_{t=0} + \frac{t^3}{6} \frac{d^3\phi}{dt^3} \Big|_{t=0} + \frac{t^4}{24} \frac{d^4\phi}{dt^4} \Big|_{t=0} + \dots$$

By substituting and differentiating this expansion, and exploiting the fact that $\phi(q_s(h))$ and $\dot{\phi}(q_s(h))$ also vanish, we can eliminate the quadratic and cubic terms from $\phi(q_c)$,

$$\phi(q_c) = \frac{233}{1142} h^4 \frac{d^4\phi}{dt^4} \Big|_{t=0} + \mathcal{O}(h^5). \tag{5.5}$$

Therefore, for sufficiently small h we have $\|\phi(q_c)\| < Ch^4$ and, similarly, $\|\dot{\phi}(q_c)\| < Ch^3$. For

notational ease, we take C to be some global constant of sufficient size. A similar expansion of $\psi(q_s(t), v_s(t))$ demonstrates that $\|\psi(q_c, v_c)\| < Ch^4$ and $\|\dot{\psi}(q_c, v_c)\| < Ch^3$. Multiplying the position component of the collocation constraint (5.3) by $J(q_c)$, and exploiting the bounds on these two values for the constraint velocity, $\dot{\phi}(q_c)$ and $\psi(q_c, v_c)$, we derive a bound on the velocity correction,

$$\|J(q_c)^T \bar{\gamma}_0\| < Ch^3. \quad (5.6)$$

Similarly, by substituting $\dot{v}_c = \bar{f}(x_c, u_c, \bar{\lambda}_0)$ into an expansion of $\alpha(q_c, v_c, u_c, \bar{\lambda}_0)$ and employing the bound on $\dot{\psi}(q_c, v_c)$, we have

$$\|\alpha(q_c, v_c, u_c, \bar{\lambda}_0)\| < Ch^3. \quad (5.7)$$

Simply put, (5.6) and (5.7) bound the defect between the derivative of the splines and the manifold tangent plane. To leverage existing results regarding collocation methods and ODEs, we extend the constrained dynamics by defining \dot{x} when x is off the manifold,

$$\dot{q}(t) = v(t) + J(q(t))^T \gamma(t) \quad (5.8)$$

$$\dot{v}(t) = f(x(t), u(t), \lambda(t)), \quad (5.9)$$

where the constraint forces are such that $J(q(t))\dot{q}(t) = 0$ and $\alpha(q(t), v(t), u(t), \lambda(t)) = 0$. Note that these extended dynamics agree with the constrained dynamics for states on the manifold, but define an ODE for all $x \in \mathbb{R}^n$. Using standard results in Hairer [44] and Betts [10], a direct collocation algorithm for these extended dynamics would have $\mathcal{O}(h^4)$ accuracy over a single timestep. Since (5.6) and (5.7) imply $\mathcal{O}(h^3)$ errors in extended dynamics when evaluated at the collocation point, this results in $\mathcal{O}(h^4)$ accumulated error over the interval $[0, h]$. \square

5.1.1 Friction Limits

By explicitly introducing the constraint forces λ as decision parameters within the optimization, we can easily require that they obey a set of nonlinear constraints as in [100]. For instance, we can require that they lie within the Coulomb friction cone with $\lambda_N \geq 0$ and $\mu\lambda_N^2 - \|\lambda_T\|_2^2 \geq 0$.

This is of particular interest when the rows of the Jacobian J are *not* linearly independent, a common case that occurs in all of the examples in this chapter. When J is full row-rank, one might directly solve for the unique λ such that $\alpha(q, v, u, \lambda) = 0$ and evaluate the friction constraints by solving a simple linear system of equations. However, when J is rank deficient, there are a subspace of such forces. Therefore, solving for a force that satisfies the constraints is equivalent to a convex optimization problem in and of itself. Explicit representation of the forces avoids this added complexity, and greatly simplifies the representation of these constraints.

5.1.2 Hybrid Collocation

As with other trajectory optimization algorithms, DIRCON can be simply extended to the hybrid case when the mode sequence is known. The hybrid trajectory optimization problem constructs one set of decision parameters and constraints per contact state, or hybrid mode. Consistency between the modes is enforced via a hybrid jump condition, described by explicitly including the impulse Λ . Letting z^j be the decision variables for the j th mode, the jump constraints are then:

$$q_1^j = q_{N^{j-1}}^{j-1} \tag{5.10}$$

$$v_1^j = v_{N^{j-1}}^{j-1} + H(q_{N^{j-1}}^{j-1})^{-1} J(q_{N^{j-1}}^{j-1})^T \Lambda^{j-1}, \tag{5.11}$$

Note that the manifold constraints for the new mode automatically imply post-impact conditions that constrain Λ . Additional constraints, defined by the gap function $\phi(q)$, prevent contact penetration and guard conditions to ensure that mode changes occur when the appropriate points are in contact.

5.2 Equality-Constrained LQR

Given a trajectory output from the collocation algorithm described in the previous section, we next address the problem of designing a tracking controller. The presentation here is similar in principle to [82], though we base the design around LQR. A powerful tool for the stabilization of both time-invariant and time-varying linear dynamical systems, LQR is also widely used for *local* stabilization of non-linear systems [2]. For a linear system, finite horizon LQR minimizes the quadratic cost,

$$x(T)^T Q_f x(T) + \int_0^T [x(t)^T Q x(t) + u(t)^T R u(t)] dt, \quad (5.12)$$

by solving the Hamilton-Jacobi-Bellman (HJB) equation. The product is an optimal controller $u(t) = -K(t)x(t)$ and the associated cost-to-go $V(t, x(t)) = x(t)^T S(t)x(t)$. To track a trajectory of a nonlinear system, a linearization of the dynamics about the nominal motion can be used to generate a feedback policy. Here, we provide a straight-forward extension of the classical notion of LQR to constrained dynamical systems. Consider the time-varying linear system

$$\dot{x} = A(t)x(t) + B(t)u(t), \quad (5.13)$$

where the dynamics constrain the state to the manifold defined by $F(t)x(t) = 0$ and $F(t)$ is full row-rank. While the derivations in this section apply to generic systems, for notational consistency, we will continue to focus on second-order plants with $F : \mathbb{R}^+ \rightarrow \mathbb{R}^{(2n-2m) \times 2n}$. The manifold constraint implies that the system is neither controllable nor stabilizable in the traditional senses. As a result, we cannot simply ignore $F(t)$ and solve the standard Riccati equation.

While we may not have a set of minimal coordinates, we can derive a time-varying basis for locally minimal coordinates and then apply traditional LQR techniques. Assume that $F(t)$ is differentiable and take some $P(0)$ to be an orthonormal basis of the nullspace of $F(0)$:

$$PP^T = I^{2m} \quad (5.14)$$

$$PF^T = 0^{2m \times (2n-2m)}. \quad (5.15)$$

To ensure that these identities hold for all time, we differentiate and write an ordinary differential equation for $P(t)$,

$$\begin{aligned}\dot{P}P^T + P\dot{P}^T &= 0 \\ \dot{P}F^T + P\dot{F}^T &= 0.\end{aligned}\tag{5.16}$$

For any $x \in \mathbb{R}^{2n}$, we can write $x = P^T y + F^T z$ for some $y \in \mathbb{R}^{2m}$ and $z \in \mathbb{R}^{2n-2m}$. However, as a result of the constraints, we know that $Fx(0) = z(0) = 0$. Additionally, along any trajectory $x(t)$, we have $\dot{z}(t) = 0$ and so $x(t) = P^T y(t)$ and $y(t) = Px(t)$. The dynamics of y are given by:

$$\dot{y} = \dot{P}x + P\dot{x} = \bar{A}y + \bar{B}u,\tag{5.17}$$

for $\bar{A} = \dot{P}P^T + PAP^T$ and $\bar{B} = PB$. We can apply classical LQR control techniques to this system in a two-step process. First, given $F(t)$, generate an appropriate $P(0)$ and then numerically integrate (5.16) to find $P(t)$. Note that some regularization of the ODE may be required to ensure that the solution does not drift from the identities (5.14)-(5.15). Second, use $P(t)$ to perform the change of coordinates from x to y . Solve the resulting Riccati equation and transform the solution back to the original coordinates.

The LQR solution from an individual mode can be projected through hybrid transitions using a linearization of the instantaneous impact dynamics, via the jump Riccati equation described in [81].

5.2.1 Example: Kinematic Constraint

We cast the case of a kinematic constraint $\phi(q)$ into the constrained LQR formulation. Given a nominal trajectory $q_0(t), v_0(t)$ that satisfies the constraints $\phi(q_0(t)) = 0$ and $\psi(q_0(t), v_0(t)) =$

0, linearize the dynamics about this trajectory:

$$\begin{aligned} q(t) &= q_0(t) + \tilde{q}(t), & v(t) &= v_0(t) + \tilde{v}(t) \\ \dot{\tilde{q}}(t) &= \tilde{v}(t) \\ \dot{\tilde{v}}(t) &= \tilde{A}(t) \begin{bmatrix} \tilde{q}(t) \\ \tilde{v}(t) \end{bmatrix}. \end{aligned}$$

Linearizing the constraint, and suppressing the dependence of q_0 and v_0 on time, we get

$$\begin{bmatrix} \phi(q) \\ \psi(q, v) \end{bmatrix} \approx \begin{bmatrix} J(q_0) & 0 \\ \frac{dJ(q_0)}{dt} & J(q_0) \end{bmatrix} \begin{bmatrix} \tilde{q}(t) \\ \tilde{v}(t) \end{bmatrix} = F(t) \begin{bmatrix} \tilde{q}(t) \\ \tilde{v}(t) \end{bmatrix} \quad (5.18)$$

which gives $F(t)$ as a kinematic function of nominal trajectory. Since we require $F(t)$ to be full rank, but $J(q)$ will often be rank deficient (though constant rank), it is necessary, but straightforward, to extract a full rank basis for J and its time derivative.

5.3 QP Feedback Controller

Kuindersma et al. [61] introduced a QP-based framework for combining an LQR policy with the unilateral constraints due to input and friction limits. To evaluate the results of DIRCON and the constrained LQR approaches, we utilize a similar approach and formulate a quadratic program. We summarize here the formulation of [61, 62], applied in our setting. Given a planned nominal trajectory $x_0(t), u_0(t)$ and LQR solution $S(t)$, we write:

$$\begin{aligned} \underset{u, \beta}{\text{minimize}} \quad & \tilde{u}^T R \tilde{u} + 2\tilde{x}^T S(A\tilde{x} + B\tilde{u}) \\ \text{subject to} \quad & H\dot{v} + C = Bu + J_\beta^T \beta \\ & \dot{J}v + J\dot{v} = 0 \\ & u_{\min} \leq u \leq u_{\max} \\ & \beta \geq 0, \end{aligned} \quad (5.19)$$

where $\tilde{x} = x - x_0$, $\tilde{u} = u - u_0$ and we have suppressed dependence on time and state. The cost function is derived from the HJB equation for the time-varying LQR system. Therefore, in the absence of unilateral constraints, the QP solution is equivalent to the optimal LQR input. The decision variables β are force coefficients that multiply a set of generating vectors that define a polyhedral approximation to the friction cone, $\lambda_j = \sum_{i=1}^{N_d} \beta_{ij} w_{ij}$. This reparameterization of the contact force results in a less variable set of active constraints, reducing computation time as the QP is solved repeatedly. The contact points included in β are determined at each control step, and any point in contact will be added to the QP (whether or not it is originally planned), giving the system the opportunity to use environmental forces to correct deviations from the desired trajectory. As compared with [61, 62], the LQR cost-to-go used here is derived using the full, whole-body robot dynamics, rather than the dynamics associated with a simplified model.

5.4 Example Applications

The components above are tested on three examples related to robotic locomotion. The algorithms were implemented in MATLAB within the Drake planning and control toolbox [133]. Trajectory optimizations were solved with the SNOPT toolbox [39]. Depending on complexity, the trajectory optimization and LQR components were solved offline on a desktop computer within ten minutes to two hours. The QP controller was solved at real-time rates during simulation. Highlights from the experiments here are shown in figures below.

5.4.1 Underactuated planar biped

We first demonstrate the approach on underactuated planar biped, where each leg has a degree of freedom in the hip, knee, and ankle. The hips and knees are actuated, but the ankle consists solely of a passive spring and damper. With a back joint and the planar floating base, this model has an 20-dimensional state space. Contact points are modeled at the toe and heel of each foot. The mass properties are similar to those of the Atlas robot [62], and the ankle spring and damping coefficients are 10 Nm/rad and 2 Nms/rad. To produce limit cycle walking, a hybrid trajectory optimization was executed with a contact sequence

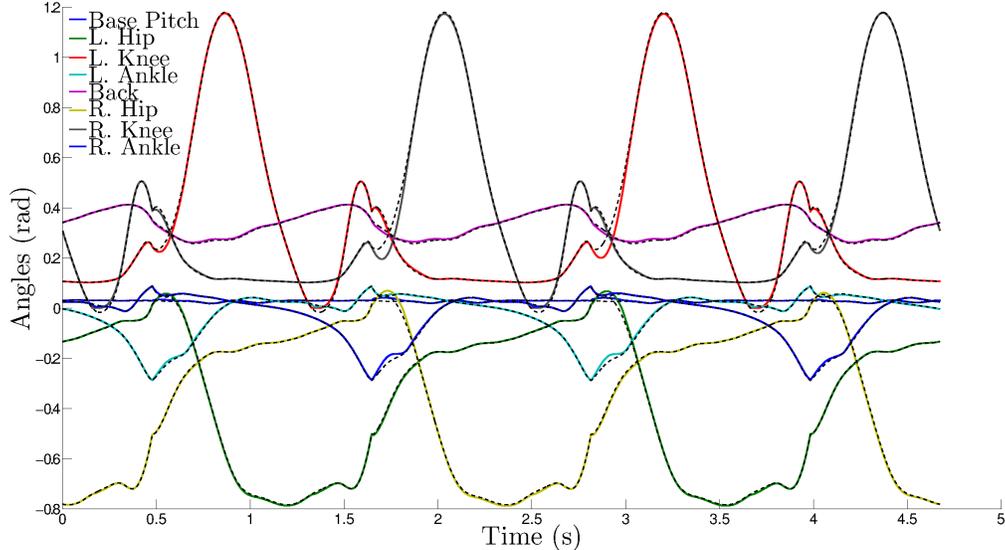


Figure 5-4: Joint angle and body pitch tracking error along four steps of the underactuated walker. The nominal trajectory is shown in the dashed lines and the executions are in the solid lines. Tracking error is worst shortly following the impacts (where the trajectories are not differentiable).

containing both single and double support phases. The objective was to minimize effort, a quadratic penalty on control input $u^T u$. Linear constraints were imposed on x_1 and x_N to produce a periodic motion and a minimum average walking speed. A small penalty was added to acceleration, $10^{-4} \sum_k \|\bar{f}(x_k, u_k, \lambda_k)\|^2$, to encourage smoothness in the solution. Additional constraints on the foot position enforced some amount of swing clearance. For the LQR component, simple, diagonal matrices were used for both Q and R . Elements of Q were 100 and 1 for the generalized positions and velocities respectively, while the diagonal R was uniformly 0.01.

Figure 5-4 shows body pitch and joint angle tracking over four steps. Overall, the controller is able to closely track the nominal motion, with deviations most noticeable shortly after impacts with the ground. One of the aims of this trajectory optimization is to create motions that are both dynamic and efficient. Mechanical cost of transport (COT) serves as a useful metric for locomotion efficiency: if M is the mass and d is the total distance traveled, we integrate the total joint work done and the unitless cost of transport is $\text{COT} = \frac{1}{Mgd} \int |\text{work}| dt$. While we did not explicitly minimize COT, minimizing effort produced an efficient nominal gait with a COT of 0.139. Execution of the trajectory should

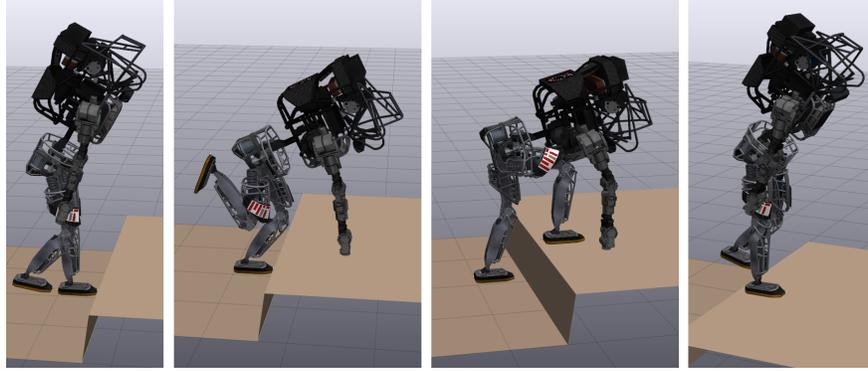


Figure 5-5: A sequence of states from the executed trajectory as the robot uses its arm to help climb up onto the step in less than 3 seconds.

increase this cost, as the controller must expend energy to eliminate error. However, as an indication of the accuracy of the nominal motion, the executed COT over the four steps in the figure was only 0.143, a marginal increase.

5.4.2 Multi-contact climbing

We examine a planar humanoid model where the biped from the previous example has been augmented with a two degree of freedom arm (shoulder and elbow joints), also based off the Atlas. Note that the previously used back joint has been eliminated for simplicity and the ankle joints are actuated, giving the plant a 22-dimensional state space. A single contact point is included at the end of the arm, for a total of five possible contacts. The restrictive joint limits of the physical Atlas robot have been relaxed to allow greater flexibility for this motion. The trajectory optimization was constrained to use both the hand and feet to climb a 0.3 meter step and then reach a stable position using a sequence of five different contact modes. As before, constraints were used to enforce swing clearance and the objective was to minimize effort. The costs used for LQR are identical in nature to those from the walking example. The duration of the resulting trajectory was less than 3 seconds, so the robot must move quickly and dynamically to execute it successfully. Figure 5-5 shows a set of illustrative key-frames from the motion.

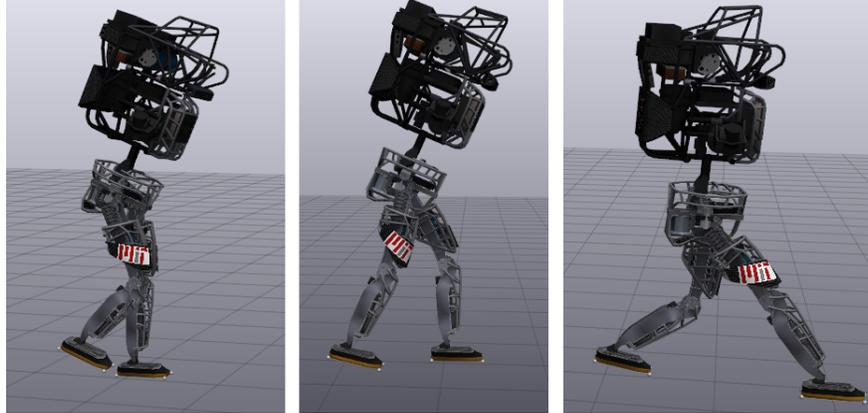


Figure 5-6: A sequence of states as the robot walks and then executes a sudden halting maneuver. The leftmost two images illustrate phases of the walking motion and the rightmost image shows the final state after the rapid stop.

5.4.3 3D biped

The final example is a biped walking in three dimensions along flat terrain. Also based upon the Atlas robot, we use a model with six degrees of freedom in each leg: three at the hip, one at the knee, and two at the ankle. Including the floating base, the model has a 36-dimensional state space with eight total contact points at the corners of the feet. As with the underactuated biped, we synthesize limit cycle walking with both single and double support phases. The objective was to minimize effort, and linear constraints enforced a periodic motion while walking at a human-like speed of over 1 m/s. Locomotion at this speed requires continuous, dynamic motion and the planned motion utilizes push-off from the ankle during double support.

To illustrate the capability to produce and execute rapid, aperiodic motions, we further synthesized a trajectory that brought the robot to a complete halt within a half a stride (starting from mid-swing). To stop this quickly, the robot must quickly propel its swing leg forward before coming to rest with its forward foot and rear toe in contact with the ground. For the LQR component, simple, diagonal matrices were again used for both Q and R . Components of Q were 200 and 1.5 for the generalized positions and velocities respectively, while the diagonal R was uniformly 0.01.

We note that, when executed in simulation, the periodic gait was not stabilized over an infinite horizon as small tracking errors in the footfall locations and timings caused eventual

instability. Over shorter distances, however, the controller produces efficient walking at human speeds. Key-frames of the stopping maneuver are shown in Figure 5-6. As evidence of the accuracy of the nominal trajectory and the efficiency of the feedback policy, the calculated COT of the executed trajectory was 0.399, only slightly larger than the COT of the nominal motion, 0.382.

To examine the response of the closed-loop controller to disturbances, randomly oriented $10N\cdot s$ impulses were applied, mid-stance, at the pelvis of the robot and a single walking step was simulated. The impulse causes a roughly 10% deviation in the center of mass velocity and substantial joint velocity errors. The LQR cost-to-go after the disturbance and after one step was used as a measure of robustness. After the disturbance, the median cost-to-go from 300 trials was 0.39 and, after a single step, the controller had reduced the median cost-to-go to 0.07. Some of the random disturbances did cause falls or other instabilities. In total, the controller was able to reduce the cost-to-go in 96% of the trials, empirically demonstrating some level of robustness.

5.5 Discussion

The most common trajectory optimization approaches utilize tools from general nonlinear optimization and are therefore sensitive to the choice of the initial, or seed, value for the unknown parameters. As an indication of the robustness of the DIRCON method, all of the optimizations in Section 5.4 were initialized with exceedingly simple trajectories: a constant, nominal pose for the states and white noise for the control inputs. Even without carefully chosen seed values, the optimizations consistently converged to high quality solutions for problems of significant size.

While we are able to execute motions over finite horizons, the walking trajectories discussed above are not stable over an infinite sequence of steps. Perturbations around the moment of impact cause slight mismatches in footfall timing and location, eventually leading to a fall. One potential solution to this issue is to eliminate the explicit dependence of the controller on time, either via use of transverse coordinates [81] or through a zero dynamics manifold [122].

In this chapter, we have presented a general purpose, end-to-end approach for synthesis and stabilization of optimal trajectories for robotic systems in contact with their environment. These contacts restrict motion of the robot to manifolds of feasible states, which can have complex geometries in multi-contact scenarios. By explicitly addressing the nature of these constraints, we design methods that seamlessly handle both non-minimal coordinates and underactuated dynamics, both of which present problems for many existing algorithms. The DIRCON algorithm is efficient, robust to initial seeds, and exhibits cubic integration accuracy. Use of lower order methods complicates the task of stabilization and can result in trajectories that do not accurately represent the true cost. As evidenced by the examples above, the combined LQR and QP control is capable of closely tracking these dynamic motions in terms of both state and control effort.

Chapter 6

Stability Verification and Control Design

In this chapter, we present a method, published in [99, 102], for applying sums-of-squares verification to rigid bodies with Coulomb friction undergoing discontinuous, inelastic impact events. While methods based on sums of squares for numerical computation of Lyapunov certificates are a powerful tool for analyzing the stability of continuous nonlinear systems, prior methods do not scale to systems with multiple potential contacts. The proposed algorithm scales tractably in the number of contacts and explicitly generates Lyapunov certificates for stability, positive invariance, and safety over admissible (non-penetrating) states and contact forces. Building off the measure differential inclusion formulation of contact, described in 3.2.3, we naturally generate semialgebraic constraints that define this admissible region. The presented algorithms can additionally be used to automatically synthesize stabilizing feedback controllers through contact events. The approach is demonstrated on multiple robotics examples, including simple models of a walking robot, a perching aircraft, and control design of a balancing robot.

6.1 Definitions

We introduce some background definitions related to measure differential inclusions and to stability analysis. Recall the definition of the admissible set \mathcal{A} in (3.6). First, we will focus on MDIs which are *consistent* (see [71], Ch. 4).

Definition 6.1. *A measure differential inclusion is consistent if every solution defined for*

t_0 is defined for almost all $t \geq t_0$, all such solutions remain within \mathcal{A} , and for each $x_0 \in \mathcal{A}$ there exists at least one solution passing through x_0 .

An *equilibrium point* for such a system is defined as any point $x_0 \in \mathcal{A}$ such that $x(t) = x_0$ is a solution. As is typical for models with dry friction, we do not expect to have unique solutions from the systems covered by this work [13, 127]. However, for systems governed by MDIs, there are natural extensions to the notions of stability and positive invariance ([71] Ch. 6).

Definition 6.2. *An equilibrium point $x_0 \in \mathcal{A}$ of a consistent MDI is stable in the sense of Lyapunov if, for each $\epsilon > 0$, there exists a $\delta > 0$ such that every solution $x(t)$ satisfying $|x_0 - x(t_0)| < \delta$ satisfies $|x_0 - x(t)| < \epsilon$ for almost all $t \geq t_0$.*

Definition 6.3. *A set $B \subset \mathcal{A}$ is positively invariant if each solution $x(t)$ satisfying $x_-(t_0) \in B$ satisfies $x(t) \in B$ for almost all $t \geq t_0$.*

In order to apply Lyapunov analysis to MDIs, we make note of the following fact (see [71], Proposition 6.3): if a function $V : D \rightarrow \mathbb{R}$ is a continuously differentiable function on a compact set $D \subset \mathbb{R}^{2n}$, and $x(t)$ is of locally bounded variation (LBV), then $V(x(t))$ is also LBV and

$$V(x_+(t)) - V(x_-(t_0)) = \int_{t_0}^t \frac{\partial V}{\partial x} \dot{x}(\tau) d\tau + \int_{t_0}^t V(x_+(\tau)) - V(x_-(\tau)) d\eta(\tau), \quad (6.1)$$

where $\dot{x}(t) = [v(t)^T \dot{v}(t)^T]^T$, as in (3.4) and (3.5). For the remainder of this paper, when we write $dV(x) \leq 0$ for certain $x \in A$ we mean that, for any solution satisfying $x_-(t) = x$, we have $\frac{\partial V}{\partial x} \dot{x}(t) \leq 0$ and $V(x_+(t)) - V(x_-(t)) \leq 0$.

6.2 Conditions for Stability

The highly-structured nature of rigid-body dynamics and the complementarity formulation of contact allow us to construct semialgebraic conditions for stability in the sense of Lyapunov and positive invariance.

6.2.1 Lyapunov Conditions for MDIs

We begin by describing sufficient conditions for stability in the sense of Lyapunov and positive invariance, stated in terms of Lyapunov functions. Recall that a function $\alpha : [0, \infty) \rightarrow [0, \infty)$ belongs to class \mathcal{K} if it is strictly increasing and $\alpha(0) = 0$. The following theorem is adapted from [71], Theorem 6.23, and is stated without proof.

Theorem 6.4. *Let x_0 be an equilibrium point for a consistent MDI and let $V : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ be a continuously differentiable function. If there exists a neighborhood U of x_0 and a class \mathcal{K} function α such that $x \in U \cap \mathcal{A}$ implies $dV \leq 0$ and $V(x) \geq \alpha(\|x - x_0\|)$ then x_0 is stable in the sense of Lyapunov.*

Given a candidate Lyapunov function $V(q, v)$, define the c -sublevel set

$$\Omega_c = \{(q, v) \in \mathbb{R}^{2n} \mid V(q, v) < c\}.$$

For a system whose solutions are continuous functions of time, $dV \leq 0$ on $\Omega_c \cap \mathcal{A}$ would be sufficient to show that each connected component of $\Omega_c \cap \mathcal{A}$ is positively invariant. However, where $v(t)$ is discontinuous, the pre- and post-impact states may lie in disjoint connected components. Furthermore, for systems where some region of state space is inadmissible with $\phi(q) < 0$, we do not necessarily expect regionally valid Lyapunov functions to be globally convex. An example of such a Lyapunov function is later illustrated in Figure 6-2b. This necessitates the focus on individual connected components of Ω_c . The following theorem provides stronger conditions which guarantee positive invariance of such a connected component.

Theorem 6.5. *Let $V : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ be a continuously differentiable function, and \mathcal{C} be a connected component of $\Omega_c \cap \mathcal{A}$ with $dV \leq 0$ on \mathcal{C} . Then \mathcal{C} is positively invariant if, for every solution $(q(t), v(t))$ with $(q(t), v_-(t)) \in \mathcal{C}$, there exists a path $\bar{v} : I \rightarrow \mathbb{R}^n$ for some interval $I \subset \mathbb{R}$ from $v_-(t)$ to $v_+(t)$, such that $V(q(t), \bar{v}(s))$ is a non-increasing function of s .*

Proof. Fix a solution $x(t) = (q(t), v(t))$ with $x_-(t_0) \in \mathcal{C}$. Let

$$\bar{\tau} = \sup\{\tau \in \mathbb{R} : x_-(t) \in \mathcal{C} \quad \forall t \in [t_0, \tau]\}.$$

Suppose, for contradiction, that $\bar{\tau}$ is finite. We have $x_-(\bar{\tau}) \in \mathcal{C}$ as it is the limit of a sequence in a connected component. The function $s \mapsto (q(\bar{\tau}), \bar{v}(s))$ provides a path from $x_-(\bar{\tau})$ to $x_+(\bar{\tau})$. This path lies in Ω_c as $dV \leq 0$ on \mathcal{C} implies $V(x_-(\bar{\tau})) \leq V(x_-(t_0)) < c$ and, by assumption, $V(q(\bar{\tau}), \bar{v}(s))$ is a non-increasing function of s . The path lies in \mathcal{A} as only the generalized velocities vary (recall our definition of \mathcal{A} in (3.6)). Thus $x_+(\bar{\tau})$ belongs to \mathcal{C} . As $V(x_+(\bar{\tau})) < c$ and V is continuous, there exists an $r > 0$ such that:

$$U_r = \{x \in \mathbb{R}^{2n} : \|x - x_+(\bar{\tau})\|_\infty < r\}$$

is contained in Ω_c (where $\|\cdot\|_\infty$ is the maximum norm). As $x_+(\bar{\tau})$ is a right limit, there exists an $\epsilon > 0$ such that $x(t) \in U_r$ for $t \in (\bar{\tau}, \bar{\tau} + \epsilon)$.

We now show that $x(t) \in \mathcal{C}$ for almost all $t \in (\bar{\tau}, \bar{\tau} + \epsilon)$, contradicting the definition of $\bar{\tau}$. Fix t such that $v(t)$ is defined and examine the following functions:

$$\begin{aligned} \sigma_1 &\mapsto (q(\sigma_1), v_+(\bar{\tau})), \\ \sigma_2 &\mapsto (q(t), \sigma_2 v(t) + (1 - \sigma_2)v_+(\bar{\tau})), \end{aligned}$$

the first defined for $\sigma_1 \in [\bar{\tau}, t]$ and the second for $\sigma_2 \in [0, 1]$. Since $q(\cdot)$ is continuous, both functions are also continuous. We see the range of both maps lie in \mathcal{A} as $\{q(t)\} \times \mathbb{R}^n \subset \mathcal{A}$ for all $t \geq t_0$. We see the ranges of the functions also lie in U_r : the first lies in U_r as $(q_1, v_1), (q_2, v_2) \in U_r$ implies $(q_2, v_1) \in U_r$ and the second as U_r is convex. Together, these functions provide a path from $x_+(\bar{\tau})$ to $x(t)$ that lies in $\Omega_c \cap \mathcal{A}$, thus $x(t) \in \mathcal{C}$. \square

Theorem 6.5 holds for general Lyapunov functions and systems where the unilateral constraints are defined by the generalized positions q . While we are most often interested in systems with friction, we briefly consider the special structure implied by rigid-body dynamics and frictionless, inelastic collisions. The following proposition shows that for such systems, and if V is a convex function in v for each fixed q , the above sufficient condition for positive invariance is also necessary. In particular, no additional conservatism is added by requiring \bar{v} to be the chord connecting $v_-(t)$ to $v_+(t)$.

Proposition 6.6. *For a rigid-body system undergoing frictionless, inelastic impacts, let*

$V : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ be a continuously differentiable function, and \mathcal{C} be a connected component of $\Omega_c \cap \mathcal{A}$ such that $dV \leq 0$ on \mathcal{C} . If V is convex in v for each fixed q , then the following conditions are equivalent:

(i) \mathcal{C} is positively invariant.

(ii) For each solution $(q(t), v(t))$, when $(q(t), v_-(t)) \in \mathcal{C}$, $V(q(t), \bar{v}(s))$ is non-increasing along the path $\bar{v}(s) = sv_+(t) + (1-s)v_-(t)$ for $s \in [0, 1]$

Proof. That (ii) implies (i) is the content of Theorem 6.5. Now assume (i) holds and fix a solution $(q(t), v(t))$ with $(q(t), v_-(t)) \in \mathcal{C}$. For convenience, let q, v_+ , and v_- denote $q(t), v_+(t), v_-(t)$. Take the path $\bar{v}(s) = sv_+ + (1-s)v_-$. Since V is convex in v and $dV \leq 0$, we know

$$V(q, v_-) \geq (1-s)V(q, v_-) + sV(q, v_+) \geq V(q, \bar{v}(s)),$$

so that the chord $(q, \bar{v}(s))$ lies in Ω_c , and clearly lies in \mathcal{A} .

We show that $\frac{dV(q, \bar{v}(s))}{ds} \leq 0$. Observe that $\{(q, \bar{v}(s)) : s \in [0, 1]\}$ are all possible pre-impact states since $\phi(q) = 0$ and the impact conditions $J_N v_- < 0$ and $J_N v_+ = 0$ imply that $J_N \bar{v}(s) < 0$. Let Λ_N be a feasible impulse such that $v_+ = v_- + H^{-1} J_N^T \Lambda_N$. Substituting into the definition of $\bar{v}(s)$,

$$\bar{v}(s) = v_- + sH^{-1} J_N^T \Lambda_N.$$

Since the constraints on Λ_N are linear, we know that the impulse $(1-s)\Lambda_N > 0$ will also be feasible. Applying this impulse to $(q, \bar{v}(s))$, we get the post-impact velocity

$$\bar{v}_+(s) = \bar{v}(s) + (1-s)H^{-1} J_N^T \Lambda_N = v_+.$$

And so v_+ is a possible post-impact velocity from an impact at any point along the chord. Since $dV \leq 0$, we then know $V(q, \bar{v}(s)) \geq V(q, v_+)$. This implies that the minimum of V along the chord is achieved at (q, v_+) and, since V is convex, the derivative along the chord must be non-positive. \square

Remark 6.7. Note that the proof above for (i) implying (ii) fails for contacts with friction.

The Routh path from pre- to post-impact states is no longer a single line segment, and so the set of pre-impact states along the chord do not all have the same post-impact state. The frictionless assumption does, however, cover a class of interesting problems including collisions due to impacting hard joint limits.

6.2.2 Conditions For Complementarity Systems

We now focus on Lagrangian mechanical systems with impacts and friction described by complementarity conditions. This section contains sufficient conditions for demonstrating $dV \leq 0$ and the additional constraint on jump discontinuities in the statement of Theorem 6.5. We partition the admissible set \mathcal{A} into three disjoint sets: \mathcal{F} , \mathcal{I} , and \mathcal{U} .

$$\mathcal{F} = \{(q, v) \in \mathcal{A} : \phi_i(q) = 0 \Rightarrow J_{N,i}(q)v > 0 \ \forall i \in \{1, \dots, m\}\}$$

$$\mathcal{I} = \{(q, v) \in \mathcal{A} : \exists i \in \{1, \dots, m\} \ \phi_i(q) = 0, J_{N,i}(q)v < 0\}$$

$$\mathcal{U} = \{(q, v) \in \mathcal{A} \setminus \mathcal{I} : \exists i \in \{1, \dots, m\} \ \phi_i(q) = 0, J_{N,i}(q)v = 0\}$$

Intuitively, \mathcal{F} is the region where there is no contact or all contacts are being broken, so all the contact forces must vanish. On \mathcal{F} , we know that $\lambda = \Lambda = 0$ and so that $dV \leq 0$ is equivalent to

$$\nabla V(q, v)^T \begin{bmatrix} v \\ -H^{-1}C \end{bmatrix} \leq 0. \quad (6.2)$$

On \mathcal{U} , there may be forces due to contact, and the condition on continuous evolution is

$$\nabla V(q, v)^T \begin{bmatrix} v \\ H^{-1}(-C + J_N^T \lambda_N + J_T^T \lambda_T) \end{bmatrix} \leq 0. \quad (6.3)$$

On \mathcal{I} , there must be a collision for a trajectory to remain within \mathcal{A} . Additionally, consistency may also require collisions on \mathcal{U} (for a detailed explanation of this, see the discussion of Painlevé's paradox in [127]). We now provide conditions on V and a path, implicitly defined, for each jump discontinuity such that the requirements of Theorem 6.5 are satisfied. In the frictionless case, this path corresponds directly to the path in Proposition 6.6. Recall that

the Routh method of 3.2.1 constructs a path through the space of contact impulses. We take $\bar{v}(s)$ to be the velocities defined by the Routh method, where s is the path parameter varying the impulses. As \bar{v} depends linearly on the forces, we can find the derivative of \bar{v} with respect to s , defined almost everywhere along each path segment:

$$\frac{d\bar{v}(s)}{ds} = H^{-1}(J_{N,i}^T \Lambda'_{N,i} + J_{T,i}^T \Lambda'_{T,i}),$$

where $\Lambda'_{N,i}$ and $\Lambda'_{T,i}$ satisfy the Coulomb friction conditions. To show V is non-increasing along the path, we require

$$\left. \frac{\partial V(q, v)}{\partial v} \right|_{(q, \bar{v}(s))} H^{-1}(J_{N,i}^T \Lambda'_{N,i} + J_{T,i}^T \Lambda'_{T,i}) \leq 0. \quad (6.4)$$

Since the Routh method for resolving impacts is memoryless, any point $(q, \bar{v}(s))$ is also a possible pre-impact state. So, the set of all possible $(q, \bar{v}(s))$ is precisely equivalent to $\mathcal{A} \setminus \mathcal{F}$ and it is equivalent to enforce (6.4) for all $(q, v) \in \mathcal{A} \setminus \mathcal{F}$ instead of along potential paths. This constraint must hold for all i , so we construct a single condition that encompasses all contact points:

$$\frac{\partial V(q, v)}{\partial v} H^{-1}(J_N^T \Lambda'_N + J_T^T \Lambda'_T) \leq 0. \quad (6.5)$$

Both (6.3) and (6.5) are defined in terms of permissible contact forces λ and slopes of the impulse path Λ' when resolving collisions. Conditions can be used to describe the set of feasible contact normal forces [13, 127, 71]:

$$0 \leq \phi_i \perp \lambda_{N,i} \geq 0, \quad (6.6)$$

$$(J_{N,i} v) \lambda_{N,i} \leq 0. \quad (6.7)$$

These constraints prohibit contact at a distance and ensure that the contact normal is a compressive and dissipative force. Note that (6.6-6.7) apply not only to the continuous force λ , but they also describe the set of feasible impulse slopes Λ' . Observing that the friction

constraints on both are also identical, we write the additional set of constraints:

$$(J_{T,i}v)\lambda_{T,i} \leq 0, \quad (6.8)$$

$$\mu^2\lambda_{N,i}^2 - \lambda_{T,i}^2 \geq 0, \quad (6.9)$$

$$(\mu^2\lambda_{N,i}^2 - \lambda_{T,i}^2)(J_{T,i}v) = 0. \quad (6.10)$$

Here, we describe the proper nonlinear friction cone and slightly diverge from the standard linear complementarity description of Coulomb friction. However, for any (q, v) , this full set of conditions is exactly equivalent to our formulations of both frictional, inelastic collisions and Coulomb friction.

6.2.3 Separability of Contacts

We now have three separate positivity conditions for stability. We require (6.2) to hold on \mathcal{F} , (6.3) on \mathcal{U} , and (6.5) on $\mathcal{U} \cup \mathcal{I}$, with the contact forces and impulses subject to (6.6) – (6.10). One issue with this formulation is that the $\mathcal{O}(m)$ contact force terms, when appearing together, will significantly increase the size of verification programs we formulate in Section 6.3, detailed in Section 6.3.5. However, the structure in the problem leads to a significant reduction in complexity. For this, we require an additional assumption: that the contact surfaces themselves are distinguishable from one another. This rather benign assumption is satisfied by most rigid-body systems of interest, even including situations where “jamming” contact may occur.

Assumption 6.8. *For all $i \in \{1, \dots, m\}$ and for any (q, v) where $\phi_i(q) = 0$ and $J_{N,i}v \leq 0$, there exists a sequence $(q_k, v_k) \rightarrow (q, v)$ where*

1. $\phi_i(q_k) = 0$ and $J_{N,i}v_k \leq 0$ and
2. $\phi_j(q_k) > 0$ or $J_{N,j}v_k > 0$ for all $j \neq i$.

Theorem 6.9. *Given Assumption 6.8, $dV \leq 0$ is equivalent to requiring (6.2) to hold on \mathcal{F} , and*

$$\frac{\partial V(q, v)}{\partial v} H^{-1}(J_{N,i}^T + J_{T,i}^T \lambda_{T,i}) \leq 0, \forall i = 1, \dots, m \quad (6.11)$$

whenever the following conditions hold:

$$\phi_i = 0, \quad (6.12)$$

$$J_{N,i} v \leq 0, \quad (6.13)$$

$$(J_{T,i} v) \lambda_{T,i} \leq 0, \quad (6.14)$$

$$\mu^2 - \lambda_{T,i}^2 \geq 0, \quad (6.15)$$

$$(\mu^2 - \lambda_{T,i}^2)(J_{T,i} v) = 0, \quad (6.16)$$

$$\phi_j \geq 0, \forall j \neq i. \quad (6.17)$$

Proof. First, observe that the original conditions on λ and Λ are identical, and they appear in distinct conditions and constraints, so we may combine them and treat the two simply as λ . To show that these conditions are sufficient is straightforward, and does not require Assumption 6.8. Since (6.6) – (6.10) and (6.5) are homogeneous in $(\lambda_{N,i}, \lambda_{T,i})$, we may fix $\lambda_{N,i} = 1$, so (6.12) – (6.17) is equivalent to (6.6) – (6.10). Summing (6.11) for all i shows that (6.11) is a sufficient condition for (6.5). By continuity, since (6.2) holds on \mathcal{F} , it also holds on \mathcal{U} . By summing (6.2) with (6.11), we have also have sufficient conditions for (6.3) on \mathcal{U} .

(6.11) – (6.17) may seem stricter than the original formulation when multiple contacts are active. However, by Assumption 6.8, if (6.5) holds in any state (q, v) , then for each active contact, (6.11) must hold on some sequence converging to (q, v) and, by continuity, must also hold at (q, v) . \square

Remark 6.10. *Theorem 6.9 provides a reduced set of conditions where each condition depends on at most one contact force indeterminate, $\lambda_{T,i}$. These concise conditions will be used throughout the remainder of this chapter to generate stability certificates. Furthermore, (6.12) – (6.17) restrict $\lambda_{T,i}$ to be within a compact region, whereas, by the original constraints,*

the forces are unbounded since λ_N could be scaled arbitrarily. Not only is it practically useful that the feasible set be compact, with respect to numerical tolerances, but compactness also has theoretical consequences for SOS-based methods like the Positivstellensatz [109].

6.2.4 Extension to Three Dimensions

For clarity of presentation, we have focused this discussion on the planar case. While the examples in this work treat two dimensions, the extension to the full three dimensional case, where $\lambda_{T,i} \in \mathbb{R}^2$ and $J_{T,i} \in \mathbb{R}^{2 \times n}$, is straightforward and is presented here without proof. Replace conditions (6.14) – (6.16) with

$$(J_{T,i}v) \circ \lambda_{T,i} \leq 0, \quad (6.18)$$

$$\mu^2 - \|\lambda_{T,i}\|^2 \geq 0, \quad (6.19)$$

$$(\mu^2 - \|\lambda_{T,i}\|^2)(J_{T,i}v) = 0, \quad (6.20)$$

$$[\lambda_{T,i}]_1[J_{T,i}v]_2 - [\lambda_{T,i}]_2[J_{T,i}v]_1 = 0, \quad (6.21)$$

where we take \circ to represent the Hadamard product and $[x]_i$ is the i th element of vector x . The first and third of these constraints are vector-valued and are treated elementwise. Note that the fourth constraint is new to the three dimensional case, and is used to ensure that the frictional force exactly opposes the direction of motion.

6.2.5 Semialgebraic Representation

To apply tools from algebraic geometry, like SOS programming, it is important that the constraints above be expressible as polynomials. While Taylor approximation of the preceding conditions can always be used, rigid-body dynamics and the manipulator equations offer a great deal of structure that we can exploit to make the problems of control and verification especially amenable to algebraic methods. For many rigid-body systems, especially those of interest in robotics, trigonometric substitutions can reduce the task of kinematics to an algebraic problem [141]. Concretely, for rotational joints, substituting new indeterminates c_i and s_i for $\cos(q_i)$ and $\sin(q_i)$ respectively, with the constraint that $s_i^2 + c_i^2 = 1$, will of-

ten result in polynomial kinematics in c_i and s_i . Prismatic (or translational) joints, require no such substitution and naturally result on polynomial kinematics. Most common robotic joints can be represented as a sequence of such rotational and prismatic transformations (see [120], Sec. 1.4), and so this polynomial representation can be easily formed. Helical joints, however, are a notable exception, since a single helical joint creates both rotational and translational motion. For simple contact surfaces, such as between a point contact and the ground, the gap function and Jacobians ϕ , J_N , and J_T are kinematic functions, and thus polynomial in c_i , s_i and the remaining translational coordinates of q .

Similarly, the various terms of the manipulator equations and constraints (H , C , and B) are also polynomial in the same position coordinates and the standard velocity vector v (see [120], Ch. 2). Several methods can be used to accommodate the appearance of $H(q)^{-1}$ in the conditions of the previous section. First we note that by explicitly introducing an additional indeterminate variable $\dot{v} \in \mathbb{R}^n$, the condition (3.1) is algebraic in \dot{v} , v , λ , the translational components of q and any introduced trigonometric variables. Alternatively, as $H(q)$ is positive definite and polynomial, its inverse is a rational function, where the denominator is the $\det(H)$, and thus strictly positive. Therefore, we can find equivalent conditions by multiplying by the denominator. These facts imply that semialgebraic conditions can be posed that are equivalent to those in 6.2.2.

6.3 Verification Algorithms

For our systems of interest, the Lyapunov conditions in 6.2 amount to non-negativity constraints on polynomials over basic semialgebraic sets. This formulation is amenable to the SOS-based techniques introduced in 3.4, which provide certificates that a polynomial can be written as a sum of squares of polynomials. Searching over polynomials which satisfy these sufficient conditions can be cast as an SDP, allowing for the application of modern convex optimization tools. For the examples in this chapter, we use the YALMIP [74, 75] and SPOT [83] toolboxes to generate programs for the semidefinite solvers SeDuMi [130] and MOSEK [90]. For a portion of our approach, we exploit the bilinear alternations described in 3.4.3. Note that in 7.3.1.1, we improve upon the alternation scheme used in this chapter.

6.3.1 Global Verification

For some dynamic systems, we can verify the Lyapunov conditions over the entire admissible set. Define \mathcal{D}_i to be the set of all (q, v, λ_i) that satisfy conditions (6.12) – (6.17). Note that this also implies $(q, v) \in \mathcal{A}$. If $(0, 0)$ is an equilibrium of the system, we can then pose the global feasibility SOS program:

$$\begin{aligned}
 & \text{find } V(q, v) && (6.22) \\
 & \text{subj. to } V(0, 0) = 0, \\
 & V(q, v) \geq \alpha(\|q\| + \|v\|) && \text{for } (q, v) \in \mathcal{A}, \\
 & \nabla V^T \begin{bmatrix} v \\ \dot{v} \end{bmatrix} \leq 0 && \text{for } (q, v) \in \mathcal{A}, \\
 & \frac{\partial V}{\partial v} H^{-1}(J_{N,i}^T + J_{T,i}^T \lambda_{T,i}) \leq 0 && \text{for } (q, v, \lambda_i) \in \mathcal{D}_i,
 \end{aligned}$$

with given $\alpha(\cdot)$ in class \mathcal{K} . SOS allows us to search over a family of polynomial Lyapunov functions via SDP, thus verifying that every sublevel set of V is positively invariant and that the origin is stable in the sense of Lyapunov. This certificate of a nested set of invariant regions is weaker than asymptotic stability but stronger than invariance of a single set.

6.3.2 Regional Verification

For many problems of interest, we would like to maximize the verified region about an equilibrium. Specifically, we aim to find a Lyapunov function that maximizes the volume of a connected component $\mathcal{C} \subseteq \Omega_1 \cap \mathcal{A}$, which is positively invariant and, for all $\rho \leq 1$, $\mathcal{C} \cap \Omega_\rho$

is also positively invariant. This leads to the following optimization problem:

$$\begin{aligned}
& \max_V \quad \text{Volume}(\mathcal{C}) && (6.23) \\
& \text{subj. to} \quad V(0,0) = 0, \\
& \quad V(q,v) \geq \alpha(\|q\| + \|v\|) && \text{for } (q,v) \in \mathcal{C}, \\
& \quad \nabla V^T \begin{bmatrix} v \\ \dot{v} \end{bmatrix} \leq 0 && \text{for } (q,v) \in \mathcal{C}, \\
& \quad \frac{\partial V}{\partial v} H^{-1} (J_{N,i}^T + J_{T,i}^T \lambda_{T,i}) \leq 0 && \text{for } (q,v,\lambda_i) \in \mathcal{D}_i \text{ and } (q,v) \in \mathcal{C}.
\end{aligned}$$

As currently posed, this problem is not amenable to convex optimization techniques. It is difficult to directly measure the volume of \mathcal{C} and, as \mathcal{C} is only one connected component of $\Omega_1 \cap \mathcal{A}$, it is not naturally described as a semialgebraic set. We approximate these regions by finding polynomials $g_I(q,v)$ and $g_O(q,v)$ such that their one sublevel sets (\mathcal{G}_I and \mathcal{G}_O respectively) are inner and outer approximations of \mathcal{C} , i.e.

$$(\mathcal{G}_I \cap \mathcal{A}) \subseteq \mathcal{C} \subseteq (\mathcal{G}_O \cap \mathcal{A}). \quad (6.24)$$

By containing \mathcal{C} within the semialgebraic set \mathcal{G}_O and verifying the Lyapunov conditions on \mathcal{G}_O , we provide sufficient conditions on \mathcal{C} . The inner approximation \mathcal{G}_I is used to estimate the volume of the verified region. In practice, we parameterize g_I and g_O as quadratic forms. For $g_I(q,v) = \begin{bmatrix} q^T & v^T \end{bmatrix} G_I \begin{bmatrix} q^T & v^T \end{bmatrix}^T$, we will use $-\text{Trace}(G_I)$ as a proxy for the volume of

\mathcal{C} . Given this, we pose the following problem:

$$\begin{aligned}
& \min_{V, G_I, G_O} \text{Trace}(G_I) && (6.25) \\
& \text{subj. to } V(0, 0) = 0 \\
& G_I, G_O \succeq 0, \\
& V(q, v) \geq \alpha(\|q\| + \|v\|) && \text{for } (q, v) \in \mathcal{A} \cap \mathcal{G}_O, \\
& \nabla V^T \begin{bmatrix} v \\ \dot{v} \end{bmatrix} \leq 0 && \text{for } (q, v) \in \mathcal{A} \cap \mathcal{G}_O, \\
& \frac{\partial V}{\partial v} H^{-1}(J_{N,i}^T + J_{T,i}^T \lambda_{T,i}) \leq 0 && \text{for } (q, v, \lambda_i) \in \mathcal{D}_i \text{ and } (q, v) \in \mathcal{G}_O, \\
& V(q, v) \geq 1 && \text{for } (q, v) \in \mathcal{A} \text{ and } g_O(q, v) = 1, \\
& g_I(q, v) \geq 1 && \text{for } (q, v) \in \mathcal{A} \setminus \Omega_1.
\end{aligned}$$

This problem verifies the Lyapunov conditions on the outer approximation \mathcal{G}_O and ensures the containment in (6.24). It is now posed in the familiar form of an optimization over polynomials that are positive on a basic semialgebraic set. As described in 3.4.3, we use a bilinear alternation technique to solve this problem. One of the potentially difficult aspects of this alternation is that we must typically supply an initial feasible Lyapunov candidate. Previous sums-of-squares-based methods have used local linearizations of the dynamics to find initial candidates [137, 134], but this approach fails when the dynamics are discontinuous. Instead, we have used two potential methods for determining an initial Lyapunov function. The simplest strategy is to use the observation that the passive rigid-body dynamics and inelastic collisions are energetically conservative, and that taking V to be the total energy provides a viable starting point for most mechanical systems. Alternatively, bilinear alternations can be initiated by choosing candidates for G_I and G_O , such as ellipsoids with \mathcal{G}_I small and \mathcal{G}_O relatively large. In the examples in this work, we will generally take the later approach, to avoid initializing the alternations at a particular Lyapunov candidate, and thus avoiding explicit bias towards energy as a solution.

Solutions to (6.25) are guaranteed to be feasible Lyapunov functions to the original problem (6.23), although they will generally be suboptimal. This method, however, provides

a tractable technique to synthesize useful regional certificates through contact discontinuities. A similar approach to bilinear alternations is to fix \mathcal{G}_O and to fix the form of \mathcal{G}_I (within a scalar factor) and pose (6.25) as a feasibility problem. The optimal scaling of \mathcal{G}_I can then be found by binary search. Though it only searches over a subset of the solutions to the first formulation, this SDP may be better conditioned numerically for some applications.

6.3.3 Verifying Safety

The algorithm above for verifying stability and invariance can be easily adapted to address questions of dynamic safety. For instance, we might wish to determine the largest set of initial conditions such that the infinite horizon reachable set does not intersect some unsafe semialgebraic set \mathcal{X}_u . We pose this problem in a manner similar (6.25), although here we do not require that V be positive definite:

$$\begin{aligned}
& \min_{V, G_I, \mathcal{G}_O} \text{Trace}(G_I) && (6.26) \\
& \text{subj. to } G_I, \mathcal{G}_O \succeq 0, \\
& \nabla V^T \begin{bmatrix} v \\ \dot{v} \end{bmatrix} \leq 0 && \text{for } (q, v) \in \mathcal{A} \cap \mathcal{G}_O, \\
& \frac{\partial V}{\partial v} H^{-1} (J_{N,i}^T + J_{T,i}^T \lambda_{T,i}) \leq 0 && \text{for } (q, v, \lambda_i) \in \mathcal{D}_i \text{ and } (q, v) \in \mathcal{G}_O, \\
& V(q, v) \geq 1 && \text{for } (q, v) \in \mathcal{A} \text{ and } g_O(q, v) = 1, \\
& V(q, v) \leq 1 && \text{for } (q, v) \in \mathcal{A} \cap \mathcal{G}_I, \\
& V(q, v) > 1 && \text{for } (q, v) \in \mathcal{A} \cap \mathcal{X}_u.
\end{aligned}$$

The optimization program in (6.26) verifies that \mathcal{C} is positively invariant and that $\mathcal{C} \cap \mathcal{X}_u = \emptyset$, so no trajectory that originates in \mathcal{C} can leave the safe region.

6.3.4 Control Design

This approach to Lyapunov analysis of autonomous systems can be naturally extended to the design of feedback control laws, as detailed in [49]. The method was later extended to verify

stability along trajectories and then experimentally tested in [78]. Traditionally applied to continuous systems, here we consider the problem of designing a polynomial feedback law $u(q, v)$ that is smoothly dependent on state. The use of such a control law would provide robustness to disturbances that may induce unexpected impacts and to uncertainties in sensing the contact state. Design of a hybrid or switching controller would additionally suffer from the same combinatorial mode enumeration issues faced by verification of such models. When controlling a robotic system near to multiple contact surfaces, a stable, mode-invariant feedback law would be both robust and simple to implement.

For control input u , the continuous dynamics are given in the standard controlled manipulator form (3.1). The task of feedback design can then be expressed as finding control law $u(q, v)$ such that the conditions in Theorem 6.9 holds. This amounts to a modification of (6.2) to be

$$\nabla V(q, v)^T \begin{bmatrix} v \\ H^{-1}(Bu(q, v) - C) \end{bmatrix} \leq 0. \quad (6.27)$$

Since both $V(q, v)$ and $u(q, v)$ are optimization parameters, this constraint is bilinear in V and u , and so, here too we employ an alternating method. When merged with the regional verification of Section 6.3.2, the problem remains bilinear, and so a two-step alternation approach suffices.

The full process detailing initialization and iterations for control design and regional verification is detailed in Algorithm 6.1. In ITERATIONA, we fix G_I and the Lyapunov function, up to a scale factor, and we search for the scale factor, the control policy $u(q, v)$, and G_O . Conversely, in ITERATIONB, we fix G_O and the control policy, and we search for the Lyapunov function and G_I . By splitting the decision parameters in this manner, each alternation is convex and can be represented by a SDP. Furthermore, the previously described versions for global analysis and safety are straightforward to implement in practice.

Algorithm 6.1 Control Design Alternations

Require: Initial locally stabilizing controller $u(q, v)$ and G_O and G_I

Require: Termination criteria ϵ

- 1: $i \leftarrow 1$
 - 2: $cost_0 \leftarrow \infty$
 - 3: INITIALIZE
 - 4: **do**
 - 5: ITERATIONA
 - 6: ITERATIONB
 - 7: $cost_i \leftarrow -Trace(G_I)$
 - 8: $i \leftarrow i + 1$
 - 9: **while** $cost_i - cost_{i-1} > \epsilon cost_i$
 - 10: **function** INITIALIZE
 - 11: Fix $u(q, v)$, G_O , G_I
 - 12: Solve a modified form of SOS program (6.25) with two changes:
 1. Remove the cost, so the program is a question of feasibility only
 2. Since V is unknown and G_I is fixed, replace the final condition $g_I(q, v) \geq 1$ with
$$V \leq 1 \text{ for } (q, v) \in \mathcal{A} \cap \mathcal{G}_I$$
 - 13: From solution, extract $V(q, v)$ and S-procedure multiplier for \mathcal{G}_O .
 - 14: **function** ITERATIONA
 - 15: $V(q, v) \leftarrow \gamma V(q, v)$
 - 16: Fix G_I and S-procedure multipliers related to \mathcal{G}_O
 - 17: Solve a modified form of SOS program (6.25), where the objective is to maximize γ
 - 18: From solution, extract $u(q, v)$, G_O , and S-procedure multiplier for Ω_1 .
 - 19: **function** ITERATIONB
 - 20: Fix $u(q, v)$, G_O , and S-procedure multiplier for Ω_1
 - 21: Solve SOS program (6.25)
 - 22: From solution, extract $V(q, v)$, G_I , and S-procedure multiplier for \mathcal{G}_O .
-

6.3.5 Complexity

To numerical precision, SDPs can be efficiently solved in practice and theoretically solved in polynomial time [139]. The difficulty of these problems is dependent on the number of variables and semidefinite constraints. Let n be the dimension of the system state space, m the number of potential contacts, and d the total degree of the polynomial representation used. Then, the approaches presented in this section construct SOS programs with $\mathcal{O}(m)$ non-negativity constraints to verify that $dV \leq 0$. The inequality (6.17) results in the generation of $\mathcal{O}(m)$ S-procedure multipliers per such constraint, each of which is transformed into a semidefinite constraint of size $\mathcal{O}((n+1)^d)$ in the SDP. Therefore, there are $\mathcal{O}(m^2)$ semidefinite constraints, and $\mathcal{O}(m^2(n+1)^d)$ variables in the SDP. Note that had we not used the formulation from Theorem 6.9 and decoupled the contact constraints, the SDP size would be $\mathcal{O}(m(n+m)^d)$. By comparison, hybrid formulations that enumerate each mode and transition will necessarily introduce variables and constraints that scale exponentially in m .

6.4 Example Applications

6.4.1 Bean Bag Toss

We first examine the simple problem of a bean bag, modeled as a planar point mass, colliding inelastically with the ground. This example serves to demonstrate the method on a system simple enough where the calculations can be easily verified by hand. With $q = \begin{bmatrix} y & z \end{bmatrix}^T$ and $v = \begin{bmatrix} \dot{y} & \dot{z} \end{bmatrix}^T$, we define $\phi(q) = z$. The dynamics are given by

$$\begin{aligned} M\ddot{y} &= \lambda_T, \\ M\ddot{z} &= -Mg + \lambda_N. \end{aligned}$$

For this simple system, the dynamics are invariant under y , so we consider the equilibrium set where the mass rests on the ground, $\{(y, z) \in \mathbb{R}^2 : z = 0\}$. Choosing our Lyapunov function candidate be equal to the total energy of the system, we will show stability in the

sense of Lyapunov and invariance of a series of nested sets. That is, each sublevel set of energy is positively invariant. Substituting $V(q, v) = E = \frac{1}{2}M\dot{y}^2 + \frac{1}{2}M\dot{z}^2 + Mgz$ and the dynamics into (6.22), we have the two conditions:

$$-\nabla V^T \begin{bmatrix} v \\ \dot{v} \end{bmatrix} = -Mg\dot{z} + Mg\dot{z} \geq 0 \text{ for } (q, v, \lambda) \in \mathcal{D}, \quad (6.28)$$

$$-\frac{\partial V}{\partial v} H(q)^{-1} (J_N(q)^T \lambda_N + J_T(q)^T \lambda_T) = -\dot{y}\lambda_T - \dot{z}\lambda_N \geq 0 \text{ for } (q, v, \lambda) \in \mathcal{D}. \quad (6.29)$$

The first condition is trivially true. Observing that $Jv = \dot{z}$ and $J_T v = \dot{y}$, we use S-procedure type multipliers to verify the second condition. Generating sums-of-squares multipliers $\sigma_i(q, v, \lambda)$ for the relevant unilateral constraints (6.7) and (6.8), replace (6.29) with

$$-\dot{y}\lambda_T - \dot{z}\lambda_N + \sigma_1\dot{y}\lambda_T + \sigma_2\dot{z}\lambda_N \text{ is SOS.} \quad (6.30)$$

Choosing $\sigma_1 = \sigma_2 = 1$, the equation above vanishes and is trivially non-negative. Thus, we have used our methods to demonstrate the rather obvious fact that every sublevel set of energy will be positively invariant. Note that the quartic Lyapunov function

$$V = E + E^2 + \frac{1}{2}\dot{z}^3 + \frac{1}{2}z\dot{z}$$

satisfies (6.22) but where we can additionally verify that $\dot{V} < -\alpha(z + \dot{y}^2 + \dot{z}^2)$ for some class \mathcal{K} function α . Combined with the condition that $dV \leq 0$, this is sufficient to verify asymptotic stability of the equilibrium set, though we do not prove this here. In general, it is difficult to find such Lyapunov functions for discontinuous mechanical systems.

6.4.2 Rimless Wheel

The rimless-wheel model is a single rigid body composed of a number of equally-spaced spokes about a simple mass. This simple model has been used extensively as a proxy for a passive-dynamic walking robot [22]. Though previous works have primarily been interested in analyzing the limit-cycle behavior of the rimless wheel, here we focus on the stability of

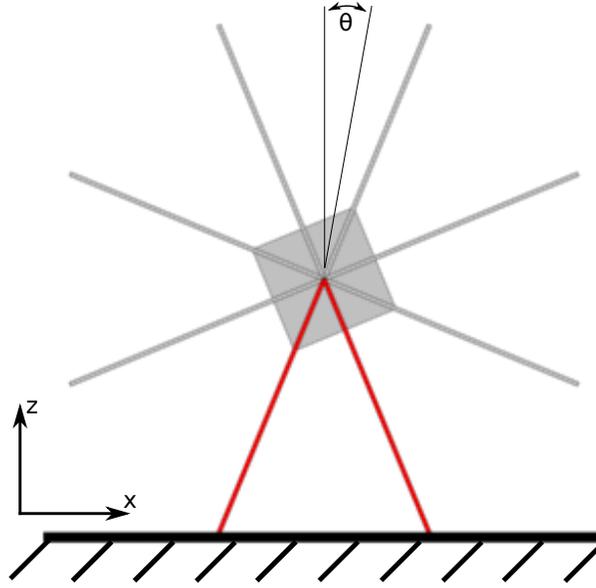
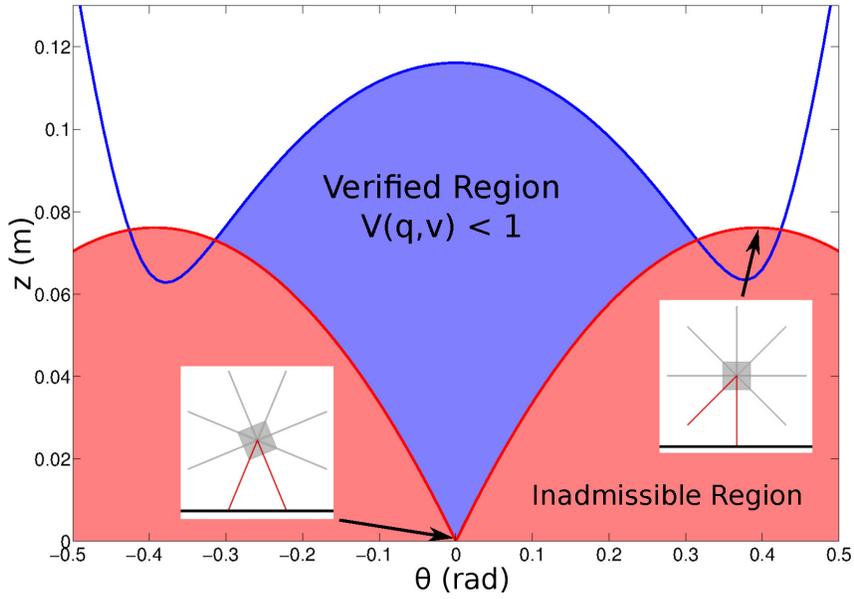


Figure 6-1: The rimless wheel shown in an equilibrium state, with two feet on the ground. Verified trajectories pass through four possible contact states (no contact, double-support, and both single-support phases).

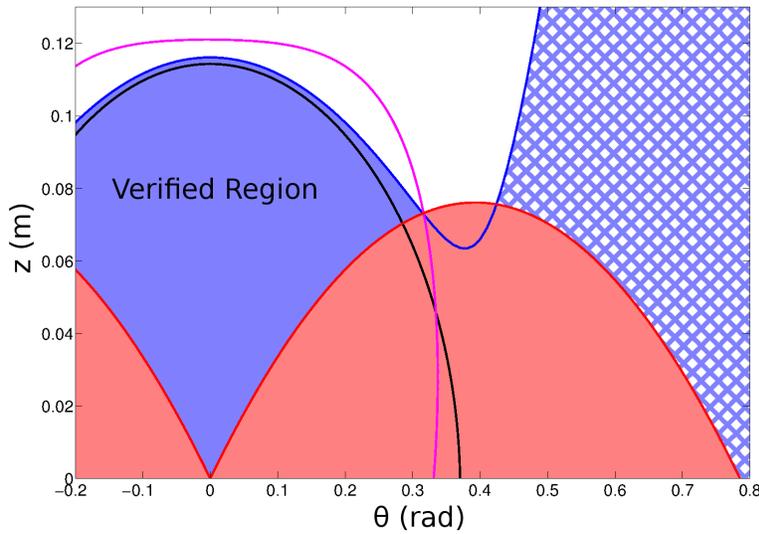
a single, static configuration of the system. We allow for frictional contacts between two of the spokes and the ground, highlighted in Figure 6-1, and we consider the equilibrium set where both of these spokes rest on a flat ground. We differentiate between resting on these two particular spokes and any other equilibrium state. Trajectories of the rimless wheel that come to rest in the equilibrium set may undergo an infinite number of collisions rocking back and forth between the two feet, in an example of Zeno phenomena.

Remark 6.11. *Not only does this simple example exhibit Zeno phenomena, but it also admits non-unique solutions from some initial conditions. For example, take $q = \begin{bmatrix} 0 & z & 0 \end{bmatrix}^T$ and all velocities to be zero. The rimless wheel will simultaneously strike the ground with two legs, and feasible resolutions to this multi-impact problem lead to a set of post-impact states. The rimless wheel can come instantly to rest, or it can rock in either direction.*

The planar floating base model of the rimless wheel has three degrees of freedom, $q = \begin{bmatrix} x & z & \theta \end{bmatrix}^T$ and $v = \begin{bmatrix} \dot{x} & \dot{z} & \dot{\theta} \end{bmatrix}^T$. With the trigonometric substitutions $s = \sin(\theta)$ and $c = \cos(\theta)$, the dynamics of the rimless wheel and the contact related elements $\phi_i(q)$, $J_{N,i}(q)$,



a The red region indicates the inadmissible set, where at least one of the contact points is penetrating the ground. Two states are highlighted: the stable equilibrium in double-support and an unstable equilibrium in single-support. The blue region below the dashed line is the connected component $\mathcal{C} \subseteq \Omega_1 \cap \mathcal{A}$ that contains the equilibrium (the verified region).



b Two additional curves are shown. The black curve is the boundary of \mathcal{G}_I , tight to $\Omega_1 \cap \mathcal{A}$, which we use to approximate the volume of the verified region. The magenta curve outlines $\mathcal{G}_O \supseteq \mathcal{C}$. \mathcal{G}_O is parameterized as an ellipsoid in the redundant state variables, including s and $(1 - c)$, which is why it is not ellipsoidal when plotted against θ . The hatched region, while a subset of $\Omega_1 \cap \mathcal{A}$, is not connected to \mathcal{C} and is not verified.

Figure 6-2: Verified regions for a slice of state space where all velocities are zero.

and $J_{T,i}(q)$ are all polynomial functions of the redundant state variables $(x, z, s, c, \dot{x}, \dot{z}, \dot{\theta})$ and the contact forces $(\lambda_{N,i}, \lambda_{T,i})$. As with the point mass example, the dynamics are invariant under x , and so the equilibrium set is defined as $\{(x, z, \theta) \in \mathbb{R}^3 : z = 0, \theta = 0\}$.

Fixing g_i and g_o , we find an initial candidate Lyapunov function as in Algorithm 6.1. We then use iterations search for a solution to (6.25) to find a nested set of invariant regions and verify stability in the sense of Lyapunov. When we parameterize V as a quartic polynomial, we verify a significant region of state space about the origin. A slice of this region is shown in Figure 6-2a where the verified region is the connected component of $\Omega_1 \cap \mathcal{A}$ containing the origin. Figure 6-2b illustrates the use of \mathcal{G}_I and \mathcal{G}_O to provide inner and outer bounds on \mathcal{C} .

It is interesting to note that if we parameterize V as a quadratic polynomial, the alternations quickly converge to verify a region that is identical to the maximal sublevel set of energy that does not contain any additional equilibrium points. Similarly, we recover a scaled version of energy as our Lyapunov candidate. The quartic parameterization, however, verifies a larger region with a Lyapunov function significantly different from energy.

Note that the true region of attraction of this model is unbounded. For instance, for any x, z , take $q = 0$ and $v = \begin{bmatrix} \dot{x} & 0 & 0 \end{bmatrix}^T$. A trajectory starting from any such state will slide along the ground and eventually come to rest. By our parameterizations of \mathcal{G}_I and \mathcal{G}_O , our regional approach is limited to ellipsoidal volumes, and so will not recover the entire region of attraction. We do find a significant volume about the equilibrium set that would be relevant to planning or control applications.

6.4.3 Perching Glider

We also examine the problem of verifying a safe set of initial conditions for a glider perching against a wall, by adapting a model first presented in [24] and [30]. We consider the instant after the glider feet, which have adhesive microspines, have first impacted the wall, and so we treat this contact as a pin joint. The glider is then modeled as a two-link body, with a spring damper connecting the bodies as shown in Figure 6-3. We allow the tail of the glider to impact the surface of the wall and slide along it. The specific problem of verification was earlier addressed by Glassman et al. [41], although the authors there used a model with a

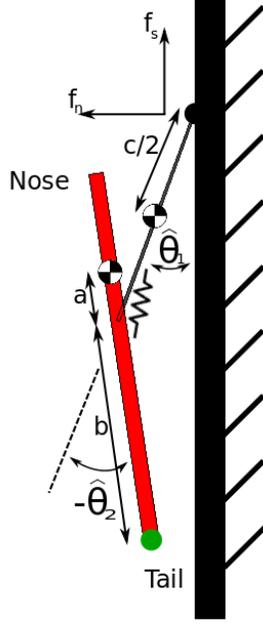
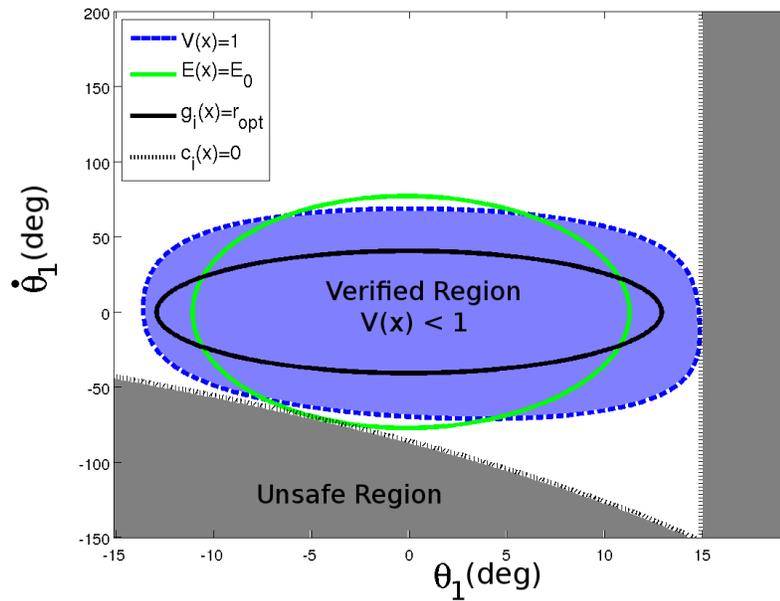


Figure 6-3: A simple model of a perching aircraft using two rigid links. The foot of the aircraft is pinned to the wall surface and there is a contact point at the tail that can collide with and slide along the wall. A torsional spring damper connects the main body of the aircraft to the foot.

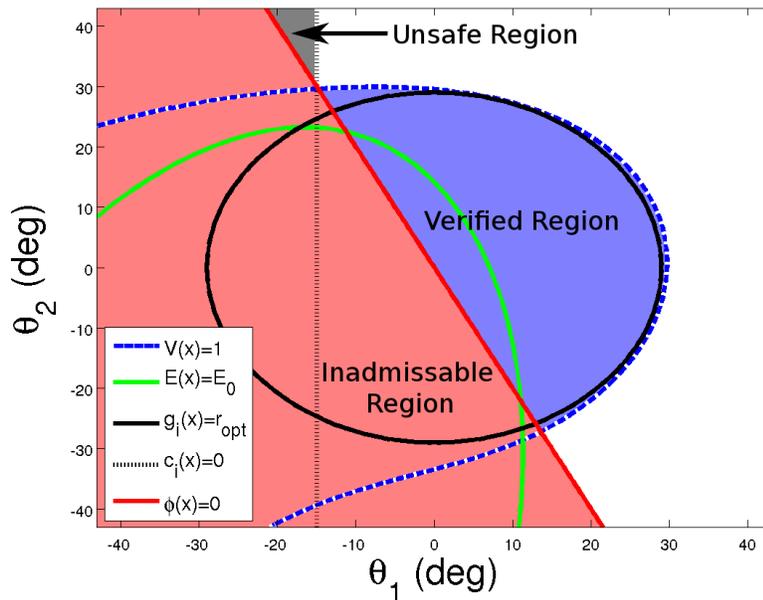
single joint and fixed the tail of the glider to slide along the wall, disallowing collisions. In this work, we verify a significantly larger region than in [41], though a direct comparison is impossible since our model is higher dimension and uses Coulomb friction instead of viscous damping at the tail contact.

There are two relevant failure modes for the perching behavior of the glider, described in more detail in [41]. In one, the nose of the glider impacts the wall, which would be a potentially damaging event. In the other, the force limit of the feet microspines is exceeded and the glider falls from the wall. The force at the feet is a rational function of the state variables, and so the force limit can be easily expressed as a semialgebraic constraint.

This is a two degree of freedom model, with $q = [\hat{\theta}_1 \quad \hat{\theta}_2]^T$, and we again use a trigonometric substitution for both angles. We also change coordinates so that $(0, 0)$ is an equilibrium point with the tail resting against the wall, substituting $\theta_1 = \hat{\theta}_1 - 0.2604$ and $\theta_2 = \hat{\theta}_2 + 0.5207$. With the rimless wheel, $H(q)$ was a constant matrix and so $H^{-1}(q)$ was also constant. Here, $H(q)^{-1}$ is rational and so, for our dynamical constraints, we clear the denominator to ensure that our conditions are algebraic.



a A slice of the glider state space where the tail is restricted to the surface of the wall. The shaded blue region within the dashed line indicates the verified region $\mathcal{C} \cap \mathcal{A}$ for a quartic Lyapunov function and the solid black ellipse outlines \mathcal{G}_I . The green ellipse indicates the maximal sublevel set of energy that does not intersect the unsafe region, shown in gray.



b A second slice of state space where the joint velocities are zero is also shown. In this slice, the general quartic Lyapunov function vastly outperforms energy and \mathcal{G}_I is tight to \mathcal{C} .

Figure 6-4: Verified regions for two different slices of the glider state space.

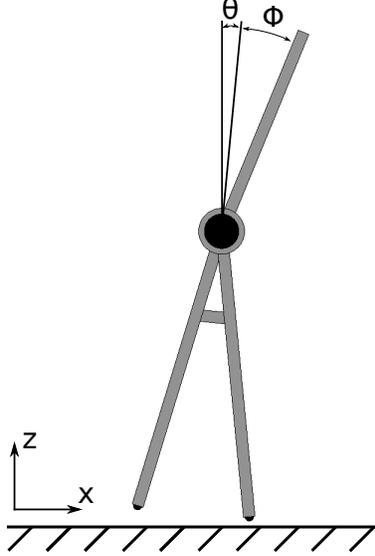


Figure 6-5: A simple balancing robot with a rigid base and actuated torso. Two feet can make contact with the ground, in a manner similar to the rimless wheel.

We search for a solution to (6.26), to maximize the set of initial conditions of trajectories that do not violate either constraint. Letting \mathcal{G}_O be all of \mathbb{R}^{2n} , define

$$\mathcal{G}_I(\rho) = \{(q, v) : 2(1 - c_1) + 2(1 - c_2) + 0.1\|v\|^2 < \rho\}.$$

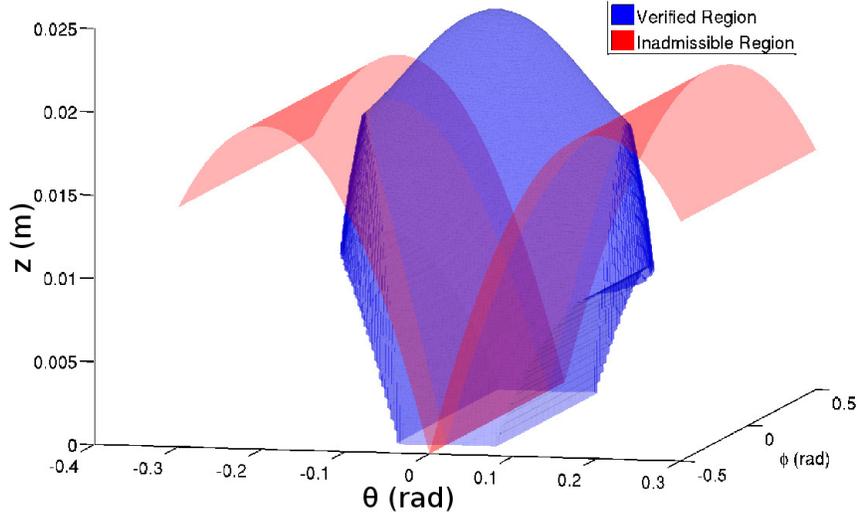
We seek to maximize ρ through a binary search, observing that $2(1 - c_i)$ well approximates θ_i^2 near $\theta_i = 0$. As with the rimless wheel, if we restrict our search to parameterizations of V of equal degree to energy, we recover the maximal sublevel set of energy that does not intersect either constraint boundary. If we expand our search to include all quartic polynomials, we find a Lyapunov function which verifies a visibly larger region. Two slices of this region are visualized in Figure 6-4a and 6-4b. Our binary search terminates finding $\rho = 0.25$, and we can verify that an upper bound on the true optimal value is $\rho = 0.327$, since there $\mathcal{G}_I(\rho)$ intersects the constraint boundary. Of course, the true optimal value might be lower still, since there is no claim that any $\mathcal{G}_I(\rho)$ is invariant. Here, we find a significant invariant region that usefully approximates the safe set of initial conditions.

6.4.4 Balancing Robot Control Design

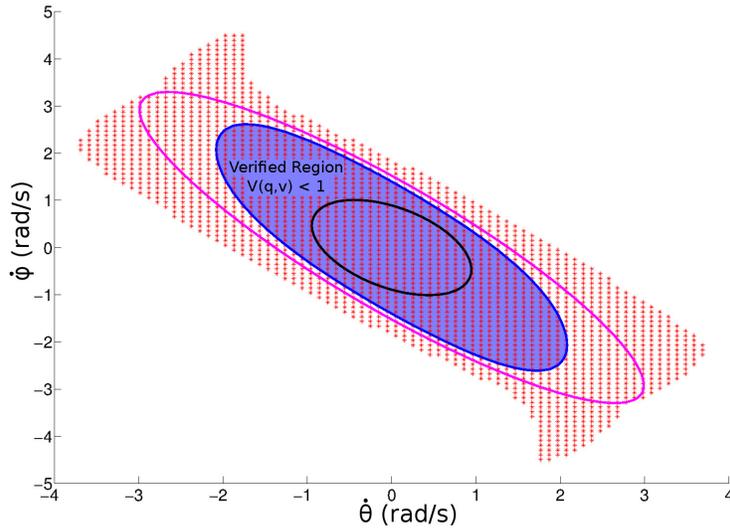
Lastly, we present a control-design example that builds on the rimless wheel. The system here is a simple balancing robot described by planar dynamics. The lower body is similar to that of the rimless wheel, where the two legs are at a fixed angle, each with a single contact point at the end. The upper mass is connected to the lower body by a single actuated joint. When both feet are motionless with respect to the ground, the robot resembles an inverted pendulum. In all other contact configurations, it has varying degrees of underactuation. As with the rimless wheel, the dynamics are invariant in the x direction, and so we can treat the robot as a 7 state system. Of note, the balancing robot also undergoes Zeno phenomena and exhibits a lack of uniqueness in some situations. Typical simulated trajectories undergo a lengthy sequence of contact transitions, including single and double support phases as well as sliding modes.

For this example, we use cubic Taylor expansions of the dynamics and contact constraints. Algorithm 6.1 is initialized with a simple PD controller on ϕ and G_i and G_o such that both locally approximate total energy. Note that because of the unilateral constraints, and the linear dependence of gravitational potential on height, the Hessian of energy must be modified slightly to form this approximation. For optimizing the verified region, the controller was parameterized as a full-state feedback cubic controller. A quartic function was used to parameterize the Lyapunov candidates, and S-procedure multiplier degrees were chosen to balance the total degree of each sums-of-squares condition.

Using SPOT to generate the SDP and MOSEK to solve it, each iteration of Algorithm 6.1 (two SOS programs) took approximately 45 minutes. For $\epsilon = .01$, convergence was reached in 30 iterations. While this total running time is substantial, the complexity is dominated by the dimensionality of the plant, not the number of contact points. For comparison, each iteration for rimless wheel, which has two fewer states and no actuation, runs in approximately 90 seconds on a desktop machine. As with the other examples, the algorithm is able to verify a significant region of state space surrounding the equilibrium, slices of which are illustrated in Figure 6-6. In a 7 dimensional state space, a true experimental evaluation of the stabilized region is intractable. Similar to the rimless wheel, the true stabilized region is unbounded in



a A three dimensional slice of the state space, where all velocities have been set to zero. Here, the blue region is the verified volume, although, in the figure, it extends slightly past the contact surfaces for visualization purposes.



b A second slice is shown with respect to the angular velocities, where positions and translational velocities set to zero. This figure additionally illustrates the boundaries of \mathcal{G}_i and \mathcal{G}_o in in black and magenta. With initial conditions in this slice, trajectories were simulated by a time-stepping method ([128]) to find the experimental region of attraction. The trajectories that converged to the equilibrium are indicated with red stars.

Figure 6-6: Verified regions for two slices of state space.

some directions. In other slices, however, sampling the space and simulating gives a rough indication that the verified region is a good approximation of the true stabilized region. For example, we discretized a slice of state space and exhaustively simulated the control policy from those initial conditions. Results are illustrated in Figure 6-6b, where the verified region captures much of the shape and size of the true region of attraction.

6.5 Discussion

The natural structure of rigid-body dynamics and the complementarity formulation of frictional impacts provide a framework for posing questions of stability and invariance as sums-of-squares optimization problems. This chapter presents a class of algorithms for numerical computation of Lyapunov certificates for such systems, as well as for the design of mode-invariant stabilizing controllers. By invoking the measure differential inclusion model of contact, we avoid directly reasoning about both the complexity of Zeno phenomena and the combinatorial number of hybrid modes associated with the set of potential contact states. Initial experiments in a simulated environment have found physically significant certificates of stability and invariance for multiple problems of interest.

In this chapter, we have been primarily interested in stability in the sense of Lyapunov and positive invariance, but we hope to extend these methods to asymptotic stability. One challenge with this extension is that friction often leads to a connected set of equilibria, which poses difficulties for methods based on analytic Lyapunov functions. That fact aside, two common approaches to verifying asymptotic stability are to find a Lyapunov candidate where \dot{V} is strictly negative or to apply LaSalle's Invariance Principle. With the former method, energy no longer provides an initial feasible candidate to begin bilinear alternation. In [71], Theorem 6.31 gives a generalization of LaSalle to discontinuous systems with the additional condition that the limit sets of trajectories also be positively invariant. The algorithms presented here might be extended to meet this condition and search for certificates of asymptotic stability.

Ongoing work in this area also centers on extending these algorithms to more complex tasks. While the examples here are relatively low dimensional, commercial SDP solvers are

relatively immature and are rapidly improving. Additionally, recent work on hierarchical relaxations of SOS problems have shown promise in solving significantly larger problems [79, 68]. In particular, we are interested in the analysis of trajectories and limit cycles of robotics systems, where previous work has demonstrated the effectiveness of SOS-based methods [81]. By extending to trajectory analysis, we might verify the stability of locomotion gait primitives [43] or that of walking motions with respect to terrain variations [115]. Mode-invariant analysis might also be applied to methods which reduce the dimensionality of walking problems, such as the form of zero dynamics explored in [65]. Furthermore, a natural extension of this work would be to include elastic impacts, where many models exist which are amenable to complementarity formulations such as in [129].

Chapter 7

Balancing and Push Recovery

A fundamental requirement for legged robots is to maintain balance and prevent potentially damaging falls, whenever possible. As a response to outside disturbances, fall prevention can be achieved by a combination of active balancing actions, e.g. through ankle torques and upper-body motion, and through reactive step placement. While it is widely accepted that stepping is required to respond to large disturbances, the limits of active motions on balancing and step recovery are only well understood for the simplest of walking models. Recent advances in convex optimization-based verification and control techniques enable a more complete understanding of the limits and capabilities of more complex models. In this chapter, we present an algorithmic approach for formal analysis of the viable-capture basins of walking robots and design of push recovery control strategies [103]. Extending beyond the classic Linear Inverted Pendulum Model (LIPM), we analyze a series of centroidal momentum based planar walking models, examining the effects of center of mass height, angular momentum, and impact dynamics during stepping on capturability.

The LIPM makes a number of key assumptions to simplify the dynamical equations of motion: (1) planar center-of-mass (COM) motion (often constant height), (2) constant angular momentum, (3) minimum step time (independent of step length), and (4) zero-impact stepping. Note that the first constraint necessarily requires the fourth, since impulsive forces when stepping would cause changes in the vertical velocity. Typical control approaches based on the LIPM plan and execute motions that satisfy these four conditions, therefore the LIPM assumptions can be seen as model simplifications or, alternatively, as restrictions

on control policies.

As detailed in 2.3, recent robotics research has examined some of these assumptions, typically in isolation. Here, we introduce a more unified approach to this analysis, based on sums-of-squares programming. As discussed previously, SOS methods for formal analysis, like those used in Chapter 6 or those discussed in 2.2 and 3.4, do not yet scale to high-dimensional models. However, centroidal-momentum based models nicely capture the dynamics relevant to push recovery and balancing and are relatively low-dimensional, making this an ideal setting for formal dynamical analysis.

Below, we present SOS algorithms for estimating the N -step viable-capture basins of walking robots. We compute both inner and outer approximations to the viable-capture basins, along with control policies that provably stabilize all states within the inner approximations. These algorithms are demonstrated on a series of centroidal models, evaluating the effects of vertical acceleration, angular momentum, and impact dynamics on the viable-capture basins.

7.1 Preliminaries

This chapter builds upon the previously introduced notions of barrier functions in 3.4. Below, we include some additional necessary background material. We define the class of walking and balancing models, discuss the concept of N -step capturability, and introduce a closely related approach to reachability analysis.

7.1.1 Model Classes

In this chapter, we consider a number of different planar models for walking robots, each based on the centroidal dynamics. As a result, we make a departure here from the second-order, rigid-body model used in the previous chapters. We will also focus on a simple, time-driven model of stepping events, and will therefore leverage the classical terminology from hybrid systems. Specifics of each model will be given in Section 7.4, but all will obey a particular structure. We restrict to systems governed by control-affine dynamics with

box-constrained control inputs,

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

with $x \in \mathbb{R}^n$ and $u \in \mathcal{U}$. For simplicity, and without further loss of generality, we take $\mathcal{U} = [-1, 1]^m$. Stepping events will be assumed to occur after a fixed period, T , and result in a discrete event with the post-step state given by a reset map $x_+ = r(x_-, s, \Lambda)$ for the foot location $s \in [-1, 1]$ and impact impulse $\Lambda \in \mathbb{R}$. Note that some models will follow the traditional LIPM approach and assume zero impulse during stepping, and others will include impulsive impact forces transmitted through a massless leg. While, for models with impacts, it might be possible to explicitly define the impulse Λ , we instead will exploit an implicit definition of inelastic impacts to reduce overall problem complexity. In this formulation, valid impulses must satisfy an implicit constraint of the form

$$h(x_-, s, \Lambda) = 0. \tag{7.1}$$

Lastly, for each model class considered, $r(x_-, s, \Lambda)$ will be affine in s .

7.1.2 Capturability

We briefly introduce the concept of N -step capturability from [58]. A state x_0 is said to be N -step capturable if there exists a dynamically feasible trajectory $x : \mathbb{R}_+ \rightarrow \mathbb{R}^n$ of at most N steps, with $x(0) = x_0$, such that $x(t) \notin \mathcal{X}_f$ for all t and $\mathcal{X}_f \subset \mathbb{R}^n$ a defined set of failed states. The set of all such states is called the N -step viable-capture basin. Therefore, the 0-step viable-capture region defines the set of states which can avoid the set of failed states by balancing actions alone (without stepping).

A useful property of N -step capturability is that it enables a recursive analysis. A state x_0 is N -step capturable if there exists a dynamically feasible trajectory $x(t)$ with $x(0) = x_0$ such that one of two conditions hold:

1. The trajectory contains zero steps and is forever disjoint from \mathcal{X}_f .
2. The trajectory is disjoint from \mathcal{X}_f until time T , when a stepping event occurs such that the reset map brings the state to the $(N - 1)$ -step viable-capture basin.

Note that this property, which we will exploit below, is identical to the classical concept of a viability kernel with target from Quincampoix and Veliov [110], Aubin et al. [6],

7.2 Reachability via occupation measures

An approach to polynomial optimization, based in semidefinite programming, derives from the work of Lasserre’s moment relaxations [67], and is known to be the dual to sums-of-squares. This formulation has led to a number of applications in nonlinear control (e.g [46, 80, 60]), with a general hybrid formulation given by Shia et al. [118]. Here, we will make use of the work of Henrion and Korda [46] to find outer approximations of dynamically reachable sets. This method was originally introduced in terms of occupation measures, enabling proofs of convergence found in [46]. For notational consistency and clarity of presentation, we briefly discuss the dual formulation over SOS polynomials.

Given a compact region of state space (e.g. a ball of some radius) $\mathcal{X} \subset \mathbb{R}^n$ and goal set $\mathcal{X}_{goal} \subset \mathcal{X}$, this approach synthesizes an outer approximation to the backwards reachable set of the goal region. Failed states, \mathcal{X}_f , are assumed to be those outside \mathcal{X} . In other words, if the goal region is the balancing equilibrium, it computes an outer approximation to the viable-capture basin. Similar to the method of Lyapunov and barrier functions, this approach synthesizes functions which prove that some set of states *cannot* ever be stabilized.

If functions $V : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$ and $p : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}^m$ can be found, such that

$$x \in \mathcal{X} \text{ and } t \in [0, T] \Rightarrow -\frac{\partial V(t, x)}{\partial x} f(x) - \mathbf{1}^T p(t, x) - \frac{\partial V(t, x)}{\partial t} > 0, \quad (7.2)$$

$$x \in \mathcal{X} \text{ and } t \in [0, T] \Rightarrow p_i(x) \geq \left| \frac{\partial V(t, x)}{\partial x} g_i(x) \right| \text{ for } i = 1, \dots, m, \quad (7.3)$$

$$x \in \mathcal{X}_{goal} \Rightarrow V(T, x) > 0, \quad (7.4)$$

then the 0-superlevel set of V is an outer approximation of the viable-capture basin. Intuitively, these constraints ensure that $\dot{V} < 0$ for all possible control inputs, with p as a bound on the influence of the control actions. Given this, trajectories starting from x with $V(t, x) < 0$ cannot ever reach the goal region, where $V > 0$, without leaving \mathcal{X} .

Objective functions are chosen to minimize the size of this outer approximation. In

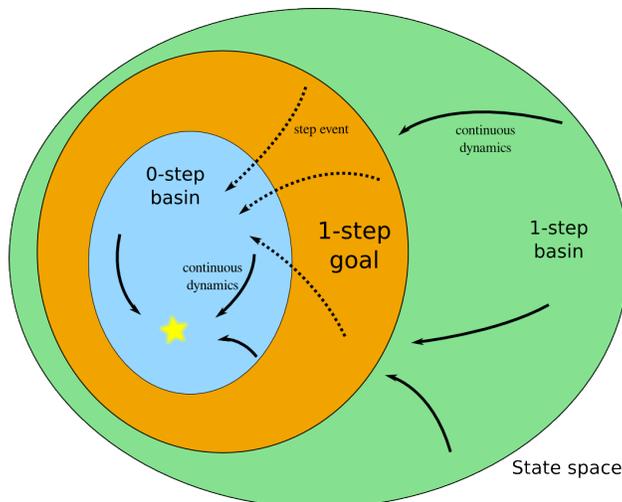


Figure 7-1: The viable-capture basins will be iteratively built, starting by finding the 0-step or balancing basin. Stepping events connect the goal for subsequent iterations to the previous basin.

particular, by introducing $W : \mathbb{R}^n \rightarrow \mathbb{R}$, with

$$x \in \mathcal{X} \Rightarrow W(x) \geq 0, \tag{7.5}$$

$$x \in \mathcal{X} \Rightarrow W(x) \geq 1 + V(0, x), \tag{7.6}$$

then minimizing $\int_{\mathcal{X}} W(x)dx$, “pushes down” on the 0-superlevel set of V , and approximates the volume of the viable-capture basin. For many common descriptions of \mathcal{X} , including ellipsoidal regions, this integral can be easily computed, and is linear in the coefficients of W , as in [36].

7.3 Approach

Here, we present sums-of-squares based algorithms for computing inner and outer approximations to the N -step viable-capture basins for walking robots. Sums-of-squares formulations are presented in detail to improve clarity and reproducibility. Both approaches will follow this simple iterative outline, cartooned in Figure 7-1, separating the continuous dynamics from the discrete events (stepping):

1. **0-step:** Approximate the infinite-horizon set of states from which a balancing controller

can stabilize the origin

2. **N -step:** Set the goal region to be the set from which a stepping event leads to the $(N - 1)$ -step viable-capture basin. Approximate the backwards reachable set of this goal region over a finite time interval.

7.3.1 Bilinear inner approximation

The approach to generate inner approximations of the viability kernel requires finding Lyapunov-like barrier functions (recall 3.4.1) and a corresponding control policy. Since, first and foremost, we are interested in fall prevention from the widest set of initial conditions, we choose to parameterize a bang-bang control policy. As a result of Pontryagin’s Minimum Principle, we expect bang-bang to be optimal for nonsingular problems [9]. Accordingly, define this policy in terms of switching surfaces $S : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}^m$ with $u_i(t, x) = \text{sgn}(S_i(t, x))$. This leads to 2^m barrier conditions, defined over (potentially not connected) domains of constant control input specified by $I \in \{-1, 1\}^m$,

$$D_I = \{(t, x) : t \in [0, T], I_i S_i(t, x) \geq 0 \text{ for } i = 1, \dots, m\},$$

Additionally, we wish to guarantee that the robot state never enters given unsafe or undesirable regions. For instance, the height and orientation of the robot should be constrained to stay within reasonable bounds. Let the unsafe region be described by $\mathcal{X}_u = \{x : \zeta_i(x) \geq 0, i = 1, \dots, k\}$. Therefore, the barrier function must separate the unsafe region from the viable-capture basins. The safety constraints and the bang-bang controller generate the polynomial optimization program:

$$\begin{aligned} & \text{minimize} && \int_{B_R} V(0, x) dx && \text{(I)} \\ & \text{s.t.} && (t, x) \in D_I, V(t, x) = \rho(t), I \in \{-1, 1\}^m \Rightarrow \dots \\ & && \frac{d\rho(t)}{dt} - \frac{\partial V(t, x)}{\partial x} (f(x) + g(x)I) - \frac{\partial V(t, x)}{\partial t} > 0, \\ & && x \in \mathcal{X}_u, t \in [0, T] \Rightarrow V(t, x) > \rho(t). \end{aligned}$$

As a proxy for volume of the ρ -sublevel set, we use the integral of V over a prescribed ball of state space. Note that other options exist, such as in Henrion et al. [47] or Chapter 6, but this approach was used here for simplicity and effectiveness. As mentioned above, this integral can be computed in closed form and is linear in the coefficients of V , and therefore is nicely compatible with SOS optimization. However, the above optimization program is bilinear in the unknown polynomials. To solve it, we adopt a two-stage technique based on bilinear alternations, similar to the approaches used in [137, 78, 102]. While these approaches offer no guarantee of optimality, they are practically effective and relatively straightforward to implement.

7.3.1.1 Strict Feasibility for Alternations

Solving a sequence of SOS programs can be computationally challenging. Solutions to these programs often lie on the boundary of the feasible set, and small numerical tolerances can lead to infeasibilities in subsequent programs. Adapting the work of Josz and Henrion [52], we improve upon the bilinear approach in 3.4.3 by writing the alternations in a manner that guarantees that (1) the feasible set always has a non-empty interior and (2) that alternating solutions lie on the interior. These simple steps greatly enhance the numerical stability of bilinear alternations. To accomplish this, we must add one additional ball constraint, that $x^T x \leq R^2$ for given R , and corresponding S-procedure multiplier. Since we are already restricted to the ρ -sublevel set of V , we simply choose a sufficiently large value for R and this does not result in any additional conservatism.

The approach, specified in Algorithm 7.1, is described here in detail. In the first stage, V, ρ and S are fixed polynomials, and we solve

$$\begin{aligned}
 & \underset{\gamma, q_I, \sigma_*}{\text{minimize}} && \gamma && \text{(A)} \\
 & \text{s.t.} && \gamma + \frac{d\rho}{dt} - \frac{\partial V}{\partial x}(f + gI) - \frac{\partial V}{\partial t} - \sigma_{I,R}(R^2 - x^T x) + \dots \\
 & && - q_I(\rho - V) - \sigma_{I,T}(Tt - t^2) - \sum_{i=1}^m \sigma_{I,i} I_i S_i \text{ is SOS,} \\
 & && \sigma_{I,R}, \sigma_{I,T}, \sigma_{I,1}, \dots, \sigma_{I,m} \text{ are SOS, } \forall I \in \{-1, 1\}^m,
 \end{aligned}$$

where $\gamma \in \mathbb{R}$ is a slack parameter. A solution with $\gamma < 0$ is feasible for (I). As shown in [52], the feasible set of this program is guaranteed to contain a non-empty interior. In the second stage, the multipliers q_I and $\sigma_{I,i}$ are held constant and we solve the program

$$\begin{aligned}
& \underset{\gamma, V, S, \rho, \sigma_{I,R}, \sigma_{I,T}}{\text{minimize}} && \gamma && \text{(B)} \\
& \text{s.t.} && \gamma + \frac{d\rho}{dt} - \frac{\partial V}{\partial x}(f + gI) - \frac{\partial V}{\partial t} - \sigma_{I,R}(R^2 - x^T x) + \dots \\
& && - q_I(\rho - V) - \sigma_{I,T}(Tt - t^2) - \sum_{i=1}^m \sigma_{I,i} I_i S_i \text{ is SOS } \forall I \\
& && \sigma_{I,R}, \sigma_{I,T} \text{ are SOS, } \forall I \in \{-1, 1\}^m, \\
& && V - \rho - \sigma_{\zeta,i} \zeta_i - \sigma_{i,T}(Tt - t^2) \text{ is SOS,} \\
& && \sigma_{\zeta,i}, \sigma_{i,T} \text{ are SOS for } i = 1, \dots, k \\
& && \int_{B_R} V(x) dx \leq c^*.
\end{aligned}$$

where c^* represents the optimal cost, found via binary search. Observe that the second stage additionally incorporates the safety constraints described above. As with the first stage, $\gamma < 0$ is used to verify feasibility.

7.3.1.2 Balancing

Algorithm 7.1 Inner Approximation

Require: Termination criteria ϵ

Require: LQR costs Q_{lqr}, R_{lqr} for initialization

- 1: $i \leftarrow 1, \quad cost_0 \leftarrow \infty$
 - 2: INITIALIZE
 - 3: **do**
 - 4: $(q_I, \sigma_{I,i}) = \text{ITERATIONA}$ \triangleright Solve (A) for multipliers
 - 5: $(V, S, \rho, c^*) = \text{ITERATIONB}$ \triangleright Solve (B) via binary search
 - 6: $cost_i \leftarrow c^*, \quad i \leftarrow i + 1$
 - 7: **while** $cost_{i-1} - cost_i > \epsilon cost_{i-1}$
 - 8: **function** INITIALIZE
 - 9: $V(x) \leftarrow \text{LQR}\left(\frac{\partial f}{\partial x}\Big|_{x=0}, g(0), Q_{lqr}, R_{lqr}\right)$
 - 10: $B(x) \leftarrow -\frac{\partial V}{\partial X}g(0)$
-

Algorithm 7.1 provides the framework for computing the N -step viable-capture basins of legged robots. As a first step, we compute the 0-step (balancing) region. For balancing, we pose the infinite-horizon problem, and eliminate explicit dependence on time t . Additionally, we specify the barrier function to take the form $V_0(x) = x^T Q x$ with $Q \succ 0$, and the control switching function $S_0(x)$ to also be quadratic in x . This condition enters as a semidefinite constraint in (B), and ensures that the origin is contained within the barrier function.

7.3.1.3 N -step

To compute the N -step viable-capture basin, we must properly encode the condition that the N -step region leads to the $(N - 1)$ -step region:

$$V_N(T, x) < \rho_N(T) \Rightarrow \exists s \in [-1, 1] \text{ s.t. } V_{N-1}(0, r(x, s, \Lambda)) < \rho_{N-1}(0) \quad (7.7)$$

For models with impacts, we first eliminate the force variable by solving the optimization program

$$\begin{aligned} & \underset{W, q_V, q_h, \sigma_R}{\text{minimize}} && \int_{B_R \times [-1, 1]} W(x, s) dx ds && \text{(R)} \\ & \text{s.t.} && q_w(W - 1) + q_V(V_{N-1}(0, r) - \rho_{N-1}(0)) - q_h h - \sigma_R(R^2 - x^T x) \text{ is SOS,} \\ & && \sigma_R \text{ is SOS,} \end{aligned}$$

where q_V, q_h , and σ_R are S-procedure multiplier polynomials, h is the implicit constraint from (7.1), and q_w is a given multiplier, typically $q_w = (1 + x^T x + s^2)^d$ as in Parrilo [94]. The 1-sublevel set of W , therefore, contains the pairing of states and step locations that, through the reset map, lead to the $(N - 1)$ -step region. In the zero-impact setting, simply take $W(x, s) = V_{N-1}(0, r(x, s, 0))$. Condition (7.7) can then be effectively encoded as a set of SOS constraints in (B), with details in Appendix 7.A. We parameterize $V_N(t, x)$ and $S_n(t, x)$ as quadratic in both state and time (with quartic cross terms), to express time-varying quadratic functions, and follow the approach of Algorithm 7.1.

Algorithm 7.1 provides a formulaic procedure for computing an inner approximation to the N -step viable-capture regions of a given model and a bang-bang control policy that

provably achieves the discovered region. The problem of control synthesis and verification is, however, nonconvex, and so no guarantees of convergence can be made. In practice, initialization with an LQR-based controller and barrier function leads to good results—although local minima do exist.

7.3.2 Outer approximations

The approach for computing outer approximations follows a similar technique as the inner approximations, utilizing the underlying method of Henrion and Korda [46]. Unlike with the inner approximations, the SOS programs are natively linear and do not require alternations.

7.3.2.1 Balancing

As in 7.3.1, we pose an infinite horizon problem, based in (7.2)-(7.6):

$$\begin{aligned}
& \underset{V, W, p, \sigma_R}{\text{minimize}} && \int_{B_R} W dx && \text{(O1)} \\
& \text{s.t.} && -\frac{\partial V}{\partial x} f - \mathbf{1}^T p - \sigma_R(R^2 - x^T x) \text{ is SOS,} \\
& && V|_{x=0} > 0, \\
& && p_i - \frac{\partial V}{\partial x} g_i - \sigma_{p,i}(R^2 - x^T x) \text{ is SOS for } i = 1, \dots, m, \\
& && p_i + \frac{\partial V}{\partial x} g_i - \sigma_{n,i}(R^2 - x^T x) \text{ is SOS for } i = 1, \dots, m \\
& && W \text{ is SOS,} \\
& && W - V - 1 \text{ is SOS,}
\end{aligned}$$

where $V : \mathbb{R}^n \rightarrow \mathbb{R}$, $p : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and the σ 's are S-procedure multipliers. We take the set of failed states to be those outside the R radius ball around the origin. Given a solution to (O1), the 0-superlevel set of V , $\{x : V(x) > 0\}$, provides an outer approximation to the 0-step viable-capture basin.

7.3.2.2 N -Step

As with the inner approximations, we must also include constraints which link the N and $(N-1)$ -step regions. To compute an outer approximation, we express the sentiment that the *goal region* ($t = T$) for the N -step calculation must include *all* states which can be brought to the beginning $N-1$ capture basin ($t = 0$). Defining

$$\mathcal{H} := \{(x, s, \Lambda) : s \in [-1, 1], h(x, s, \Lambda) = 0, V_{N-1}(0, r(x, s, \Lambda)) \geq 0\},$$

then this condition can be written

$$(x, s, \Lambda) \in \mathcal{H} \Rightarrow V_N(T, x) \geq 0. \quad (7.8)$$

Unlike with the inner approximations, no secondary step is required, and this constraint can be directly incorporated in a single stage. To compute the N -step outer approximation, solve the SOS program described by (7.2)-(7.6) and (7.8):

$$\begin{aligned} & \underset{V_N, W, p, q_h, \sigma_*}{\text{minimize}} && \int_{B_R} W dx && (O2) \\ & \text{s.t.} && -\frac{\partial V}{\partial x} f - \frac{\partial V}{\partial t} - \mathbf{1}^T p - \sigma_R(R^2 - x^T x) - \sigma_T(Tt - t^2) \text{ is SOS,} \\ & && V_N(T, x) - \sigma_V V_{N-1}(0, r(x, s, \Lambda)) - \sigma_s(1 - s^2) - q_h h \text{ is SOS,} \\ & && p_i - \frac{\partial V_N}{\partial x} g_i - \sigma_{Rp,i}(R^2 - x^T x) - \sigma_{Tp,i}(Tt - t^2) \text{ is SOS for } i = 1, \dots, m, \\ & && p_i + \frac{\partial V_N}{\partial x} g_i - \sigma_{Rn,i}(R^2 - x^T x) - \sigma_{Tn,i}(Tt - t^2) \text{ is SOS for } i = 1, \dots, m, \\ & && W \text{ is SOS,} \\ & && W - V_N(0, x) - 1 \text{ is SOS,} \\ & && \sigma_R, \sigma_T, \sigma_V, \sigma_s, \sigma_{Rp,i}, \sigma_{Rn,i}, \sigma_{Tp,i}, \sigma_{Tn,i} \text{ are SOS,} \end{aligned}$$

where q_h and the σ 's are multipliers. Solutions to (O2) provide an outer approximation of the N -step viable-capture basin as the 0-superlevel set of $V(0, x)$. Tightness of these approximations is governed by the total polynomial degree used for V and W , with convergence results found in [46].

Table 7.1: Example model properties

	Value		Value		Value
T	0.3 s	\bar{z}_{cm}	1 m	J	0.125 m ²
r_{step}	0.7 m	$U_{z,max}$	5 m/s ²	$U_{x,max}$	1 m/s ²
r_{foot}	0.05 m	z_{max}	0.5 m	θ_{max}	90 deg

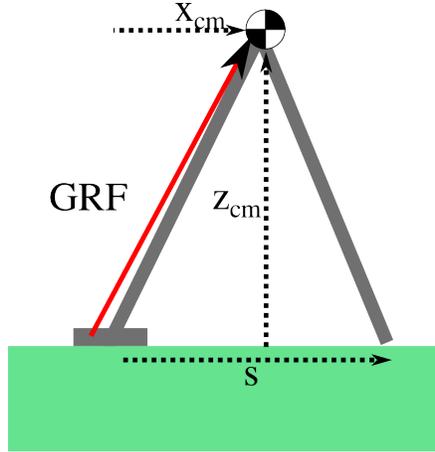


Figure 7-2: Here, z_{cm} remains constant for the LIPM, and the ground reaction forces are chosen to point from the center of pressure on the foot toward the center of mass.

7.4 Example Applications

The approach is demonstrated on four examples, computing inner and outer approximations to the 0-step and 1-step basins. These examples explore the effects of three of the primary assumptions made in the standard LIPM. Solutions are generated in MATLAB, using Spotless [136] to formulate SOS programs and MOSEK [90] to solve the resulting semidefinite optimizations. Depending on model complexity, solutions were computed over a period of minutes to hours on a desktop computer. Comparisons across models for the balancing regions are also illustrated at the end of the section in Figure 7-9. Model properties were chosen to emulate those of a prototypical walking robot and are listed in Table 7.1. Sampling and numerical simulations, not depicted, have also been used as evidence that the true viable-capture basins lie between the inner and outer approximations.

7.4.1 LIPM

For validation, we present approximations for the basic planar LIPM (constant center of mass height, \bar{z}_{cm}), where the true viable-capture basins are explicitly known. The ground reaction forces (GRF) of the LIPM are constrained so that $\ddot{z}_{cm} = 0$, and the angular momentum is also constant. A cartoon illustration of the model is seen in Figure 7-2. A single control input, u_1 , governs the location of the center of pressure with respect to the robot's foot, and is bounded by the foot radius r_{foot} . The model has one degree of freedom, $x = \begin{bmatrix} x_{cm} \\ \dot{x}_{cm} \end{bmatrix}$, with dynamics.

$$\ddot{x}_{cm} = \frac{g}{\bar{z}_{cm}}(x_{cm} + r_{foot}u_1.) \quad (7.9)$$

Stepping, up to distance r_{step} , occurs without impact, and so the reset map is given by

$$r(x_-, s) = \begin{bmatrix} x_{cm-} - r_{step}s \\ \dot{x}_{cm-} \end{bmatrix}. \quad (7.10)$$

Results are illustrated in Figure 7-3, where the balancing and 1-step approximations are nearly identical to the explicit calculations from Koolen et al. [58]. For this simple model, both inner and outer approximations do an excellent job of capturing the true viable-capture basins with minimal gap.

7.4.2 Variable Height

As a first extension, we relax the assumption that $\ddot{z}_{cm} = 0$, replacing it with the control authority to generate bounded vertical accelerations. The resulting model has two degrees of freedom, x_{cm} and z_{cm} , along with a second control input that determines the vertical acceleration. The equations of motion are:

$$\ddot{x}_{cm} = \frac{g + U_{z,max}u_2}{z_{cm}}(x_{cm} + r_{foot}u_1), \quad (7.11)$$

$$\ddot{z}_{cm} = U_{z,max}u_2 \quad (7.12)$$

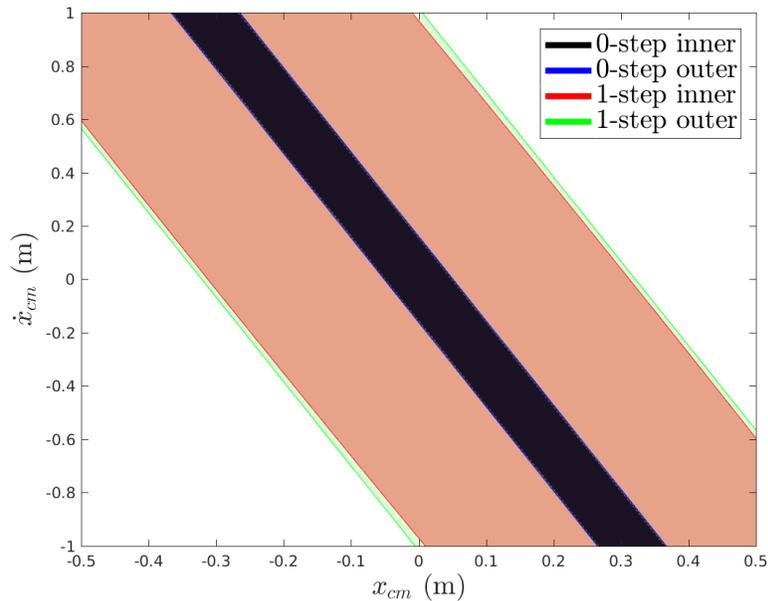


Figure 7-3: There is a minimal gap between inner and outer approximations for the LIPM. Additionally, the explicitly known viable-capture basins are visually indistinguishable from the outer regions, and so are not plotted here.

Note that (7.11) is not natively polynomial in x and control affine, and so we approximate using a quadratic Taylor expansion of z_{cm}^{-1} about the nominal height (\bar{z}_{cm}) and linear expansion in the control input variables (eliminating the comparatively small second-order term $u_1 u_2$). Additionally, we enforce the physical constraint that the variation in height satisfy $|z_{cm} - \bar{z}_{cm}| \leq z_{max}$, defining a corresponding unsafe region. As with the LIPM, this is a zero impact model, and so the reset dynamics remain unchanged from (7.10), with the addition that $z_{cm+} = z_{cm-}$ and $\dot{z}_{cm+} = \dot{z}_{cm-}$.

Figure 7-4 illustrates a slice of the the viable-capture regions for this model, where $z_{cm} = \bar{z}_{cm}$ and $\dot{z}_{cm} = 0$, demonstrating the marginal improvement in control authority gained by this additional control authority. Koolen et al. [59] also analyzed a variable height model, for balancing only, although there are a few key differences between that work and this. The approach there was able to exactly calculate the 0-step viable-capture basin, without approximation. However, their model was significantly more permissive in both input and state constraints, using the bounds $\ddot{z}_{cm} \geq 0$ and $z_{cm} \geq 0$ with no upper bounds. The ability to include non-zero limitations on inputs and states largely accounts for the more limited

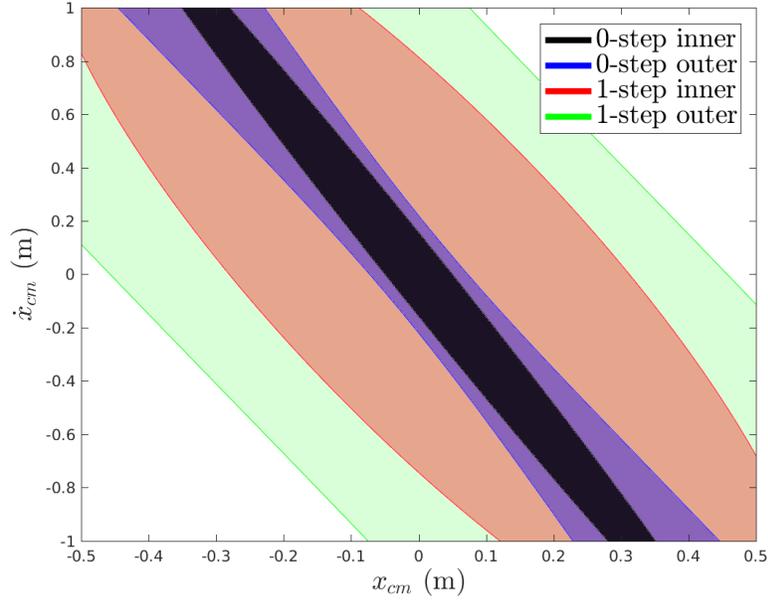


Figure 7-4: The ability to vary the vertical COM acceleration has a small effect on the viable-capture basins, though the gap between inner and outer approximations is larger for the higher dimensional model.

benefits to variable height that are demonstrated here.

7.4.3 Incorporating Impact Dynamics

As presented in Section 7.3, we are also able to include impact dynamics in our models. Using an assumption of a massless leg, an impact generates an impulsive force from the landing foot through the center of mass, illustrated in Figure 7-5. The impulse must satisfy the post-impact constraint that the center of mass velocity be orthogonal to the new stance leg. The reset map and reset constraints are therefore

$$x_+ = r(x_-, s, \Lambda) = \begin{bmatrix} x_{cm-} - r_{step}S \\ z_{cm-} \\ \dot{x}_{cm-} + \Lambda(x_{cm-} - r_{step}S) \\ \dot{z}_{cm-} + \Lambda z_{cm-} \end{bmatrix}, \quad (7.13)$$

$$h(x_-, s, \Lambda) = (\dot{x}_{cm+}x_{cm+}) + (\dot{z}_{cm+}z_{cm+}). \quad (7.14)$$

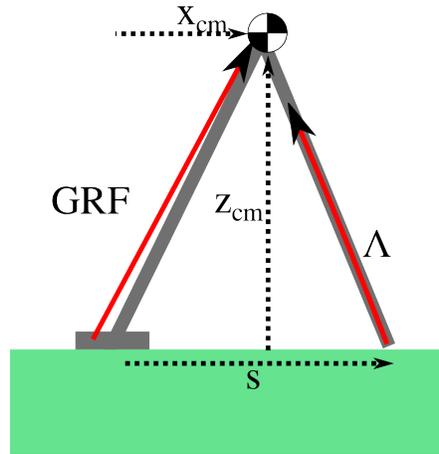


Figure 7-5: An impulsive force during footstrike causes an instantaneous change in the horizontal and center of mass velocities.

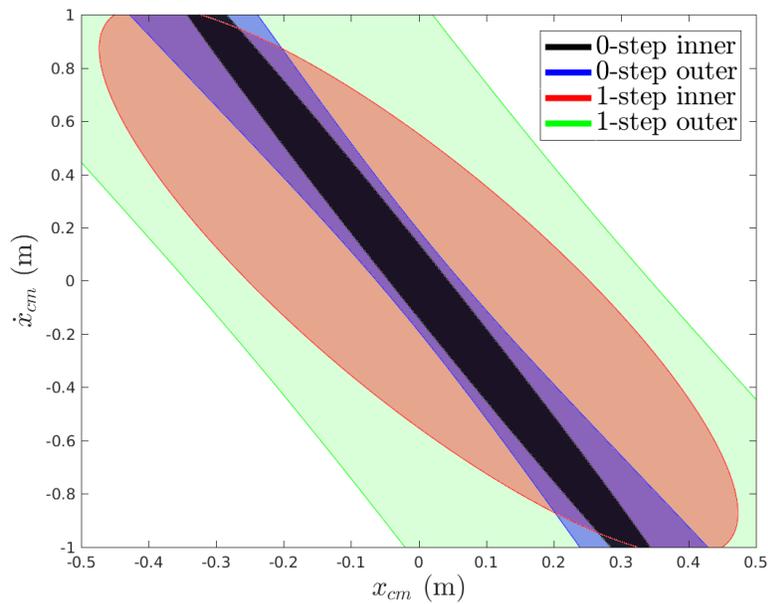


Figure 7-6: A slice of the viable-capture basins for a variable height model with impact dynamics.

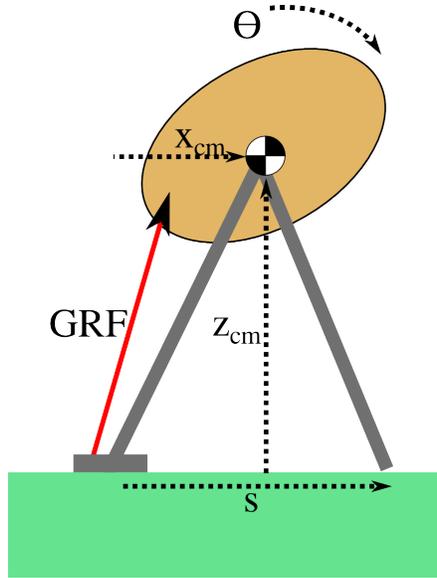


Figure 7-7: With variable angular momentum, the ground reaction forces are no longer required to point directly at the COM.

As before, a slice of the viable-capture basins is shown in Figure 7-6. While one might expect the impacts, by dissipating energy, to have a strong stabilizing effect, there appears to only be a minimal increase in capturability.

7.4.4 Variable Height and Rotational Inertia

The final model under consideration incorporates variable rotational inertia to the variable height model of 7.4.2. Rotational inertia is captured in a reaction wheel style model, similar to the one studied in [58]. An additional degree of freedom, θ , captures the orientation of the reaction wheel with a third input that governs the additional lateral acceleration. Parameter J encodes the ratio between the moment of inertia and mass. As shown in Figure 7-7, the ground reaction force is no longer limited to point directly at the center of mass. The limit on the third input, in Table 7.1, realistically restricts the angular acceleration to 8 rad/s^2 . Safety constraints additionally enforce that $|\theta| \leq \theta_{max}$, to capture the fact that the robot

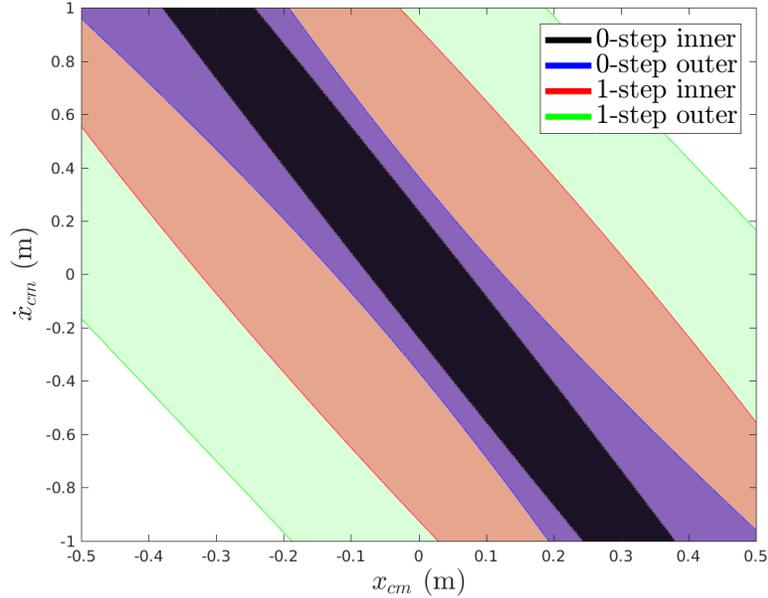


Figure 7-8: The viable-capture regions for a model including variable height and inertia illustrate significantly larger regions than with simpler approaches, particularly the 0-step basin. However, with increased model complexity, the gap between inner and outer approximations is also larger.

torso is restricted in its range of movement. The nominal equations of motion are

$$\ddot{x}_{cm} = \frac{g + U_{z,max}u_2}{z_{cm}}(x_{cm} + r_{foot}u_1) + U_{x,max}u_3, \quad (7.15)$$

$$\ddot{z}_{cm} = U_{z,max}u_2, \quad (7.16)$$

$$\ddot{\theta} = \frac{z_{cm}}{J}U_{x,max}u_3. \quad (7.17)$$

As with (7.15), we use Taylor approximations to pose a polynomial, control-affine problem. Figure 7-8 illustrates the viable-capture basins for this model, demonstrating a significant gap in capturability as compared with simpler approaches (see Figure 7-9). However, even with these additional degrees of freedom, stepping remains the only mechanism for recovering from large disturbances. For this higher dimensional model with larger optimization problems, the gap between inner and outer approximations is also larger. Computational considerations limited the outer approximations to fourth degree polynomials, whereas the other examples were carried out with polynomials of degree six and above.

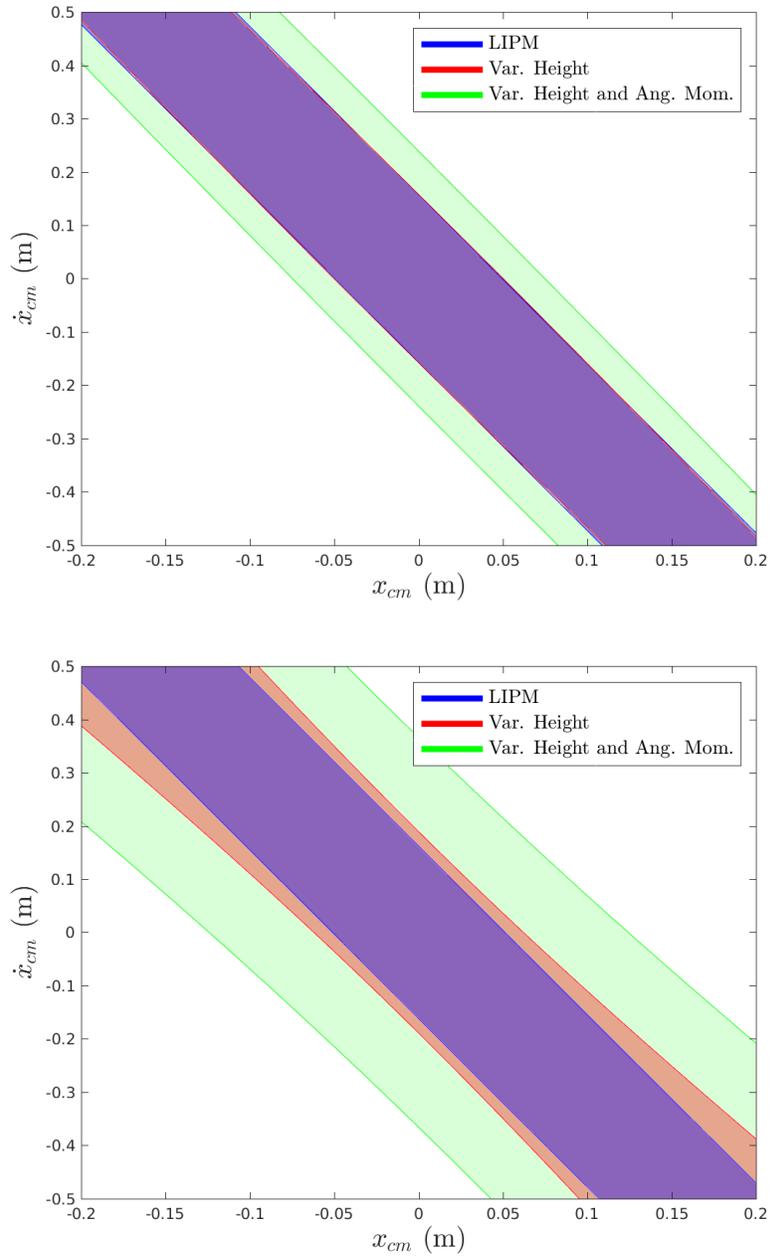


Figure 7-9: Comparisons between inner (top) and outer (bottom) approximations to the 0-step basins are shown. While neither modification greatly expands the capture basin beyond that of the LIPM, the effect of angular momentum is noticeably greater than that of vertical acceleration.

7.5 Discussion

The ability to formally analyze multiple walking models enables a deeper understanding of the advantages and limitations of different control approaches. The results above explicitly bound the potential benefits from leveraging center of mass height, angular momentum, and impact dynamics, demonstrating the comparatively larger effect of angular momentum. Furthermore, these optimization tools also offer algorithmic approaches to control synthesis. Along with the inner approximations, we have designed bang-bang control policies with provable guarantees of performance. The barrier functions themselves could be also be used in a similar manner to that of control Lyapunov functions—describing a broader set of control actions which are also provably effective. Future work will explore the effectiveness of these policies along with alternate control implementations. The occupation measure methods for outer approximations have also led to work on control synthesis [80, 60]. Future directions also include analysis of additional walking models. For example, models which capture the left-right asymmetry foot placement for lateral stability and as well as an examination of swing-leg dynamics.

7.A Appendix

Details for expressing (7.7) as SOS constraints are given here. While not strictly necessary, it is efficient to leverage the fact that $W(x, s)$ is a convex quadratic in s . Observe that (7.7) can be equivalently written as the statement

$$\text{If } W(x, s) = 1 \Rightarrow |s| > 1, \text{ then } V_N(T, x) \geq \rho_N(T). \quad (7.18)$$

Rewriting $W(x, s) - 1 = as^2 + b(x)s + c(x)$, where $a > 0$ is a scalar, we solve for s and can manipulate the quadratic formula to express the condition $|s| > 1$ as

$$\begin{aligned} &(-b^2 + 4ac > 0) \vee \dots \\ &((b - 2a > 0) \wedge (a - b + c > 0)) \vee \dots \\ &((-b - 2a > 0) \wedge (a + b + c > 0)) \end{aligned}$$

Defining $V_T(x) := V_N(x, T)$ and $\rho_T := \rho_N(x, T)$, these conditions are naturally incorporated via the S-procedure as additional SOS constraints in (B):

$$\begin{aligned} &q_{V,1}(V_T - \rho_T) - \sigma_{r,1}(4ac - b^2) \text{ is SOS,} \\ &q_{V,2}(V_T - \rho_T) - \sigma_{r,2}(-2a + b) - \sigma_{r,3}(a - b + d) \text{ is SOS,} \\ &q_{V,3}(V_T - \rho_T) - \sigma_{r,4}(-2a - b) - \sigma_{r,5}(a + b + d) \text{ is SOS,} \end{aligned}$$

where $q_{V,i}$ are fixed multipliers (like q_w) and the σ 's are new S-procedure multipliers.

Chapter 8

Conclusion

In this thesis, we provide computationally tractable methods for control and planning of robotic systems making and breaking contact with the environment. To address these fundamentally hard problems, we leverage the natural interplay between the measure differential inclusion models of non-smooth dynamics, nonlinear control theory, and modern optimization algorithms. The primary challenges related to acting in contact-rich environments result from the discontinuities induced by impact events and frictional forces. While these discontinuities can be represented with the language of hybrid dynamical systems, use of this general purpose framework risks neglecting the structure provided by mechanical systems.

In Chapters 4 and 6, we avoid the typical hybrid-systems framework, instead exploiting the complementarity and algebraic structure of the MDI formalism. In Chapter 4, we introduce a contact-implicit trajectory optimization algorithm capable of synthesizing new trajectories, where the algorithm implicitly searches over the set of possible contact sequences. This method is inspired by, and appropriate for, situations with a large number of potential contacts where a precise sequencing of contact events is an unrealistic expectation. We demonstrate the contact-implicit algorithm on a number of examples, scaling to models with dozens of joints and dozens of potential contacts. When it came to executing motions generated in this fashion, a primary obstacle lay in the first-order numerical integration scheme embedded in the algorithm. Therefore, given a nominal motion and contact sequence, we look to further refine optimal trajectories. While there is an extensive literature of trajectory optimization algorithms for smooth or hybrid systems, standard ap-

proaches do not address the manifold constraints that arise from closed kinematic chains (like a four-bar linkage or double support). In Chapter 5, we introduce an extension to the direct collocation algorithm to perform trajectory optimization constrained to a kinematic manifold. The DIRCON approach maintains third-order accuracy while guaranteeing the manifold constraints.

In addition to trajectory optimization, we have also investigated questions of local control synthesis and stability analysis. In Chapter 6, we embed the algebraic structure of implicit contact mechanics into sums-of-squares algorithms for Lyapunov analysis and control design. By doing so, we reason through contact events (including the possibility of non-unique solutions) to optimize over control policies and generate numerical proofs of stability and invariance. The presented algorithm avoids explicit reasoning about the *combinations* of contacts, and therefore scales tractably in contact-rich settings.

Looking beyond full, rigid-body dynamical description, we lastly analyze some of the simpler, centroidal momentum-based models that have proven to be enduring and effective tools for control of bipedal robots. In Chapter 7, we explore the fruitful middle ground between fully articulated models (intractable for formal, SOS analysis) and the simplified, linear models (amenable to a variety of classical techniques). We describe SOS approaches to compute both inner and outer approximations to the N -step viable capture basins. This allows formal analysis of the balancing and push recovery capabilities of control strategies that incorporate angular momentum, center of mass height, and impact dynamics.

8.1 Challenges and Open Questions

8.1.1 Computational Challenges

As with any numerical algorithm, computational scaling is a critically important topic. While our approaches above scale quite favorably when compared to prior art, issues like computational runtime, memory usage, and numerical stability continue to pose challenges. Both the trajectory optimization and sums-of-squares algorithms are offline procedures, with runtimes ranging from tens of seconds to hours. The mathematical program with complementarity

constraint formulation used in Chapter 4 carries with it significant numerical complexity. While mitigating approaches exist, as discussed in 4.1.2, for particularly high dimensional problems (or long trajectories) we often struggle to converge to high quality solutions. With the understanding that these nonlinear programs are limited to local searches, the challenge remains to reliably identify good solutions for the larger optimization problems.

While interior point algorithms can solve semidefinite programs in polynomial time (to numerical accuracy), both memory usage and numerical stability limit the dimensionality and the degree of the polynomials used in SOS methods. It should be noted that SDP remains a relatively new and active area of research, and it is reasonable to expect significant improvement in the coming years. For example, the introduction of the commercial MOSEK SDP solver [90] has had a significant impact since its recent release. Numerical challenges are especially critical for bilinear alternations, where the numerical tolerance in a given solution can lead to problems in later iterations. While we make progress in this regard, discussed in 7.3.1.1, stable alternations are by no means guaranteed for all problems. As with trajectory optimization, bilinear SOS problems require local solutions. Principled approaches to choosing initial seeds that improve solution quality remains an open area of interest.

8.1.2 Reliance Upon Models

The work in this thesis falls under the class of model-based control and planning, where we assume the existence of an accurate (and known) dynamical model. However, despite one’s best effort to identify such a model, there will doubtless remain unmodeled phenomena and other errors. Despite this limitation, models have played a major role in the practical implementation of robotic control policies. As but one example, many successful teams at the DARPA Robotics Challenge based their strategies on a rigid-body model (e.g. [33, 62, 57]). In this work, contact between rigid bodies has occurred at a finite set of points. Toward relaxing this assumption, recent work has explored the boundary between data-driven approaches and rigid-body models to characterize surface-surface contact [148]. This and similar approaches might be incorporated into the algorithms presented here.

Nonetheless, an rigorous treatment of model uncertainty (both stochastic and parametric), particularly for systems in contact, remains an open problem. In the context of the

work presented here, the existence of Lyapunov and barrier functions provides some notion of robustness. However, we do not explicitly address model uncertainty in this thesis. Possible extensions to our work might directly incorporate bounded uncertainty, as in Aylward et al. [7], or stochastic models as in Steinhardt and Tedrake [124]. Similarly, planning and trajectory optimization with uncertainty remains a challenging task, though methods based on simple models [17, 144] and sampling [28, 29, 88, 51] have recently been developed.

8.1.3 Real-time Control

In the final chapters of this thesis, we have presented approaches for feedback policy synthesis in a few different settings. The methods in Chapter 6 represent the most comprehensive algorithm for control design through contact, generating policies valid in a neighborhood of states that include all possible contact modes. However, this work currently scales only to systems with a modest number of degrees of freedom. While the LQR and QP based methods in Chapter 5 do address high-dimensional systems, these approaches struggle to handle unplanned contact. Stabilization in the presence of contact uncertainty, particularly contact modes that are *not* part of the original plan, remains an open problem. While the QP controller reasons about the current contact state, the cost-to-go function from LQR provides no useful information in directions normal to the planned manifold.

While we have made notable progress in this area, all of these methods rely upon offline computation to design controllers for specific tasks and environments. To enable robots to succeed at practical dynamic tasks, algorithms capable of real-time control in a previously unknown environment are required. The QP approach is naturally extended in a contact-implicit form, as a quadratic program with complementarity constraints (QPCC) or as a mixed-integer quadratic program (MIQP). However, both of these formulations are computationally expensive and cannot currently be solved at practically useful speeds. Model predictive control approaches, based on smoothed approximations to the discontinuous dynamics, have proven successful in simulation [63, 132]. However, it remains unknown whether these approximations result in experimentally valid solutions. An exciting direction for future research is to develop algorithms, that scale to high-dimensional systems, capable of making real-time decisions to make and break contact as necessary in contact-rich settings.

Bibliography

- [1] V. Acary and B. Brogliato. *Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics*, volume 35. Springer, 2008.
- [2] B. D. O. Anderson and J. B. Moore. *Optimal control: linear quadratic methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990. ISBN 0-13-638560-5.
- [3] M. Anitescu. On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 15(4):1203–1236, 2005.
- [4] M. Anitescu and F. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14(3): 231–247, 1997.
- [5] J.-P. Aubin and A. Cellina. *Differential inclusions: set-valued maps and viability theory*, volume 264. Springer Science & Business Media, 2012.
- [6] J.-P. Aubin, J. Lygeros, M. Quincampoix, S. Sastry, and N. Seube. Impulse differential inclusions: A viability approach to hybrid systems. *IEEE Transactions on Automatic Control*, 47(1):2–20, 2002. ISSN 00189286.
- [7] E. M. Aylward, P. A. Parrilo, and J.-J. E. Slotine. Stability and robustness analysis of nonlinear systems via contraction metrics and sos programming. *Automatica*, 44(8): 2163 – 2170, 2008.
- [8] S. Berard, B. Nguyen, K. Anderson, and J. Trinkle. Sources of error in a simulation of rigid parts on a vibrating rigid plate. *ASME Journal of Computational and Nonlinear Dynamics*, 5(4):041003–1, 2010.
- [9] D. P. Bertsekas. *Dynamic Programming & Optimal Control*, volume I and II. Athena Scientific, 3rd edition, May 1 2005. ISBN 1886529264.
- [10] J. T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM Advances in Design and Control. Society for Industrial and Applied Mathematics, 2001.
- [11] V. Bhatt and J. Koechling. Three-dimensional frictional rigid-body impact. *Journal of Applied Mechanics*, 62:893, 1995.

- [12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 9780521833783.
- [13] B. Brogliato. *Nonsmooth mechanics: models, dynamics, and control*. Springer Verlag, 1999.
- [14] F. Bullo and A. D. Lewis. *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*. Texts in Applied Mathematics. Springer, Nov 4 2004. ISBN 0387221956.
- [15] M. Buss, M. Hardt, J. Kiener, M. Sobotka, M. Stelzer, O. von Stryk, and D. Wollherr. Towards an Autonomous, Humanoid, and Dynamically Walking Robot. In *Proceedings of the 3rd International Conference on Humanoid Robotics*, pages 2491–2496, 2003.
- [16] K. Byl and R. Tedrake. Approximate optimal control of the compass gait on rough terrain. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2008.
- [17] K. Byl and R. Tedrake. Metastable walking machines. *International Journal of Robotics Research*, 28(8):1040–1064, August 1 2009.
- [18] A. Chatterjee and A. Ruina. A new algebraic rigid-body collision law based on impulse space considerations. *Journal of Applied Mechanics*, 65(4):939–951, 1998.
- [19] C. Chen and O. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications*, 5(2):97–138, 1996.
- [20] K. B. Cheng, Y.-C. Huang, and S.-Y. Kuo. Effect of arm swing on single-step balance recovery. *Human movement science*, 38:173–184, 2014.
- [21] S. Chernova and M. Veloso. An evolutionary approach to gait learning for four-legged robots. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2562–2567. IEEE, 2004.
- [22] M. J. Coleman, A. Chatterjee, and A. Ruina. Motions of a rimless spoked wheel: a simple 3d system with impacts. *Dynamics and Stability of Systems*, 12(3):139–160, 1997.
- [23] S. H. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, February 18 2005.
- [24] R. Cory and R. Tedrake. Experiments in fixed-wing UAV perching. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 1–12. AIAA, 2008.
- [25] R. W. Cottle, J.-S. Pang, and R. E. Stone. *The linear complementarity problem*, volume 60. Siam, 2009.

- [26] S. Cotton, I. Oлару, M. Bellman, T. van der Ven, J. Godowski, and J. Pratt. Fastrunner: A fast, efficient and robust bipedal robot. concept and planar simulation. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [27] E. A. Cross and I. M. Mitchell. Level set methods for computing reachable sets of systems with differential algebraic equation dynamics. In *American Control Conference, 2008*, pages 2260–2265. IEEE, 2008.
- [28] H. Dai and R. Tedrake. Optimizing robust limit cycles for legged locomotion on unknown terrain. In *Proceedings of the IEEE Conference on Decision and Control*, page 8, Maui, Hawaii, December 2012.
- [29] H. Dai and R. Tedrake. L2-gain optimization for robust bipedal walking on unknown terrain. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [30] A. L. Desbiens, A. Asbeck, and M. Cutkosky. Hybrid aerial and scansorial robotics. *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 72–77, May 2010.
- [31] T. Erez and E. Todorov. Trajectory optimization for domains with contacts using inverse dynamics. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4914–4919. IEEE, 2012.
- [32] R. Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, January 1987. ISBN 0898382300.
- [33] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson. Optimization-based full body control for the darpa robotics challenge. *Journal of Field Robotics*, 32(2):293–312, 2015.
- [34] A. Fischer. A special newton-type optimization method. *Optimization*, 24(3-4):269–284, 1992.
- [35] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes. Local convergence of SQP methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 17(1):259–286, 2006.
- [36] G. B. Folland. How to integrate a polynomial over a sphere. *The American Mathematical Monthly*, 108(5):446–448, 2001.
- [37] M. Fukushima, Z. Luo, and J. Pang. A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints. *Computational Optimization and Applications*, 10(1):5–34, 1998.
- [38] T. Geijtenbeek, M. van de Panne, and A. F. van der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)*, 32(6):206, 2013.

- [39] 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.
- [40] J. H. Gillula, G. M. Hoffmann, H. Huang, M. P. Vitus, and C. Tomlin. Applications of hybrid reachability analysis to robotic aerial vehicles. *The International Journal of Robotics Research*, 2011.
- [41] E. L. Glassman, A. L. Desbiens, M. Tobenkin, M. Cutkosky, and R. Tedrake. Region of attraction estimation for a perching aircraft: A Lyapunov method exploiting barrier certificates. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [42] A. Goswami and V. Kallem. Rate of change of angular momentum and balance maintenance of biped robots. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 4:3785–3790, Apr 2004.
- [43] R. D. Gregg, A. K. Tilton, S. Candido, T. Bretl, and M. W. Spong. Control and planning of 3-d dynamic walking with asymptotically stable gait primitives. *IEEE Transactions on Robotics*, 28(6):1415–1423, 2012.
- [44] E. Hairer and G. Wanner. Solving ordinary differential equations ii: Stiff and differential-algebraic problems. *Springer series in computational mathematics*, 14, 1996.
- [45] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *J Guidance*, 10(4):338–342, July-August 1987.
- [46] D. Henrion and M. Korda. Convex computation of the region of attraction of polynomial control systems. *IEEE Transactions on Automatic Control*, 59(2):297–312, 2014.
- [47] D. Henrion, J. Lasserre, and C. Savorgnan. Approximate volume and integration for basic semialgebraic sets. *SIAM Review*, 51(4):722–743, 2009.
- [48] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames. 3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *IEEE International Conference on Robotics and Automation (ICRA)*. *IEEE*, 2016.
- [49] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard. Some controls applications of sum of squares programming. In *42nd IEEE Conference on Decision and Control*, volume 5, pages 4676 – 4681, December 2003.
- [50] K. H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry. On the regularization of zeno hybrid automata. *Systems & Control Letters*, 38(3):141–150, 1999.
- [51] A. M. Johnson, J. E. King, and S. Srinivasa. Convergent planning. *IEEE Robotics and Automation Letters*, 1(2):1044–1051, 2016.

- [52] C. Jozs and D. Henrion. Strong duality in lasserre’s hierarchy for polynomial optimization. *Optimization Letters*, 10(1):3–10, 2016.
- [53] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. volume 2, pages 1405 – 1411. IEEE International Conference on Robotics and Automation (ICRA), 1991.
- [54] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. pages 239 – 246. IEEE International Conference on Intelligent Robots and Systems (IROS), 2001.
- [55] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai. Staggered projections for frictional contact in multibody systems. In *ACM Transactions on Graphics (TOG)*, volume 27, page 164. ACM, 2008.
- [56] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, December 2001. ISBN 0130673897.
- [57] Koolen, Twan, Bertrand, Sylvain, Thomas, Gray, D. Boer, Tomas, Wu, Tingfan, Smith, Jesper, Engelsberger, Johannes, Pratt, and Jerry. Design of a momentum-based control framework and application to the humanoid robot atlas. Aug 2015.
- [58] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt. Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models. *The International Journal of Robotics Research*, 31(9):1094–1113, 2012.
- [59] T. Koolen, M. Posa, and R. Tedrake. Balance control using center of mass height variation: limitations imposed by unilateral contact. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, pages 8–15. IEEE, 2016.
- [60] M. Korda, D. Henrion, and C. N. Jones. Controller design and region of attraction estimation for nonlinear dynamical systems. *The 19th World Congress of the International Federation of Automatic Control (IFAC)*, 2014.
- [61] S. Kuindersma, F. Permenter, and R. Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *Proceedings of the International Conference on Robotics and Automation*, Hong Kong, China, May 2014. IEEE.
- [62] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake. Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot. *Autonomous Robots*, 40(3):429–455, 2016.
- [63] V. Kumar, Y. Tassa, T. Erez, and E. Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6808–6815. IEEE, 2014.

- [64] A. D. Kuo. An optimal control model for analyzing human postural balance. *IEEE Transactions on Biomedical Engineering*, 42(1):87–101, January 1995.
- [65] J. Lack, M. J. Powel, and A. D. Ames. Planar multi-contact bipedal walking using hybrid zero dynamics. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2582–2588. IEEE, 2014.
- [66] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal of Optimization*, 11(3):796–817, January 19 2001.
- [67] J. B. Lasserre. *Moments, Positive Polynomials and Their Applications*, volume 1. World Scientific, 2010.
- [68] J.-B. Lasserre, K.-C. Toh, and S. Yang. A bounded degree SOS hierarchy for polynomial optimization. *arXiv preprint arXiv:1501.06126*, 2015.
- [69] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. *Emerging applications of algebraic geometry*, 149:157–270, 2009.
- [70] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [71] R. Leine and N. van de Wouw. *Stability and convergence of mechanical systems with unilateral constraints*. Springer Verlag, 2008.
- [72] M. A. Lewis, A. H. Fagg, and A. Solidum. Genetic programming approach to the construction of a neural network for control of a walking robot. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2618–2623. IEEE, 1992.
- [73] R. Lind, G. Balas, and A. Packard. Evaluating D-K iteration for control design. In *American Control Conference, 1994*, volume 3, pages 2792 – 2797 vol.3, 29 June-1 July 1994.
- [74] J. Lofberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.
- [75] J. Lofberg. Pre- and post-processing sum-of-squares programs in practice. *IEEE Transactions On Automatic Control*, 54(5):1007–, May 2009.
- [76] Z. Luo, J. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.
- [77] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.
- [78] A. Majumdar, A. A. Ahmadi, and R. Tedrake. Control design along trajectories with sums of squares programming. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4054–4061, 2013.

- [79] A. Majumdar, A. A. Ahmadi, and R. Tedrake. Control and verification of high-dimensional systems with DSOS and SDSOS programming. In *Proceedings of the 53rd Conference on Decision and Control (CDC)*, 2014.
- [80] A. Majumdar, R. Vasudevan, M. M. Tobenkin, and R. Tedrake. Convex optimization of nonlinear feedback controllers via occupation measures. *International Journal of Robotics Research (IJRR)*, 33(9):1209–1230, August 2014.
- [81] I. R. Manchester. Transverse dynamics and regions of stability for nonlinear hybrid limit cycles. *Proceedings of the 18th IFAC World Congress, extended version available online: arXiv:1010.2241 [math.OC]*, Aug-Sep 2011.
- [82] H. N. McClamroch and D. Wang. Feedback stabilization and tracking of constrained robots. *Automatic Control, IEEE Transactions on*, 33(5):419–426, 1988.
- [83] A. Megretski. Systems polynomial optimization tools (SPOT), available online: <http://web.mit.edu/ameg/www>. 2010.
- [84] I. M. Mitchell and Y. Susuki. Level set methods for computing reachable sets of hybrid systems with differential algebraic equation dynamics. In *International Workshop on Hybrid Systems: Computation and Control*, pages 630–633. Springer, 2008.
- [85] K. D. Mombaur. Using optimization to create self-stable human-like running. *Robotica*, 27(3):321–330, 2009.
- [86] I. Mordatch, Z. Popović, and E. Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 137–144. Eurographics Association, 2012.
- [87] I. Mordatch, E. Todorov, and Z. Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4):43, 2012.
- [88] I. Mordatch, K. Lowrey, and E. Todorov. Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5307–5314. IEEE, 2015.
- [89] J. J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In *Nonsmooth mechanics and Applications*, pages 1–82. Springer, 1988.
- [90] A. Mosek. The mosek optimization software. *Online at <http://www.mosek.com>*, 54, 2010.
- [91] C. Mummolo, L. Mangialardi, and J. H. Kim. Numerical estimation of balanced and falling states for constrained legged systems. *Journal of Nonlinear Science*, pages 1–33, 2017.
- [92] Y. Or and A. Ames. Stability and completion of Zeno equilibria in Lagrangian hybrid systems. *IEEE Transactions on Automatic Control*, 56(6):1322–1336, June 2011.

- [93] A. Papachristodoulou and S. Prajna. Robust stability analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 54(5):1035–1041, May 2009.
- [94] P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, May 18 2000.
- [95] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.
- [96] F. Permenter and P. A. Parrilo. Dimension reduction for semidefinite programs via jordan algebras. *arXiv preprint arXiv:1608.02090*, 2016.
- [97] F. Permenter, H. A. Friberg, and E. D. Andersen. Solving conic optimization problems via self-dual embedding and facial reduction: a unified approach. *Optimization Online*, September, 2015.
- [98] M. Posa and R. Tedrake. Direct trajectory optimization of rigid body dynamical systems through contact. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, pages 527–542, Cambridge, MA, June 2012.
- [99] M. Posa, M. Tobenkin, and R. Tedrake. Lyapunov analysis of rigid body systems with impacts and friction via sums-of-squares. In *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control (HSCC 2013)*, pages 63–72. ACM, April 8-11 2013.
- [100] M. Posa, C. Cantu, and R. Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *International Journal of Robotics Research*, 33(1):69–81, January 2014.
- [101] M. Posa, S. Kuindersma, and R. Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 1366–1373, Stockholm, Sweden, May 2016.
- [102] M. Posa, M. Tobenkin, and R. Tedrake. Stability analysis and control of rigid-body systems with impacts and friction. *IEEE Transactions on Automatic Control (TAC)*, 61(6):1423 – 1437, June 2016.
- [103] M. Posa, T. Koolen, and R. Tedrake. Balancing and step recovery capturability via sums-of-squares optimization. *To appear in Robotics: Science and Systems*, 2017.
- [104] S. Prajna and A. Rantzer. Convex programs for temporal verification of nonlinear dynamical systems. *SIAM Journal of Control and Optimization*, 46(3):999–1021, 2007.
- [105] S. Prajna, A. Jadbabaie, and G. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415 –1428, Aug 2007.

- [106] J. Pratt. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. PhD thesis, Computer Science Department, Massachusetts Institute of Technology, 2000.
- [107] J. Pratt and G. Pratt. Intuitive control of a planar bipedal walking robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1998.
- [108] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture point: A step toward humanoid push recovery. In *2006 6th IEEE-RAS international conference on humanoid robots*, pages 200–207. IEEE, 2006.
- [109] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.
- [110] M. Quincampoix and V. Veliov. Viability with a target: theory and applications. *Applications of Mathematics in Engineering (Sozopol, 1997)(B. Cheshankov and M. Todorov, eds.)*, Heron Press, Sofia, pages 47–54, 1998.
- [111] N. Ratliff, M. Zucker, J. A. D. Bagnell, and S. Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2009.
- [112] C. D. Remy. *Optimal Exploitation of Natural Dynamics in Legged Locomotion*. PhD thesis, ETH ZURICH, 2011.
- [113] C. D. Remy, K. Buffinton, and R. Siegwart. Stability analysis of passive dynamic walking of quadrupeds. *The International Journal of Robotics Research*, 29(9):1173–1185, 2010.
- [114] E. Routh. *Dynamics of a system of rigid bodies*. MacMillan and co. London, 1891.
- [115] C. O. Saglam and K. Byl. Switching policies for metastable walking. In *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 977–983. IEEE, 2013.
- [116] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: Science and Systems*, volume 9, pages 1–10. Citeseer, 2013.
- [117] G. Schultz and K. Mombaur. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics*, 15(5):783–792, Oct. 2010.
- [118] V. Shia, R. Vasudevan, R. Bajcsy, and R. Tedrake. Convex computation of the reachable set for controlled polynomial hybrid systems. In *Proceedings of the 53rd Conference on Decision and Control (CDC)*, page 8, 2014.
- [119] A. Shkolnik, M. Levashov, I. R. Manchester, and R. Tedrake. Bounding on rough terrain with the littledog robot. *The International Journal of Robotics Research (IJRR)*, 30(2):192–215, Feb 2011.

- [120] B. Siciliano and O. Khatib, editors. *Handbook of Robotics*. Springer, 2008. ISBN 978-3-540-23957-4.
- [121] G. V. Smirnov. *Introduction to the theory of differential inclusions*, volume 41. American Mathematical Soc., 2002.
- [122] Sreenath, K., Park, H.W., Poulakakis, I., Grizzle, and JW. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *International Journal of Robotics Research*, 2010.
- [123] M. Srinivasan and A. Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439:72–75, January 5 2006.
- [124] J. Steinhardt and R. Tedrake. Finite-time regional verification of stochastic nonlinear systems. *International Journal of Robotics Research*, 31(7):901–923, June 2012.
- [125] G. Stengle. A nullstellensatz and a positivstellensatz in semialgebraic geometry. *Mathematische Annalen*, 207(2):87–97, 1974.
- [126] B. Stephens. Humanoid push recovery. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 589–595. IEEE, 2007.
- [127] D. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, 2000.
- [128] D. Stewart and J. 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.
- [129] W. Stronge. Rigid body collisions with friction. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 431(1881):169–181, 1990.
- [130] J. F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625 – 653, 1999.
- [131] Y. Tassa and E. Todorov. Stochastic complementarity for local control of discontinuous dynamics. In *Proceedings of Robotics: Science and Systems (RSS)*. Citeseer, 2010.
- [132] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4906–4913. IEEE, 2012.
- [133] R. Tedrake. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems. <http://drake.mit.edu>, 2014.
- [134] R. Tedrake, I. R. Manchester, M. M. Tobenkin, and J. W. Roberts. LQR-Trees: Feedback motion planning via sums of squares verification. *International Journal of Robotics Research*, 29:1038–1052, July 2010.

- [135] M. M. Tobenkin, I. R. Manchester, and R. Tedrake. Invariant funnels around trajectories using sum-of-squares programming. *Proceedings of the 18th IFAC World Congress, extended version available online: arXiv:1010.3013 [math.DS]*, 2011.
- [136] M. M. Tobenkin, F. Permenter, and A. Megretski. Spotless polynomial and conic optimization, available online: <https://github.com/spot-toolbox/spotless>. 2013.
- [137] U. Topcu, A. Packard, and P. Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44(10):2669 – 2675, 2008.
- [138] J. Trinkle, J. Pang, S. Sudarsky, and G. Lo. On dynamic multi-rigid-body contact problems with coulomb friction. *ZAMM-Journal of Applied Mathematics and Mechanics*, 77(4):267–279, 1997.
- [139] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.
- [140] P. L. Várkonyi and Y. Or. Lyapunov stability of a rigid body with two frictional contacts. *arXiv preprint arXiv:1603.09672*, 2016.
- [141] C. W. Wampler and A. J. Sommese. Numerical algebraic geometry and algebraic kinematics. *Acta Numerica*, 20:469–567, 2011.
- [142] K. Wampler and Z. Popović. Optimal gait and form for animal locomotion. In *ACM Transactions on Graphics (TOG)*, volume 28, page 60. ACM, 2009.
- [143] Y. Wang and M. T. Mason. Two-dimensional rigid-body collisions with friction. *ASME Journal of Applied Mechanics*, 59:635–642, 1992.
- [144] A. Wu and H. Geyer. The 3-d spring–mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments. *IEEE Transactions on Robotics*, 29(5):1114–1124, 2013.
- [145] W. Xi and D. C. Remy. Optimal gaits and motions for legged robots. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3259–3265. IEEE, 2014.
- [146] K. Yunt and C. Glocker. Trajectory optimization of mechanical hybrid systems using sumt. In *Advanced Motion Control, 2006. 9th IEEE International Workshop on*, pages 665–671. IEEE, 2005.
- [147] P. Zaytsev, S. J. Hasaneini, and A. Ruina. Two steps is enough: no need to plan far ahead for walking balance. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6295–6300. IEEE, 2015.
- [148] J. Zhou, R. Paolini, J. A. Bagnell, and M. T. Mason. A convex polynomial force-motion model for planar sliding: Identification and application. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 372–377. IEEE, 2016.