A Contact-Implicit Direct Trajectory Optimization Scheme for the Study of Legged Maneuverability

UNIVERSITY OF CAPE TOWN
DEPT. OF ELECTRICAL ENGINEERING

Presented by:
Stacey Shield

Prepared for:
A/Prof. Amir Patel

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Doctor of Philosophy degree in Engineering.

July 1, 2022

Acknowledgments

Sometimes the light's all shinin' on me
Other times, I can barely see
Lately, it occurs to me
what a long, strange trip it's been

I will now do my best to express the inexpressible gratitude I feel for my "Fellowship of the Thesis", who supported me in so many ways throughout this long adventure.

Firstly, my supervisor, Amir Patel: thank you for your guidance, encouragement and enthusiasm. Whenever I was stuck in mental quicksand, your undauntable optimism is what made me pull myself out and give the problem one last good go. It has been incredible to watch us all grow over this time: me – from confused undergraduate to fledgling scientist, you – from newly-minted Doctor to esteemed Associate Professor, and our lab – from dusty darkness of the Control Lab storeroom to recognized research unit. I can't wait to see what we'll do next!

I would also like to thank my colleagues in the electrical engineering department, especially Robyn Verrinder and Fred Nicolls, and my co-authors and collaborators, especially Monica Daley. You were the unofficial mentors who taught me so many unwritten lessons about academia, and the sounding board that always bounced ideas back better. I'm delighted to keep working with you in the future.

To my friends from the lab, the Pole Project, the Ferdies Art Group and elsewhere: seeing you is what kept me from turning into a robot myself! Thank you for all the fun distractions, creative outlets, shoulders to cry on and Turkish-delight-flavored energy drinks that kept me going, and for putting up with me through these years of missed plans, deadline-related periods of radio silence and endless whining.

And finally, to my parents: this would never have been possible without your unwavering support, from my first Lego robot to the end of my Ph.D. Thank you for always giving me a home to come back to, and for your patience throughout the most grueling stretches of this degree.

Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signature:
Stacey Leigh Shield

Date: October 16, 2022

A Contact-Implicit Direct Trajectory Optimization Scheme for the Study of Legged Maneuverability

Stacey Shield

Abstract

In recent years, impressive strides have been made in the development of legged robots (both literally and figuratively), allowing the first commercially-available platforms to step out of the lab and into homes and workplaces. The speed and agility of these robots still lags far behind their fastest animal counterparts, however. One reason for this is that non-stationary legged locomotion – particularly at the extremes of speed and rapid execution – remains a largely untapped research area.

Simulation presents an attractive option for making inroads into this field, as it addresses two key challenges:

1. the sensitivity of these maneuvers to wide range of internal and external factors, and
2. the danger of performing these maneuvers experimentally.

In this dissertation, we explore the prospect of studying rapid, high-speed legged maneuvers using trajectory optimization – a simulation method that allows feasible motions to be generated and optimized from almost no *a priori* information. Specifically, we will focus on the *collocation* approach, where the trajectory optimization problem is transcribed to a nonlinear program that can then be solved using an established open-source algorithm.

The first part describes our approach of using trajectory optimization primarily as a synthetic data generation method, with each problem being solved many times from different random starting points to obtain a large set of solutions of varying quality. This collection of local minima can then be analyzed in aggregate, to extract relationships between features-of-interest and solution quality, and identify the characteristics that separate more successful maneuvers from less successful ones. We demonstrate this approach through two case studies: an investigation into the role of arms in assisting gait termination in bipeds, and a general investigation into the rapid termination of

galloping in a fast quadruped.

Solving these large, nonlinear optimization problems hundreds of times from random points is a significant technical challenge that required advances in the problem formulation to better adapt this method to our demanding application. These advances are covered in the second part of the dissertation. The particular areas that we examined include:

- the use of high-order orthogonal collocation methods to construct the trajectory,
- the incorporation of unscheduled unscheduled contact dynamics,
- the coordinates used to model the system, and
- the method of generating random initial seeds.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation: Studying Rapid maneuverability

Fast, dynamic gait is certainly not a solved problem in legged robotics, but compared to maneuverability, it is a well-defined one. Although we have yet to build robots which can execute constant-speed gaits with the same power, speed and robustness as animals do, we have a vocabulary of words to describe them, like "trot", "gallop" and "canter", which at least tell us which foot should touch the ground next. This is more information than we typically have for transient maneuvers.

Consider gait termination, the primary case study for this project. Footage of dogs, cheetahs and horses stopping from high-speed gaits shows the maneuver being performed in various ways (Figure 1.1), but without many more examples, it is impossible to determine whether these differences should be attributed to speed, morphology, external conditions, or the state of the body when the action is initiated. It is also unclear whether these examples represent the *most rapid* way for each animal to stop, as they could also be prioritizing a variety of objectives including energy efficiency, stability and limb safety.

Acquiring enough examples to eliminate all these possible confounding variables would be an immense challenge, however: for fast-moving animals or robots, the sudden stop is an emergency response typically reserved for situations so desperate that the danger inherent in performing the movement is only surpassed by the danger of *not* performing it. Very little experimen-

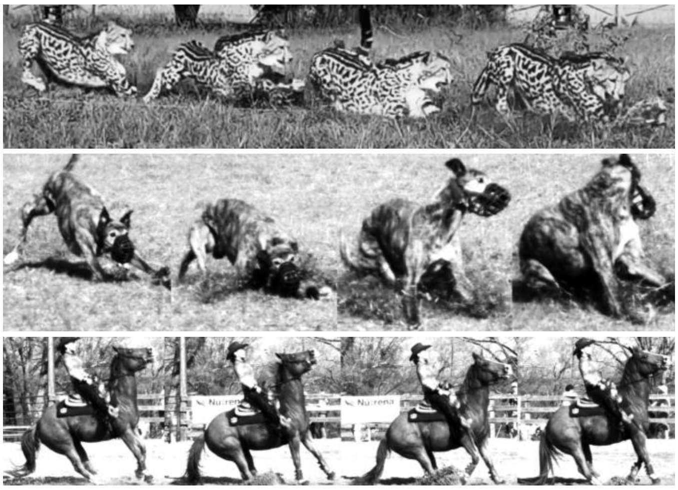

Figure 1.1: Cheetah, greyhound and horse stopping. Footage courtesy of Dr. Robert Gilette and the Canine Performance Sciences Program at the Auburn University College of Veterinary Medicine.

tal research on fast gait termination has been conducted, and consequently, the factors that contribute to successful, rapid execution of this maneuver remain unknown. This is true of factors related to the motion itself (gait termination strategies) and morphological factors.

Although the stated difficulties limit experimental research into rapid, high-speed gait termination, they make it an ideal candidate for simulation-based studies. In simulation, the initial state, environmental conditions and the parameters of the subject's morphology and actuation are entirely known and controllable. There is also no risk of injury or damage as there would if real-life humans, animals or robots were involved.

This does not mean that simulation of legged maneuvers is a trivial task, however: legged systems tend to be high-dimensional, nonlinear, nonholonomic and underactuated - a combination of characteristics that confounds traditional control techniques. They are also hybrid dynamic systems, with nonsmooth transitions occurring between foot contact states. The number of possible contact states increases quadratically with the number of defined contact points, becoming unwieldly for quadrupeds even if the feet are modeled as single contact points.

1.2 The Case for Trajectory Optimization

Trajectory optimization has exciting potential as a method of studying legged maneuverability. Optimal control is widely-applied in legged robotics, as it overcomes the problematic mathematical properties that make these systems difficult to treat with traditional feedback control. Trajectory optimization can be thought of as an extension of optimal control to long time horizons encompassing complete motion tasks.

As a simulation method, it is well-suited to the poorly-defined nature of legged maneuvers, as it allows physically-feasible motions to be generated when neither the pose sequence nor actuation profile is known. This is useful even if the aim is simply *trajectory generation*, but it is made vastly more powerful by the ability to *optimize* the motion according to a given objective. This makes it possible to study the relationship between the motion and high-level goals such as energy efficiency, limb safety or rapid task completion. The locomotion of systems as complex and highly redundant as legged systems can only be meaningfully studied with an optimization objective in mind, so the impossible variety of feasible movements can be limited to the favourable few worth considering.

1.3 Scope and Objectives

Of the many possible methods to optimize locomotion, the ones we refer to as "trajectory optimization" are gradient-based approaches based on optimal control theory. Gradient-free approaches such as genetic algorithms and metaheuristics are often discussed alongside trajectory optimization [13], and have demonstrated applicability to legged locomotion problems such as gait optimization [14], but they will not be discussed in this dissertation. Likewise, learning approaches have been shown to be effective at discovering and improving agile locomotion in legged robots (see: the recent work by Hwangbo et al. [15] and Lee et al. [16] on the ANYmal robot, for instance) but these methods are also outside the scope of this project.

Trajectory optimization is executed using various techniques across a range of research fields [17], but our specific focus is the *transcribed* method, where the trajectory optimization problem is posed as a constrained nonlinear programming problem (CNLP). This has become an especially accessible option for researchers without extensive knowledge of numerical optimization, due to the development of general-purpose commercial and open-source solver algorithms for CNLPs. (A recent review by Malyuta, et al. [18] provides a thorough list and comparison of these solvers in the context of space vehicle control.) This has created an interest in methods of improving the quality of solutions and tractability of problems that can be applied at the level of the problem formulation, without modifying the solver itself.

This project has two aims:

1. to develop a framework for studying rapid legged maneuvers using trajectory optimization, and
2. to design a formulation for trajectory optimization problems that is suitable for this application, for use with an existing open-source solver.

In the design stage, we consider only single-level, deterministic trajectory optimization problems. The focus is primarily on maximizing solver performance and solution quality within a single solving stage. This tight scope excludes some key areas of recent technical development, including

- the inclusion of stochastic elements into the optimization problem for more robust solutions,
- bilevel optimization incorporating model-free methods, or subproblem approaches, and
- multi-stage or iterative trajectory optimization involving stages of increasing complexity.

The aim is to provide a comparison between formulations of the most fundamental problem components, which can provide a helpful base for implementing more complicated trajectory optimization methods in future work.

1.4 Outline and Contributions

1.4.1 Publications

This dissertation covers work that has been presented in eight peer-reviewed publications. A further paper is currently under review.

(A) **Balancing stability and maneuverability during rapid gait termination in fast biped robots (2017)** [19]
Shield, S. and Patel, A.
2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017

(B*) **Contact-implicit trajectory optimization using orthogonal collocation (2019)** [20]
Patel, A., Shield, S., Kazi, S., Johnson, A.M. and Biegler, L.T.
IEEE Robotics and Automation Letters 4.2 (2019): 2242-2249.

(C) **On the effectiveness of silly walks as initial guesses for optimal legged locomotion problems (2020)** [21]
Shield, S. and Patel, A.
2020 International SAUPEC/RobMech/PRASA Conference. IEEE, 2020.

(D*) **The Ollie: A Case Study in Trajectory Optimization with Varied Contacts (2020)** [22]
Anderson, N., Shield, S. and Patel, A.
2020 International SAUPEC/RobMech/PRASA Conference. IEEE, 2020.

(E*) **Minor change, major gains: The effect of orientation formulation on solving time for multi-body trajectory optimization (2020)** [23]
Knemeyer, A., Shield, S. and Patel, A.
IEEE Robotics and Automation Letters 5.4 (2020): 5331-5338.

(F) **Waste Not, Want Not: Lessons in Rapid Quadrupedal Gait Termination from Thousands of Suboptimal Solutions (2020)** [24]
Shield, S. and Patel, A.
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

(G) **Tails, flails and sails: How appendages improve terrestrial maneuverability by improving stability (2021)** [25]
Shield, S., Jericevich, R., Patel, A. and Jusufi, A.
Integrative and Comparative Biology, 61(2) (2021), 506-520.

(H) **Contact-Implicit Direct Collocation with a Discontinuous Velocity State (2022)** [26]
Shield, S., Johnson, A.M. and Patel, A.
IEEE Robotics and Automation Letters 7.2 (2022): 5779-5786.

(I) **Minor Change, Major Gains II: Are Maximal Coordinates the Fastest Choice for Trajectory Optimization?**
Shield, S. and Patel, A.
(SUBMITTED) 2022 IEEE/RSJ International Conference on Intelli-

gent Robots and Systems (IROS). IEEE, 2022

The starred (*) entries are not first-authored, but contain substantial contributions from work described and expanded upon within this dissertation.

1.4.2 Outline

This dissertation is divided into two sections according to the two stated aims of the project: the first focuses on the application of trajectory optimization to the study of legged maneuverability, and the second examines the formulation of transcribed trajectory optimization problems with legged locomotion problems in mind. A more conventional structure might lead with the method rather than its applications, but the order of these chapters better reflects the order in which the work was conducted. Our attempts to advance the problem formulation were primarily motivated by a need to improve the suitability of large-scale trajectory optimization for the repetitive, randomized solving approach we devised, so presenting the work in this way better establishes the context for these investigations. It also removes the need to explain method choices in our early studies that contradict our subsequent findings.

A visual summary of the content of this dissertation is given in Figure 1.2, relating each chapter to elements of the trajectory optimization problem.

Studying Legged Locomotion with Trajectory Optimization

Chapter 2 expands upon the case for trajectory optimization as a method of studying legged maneuverability by reviewing how it has been applied to legged locomotion problems before, and the key advantages that these past uses demonstrate.

Chapter 3 consolidates the advantages established in the previous chapter into a suggested framework for studying legged locomotion using large collections of sample motions generated through trajectory optimization. This approach is demonstrated through two case studies from our work:

1. the use of trajectory optimization to investigate the contribution of arms to bipedal deceleration, originally published in (A) but updated in (G), and

2. the use of iterative trajectory optimization to identify effective strategies for quadrupedal deceleration, published in (F).

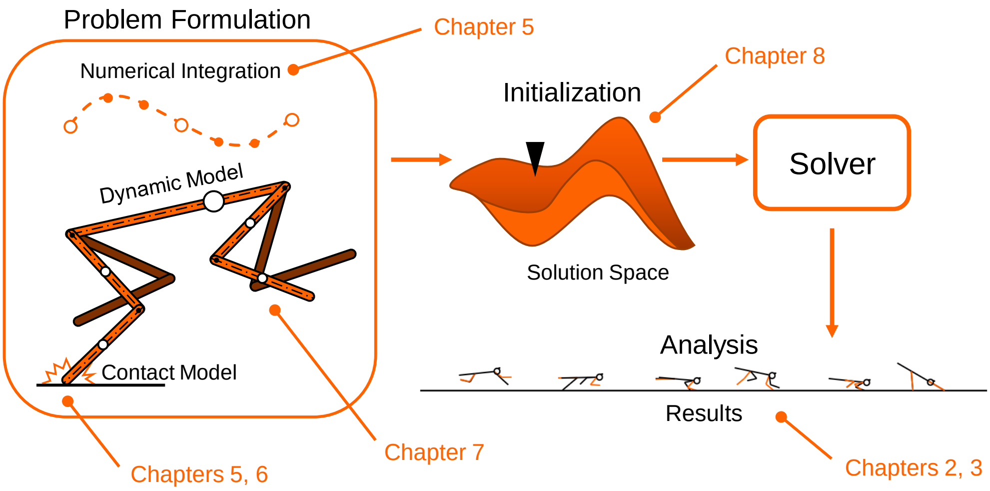

Figure 1.2: Visual summary of the aspects of the trajectory optimization problem covered by each chapter of this dissertation.

The common features of these studies are the use of whole-body models, and the generation of a large dataset of solutions of varying quality, with the aim of extracting information from many solutions in aggregate rather than individual solutions. This approach is inspired by the Monte Carlo framework proposed by Haberland [27], and represents a novel extension of these ideas to questions of strategy rather than morphology.

Formulating Legged Locomotion Problems

Chapter 4 provides the groundwork for the design phase of the project by describing the trajectory optimization problem in detail, and giving a broad overview of possible variations on the method.

Chapters 5 and 6 present our primary contributions regarding the formulation of transcribed trajectory optimization problems for legged locomotion studies. The formulation they describe is the first to expand a complementarity-based implicit contact model to a numerical integration scheme of arbitrary order. It is also the first high-order implicit integration scheme to handle partially-elastic collisions, and impacts without collision (tangential impacts).

Chapter 5 focuses on transcription using orthogonal collocation. It expands on the work described in (B*), by comparing the performance and accuracy of different orthogonal collocation methods at varying orders of polynomial approximation for smooth and non-smooth dynamic systems.

Chapter 6 modifies the scheme presented in the preceding chapter to allow discontinuities in the velocity state, thereby producing a truly impulsive collision model that is able to handle a wider range of contact scenarios. This work was published in (H).

Chapters 7 and 8 present preliminary results on the coordinate formulation and initialization of legged locomotion problems, which suggest promising directions for our future work.

Chapter 7 looks at the impact of the selected coordinate system on problem tractability. This consolidates work described in two papers: in (E*), we showed that performance can be improved by referencing joint angles to the world frame, rather than relative to the parent link. In (I), we took this idea to its logical conclusion by considering a maximal coordinate system, where the translational degrees of freedom are also defined absolutely. When (E*) was published, it was the first study examining the effect of coordinate selection on computational efficiency in the context of trajectory optimization.

Chapter 8 addresses problem initialization, an aspect of trajectory optimization that has yet to be widely researched. The motivation for this line of inquiry is established by attempting to generate a skateboarding trick called an *Ollie* using the impulsive contact-implicit formulation described in Chapter 4. We find that the solver cannot spontaneously "discover" the complicated sequence of varied collisions needed to perform the maneuver, unless one key collision is scheduled, or the problem is initialized from a "good" seed i.e. one close to a feasible solution. This leads us to ask, *"how can a "good" seed be obtained for an unknown problem?"*

We then present a novel initialization technique we devised in an attempt to answer this question: the stochastic generation of loosely gait-like motions we termed *silly walks*.

Our original attempt to model the Ollie using a combination of complementarity and hybrid-dynamic techniques was published in (D*), while the updated version was published in (H). The silly walk method was published in (C).

Chapter 2

Studying Legged Locomotion with Trajectory Optimization

Optimality is a unifying principle in the study of locomotion. According to pivotal works by Full and Koditschek [5] and Raibert [28], the fundamental dynamics of running are akin to bouncing on a single leg, so any multi-legged runner is essentially redundant. Consequently, motions recognizable as 'running' are just a small subset of the ways they are capable of traversing distance. If you (a biped) can run, you can theoretically also walk, hop, skip, leap or shuffle from one place to another, so why is running almost certainly the motion that will emerge when you see your bus pulling away from the stop without you? The assumption is that running is selected because it is, in some way, *optimal*.

This was put eloquently by Laumond et al. [29], who described optimization as a *motion selection principle* in the behavior of redundant systems: "An action is viewed as the result of an optimization process whose cost represents the *signature* of the action." Explaining, planning or controlling legged locomotion will therefore always come down to an optimization problem.

The work in this dissertation fits into a broader effort to understand how the maneuverability of legged systems can be optimized. The fundamental mechanics of legged maneuvers are relatively well-understood: Raibert's [28] monopedal theory of legged locomotion control readily explains velocity changes under the assumption that the stance foot remains stationary, while Hubicki et al. [30] and Fisher et al. [31] advocate for the more general template of a sliding mass to model acceleration and deceleration. The devil is in the details of how these generalized dynamics can best be manifested by redundant systems for rapid execution. Identifying motion patterns and physiological features that can improve the time or distance required to perform maneuvers will improve our understanding of agile locomotion in animals, and our ability to design and control agile legged robots.

Explicitly framing the synthesis of physically-feasible motion as an optimization problem defines trajectory optimization. In this chapter, we illustrate the value and versatility of this method by reviewing key examples of its application to legged locomotion problems. The aim is to identify the types of questions it is useful for answering, and how it can be adapted to the generalized motion studies we are interested in conducting. We will then define a general framework for investigating rapid legged maneuvers with trajectory optimization.

2.1 Applications of Trajectory Optimization

2.1.1 Devising Optimal Locomotion Strategies

The most straightforward question that trajectory optimization answers is "what is the optimal way to perform this task?". The solution it gives will not necessarily be the true answer - that is, the globally optimal solution that is executable by the real system. In this context, "optimal" locomotion really means "locomotion produced by the optimization process" [32].

Besides the *reality gap* that is a factor in all simulations, the nonlinear programming problems formulated to optimize legged locomotion are unlikely to fall into a class where the global optimum can be definitively determined, so solutions should only be regarded as locally optimal [32]. The assumption at the heart of trajectory optimization studies is that, despite this caveat, the *optimized* motion will still be a good enough approximation of *optimal* motion to provide a useful answer.

The simplest version of the posed question is asked for a specific, well-defined system and a similarly well-parameterized task. An example is the use of trajectory optimization to produce a jumping trajectory for the MIT Cheetah 3 robot to leap onto a desk of known height [1], as shown in Figure 2.1. The focus was more on the trajectory than the optimization in this case: the cost function primarily served to reduce unnecessary variation in the joint positions or actuation profiles, as the only aim was to synthesize motion that would produce a successful jump when tracked by the robot.

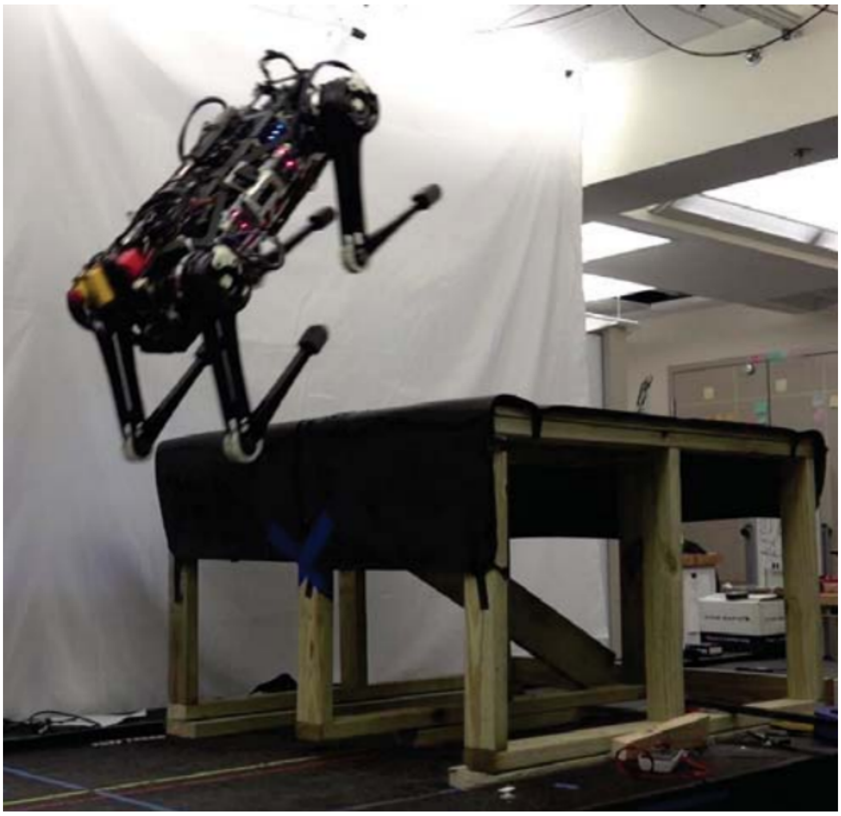

Figure 2.1: The MIT Cheetah 3 robot was able to leap onto a 30-inch desk using a jumping motion designed with trajectory optimization. Image from [1].

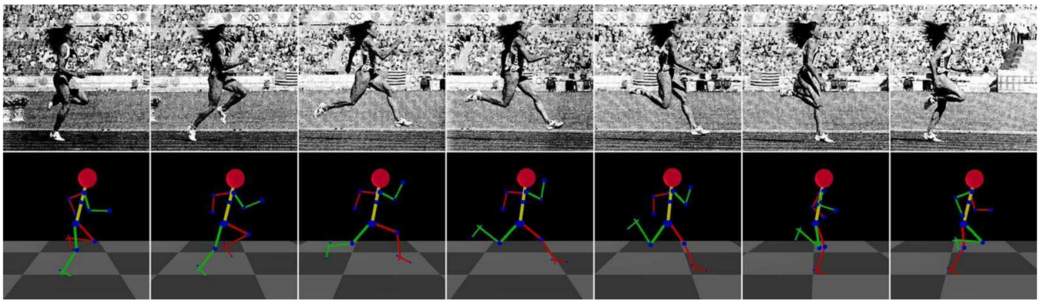

Figure 2.2: A self-stable running gait produced by trajectory optimization with an embedded stability criterion resembles the gait of an athlete. From [2].

If the goal is execution by a robot, the question might be complicated by uncertainty in the task parameters. Outside of impressive technical demonstrations like the aforementioned desk jump, it is seldom useful to work with such a tightly-specified task trajectory. To extend that example to a realistic motion planning application, we would ultimately need to ask how to jump onto *any* desk, or a desk with a height that is not precisely known.

Trajectory optimization does not seem suited to answering broader, vaguer versions of the question, as we need defined parameters to formulate problem. One approach to handling this uncertainty is to incorporate stability into the optimization criteria. This was pioneered by Mombaur, who used trajectory optimization to synthesize *open-loop stable* periodic walking gaits - that is, gait cycles that reject disturbances through passive mechanical properties of the system, rather than active feedback control [33].

Initially, this was accomplished using a *bilevel* method: gradient-based trajectory optimization acted at the lower level to minimize actuator effort, while a direct search algorithm acted above this to improve the stability properties of the resulting limit cycle [33]. Subsequent work defined a smooth stability criterion that could be incorporated into the trajectory optimization problem directly [34]. This was then used to generate self-stable humanlike running gaits [35, 2]. The resemblance between the resulting motion and human running is evident in the comparison with the gait of Olympic sprinter Florence Griffith Joyner shown in Figure 2.2. Dai et al [36] used a similar approach to generate robust trajectories for traversing uneven ground. The MATLAB optimization framework for synthesizing energy-efficient gaits, proposed by Remy et al. [37], also incorporates limit cycle stability analysis to ensure robust locomotion.

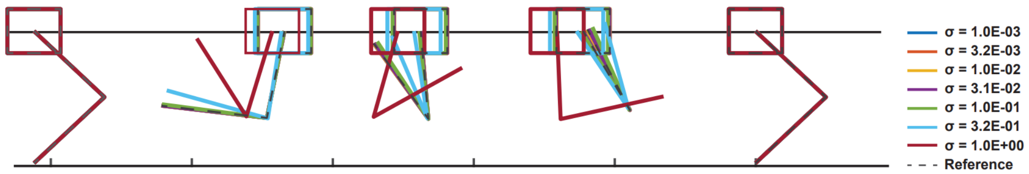

Figure 2.3: Optimal trajectories of a contact-driven cart with terrain uncertainty (σ) incorporated using stochastic complementarity. The ground clearance of the foot tended to increase in response to increasing uncertainty. From [3].

The drawback of this approach is that it assumes that the nominal state of locomotion conforms to a limit cycle, which is only the case for periodic gait. A more versatile approach to uncertainty is to incorporate stochastic elements into the optimization problem. An example of this is the work of Drnach and Zhao [3], who applied stochastic complementarity constraints to encode terrain and friction uncertainty into the optimization problem. Robustness was optimized by incorporating the expected residual into the objective function, resulting in trajectories that exhibited shorter sliding distances in response to greater friction uncertainty, and larger step clearances when terrain uncertainty was increased [3]. The latter result is illustrated in Figure 2.3, which shows the trajectories of a contact-driven cart system produced in response to varying degrees of terrain uncertainty.

Inspired by similar work on the avoidance of randomly-placed obstacles [38] and climbing on surfaces with uncertain friction [39], a later version of this framework incorporated *chance constraints* to limit possible violations of the contact model, making it less likely that infeasible trajectories would be produced in highly uncertain conditions [40].

In addition to broadening the parameters of the task we are inquiring about, we might also want to extend the optimization question to a wider range of models. Rather than investigating optimal motion for some specific system, we could be interested in a larger class of systems (for example, quadrupeds or bipeds) or even results that can be generalized to all legged locomotion. One way to apply trajectory optimization to a wide range of systems is to literally perform it using more models within the set under consideration. The 'Ensemble Contact-Implicit Optimization' method proposed by Mordatch et al. [41] is an example of how this can be implemented. It involves repeated trajectory optimization of models with contact and morphological parameters perturbed around nominal values [41]. The motivation in this case is robust control of a specific system's locomotion rather than general-

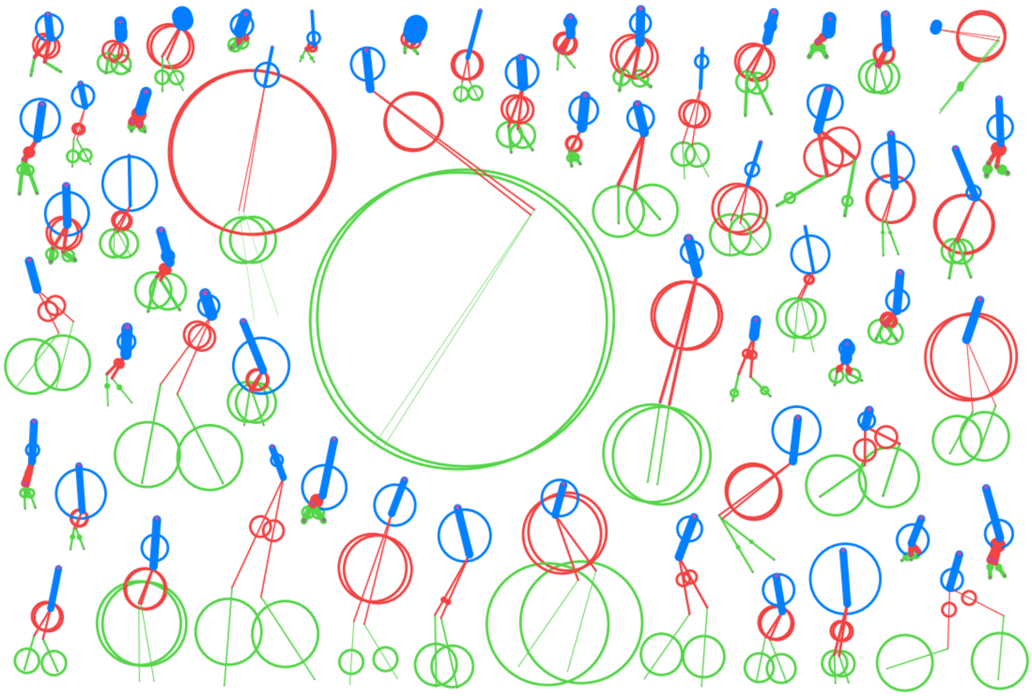

Figure 2.4: Haberland, et al. [4] generated optimal running gaits for these randomized bipedal models to investigate whether forward- or rearward-facing knees tend to confer a lower cost of transport.

izing the results to a range of systems, but the method could theoretically be applied to this purpose. This is demonstrated by Haberland and Kim's Monte Carlo optimization framework [42, 4], which uses repeated trajectory optimization of models with randomly-varying parameters to generalize the findings to all bipedal locomotion. For example, Figure 2.4 shows all the randomized model configurations for which optimal trajectories converged when this framework was used to test the effect of knee direction on the efficiency of bipedal running. This method was proposed with morphological design in mind, but there is no reason it cannot also be used to investigate optimal motion strategies for a broad class of systems.

Rather than optimizing many specific models, the alternative way to generalize trajectory optimization is to generalize the model itself so it can represent all systems of interest. This is based on the concept that highly simplified *template* models can represent the fundamental dynamics of legged locomotion across many possible configurations [5]. Once the optimal motion of the underlying template is discovered, a more detailed model can be used

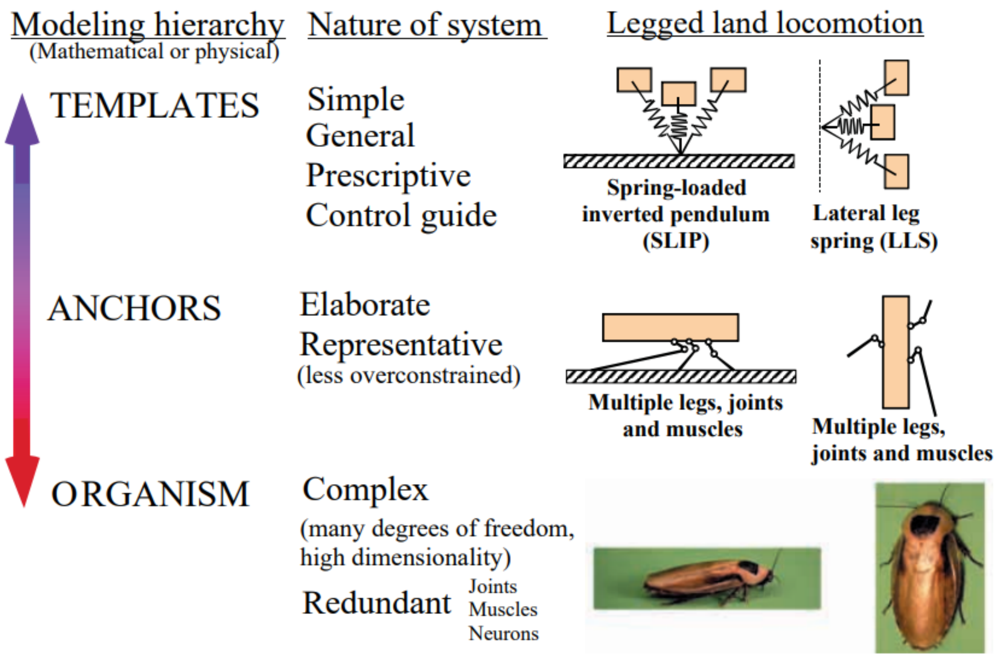

Figure 2.5: Modeling hierarchy describing the relationship between *template* and *anchor* models. Image from [5]

to *anchor* this behavior to a specific system [5]. Figure 2.5 summarizes the relationship between these two categories of model into a hierarchy.

Minimum-order models, such as the Spring-Loaded Inverted Pendulum (SLIP) and Linear Inverted Pendulum Model (LIPM) have become mainstays in the control of legged robots, as epitomized by the "Bible" of monopedal-model-based motion control: Raibert's *Legged Robots that Balance* [28]. Later studies demonstrated the ability of a passive SLIP model to reproduce all common bipedal gaits [43], and quadrupedal bounding [44], further establishing the bouncing monopod as a unifying model for legged locomotion.

Naturally, monopedal template models have also been used to investigate optimal locomotion at the most fundamental dynamic level. An especially broad example is a study by Hubicki et al. [30], which used the SLIP model to assess whether limit cycles actually emerge in energy-optimal locomotion over long-horizon tasks starting or ending from rest. Srinivasan and Ruina [45] optimized the motion of a bipedal adaptation of the SLIP to find energy-efficient gaits at different speeds. A novel intermediate gait that emerged between walking and running in this study (termed *inverted pendular running*) was later observed in birds [46], showing that these reduced-order models can capture real-world phenomena despite their simplicity.

The obvious sacrifice made in pursuit of generality is information about how specific morphological characteristics can contribute to optimal locomotion. Consider the role of non-contacting limbs in stabilizing the body [25]: this could be incorporated into the generalized model by adding an inertial mass as a stand-in for the combined action of the free appendages (see: Lee and Goswami's reaction mass pendulum [6] - Figure 2.6), or by sacrificing some generality to devise broad, but configuration-specific models. The latter approach can be implemented by normalizing the model parameters as far as possible. This is demonstrated in the work of Xi et al. [47, 48], where energy-optimal gaits are generated for generalized bipedal and quadrupedal models with all parameters normalized in terms of the total body mass, uncompressed leg length and gravity.

In the discussion thus far, we have tacitly assumed that the answer to the question of optimal motion would be answered in the form of an optimized trajectory, but the insights provided by trajectory optimization can also be obtained less directly. Even if a given solution is not close to the true optimum, comparison with a *less optimal* trajectory might still give some information about the characteristics of optimal locomotion for the task-of-interest. Studies of sprinting from rest by Celik and Piazza on bipeds [7], and Steenkamp and Patel on quadrupeds [49], compared minimum-time

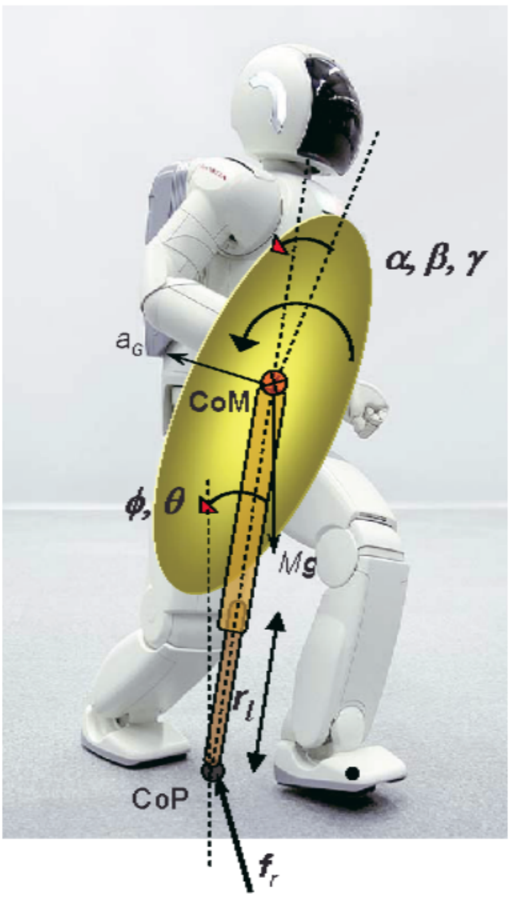

Figure 2.6: The reaction mass pendulum model is a modification of the monopedal format of template that captures the aggregated rotational momentum of the system using an ellipsoid. Figure from [6].

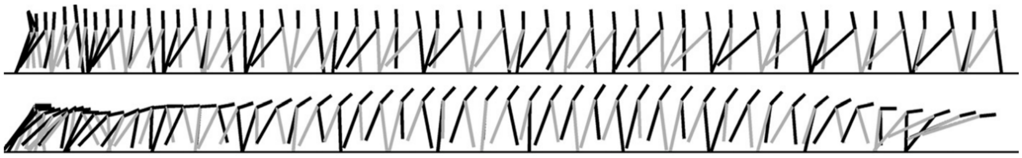

Figure 2.7: Bipedal sprinting motions generated using proportional-derivative control (top) and trajectory optimization maximizing acceleration (bottom). The increased forward pitch of the torso and final diving motion that emerged in the optimized sprint are observed in human runners. Figure from [7].

trajectories to feasible trajectories synthesized without an overarching objective, with the aim of identifying features contributing to greater acceleration. The feasible and optimal sprints produced in [7] are contrasted in Figure 2.7.

In both studies, the characteristics that emerged in the time-optimal solutions matched observations of human sprinters and racing greyhounds, indicating that the conclusions obtained from this method do reflect reality. Although individual instances of trajectory optimization are unable give the information about possible directions of improvement that might be attainable from say, evolutionary computation methods [14], these studies suggest that the aggregated results of many instances could provide similar insights.

2.1.2 Optimization-Assisted Design

One of the reasons that trajectory optimization is such a versatile method is the ability to convert values normally considered to be immutable parameters into variables. In the preceding section, we discussed how this flexibility can be used to incorporate uncertainty in the environment [3, 40, 38, 39] or system model [41] to generate robust locomotion. We also considered the use of generalized models to make results agnostic to specific system parameters [30, 45, 47, 48]. In this section, the system parameters truly become decision variables in the problem: we look at how trajectory optimization can be used to actively select them, or otherwise contribute to the design of the system.

A clear advantage of trajectory optimization in this regard is its simultaneous nature. When a proposed controller is tested through forward simulation, its performance will be limited by the specified system parameters, and likewise, when attempting to design a mechanical system with a particular control

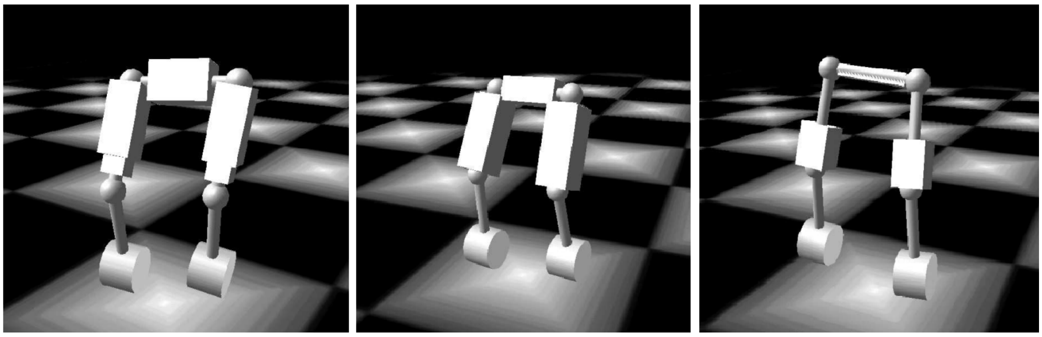

Figure 2.8: Bipeds produced through simultaneous optimization of the controller and mechanical system. The amount of variation allowed in the mass distribution of the limbs increases from left to right. Figure from [8].

policy in mind, the result can only be as good as that control policy allows. Trajectory optimization allows for the *co-design* of mechanical and control elements [50, 51], theoretically leading to an optimal combination of the two.

Although it uses a combination of evolutionary and neural network methods rather than the gradient-based trajectory optimization that is the subject of this dissertation, the work of Paul and Bongard [8] provides an example of this simultaneous design approach that could theoretically be replicated using the methods we consider. In this study, the mass distribution of the limbs of a bipedal robot was co-designed with the control trajectories, resulting in optimal pairings of controller and morphology [8] (see: Figure 2.8). The discrete nature of these pairings was also observed by Yesilevsky et al. [52], who used trajectory optimization to compare the best possible performance of series- and parallel-elastic actuation for energy-efficient hopping on a monopod. They stated, "Throughout our analysis, it became evident that each configuration had a unique optimal motion profile and a set of parameters that differed greatly between the two actuation concepts. This clearly illustrated the necessity of our optimization approach... We strongly believe that such a combined and simultaneous optimization of robot and motion will be at the core of future robotic design. [52]"

Gradient-free methods are often incorporated as an upper level to select the morphology in bilevel schemes, with trajectory optimization applied to discover feasible control profiles at the lower level. Wampler and Popovic [53] used this approach to devise a framework for generating optimal gait for a given morphological configuration, or alternatively, optimizing the morphology to perform a specified motion task. More recently, Fadini et al [54, 55] used it to optimize the structure and actuator properties of a hopping robotic leg.

Trajectory optimization can also assist the mechanical design process by providing a means of testing the maximal capabilities of a proposed design. It plays this role in the first stage of the design process for fast bipedal robots described by Luksch et al. [56], indicating the maximum hopping height that can be achieved with the proposed design.

Besides testing or solving for the parameters of a fixed design, trajectory optimization can also be used to compare discrete mechanical configurations. Haberland and Kim's framework for adapting design principles from biology [42, 4] was devised for this purpose. They propose the use of repeated, randomized trajectory optimization to evaluate whether a given biologically-inspired feature conveys an advantage to a broad class of systems. The case studies they used to demonstrate this concept were the comparison of telescoping vs. rotary knee joints with respect to cost of transport in a monopod [42], and the comparison of forward- vs. rearward-facing knees with respect to energy efficiency in running bipeds [4] (Figure 2.4). The knee direction comparison was subsequently repeated for the robot RAMone [57] with the same basic result, while their framework was used to compare different spine designs [58, 59] and leg morphologies [60] for facilitating maximal acceleration in quadrupedal robots.

2.1.3 Inverse Optimization

At the start of this chapter, optimality was posited as a foundational principle in understanding the locomotion of redundant systems, as it is vital to explain why any one motion trajectory emerges from the vast space of possibilities. The problems discussed so far have used optimization to find unknown motion trajectories or system parameters according to given objectives, but it can also be used to better understand the observed locomotion of a well-defined system, by identifying the objective guiding it. This is referred to as *inverse* optimization. Inverse optimization problems are not covered in this dissertation, so they will not be discussed in detail here, but they use variations of the same techniques applied in forward trajectory optimization, and are therefore worth mentioning to complete the picture of how these methods can be applied to locomotion problems.

Situations where the motion trajectory and system parameters are defined, but the objective behind them is not, are more likely to arise in biomechanical or neuromechanical applications than in robotics. These problems

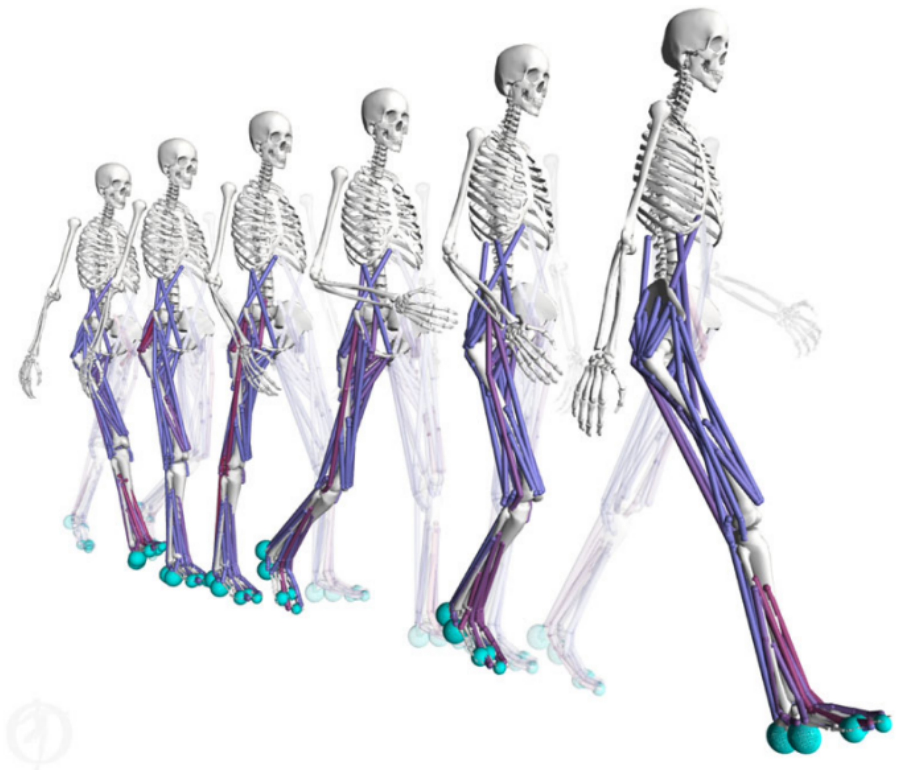

Figure 2.9: Predictive simulation of human walking generated using trajectory optimization [9] of an OpenSim musculoskeletal model. Image from [10].

have motivated developments in *predictive simulation* – a close relative of trajectory optimization that uses highly detailed musculoskeletal models to determine how the motion of humans and animals is controlled. In their review of the field, De Groote et al. [10] refer to predictive simulation as trajectory optimization when it is used to devise open-loop muscle control profiles, distinguishing it from *control policy optimization*, where it is used to identify feedback control relationships linking muscle activity to the state of the musculoskeletal system. Open-source platforms such as OpenSim Moco [9, 61] have contributed to the advancement and widespread accessibility of predictive simulation in recent years. An example of a walking gait generated using trajectory optimization [9] of a muscle-driven OpenSim human model is shown in Figure 2.9.

A study by Nguyen et al. [62] demonstrates how predictive simulation can be applied to the inverse optimization case. They used a bilevel approach to investigate the cost function driving human walking. The lower level used trajectory optimization of a detailed human model to generate walking gaits according to a cost function defined as a weighted sum of different candidate objectives. These weights were controlled on the upper level by a genetic algorithm, which aimed to match the parameters of observed human gait as closely as possible.

Inverse optimization always involves these two steps:

1. the **generation** of trajectories based on the candidate objectives, and
2. the **comparison** of these trajectories with the observed motion to determine the best match.

In the aforementioned study, the comparison step was automated through the genetic algorithm, and the weighted sum format of the cost function allowed combinations of objectives to emerge. The objectives might also be considered individually, which facilitates manual comparison. This is the approach taken in two inverse optimization studies into obstacle navigation in running birds: Blum et al. [63] considered the trajectory of the swing leg in guinea fowl traversing a small terrain drop, aiming to determine whether the birds prioritized rejecting disturbance to the gait cycle or preventing injury. Birn-Jeffrey et al. [64] extended a similar question to birds of various sizes running on uneven terrain, investigating whether they prioritized disturbance rejection, injury prevention or energy economy. In both studies, trajectory optimization of a SLIP model was performed for each of the objectives under consideration, and the results were then compared to the locomotion and force data to evaluate which objective best matched the observed activity.

A study by Koch et al. [65] similarly compared optimized walking under different objectives on a spatial humanoid model, but the model used actually represented the robot HRP-2, as the aim was to identify the optimal control strategy that produced the best approximation of humanlike walking. This followed on from work by Mombaur et al. [66, 67] concerning path generation for the same robot. The inverse optimization approach used there was more akin to the bilevel one illustrated by Nguyen et al. [62], as the comparison was also performed using a derivative-free upper level. As these studies show, the use of inverse optimization in the context of robotics tends to be in pursuit of biomimicry.

2.2 Optimization of Legged Maneuverability

2.2.1 Use Cases

The studies discussed in this chapter demonstrate that trajectory optimization is a powerful and flexible method that can handle the control challenges associated with legged locomotion effectively, and provide useful insights into questions of both strategy and design. Based on the ways it has been applied before, we intend to use it to investigate the following aspects of legged maneuverability:

- **Optimal Strategies for Rapid Execution:** The majority of the studies discussed focused on constant-speed locomotion, and objectives related to energy efficiency or stability. This leaves a clear gap for studies into transient motions, or prioritizing performance objectives such as rapid execution (execution in the shortest possible time or distance) or maximal acceleration.

  The ability to synthesize motion from nothing more than partly-specified boundary constraints is a key advantage of trajectory optimization with respect to these maneuvers, as motion patterns for transient locomotion are not well-established. We are interested in identifying these motion patterns for broad classes of systems such as bipeds and quadrupeds, and discovering the features that contribute to rapid performance.

- **Adaptation of Rapid maneuvers to Varying Conditions:** The limit on achievable acceleration during legged maneuvers is set by the properties of contact between the foot and ground. Consequently, we expect rapid transient motion to be especially sensitive to these conditions. Many maneuvers are executed from constant-speed gait, so variation in the initial state is also important to consider. In many of the studies discussed in this chapter, parameter sensitivity was addressed by incorporating the uncertainty in the problem to identify strategies that are robust to a range of conditions. The downside to this approach is that it does not directly interrogate the effect of a particular change. Trajectory optimization is an ideal method of isolating these effects that has been underused in this capacity so far. We aim to use it to describe specific ways that strategies for rapid maneuverability vary in response to the ground conditions and initial state.

- **Optimal Morphology for maneuverability:** As demonstrated in the discussed examples of optimization-assisted design, trajectory optimization is also a useful method of isolating the effects of varying model parameters. Our specific interest is in the more "macroscopic" version of this application exemplified by Haberland and Kim's work [42, 4], where it allows differences in the configuration of the model to be compared. Like these studies, we intend to use trajectory optimization to evaluate the contribution of specific morphological features to rapid maneuverability.

2.2.2 General Approach

We intend to apply trajectory optimization to the problems described using a randomized repetition approach that is conceptually similar to that of Haberland and Kim [42, 4]. Their primary focus was comparing configurations, so they randomized the physical parameters of the model within the space of each candidate. We are more interested in identifying motion strategies, so we need to introduce randomness to the movement of the model to ensure that the solution space of possible motions is adequately explored.

Because we are dealing with *optimized* rather than *optimal* locomotion – and *locally optimized* locomotion at that – no one solution is especially useful with respect to the questions we are concerned with. Initiating the solver from different seeds might yield solutions of widely-varying quality, so it is impossible to tell how good that one solution is without the context of many others seeded from diverse points. We will therefore repeat the optimization many times for each case under consideration, from random initial seeds. Rather than selecting the best result of the bunch as the *One True Solution* and disregarding the others, we will use these discards as data: by analyzing them in aggregate, we hope to identify the recurring motion features that correlate with better performance and thereby extract lessons about how legged maneuvers should be executed. Our proposed approach is visually summarized in Figure 2.10.

The following chapter will demonstrate this approach through our work concerning rapid gait termination.

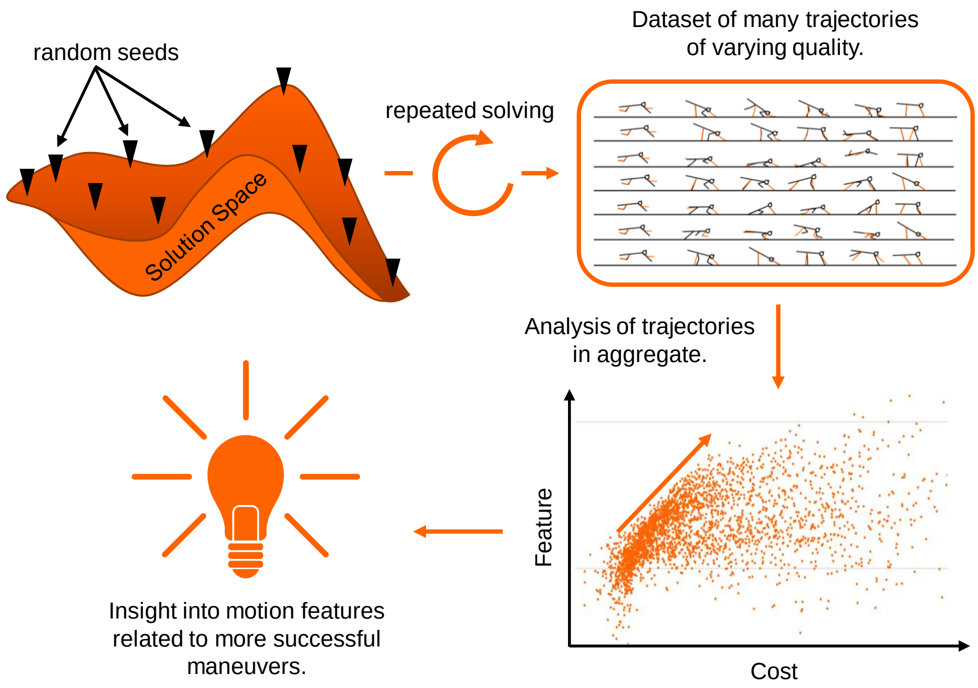

Figure 2.10: Visual summary of our proposed framework for studying legged maneuverability with trajectory optimization. By solving the problem many times from different random seeds, we will generate a large dataset of trajectories with varying costs, which can be analyzed to determine the features of motion that tend to produce more successful maneuvers.

Chapter 3

Case Studies: Rapid High-Speed Gait Termination

In the preceding chapter, we outlined an approach to studying legged maneuvers with trajectory optimization largely inspired by the repeated randomized optimizations of Haberland and Kim [42, 4]. We will now provide a proof-of-concept for this approach using two previously-published case studies where it was applied to the rapid termination of fast gaits. The first, which examines the contribution of arms to the termination of bipedal sprinting, was published in a preliminary form [19] and subsequently repeated with an updated trajectory optimization method as part of a broader review regarding the role of free limbs in maneuverability [25]. The second, which considers gait termination from a rotatory gallop, was published in [24].

The word "rapid" could suggest either a minimum time or minimum distance objective, but we will focus on stopping distance, as outside the context of a barrel race, the stopping time is less likely to be important in practical situations.

3.1 Rapid High-Speed Gait Termination

As discussed in the introduction of this document, the sudden termination of high-speed gait is difficult to study experimentally and largely unresearched. Our overarching research interest is in understanding and imitating the agility of the fastest animals [68, 69, 70, 71, 25], but the few studies that address deceleration at all primarily concern a case as far-removed from the sprinting cheetah as it is possible to be: termination of bipedal walking [72][73][74][75][76][77], often in the elderly or pathological cases [78][79]. Still, we can extract some fundamental principles and extrapolate how they might transfer from grandmothers to greyhounds.

In Raibert's spring-mass conception of legged locomotion [28], deceleration is achieved by placing the center of pressure (COP) further ahead of the center of mass (COM) than the *neutral point* - the position that would result in zero net acceleration over the stride [28][72]. This causes the portion of the stride where the leg functions as a damper (compressing and absorbing energy) to be longer than the portion where it functions as a motor (extending and expelling energy) [28]. The necessary geometry of the COM and COP positions restricts the poses from which it is possible to decelerate. This creates a *critical region* of the gait cycle within which the termination maneuver can be initiated [76]. Once the COM has passed ahead of the COP, the current cycle must be completed before deceleration can occur.

Assuming a favorable initial condition, two factors restrict the velocity that can be reduced to rest in one stride: balance and actuator power [80][81]. If these limits are exceeded, the subject must take another step or they will fall [80]. In this case, "balance" refers to the maintenance of *dynamic stability* – a looser concept of stability than the static ideal of keeping the COM contained within the base of support. The Centroidal Angular Momentum (CAM) - the instantaneous angular momentum of the body about the COM - is a widely-used metric for quantifying dynamic stability [82]. The associated stability criterion states that the system is dynamically stable if it experiences zero rate of change in angular momentum (ZRAM) [83].

Robotic gaits are frequently designed to place the foot at the *zero moment point* - the position that aligns the ground reaction force vector and COM, ensuring this criterion is satisfied throughout the stride [84]. Animal gaits are less stringent: while the angular acceleration is unlikely to be zero at any given instant, we expect that the moments created by ground reaction forces should integrate to zero, assuring the ZRAM criterion is satisfied at the stride level. Deceleration demands unbalanced forces, so it tends to produce unbalanced moments that risk toppling the subject if left unchecked. This is illustrated in Figure 3.1 for the case of a decelerating human: the ground reaction force vector passes behind the center of mass, inducing forward pitch. Research on greyhounds and polo ponies by Williams et al. [81] indicates that pitch avoidance is the primary limit on straight-line acceleration before they reach speeds that challenge their muscle power. Motion of other appendages

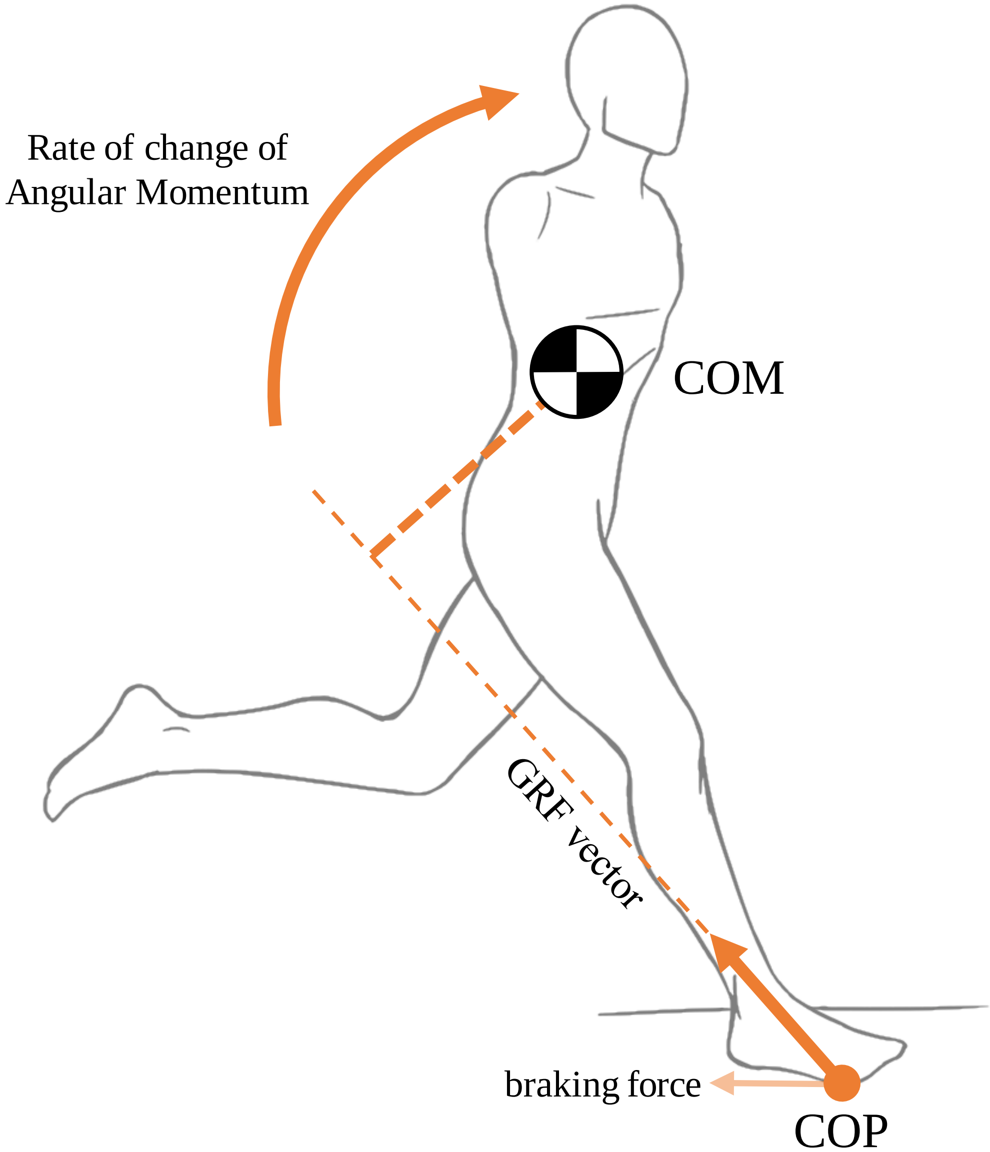

Figure 3.1: Large horizontal braking forces tend to create a forward pitch, as they cause the ground reaction force vector to pass behind the center of mass.

such as the arms [74] or a tail [69, 25] can potentially assist with balance by absorbing undesired angular momentum.

This trade-off between acceleration and stability is heightened when *rapid* stopping is prioritized. Larger braking forces have the potential to create larger destabilizing moments, increasing the chance of a balance failure, but taking additional steps could mean the difference between hitting or avoiding an obstacle. Strategies for decreasing stopping distance might involve generating larger peak braking forces, but they could also maximize braking time within the stride by reducing time in non-contact states, or by extending the safe duration over which these forces can act through careful foot placement and dynamic compensation.

3.1.1 Limitations of Previous Studies

Prior studies on gait termination have made the assumption that the stance foot must remain stationary. When low-friction surfaces have been included [74], avoiding slipping is the primary concern. Failure to consider sliding may be the result of failure to consider velocity: all the animals in the footage sampled for Figure 1.1 skidded to a halt, indicating that slipping might be unavoidable at higher speeds. With balance in mind, slipping might even be desirable earlier in the maneuver, as stopping the feet while the body is still moving fast seems likely to end in the subject tumbling posterior over paws.

Another limitation of past work is the widespread use of spring-mass monopedal templates [5]. Although these models can describe changes in velocity [28], including transient deceleration within gait, or even gradual gait termination, they were devised with periodic strides in mind. It is possible that a sudden stop from high-speed could depart from the basic form of constant-speed gait drastically enough to be described more effectively using a different template. These simple models also miss morphology-specific strategies that could improve performance, such as the use of limbs to assist in balancing, and contact with multiple feet.

3.2 The Contribution of Arms to Human Gait Termination

Balance is a more pressing concern in bipeds over quadrupeds, as their inherent stability is reduced by a smaller base of support, and a longer moment arm between the COP and COM due to a higher COM position. For this reason, it might make more sense to consider bipedal gait termination as being closely-aligned to fall avoidance. If a surface has a high enough coefficient of friction for the foot to sick while the body's center of mass keeps traveling forward, the maneuver strongly resembles tripping, but a sliding foot is also hazardous.

Humans have been observed to employ vigorous arm movement to recover after an unexpected slip or stumble during locomotion [85, 86, 87, 88], so we will now investigate whether they are similarly important to the successful execution of gait termination.

There are two possible ways that the arms could bring about a larger braking impulse:

1. Increasing the **duration** of the braking force by regulating body pitch, so the model does not have to break contact to avoid toppling.
2. Increasing the **magnitude** of the braking force by contributing to the vertical impulse.

The potential of the arms to increase the vertical impulse is supported by research into the effect of arm motion on jumping performance: the vertical ground reaction force has been found to be larger in jumps executed with arm swing, compared to those without [89, 90]. It has been theorized that this could be because the arm motion exerts a downward force on the rest of the body, but the more prevalent theory is that the increased vertical force is primarily a consequence of the stabilized torso position, as this allows the hip joint to remain better positioned for maximal activation [91, 92, 93].

3.2.1 Method

Model

This study uses a simple planar biped model with nine degrees of freedom. It has three actuated joints on each side: shoulder, hip and knee. This model is illustrated in Figure 3.2. The mass (m), length (l), moment of inertia (I) and distance from the preceding joint to the COM (d) of each rigid segment are given in Table 3.1, while the joint ranges of motion (ROM), and torque and power limits, are given in Table 3.2. The model is based on a human, with the segment parameters derived from [94] and joint torque and power limits selected to be within the ranges described in [95], [96] and [97]. These limits are simple upper and lower bounds, unrelated to the state of the joint. Hard stops are enforced at the joint limits as frictionless contacts.

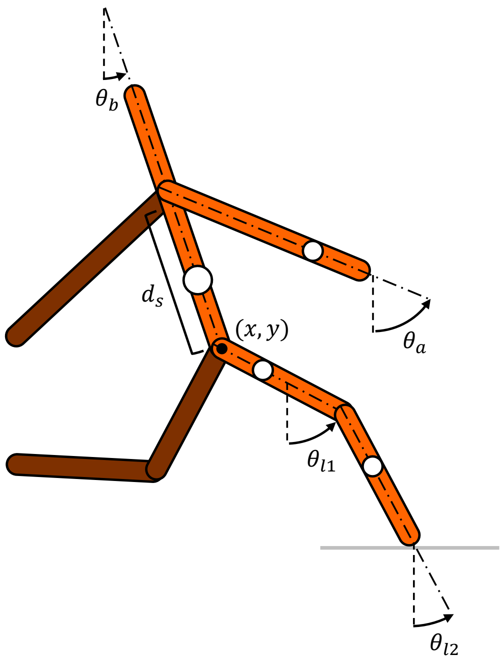

Figure 3.2: 9 degree-of-freedom planar biped model

Trajectory Optimization

The transcribed trajectory optimization problem is formulated using the contact-implicit direct collocation scheme [20] detailed in subsequent chapters of this dissertation. The key parameters of the problem are as follows:

- **Numerical integration:** the trajectory is discretized into 100 timesteps of maximum duration 0.025 seconds using a second-order implicit Runge Kutta method with collocation points placed according to the roots of Radau polynomials [20].
- **Initial condition:** the initial state is sampled from a simulation of steady-state sprinting with an average velocity of 10 m/s.
- **Final condition:** the final state must have no forward velocity or torso pitch, all other velocities less than five percent of their initial values, and both feet grounded. These conditions are imposed over the last five timesteps, to ensure a sustainable final position.
- **Objective:** to minimize the stopping distance, we create a variable x_{max} to serve as the objective value, and constrain the horizontal position at all timesteps to be less than this value.

Table 3.1: Segment parameters of biped model

link	m^1	l [m]	I^1	d^2
torso	0.5040	0.8484	0.0287	0.4183
arm	0.0494	0.7730	0.0020	0.4305
thigh	0.1416	0.4222	0.0027	0.4095
shank	0.0433	0.4340	0.0005	0.4459
$d_s = 0.5319$ of the torso length				

[1] Inertial parameters are scaled such that the model has unit mass in total.
[2] d is stated as a fraction of l.

Table 3.2: Joint limits of biped model

joint	ROM [deg.]	torque [Nm]	power [W]
shoulder	$-\infty$	-0.8	-1.5
	∞	1.1	1.5
hip	-20	-2.5	-41.1
	90	3.7	23.3
knee	-90	-3.7	-8.6
	0	2.1	15.1

- **Initialization:** the optimization was repeated from smooth-random seeds [21] until at least 50 trajectories had been obtained for each combination of configuration and initial condition.
- **Solver:** the CNLP was written using Pyomo [98], an algebraic modeling and optimization library for Python, and solved using the IPOPT algorithm [99] equipped with the Harwell linear solver, ma97 [100].

Tests

The effect of the arms on stopping distance will be evaluated by comparing the performance the model with and without arms over three test conditions:

1. **midstance-initiated, baseline friction:** This is the baseline test. We selected midstance as the point of initiation, as the body leads both feet at this point, meaning it is outside the critical region of the gait cycle where gait termination can be initiated [76] successfully. Both models will be required to take another step, allowing them to select a favourable foot position for braking. A dynamic friction coefficient of $\mu_k = 0.6$ and static friction coefficient $\mu_s = 1.0$ were selected as the baseline friction conditions.

2. **touchdown-initiated, baseline friction:** This time, gait termination is initiated from a point where the foot is ahead of the body (hence, within the critical region) so gait termination is technically possible, but as the foot was positioned for steady-state motion, it might not be placed far enough forward for prolonged braking.

3. **midstance-initiated, high friction:** The dynamic friction coefficient is increased to $\mu_k = 1.2$ and the static friction coefficient to $\mu_s = 1.8$. These high coefficients of friction are still within the range measured for athletic shoes on a variety of common playing surfaces [101]. Higher friction increases the ratio of the horizontal ground reaction component to the vertical one. This will tend to pull the ground reaction force vector further behind he center of mass, increasing the pitching moment created and destabilizing the body more quickly.

We hypothesize that the addition of arms will allow braking to take place over a longer duration. If this is so, we would expect the arms to deliver a greater improvement in cases where the model is less able to regulate its posture through foot placement alone, namely Test 2, where foot repositioning is not required prior to braking, and Test 3, where the extremely high coefficient of friction will tend to induce more forward pitch.

If the arms are able to increase the magnitude of the braking force by exerting a significant downward force on the body, we would expect them to improve deceleration performance across all three tests. Effects on the vertical impulse related to hip posture are excluded from this experiment, however, as they cannot captured by the simple, pose-independent joint power limit applied in the model.

3.2.2 Results and Discussion

The stopping distances for each model and condition are shown in Figure 3.3. To facilitate a clearer comparison across conditions with different initial velocities and friction coefficients, we scale the results according to a metric we call the *box benchmark*. This is the distance that an equivalent rigid mass (the "box") would require to stop from the same velocity, subject to the same coefficient of sliding friction. Because a statically stable model would be able to perform at least as well as the box does by sliding in a fixed posture, stopping in a longer distance than the box benchmark indicates that the model was unable to maintain consistent contact, implying a stability failure. Stopping in a shorter distance indicates that the model was able to increase the magnitude of the normal force beyond its weight (through limb motion,

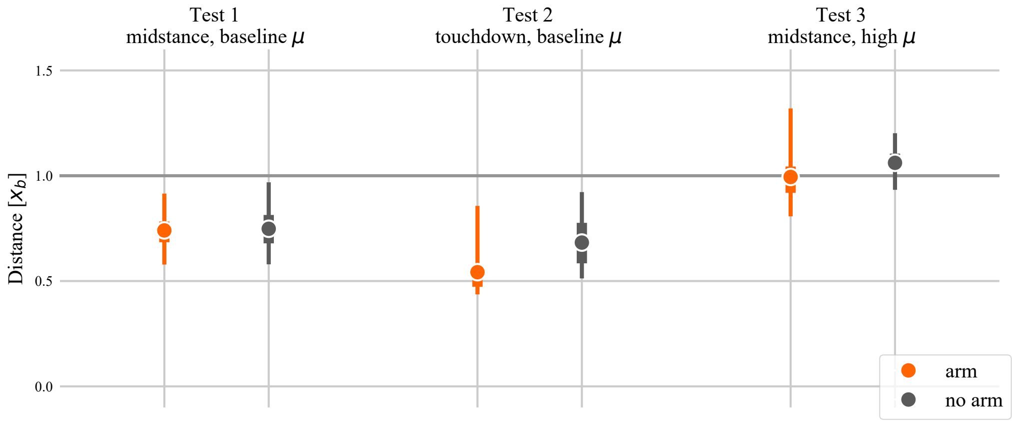

Figure 3.3: Stopping distance in bipedal gait termination trajectories with and without the action of arms. The distance is scaled using the *box benchmark* x_b, the distance a rigid body of equivalent mass would take to slide to a standstill from the same initial velocity on the same surface.

for instance) or take advantage of the larger static coefficient of friction by avoiding slipping.

The performance across the different conditions is consistent with the hypothesis that the arms primarily improve deceleration performance by prolonging duration of stable braking. The arms improved stopping performance in all conditions: the small improvement in Test 1 was not significant ($P < 0.22$), but significant improvements were noted in Test 2 ($P < 3.25\text{e}{-28}$) and Test 3 ($P < 5.08\text{e}{-6}$). In Tests 1 and 3, the model must change its foot position to brake, and therefore, it is able to choose a placement far ahead of the body that minimizes the offset between the ground reaction force vector and center of mass. With moderate friction, foot placement alone is sufficient to avoid toppling, but the failure of most armless trajectories to surpass the box benchmark in the high friction test suggests that it reaches a limit as friction is increased, allowing the stabilizing action of the arms to make a positive difference.

The most interesting case is when the maneuver is initiated with the foot already placed ahead of the body. Figure 3.4 compares the motion in representative trajectories from the touchdown-initiated dataset. The foot is not placed far enough ahead to sustain braking without toppling, so the armless model is eventually forced to take another step, which increases its stopping distance. When the arms are available, it can pinwheel them forwards to exert rearward torque on the body, opposing the forward moment produced

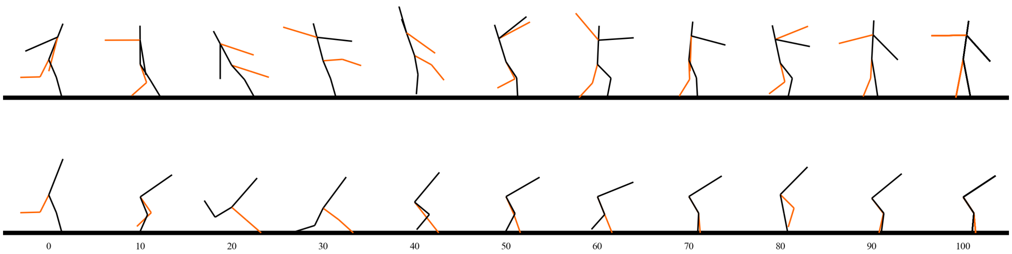

Figure 3.4: Comparison between representative trajectories in the touchdown-initiated test. The model with the arm retains the same foot placement by pinwheeling the arms forward to counteract toppling, while the model without arms must take a second step.

by the braking forces and allowing the foot to remain on the ground.

In almost all trajectories, the arms converged to this pinwheeling motion, spinning forwards 180 degrees out of phase. In this idealized, perfectly symmetrical model, this exactly mimics a reaction wheel, which indicates that they function predominantly by applying torque to the torso, rather than by redirecting the centre of mass rearwards, or creating translational forces. The behaviour of the arms and torso resembles a reaction wheel pendulum [102]: the spinning arms act as a sink for angular momentum, keeping the body from toppling.

Limitations

The feet play a vital role in stabilizing and redirecting the kinetic energy of the body during gait termination [75], so the use of point feet in this study is a notable limitation.

Due to the nature of direct trajectory optimization, the model is able to place its feet through perfect calculation of ground reaction force angle and center of mass position predicted over the full time interval. Foot placement would be far less accurate in a real human, and therefore, this mechanism of pitch control would be less effective. It is possible that the arm model would show a greater improvement in the baseline and high friction cases if some uncertainty (for instance, in the value of the friction coefficient) was incorporated into the test.

Finally, these tests should also be repeated using a spatial model, as the planar case drastically limits the possible ways that the body could be destabilized, and ways that the arms could redirect momentum to prevent falling. Typical arm motions during gait termination have not been described, but a study by [74] on the termination of walking on slippery surfaces indicated that the arms primarily functioned to redirect the motion of the body laterally, preventing it from falling forwards. In trip recovery, the arms were also often moved laterally to increase the moment of inertia in the frontal and transverse planes [87] with the largest effect of arm-swinging occurring in the transverse plane [85]. Based on these studies, we would not expect the forward pinwheeling motion occurring in these tests to be observed in real-life examples of bipedal gait termination.

3.2.3 Conclusion

These results illustrate that pitch stabilization through arm swinging allows the model to maintain braking contact in an otherwise-unsustainable position, thereby improving gait termination performance.

3.3 Iterative Optimization of Quadrupedal Gait Termination

The defining feature of the approach we demonstrate in this chapter is the repetition of each trial from randomized initial seeds. Beyond combating the problem of locally minimal solutions, we believe this could be a valuable method of generating informative data. By contrasting the better solutions against the worse ones, we aim to infer which characteristics of the motion lead to better performance. With enough trajectories available, each one can be reduced to a single datapoint and used to establish a relationship between some feature-of-interest and solution quality. In this study, we attempt to maximize the potential for meaningful comparison of results through an iterative optimization approach that produces families of increasingly effective solutions.

The test case we select is rapid gait termination from high-speed galloping in quadrupeds. Using both a whole-body quadrupedal model and a simplified monoped, we generate a dataset of over 3000 stopping motions and use it to investigate the general features of a successful braking strategy.

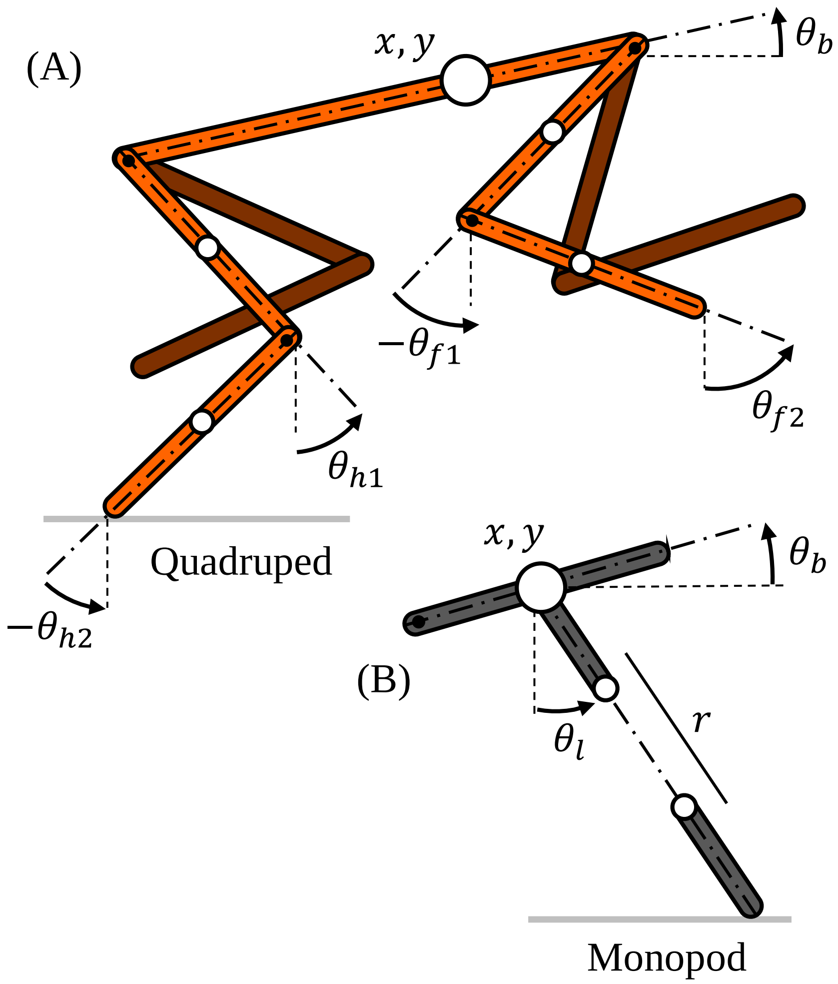

Figure 3.5: Planar quadruped (A) and half-quadruped (B) models.

3.3.1 Method

Models

The models used in this paper are shown in Figure 3.5. The primary model (Figure 3.5A) is an 11-DOF planar quadruped. Each leg is actuated by two revolute joints, with the directions of the second joints corresponding to an 'X' configuration. The angles of all segments are referenced counter-clockwise from the global vertical axis [23]. The leg segments are of equal length, and the fully-extended leg is the same length as the body link, l_b. The mass of the body and leg links are $0.6m$ and $0.05m$, respectively, where m is the total mass of the model. All links have the COM in the middle, except the body link, where it is situated $0.4l_b$ from the shoulder joint.

The force and power limits for the models were selected to be the minimal values necessary for each model to move at the desired average velocity of 30 body lengths per second - equivalent to the speed of a greyhound [103]. These values were identified through the generation of force-optimal and power-optimal trajectories for two tasks: a symmetrical gait cycle at that speed (constrained to match the characteristics of a rotatory gallop for the quad) and acceleration from rest to that speed. The maximum normal force acting on the feet was constrained to three body weights, the peak value observed in galloping animals [103].

Trajectory Optimization

The trajectory optimization problem was transcribed using the same scheme as in the previous study, with the following parameters:

- **Numerical integration:** each trajectory was discretized into $N = 100$ finite elements, consisting of three collocation points placed according to the roots of a Radau polynomial. The duration of the timestep was allowed to vary within 20% of a master timestep, h_m.
- **Initial condition:** the initial condition was sampled from a galloping gait cycle generated using the same quadrupedal model. In a previous study conducted using a simplified quadrupedal model [11], we determined that the critical region for initiating gait termination from a rotatory gallop falls between hind stance and foreleg touchdown, whereas initiating the maneuver from contracted positions requires more corrective motion. The effects of initiation point on performance are illustrated in Figure 3.6. Based on this result, we selected the apex of the extended flight phase as the initial state. Likewise, the apex of

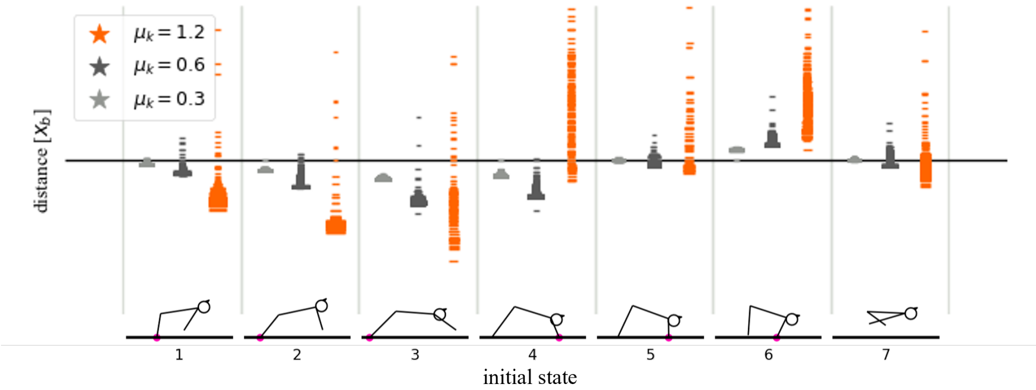

Figure 3.6: Stopping distances on surfaces with different coefficients of friction (μ_k) for termination motions initiated from various points in the galloping gait cycle. The results are normalized using the box benchmark x_b. A wider bar indicates that distances near that value occurred more frequently in the dataset. Adapted from [11]

.

the flight phase was sampled from a hopping trajectory as the starting point for the monopod.

- **Final condition:** For the final condition, we considered the gait to be terminated when the body was no longer moving forward, i.e. when the velocity of the body link, $\dot{x}_b$, had been reduced to zero (or less) and all feet were grounded. We did not want the final stabilizing motions to affect the results, so in our analysis, we cropped the trajectories to the point that $\dot{x}_b$ first reaches zero. The stopping time and distance metrics used throughout the paper therefore refer to the change in time and COM x position from the initial state until this moment.

- **Contact model:** a complementarity-based implicit contact scheme [20, 12] was used to model ground interactions and hard stops at the ROM limits of the joints. A velocity-dependent friction model was used to incorporate a larger static coefficient of friction [24]. Both coefficients were very high: $\mu_k = 1.2$ for kinetic friction and $\mu_s = 2.4$ for static friction.

- **Objective:** rather than minimizing the stopping distance by setting the upper bound on x as the objective, as in the previous case study, we decreased it through the iterative process described subsequently. The only objective in each individual run of the optimization problem was minimizing penalties associated with the complementarity constraints in the contact model.

- **Initialization:** each iterative optimization was initialized from a smooth-random initial trajectory [21].
- **Solver:** the same software and algorithm were used to formulate and solve the problem as in the previous example (3.2.1).

Iterative Minimization of Stopping Distance

We decreased the stopping time and distance iteratively with the goal of producing gradually evolving, incrementally improving motions supporting comparative analysis. There are also technical reasons why this technique could possibly lead to superior solutions, compared to the previous option of minimizing the distance directly as the problem's objective.

Stopping distance is a challenging objective to minimize as it is closely linked to time. Although we do use a variable timestep, the aim is allowing the contact state to change with greater flexibility, not facilitating wide variation in the total simulation time. Letting the timestep vary by orders of magnitude would lead to a poorly-conditioned problem. There is nothing to prevent the model from stopping in much less than the available time, but setting an intentionally over-generous value is not a desirable option, as it effectively reduces the resolution of the solution by decreasing the number of nodes used for the maneuver. The problem gets more difficult to solve as the time is decreased, however, as this restricts the solution space to trajectories close to the theoretical global time-optimum. These solutions may be difficult to find from a random initial seed that is far from feasible. "Warm starting" the process from a feasible solution is likely to improve solver performance, even if that feasible solution is not an especially good one.

We *squeeze* time and distance in an outer loop until a feasible solution can no longer be found:

1. The solver is initialized with a two-step process: first, a procedurally-generated smooth-random *silly walk*[21] is given as a guess to solve a simplified version of the problem, where two of the collocation points are deactivated and first-order integration is used. This solution then initializes the first attempt to solve the full-scale problem.
2. For the first iteration, the simulation time is assigned a random value. If it converges, the master timestep h_m for the next attempt is decreased

by 10 percent, and an upper bound is placed on x, restricting it to 0.98 of the previous stopping distance.

3. The previous solution becomes the guess for the next iteration, and the process is repeated until the problem fails to converge, or the complementarity penalties can no longer be minimized to acceptable values.

The flow diagram in Figure 3.7 illustrates this procedure. The only objective applied in each solving iteration is minimizing the complementarity penalties, so in terms of the stopping distance problem, each solution should be regarded as feasible result, rather than even a local minumum, but the overall effect of the itertively-decreasing upper bound on x is to minimize distance. Anecdotally, we did find that the final distances were comparable to those achieved when a distance-minimizing cost function was used in conjunction with total simulation times similar to those of the ultimate solutions, but the iterative process was much less failure-prone.

3.3.2 Results and Discussion

As an overview of the solutions we obtained, the stopping times and distances for both models are plotted in Figure 3.8. To facilitate a clearer comparison between the two models, we again normalized their performance by comparing it to the box benchmarks: the time T_b and distance x_b.

Static stability is one aspect of the quadrupedal configuration that monopedal templates cannot capture. Due to this limitation, the monopod was unable to beat the box's distance, while the worst quadruped solutions at least matched it. The limitations of the template are also clear in the relatively small improvement from its worst results to its best, compared to the much wider range of performance for the model with more degrees of freedom.

Because the monopod must lift and re-position its leg to maintain balance, it cannot apply a consistent braking force throughout the maneuver. This is shown in the upper half of Figure 3.9, which plots the portion of the total time that was spent actively braking. While the monopod was forced to spend, at minimum, around 10 percent of its time in the air, it was possible for the quadruped to maintain contact for the full duration. This does not, however, mean that doing so is necessarily favorable, as prolonged braking does not appear to lead better results for either model. As might be expected, the solutions that do maintain complete contact tend to fall close to the box benchmark. Of course, it is not desirable to spend the majority of the time in flight, either, so the solutions at the extremes - spending either the

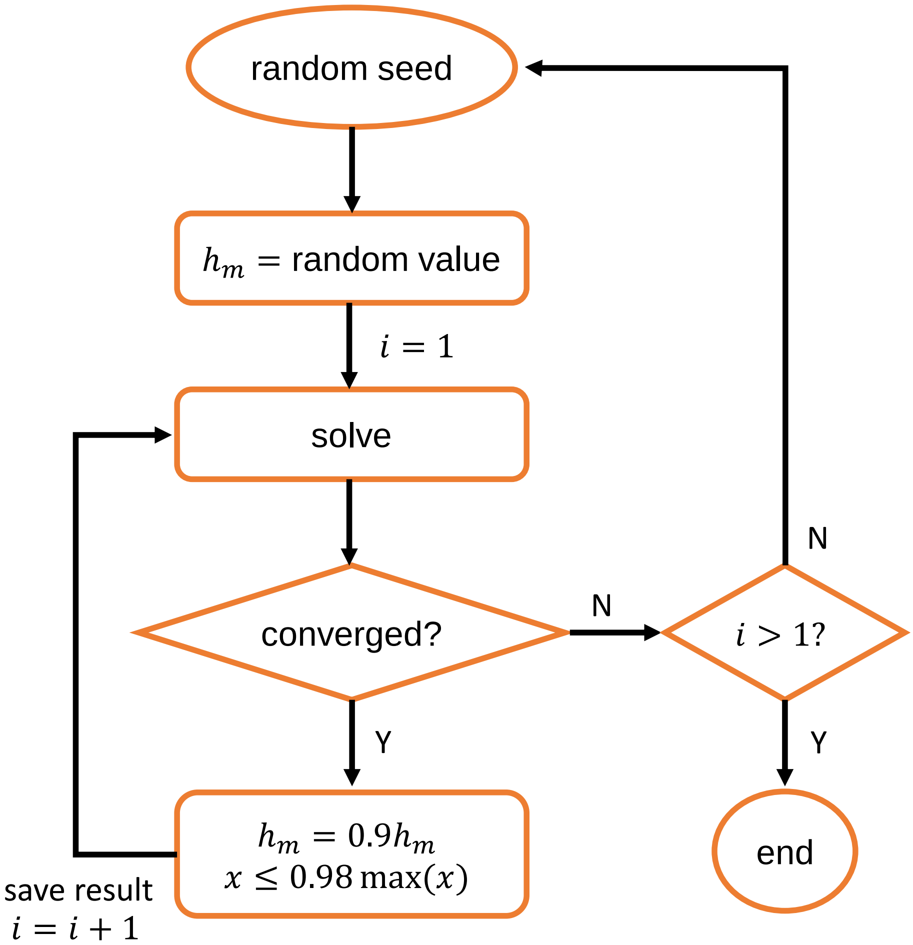

Figure 3.7: Flow diagram illustrating the iterative trajectory optimization procedure used to generate a family of incrementally-improving solutions from a random seed.

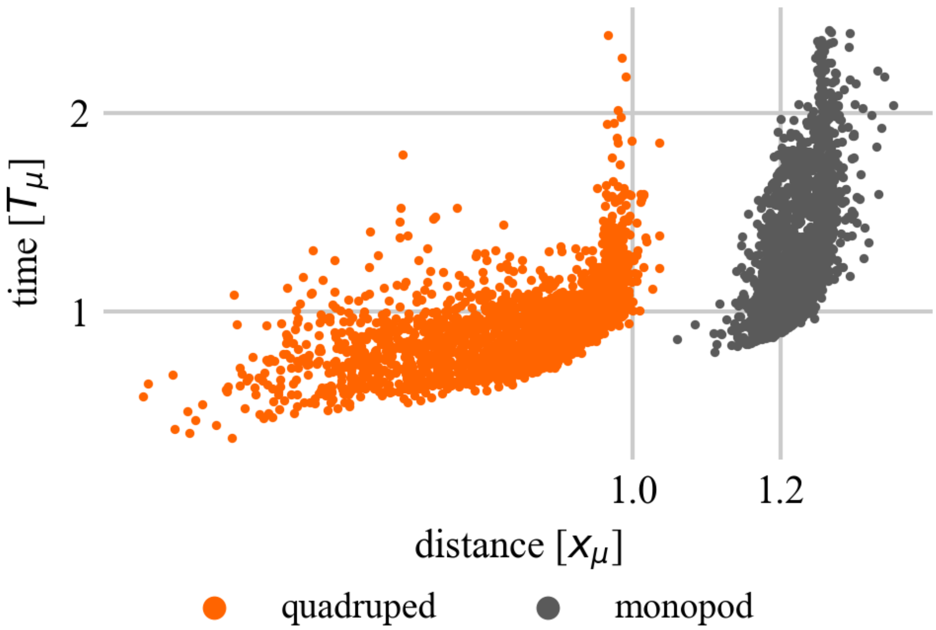

Figure 3.8: Stopping distance and time for quadrupedal and monopedal models vs. an equivalent mass sliding on a surface with the same friction coefficient.

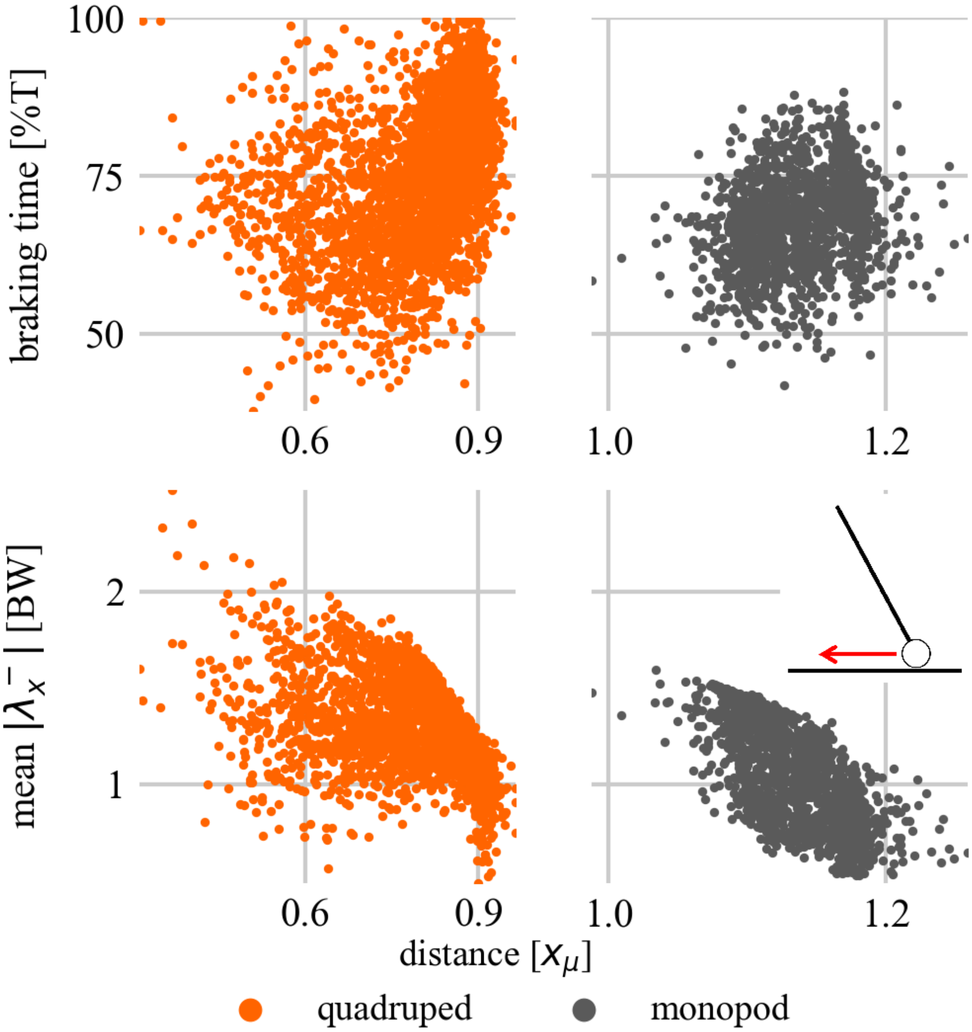

Figure 3.9: Percentage of the stopping time spent braking (top) and time-averaged magnitude of the total braking force applied to each model (bottom). Maintaining contact throughout the maneuver does not necessarily decrease the stopping distance, but increasing the braking force does.

most or the least time in the air - tend to fall on the less-successful half of the stopping distance spread.In combination with the results shown in the lower part of Figure 3.9, which plots the time-averaged magnitude of the decelerating forces, it is clear that braking harder is a more effective strategy than braking longer.

The concern with this strategy is that the application of larger braking forces could come with the drawback of decreased stability. We used the centroidal angular momentum (CAM)[82], plotted in Figure 3.10[82, 104], as a metric for the dynamic stability of the model. A large value in either direction is undesirable, with forward rotation being the most critical, as this would indicate the forward toppling that these rearward-directed forces tend to induce. We see that the model was able to stop in a shorter distance without increasing the peak CAM beyond that experienced during less-effective motions, and even the most rapid trajectories still maintained a mean CAM around zero. (They did, however, tend to lead to larger peak values for rearward rotation, for a reason that will be discussed later in this section.)

These results lead to two follow-up questions:

1. How are larger braking forces generated in the superior solutions?
2. How is the model able to maintain stability under the effect of those forces?

There was no single variable that correlated directly with the average braking force or with the forward CAM, but we can identify some contributing factors:

Actuator force

Figure 3.12 shows the mean actuator force and power exerted by the monopod's prismatic leg for the time window in which the largest deceleration occurred. As would be expected, pushing harder into the ground is an effective way to generate larger braking forces. The results for the quadruped support this, but they do not make for a compelling plot, as both joints hit their torque limits even for trajectories showing only modest improvements over the box distance. So how are the solutions that do significantly better able to exert more force once their actuators have saturated?

Hind leg swing

An advantage of the iterative way we generated the trajectories is that it allows for the identification of specific features that emerge incrementally in

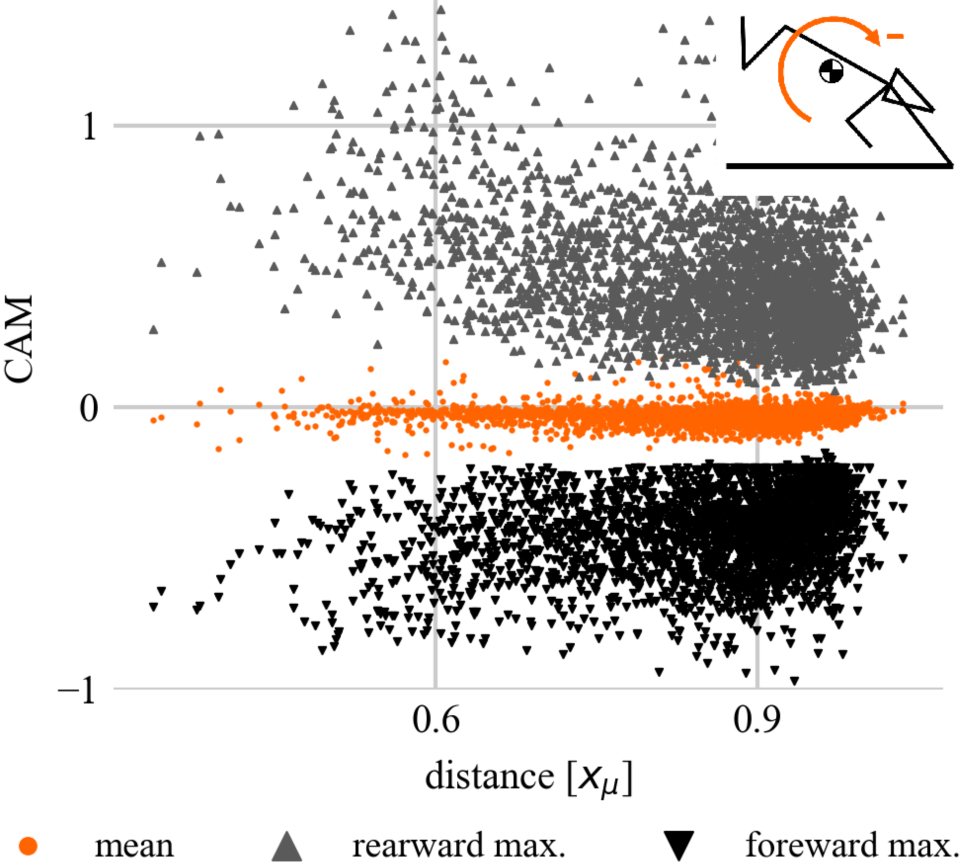

Figure 3.10: Centroidal angular momentum of the quadruped. The forward maximum is of particular interest as this represents the dangerous toppling that large, rearward ground reaction forces could cause.

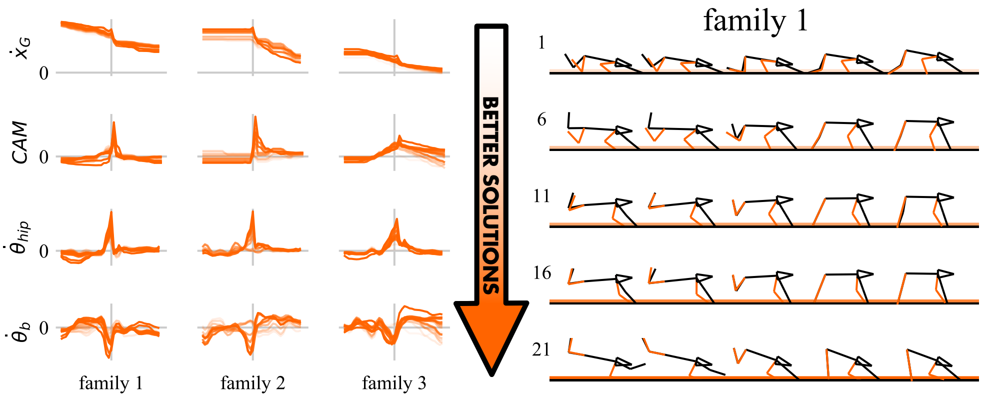

Figure 3.11: Examples of the hind leg swing motion that emerged in many of the solution families during maximal acceleration, and its effects on centroidal angular momentum (CAM) and the angular velocity of the body.

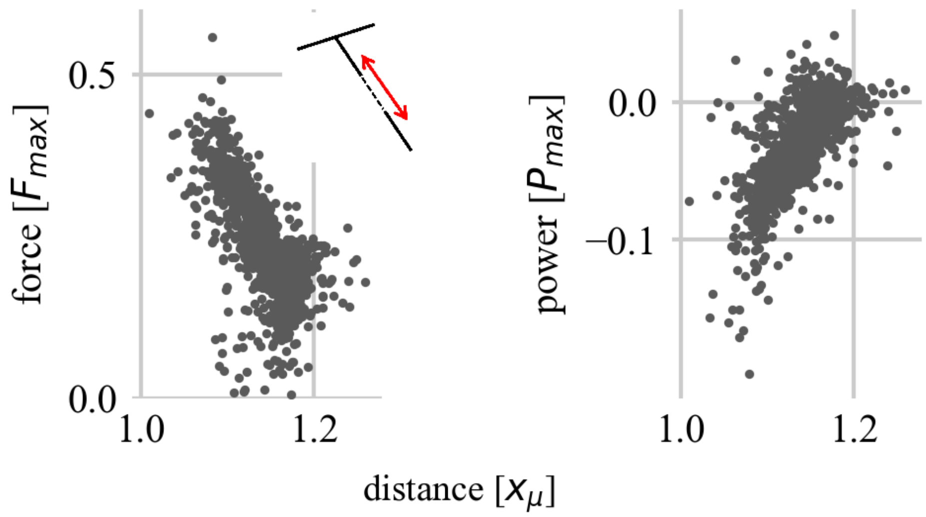

Figure 3.12: Mean force and power in the monopod's prismatic joint during maximal acceleration. (The quadruped is excluded as both its joints reached their torque and power limits for nearly all the trajectories.)

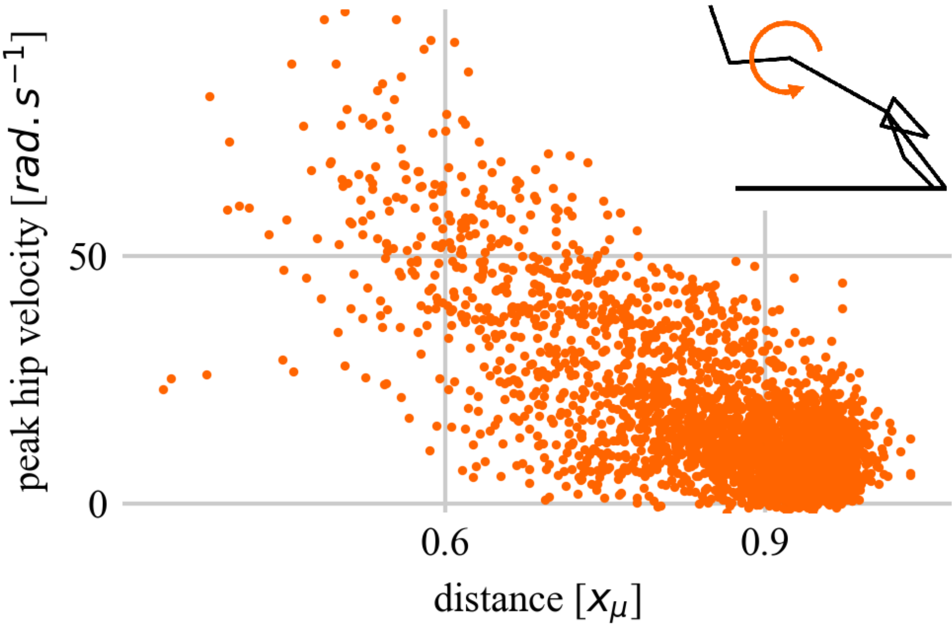

Figure 3.13: Peak side-averaged hip velocity during maximal braking for the quadruped. Better-performing trajectories tended to exhibit higher hip velocities, suggesting the forward-swinging action of the hind legs illustrated in Figure 3.11.

the gait waveforms as the performance improves. A feature that developed in many of the solution families was a rapid forward swing of one or both hind legs occurring at the same time as a sudden, steep deceleration. Often, the instantaneous braking force at this moment was the largest achieved in the trajectory. Representative examples of the leg swing are illustrated in Figure 3.11 for three families.

Plotting the peak hip velocity in a window around the largest instantaneous deceleration value (Figure 3.13) suggests that this feature is widespread in the data, and correlated with improved stopping distance.

As with the role of the arms in bipedal deceleration, the leg swing potentially performs two functions: firstly, it is responsible for the large braking force, as the opposing reaction of the front half of the body acts to push the forelimbs down, increasing the normal force and, consequently, the friction. If the feet slam into the ground at the end of the swing, this further contributes to the

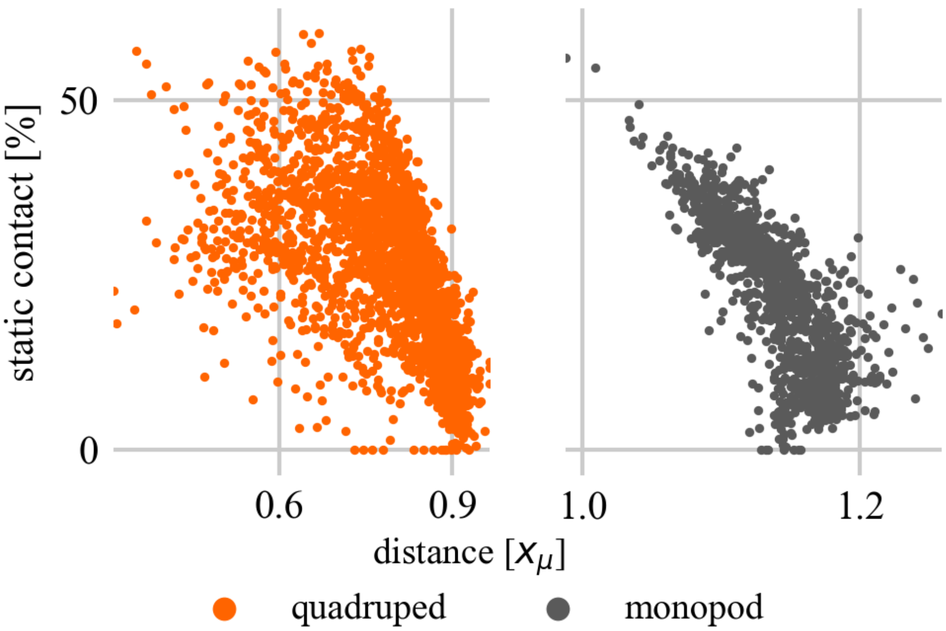

Figure 3.14: Percentage of the total braking force exerted in static contact mode.

decelerating force. Secondly, it counteracts the external pitching moment caused by this force to the extent that the centroidal angular momentum is directed rearward during the swinging motion - hence, the tendency for the peak CAM to be larger for the better-performing solutions. Although the plots of the body's angular velocity show that it does experience some forward pitching due to the opposing torque at the hip, it can immediately be corrected following the swing by the now-grounded hind legs.

When the limb loading is considered (Figure 3.15), it seems that the hind legs are more useful in this ballast role than as brakes: predictably, the majority of the braking force was exerted by the forelegs, though the model did tend to spend similar amounts of time in double stance (with both a hind- and a foreleg on the ground) and front stance. The model is even able to stop using foreleg braking exclusively, but the solutions which did this were not especially successful. This potentially advocates for the addition of a dedicated ballast limb, such as a tail [69].

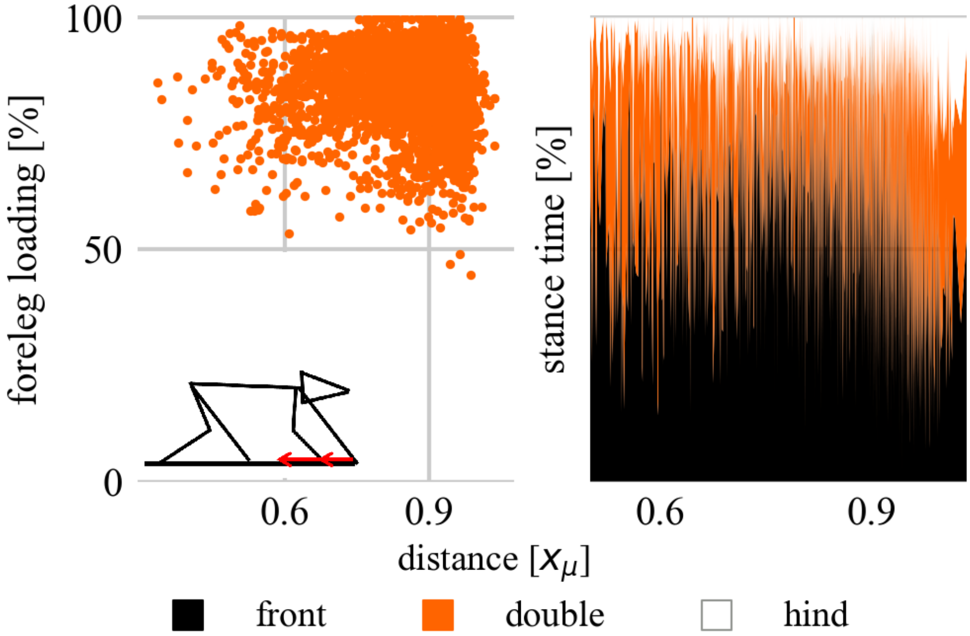

Figure 3.15: Percentage of the total braking force exerted by the forelimbs (left) and time spent in different stance configurations (right) for the quadruped.

3.3.3 Sliding

Another way that the braking force could be increased was through increased use of static braking. This is shown in Figure 3.14, which plots the portion of the total applied braking force that was exerted while the foot was stationary. Despite this trend, and the high coefficients of friction we selected for these experiments, all but a few solutions slid more than they stuck. This indicates that sliding should be incorporated into an effective high-speed stopping strategy, as it is either advantageous to some extent or nearly impossible to avoid.

As in the bipedal case, foot placement appears to be the primary method of preventing pitching during deceleration. With the exception of some of the low-quality quadruped motions, the COM angle converges around the angle of friction, indicating that these trajectories tended to adhere to the ZRAM criterion [83] by keeping the COM in line with the ground reaction force vector. They also avoided large differences between the velocities of the COM and COP, with these discrepancies typically falling within 10 percent of the COM velocity's magnitude. These quantities are plotted in Figure 3.16.

3.3.4 Conclusions

We were able to extract the following lessons in rapid gait termination from a dataset of suboptimal solutions:

- Maximizing the magnitude of braking forces is more effective than maximizing the stride-relative duration of contact.
- Once the forelegs are pushing into the ground at the maximum capacity of their actuators, the normal force can be increased by rapidly swinging the hindlegs forward. The forelegs perform most of the braking function, so a control scheme could conceivably prioritize keeping the hindlegs free to use as a ballast.
- More friction can be generated if the feet stick rather than slide, and though this should be taken advantage of, sliding might be impossible to avoid altogether. To maintain dynamic stability during slipping, the body should be positioned so the COM angle matches the angle of friction, and the relative velocity between the feet and COM should be minimized.

The swinging motion of the hind legs and tendency of the model to adhere to the ZRAM condition while decelerating match observations from the bipedal

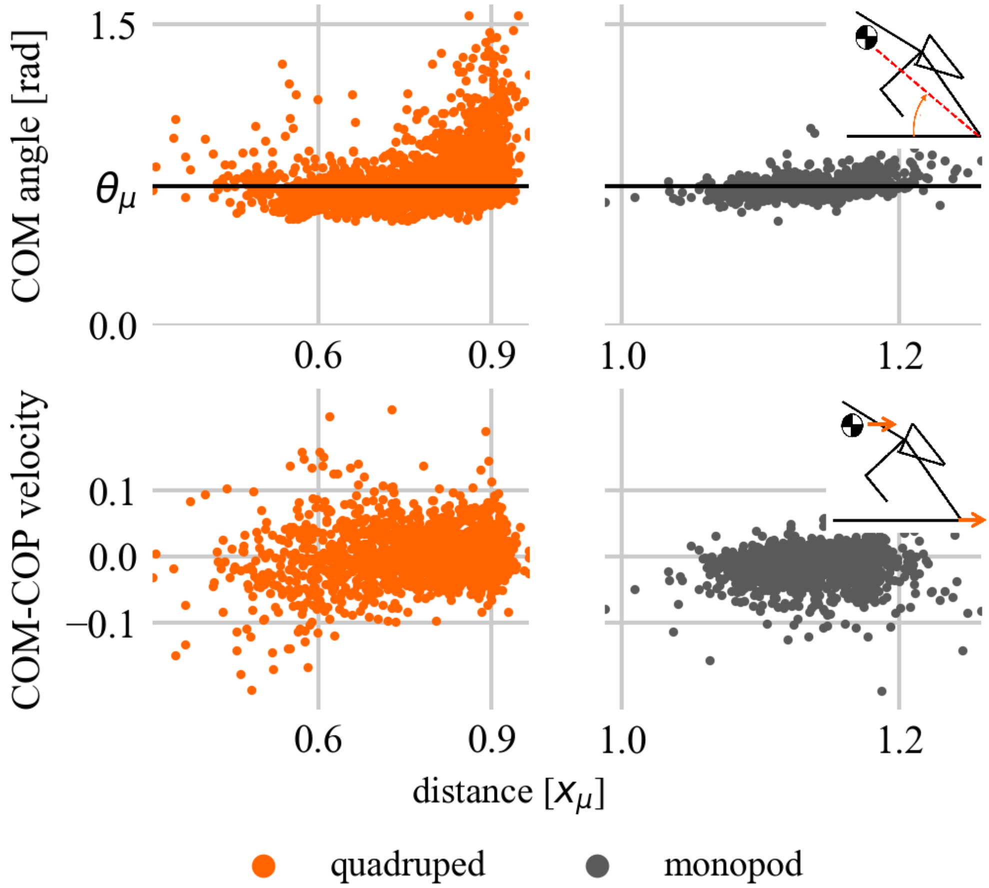

Figure 3.16: Median angle of the COM vector, relative to the ground, compared to the angle of friction (top), and median velocity of the COM relative to the COP (bottom). The relative velocities are scaled to the COM velocity: for each point in the trajectory, the difference between the COM x velocity and COP x velocity was divided by the COM x velocity, and the median of these values was plotted.

case, namely that free limbs make an important contribution to rapid performance, and that dynamic stability should primarily be maintained through foot placement.

A major limitation of this study is that these results have not been transferred to control strategies that can be tested experimentally, so we cannot confirm that their practical usefulness. The primary goal of the study was to determine whether a large set of incrementally-improving solutions produced by trajectory optimization would yield identifiable trends, and we assess it to be a successful proof-of-concept. We expect that this approach could also integrate well with learning-based methods of motion planning, as it offers a way to synthesize large datasets with easily-quantifiable quality variation. The use of trajectory optimization to assist in training complex policies has been demonstrated by Levine et al. [105] for motions including walking, while the work of Hwangbo et al. [15] and Lee et al. [16] on the ANYmal robot shows the potential for challenging locomotion tasks to be learned primarily through simulation.

3.4 Conclusions

This chapter used two case studies to show that repeated, randomized trajectory optimization is an effective process for synthesizing informative data regarding a maneuver that would be difficult and dangerous to study experimentally. While the results in this chapter are promising, they are also preliminary, and there is still much room to refine this approach further in future work. We reflect on these limitations and possibilities in the concluding chapter. The next stage of the project will design a transcribed problem formulation to implement this approach effectively using an established, open-source solver.

Chapter 4

Trajectory Optimization Methods: A Taxonomy

The first stage of this project defined how we intend to apply trajectory optimization to the study of legged maneuverability. The approach we selected hinges on the generation of large datasets of solutions, so it is imperative that the formulation of the trajectory optimization problem facilitates fast, reliable solving. The design of the problem formulation is the topic covered by the remainder of this dissertation.

In this chapter, we lay the groundwork for the technical chapters that follow by giving a brief overview of the trajectory optimization problem, and possible approaches to formulating each of its key components.

A 2018 review of trajectory optimization as applied to the field of spacecraft control by Shirazi et al. [17] proposes decomposing problems into four components to aid in comparison:

1. Model
2. Approach
3. Objective
4. Solution

We will use these categories to structure this overview and highlight the primary areas that our problem design work will focus on.

4.1 Model

The first step in composing a trajectory optimization problem is modelling the essential dynamics of the system under consideration. For a legged locomotion problem, this model will have three main components:

1. the dynamic model
2. the contact model
3. the actuator model

4.1.1 Dynamic Model

Generally, a continuous dynamic system can be modelled using an ordinary differential equation (ODE) of the form

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right) \tag{4.1}$$

where $\mathbf{x}$ is the vector of state variables, $\dot{\mathbf{x}}$ contains their time derivatives, and $\mathbf{u}$ is the input vector. This set of ODEs is referred to as the *equations of motion* (EOM) of the system.

The robots and animals we are concerned with are typically modelled as tree-like systems of interlinked rigid bodies. In legged locomotion problems, this will be a *floating base* model, where the base body is free to translate in the world frame. The position of such a system can be thought of as a combination of the position of the floating base in the world frame, and the *pose* – the position of the other bodies with respect to the floating base. The coordinates describing the position ($\mathbf{q}$) and their velocities ($\dot{\mathbf{q}}$) make up the state vector. The velocities are also included in the derivative vector, together with the accelerations ($\ddot{\mathbf{q}}$).

Many different coordinate systems can be used to describe the same model [106], with the choice affecting the size and sparsity of the problem [23, 107]. Recent work [108] suggests that the structure of the EOM also affects performance: when the equations were defined using the inverse dynamics formulation rather than the forward one (that is, when the ODE were written with the actuator torques as their output rather than the state derivatives) the solver was observed to process the problem more effectively.

The input vector is often referred to as the *control* vector, as it contains the variables that control the dynamics of the system. For a legged locomotion problem, these will usually be forces actuating the joints between bodies.

Assuming no external forces act on the system, the only other forces present are the *constraint forces* (λ). These are necessary to restrict the motion of the system in certain contact states, or when the number of coordinates exceeds the number of independent *degrees of freedom* (DOFs). The use of non-minimal coordinate systems, and *closed* kinematic chains (chains of rigid bodies which intersect at more than one point, forming loops) are common cases where the system would have more coordinates than DOFs. Mathematically, λ can be regarded as a vector of *Lagrange multipliers* related to these motion constraints, rather than explicitly-defined forces, but conceptualizing them as forces better supports the intuitive *manipulator* form of the EOM [106]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{J_a}(\mathbf{q})\mathbf{u} + \mathbf{J_c}(\mathbf{q})\lambda \tag{4.2}$$

Here, $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{G}$ respectively represent the inertial, Coriolis and gravitational matrices, while and $\mathbf{J_a}$ and $\mathbf{J_c}$ are the Jacobians mapping the control and constraint forces to the system coordinates.

As noted in Chapter 2, the complexity of the models used varies widely depending on the application. Minimum-complexity template models [5] represent the fundamental dynamics of the motion using the fewest DOFs possible, providing generalized insights extending across gaits and morphologies. Whole-body models might be based on specific systems, or normalized to represent a broad category of systems with the same basic configuration.

Models may also be simplified to improve solver performance. Approximating the dynamics as *quasistatic* (velocity-independent) removes the coriolis terms, which are computationally cumbersome for articulated systems with many rigid bodies connected in series. C-FROST [109] is an example of a trajectory optimization package designed with legged locomotion in mind that uses quasistatic dynamic models. This is not a feasible option for studying high-speed maneuverability, however, as the velocity-dependent components are significant for fast, dynamic locomotion.

Another option is to consider only the dynamics of the centroid. Some notable examples of this extensively-used approach include the following:

- Dai et al. [104] perform whole-body trajectory optimization using centroidal dynamics combined with a full kinematic model, so concerns such as obstacle avoidance and reachability can be addressed.
- Herzog et al. [110] use centroidal trajectory optimization to determine linear and angular momentum trajectories, which can then be tracked by a robot using a hierarchical whole-body optimal control approach.

- Kudruss et al. [111] used the humanoid robot HRP-2 to demonstrate that an optimized centroidal trajectory could be tracked to successfully climb stairs while grasping a handrail.
- Ponton et al. [112] proposed a convex relaxation of the centroidal dynamics that can be solved fast enough to be applicable in real time.
- Li et al. [113] model the ground reaction force and leg swing trajectories using bezier polynomials, which allows the centroidal dynamics to be calculated analytically rather than through numerical integration constraints.
- Mordatch et al. [114] employ a similar form of dynamic simplification to the centroidal model, where the model is assumed to have all its mass concentrated in the base body.

Centroidal models are not sufficient for all applications, but they can also contribute to the solution of less-tractable whole-body problems. Budhiraja et al. [115] used the Alternating Direction Method of Multipliers (ADMM) to establish consensus between the centroidal and whole-body dynamics, providing a mathematical framework for separating these into a bilevel optimization problem for more effective solving. Their proposed method was used to generate walking trajectories for the humanoid robot HRP-2, and later expanded upon by Zhou and Zhao [116]. Zhou et al. [117] subsequently adapted this approach into an alternating framework aimed specifically at agile legged locomotion, which switches between the centroidal and whole-body problems until dynamic consensus is achieved. This alternating approach can also be observed in earlier work by Herzog et al. [118] and Ponton et al. [119], but these frameworks do not have a true whole-body dynamics stage as they only consider the full kinematic model.

4.1.2 Contact Model

Contact between the feet and ground is an essential feature of legged locomotion that must be incorporated into the trajectory optimization problem. The difficulty is that it introduces discontinuities in the the allowable directions of motion corresponding to unilateral inequality constraints. In forward time-stepping simulation, such constraints could be implemented using conditional statements, but this is not possible when the forward and reverse kinematics must be solved simultaneously.

If it is possible to plan the contact sequence upfront, a viable approach is to treat these constraints as bilateral on the timesteps where they are designated to be active, creating a *hybrid dynamic model.* The hybrid dynamics approach has been shown to be highly effective for generating periodic locomotion: examples of its successful use include the synthesis of humanlike running and walking gaits by Schultz and Mombaur [2] and Felis and Mombaur [120] respectively, the framework for optimizing bipedal robotic gaits developed by Hereid et al. [121, 122, 123] and the efficient MATLAB gait creation framework developed by Remy et al. [37]. Predetermined contact sequences can also be taken advantage of to simplify the dynamics for more efficient computation, as in work by Pardo et al. [124, 125], where more compact EOM are obtained by projecting the rigid body dynamics onto the null space of the contact constraint Jacobian for each state.

Hybrid dynamics is not a practical option for our intended application. Transient maneuvers do not necessarily follow the well-defined footfall sequences of constant-speed gait, so we need a method that can discover an appropriate contact pattern. The phase-based parameterization of contact variables proposed by Winkler et al. [126] modifies the hybrid dynamics approach to make varied contact sequences possible, by letting the solver choose the duration of each foot's stance and flight phases. It still requires the number of steps to be specified upfront, however. Fully unspecified contact methods come in two classes:

1. **Explicit** contact methods define the ground reaction forces as a function of the state variables.
2. **Implicit** contact methods impose constraints preventing interpenetration between bodies and defining a friction cone, and evaluate the ground reaction forces as necessary to uphold them.

Xi et al. [48] made a case for unscheduled contact planning even in studies of constant-speed locomotion. In their study into efficient quadrupedal gait, they found that the results they obtained through an implicit contact method contradicted past results achieved with set footfall sequences, remarking "This inconsistency of gait choice found in the various quadrupedal models emphasizes the importance of using an optimization approach in which the contact sequence is not pre-determined [48]."

Explicit contact methods require two components: an activation function to determine the mode of each contact in a given instant, and a reaction force function to calculate the forces based on the state and mode. An example of this is the optimization of aperiodic sprinting conducted by Celik and Piazza [7], which used a smooth approximation of the Heaviside step based on the hyperbolic tangent function to activate contacts, combined with a compliant ground model proposed by Marhefka and Orin [127] to calculate the reaction forces based on interpenetration between the foot and ground. There are two drawbacks to the smooth approximation, or *soft* contact method:

1. **Intractability:** adding a smooth contact model requires the introduction of many challenging, nonlinear constraints to the problem. The derivatives of the smooth approximation functions become infinitely large as they approaches the true, discontinuous model of rigid-body collisions, so the models become more poorly-conditioned as they become more accurate. Computational intractability is therefore a major concern in sensitive applications.
2. **Sensitivity:** the level of relaxation is user-selected, and may require time-consuming tuning to adjust it to a specific problem.

A possible method of dealing with these issues was proposed by Onol et al. [128] for grasping problems. They use an iterative process to adjust the relaxation of the contacts, and additionally use information gleaned from the relaxation variables to improve the solution.

Contact-implicit optimization (CIO) methods define the reaction forces as decision variables in the problem, and solve for them such that a set of *complementarity* constraints defining a *hard* collision model is satisfied. A complementarity constraint essentially creates an 'exclusive OR' relationship between two positive variables, A and B, by forcing their product to be zero:

$$AB = 0 \quad A \geq 0 \quad B \geq 0 \tag{4.3}$$

To prevent interpenetration, for instance, a complementarity relationship would be defined between the contact distance and normal reaction.

Stewart and Trinkle [129] devised a time-stepping simulation scheme for rigid-body dynamics that implicitly calculated reaction forces to satisfy complementarity constraints describing contact with Coulomb friction. Mordatch et al. [130, 114, 131] subsequently adapted this scheme for CIO of articulated models with simplified dynamics, allowing the discovery of contact patterns for complex, contact-rich behaviours such as legged locomotion, sit-to-stand transitions, object manipulation and obstacle navigation. Posa et al. [12] adapted it specifically for motion planning in legged robots, preserving the full system dynamics.

In theory, complementarity-based models are true to the discontinuous nature of rigid-body collisions, but in practise, they can be challenging to solve without some relaxation. The zero product constraint is especially difficult for interior point algorithms such as IPOPT [99], as it has no interior. An interior can be created by setting the product to be less than some small positive penalty value, ε:

$$AB \leq \varepsilon \tag{4.4}$$

The penalty can be compressed over multiple solve attempts, or included as an additional term in the cost function and minimized [132]. A similar penalty method is demonstrated in a trajectory optimization scheme proposed by Neunert et al. [133], where the relaxed complementarity constraints are used to implement a scheduled contact sequence.

A related problem with CIO is that the solver might not explore the solution space effectively due to the presence of these difficult constraints. This was observed in the aforementioned study by Xi et al. [48]: when seeded with a four-beat gait, the solver was unable to detect two-beat solutions, and likewise, two-beat seeds always produced two-beat gaits. Initializing the problem from diverse points is therefore essential to realizing the apparent motion discovery benefit of these methods.

Computational efficiency is the primary challenge in both explicit and implicit approaches to optimization with unscheduled contacts. To improve the tractability of these methods, there have been several attempts to formulate contact as a convex problem within the broader nonlinear optimization problem. Todorov [134] took the approach of minimizing the magnitude of the contact velocity, subject to smoothed complementarity constraints describing the friction cone and preventing interpenetration. Erez and Todorov subsequently used this method to optimize running in a spatial humanoid model [135]. This technique is conceptually similar to recent work by Chatzinikolaidis et al. [136] where a smoothed nonpenetration constraint is combined with an analytically-solvable friction model devised as smooth approximation of the Maximum Dissipation Principle [137].

In recent years, this 'problem within a problem' approach has been made more explicit in bilevel formulations that separate the contact problem from the optimal control problem entirely. This allows each component to be solved independently using routines specialized for the purpose, improving computation time. Some examples of this include

- Carius et al. [138] use an adaptation of Moreau's time-stepping scheme for nonsmooth dynamic systems [139] on the lower level to calculate the dynamics over each timestep, with the contact dynamics posed as a minimization problem. The gradients of this problem are obtained via backpropagation and passed to the upper level – a single-shooting algorithm that determines the input variables.

- Landry et al. [140] solve for the friction forces using quadratic optimization based on the Maximum Dissipation Principle, and obtain the gradients of this lower level problem analytically. The upper level is a direct collocation problem that calculates the system dynamics, inputs and normal contact forces.
- Zhu et al. [141] also use direct collocation on the upper level, but quadratic optimization problem on the lower level incorporates the full system dynamics, and a learned model of granular surface contact.
- Howell et al. [142] use the same single-shooting method as Carius et al. [138] do on the upper level, and a path-following algorithm on the lower level to efficiently calculate the dynamics.

4.1.3 Actuator Model

Realistic actuators – be they muscles or motors – have peak force and power limits that are dependent on the state of the joint they actuate. The complexity of actuator models in the legged locomotion literature varies widely, as illustrated by two examples simulating human gait: Schultz and Mombaur's work on optimal control of humanlike running [2] simply uses ideal torque actuators with fixed upper and lower bounds, while predictive simulation [10] frameworks, such as OpenSim Moco [61], employ highly detailed musculoskeletal models. Power-limited actuators are also common, with the dimensionless power ratio used by Haberland et al. [42] being an example of how such a constraint can be applied in a scale-independent manner.

Joints are frequently modelled with passive actuation components. This is particularly important in studies concerning energy-efficient locomotion, such as the work of Xi et al. [47, 48]. A collision model can also be incorporated to add hard stops at the range of motion (ROM) limits, so these bounds do not have to be enforced by the actuator forces. This can only be feasibly implemented through an unscheduled contact scheme – the method suggested by Posa et al. [12], for instance – as it would not be viable to plan these collisions preemptively.

4.1.4 Approach

The general trajectory optimization problem is a constrained boundary value problem of the form:

$$
\begin{aligned}
min_{\mathbf{X}} \quad & J\left(\mathbf{X}\right) && (4.5a) \\
\text{s.t.} \quad & \mathbf{x}(0) = \mathbf{x_0} && (4.5b) \\
& \mathbf{x}(T) = \mathbf{x_T} && (4.5c) \\
& \mathbf{g_{eq}}\left(\mathbf{X}\right) = 0 && (4.5d) \\
& \mathbf{g}\left(\mathbf{X}\right) \geq 0 && (4.5e)
\end{aligned}
$$

where $\mathbf{x_0}$ and $\mathbf{x_T}$ are the initial and final conditions, and $\mathbf{g_{eq}}$ and $\mathbf{g}$ are the equality and inequality constraints describing the model, environment, task and variable bounds. The *approach* refers to the way this problem is formulated into a solvable format.

The systems we consider are not simple enough to yield an analytical solution, so numerical simulation is an inextricable component of the approach. Approaches to the discretization of the trajectory optimizaion problem are classified as either *direct* or *indirect*:

- **Direct** methods discretize the state and control trajectories into time-series of variables, and then seek the feasible combination of variables resulting in the lowest value of the cost function.
- **Indirect** methods construct the necessary conditions for a solution to be considered optimal, and then discretize these conditions. This results in a problem involving both variables describing the trajectory, and additional *costate* variables – essentially Lagrange multipliers of the problem's constraints.

In his introductory tutorial on trajectory optimization, Kelly [143] summarizes the difference between direct and indirect methods as follows: "a direct method discretizes and then optimizes, while an indirect method optimizes and then discretizes." A more detailed description of the two categories is given by Betts [144].

Indirect methods can be difficult to apply, as analytical expressions for the optimality conditions are often hard to derive and compute [144]. They are also difficult to initialize and solve, as suitable initial conditions for the costate variables are seldom known [144], so they tend to converge less reliably than direct problems [143]. For this reason, we will focus our attention on direct trajectory optimization.

Direct trajectory optimization methods come in two common varieties:

- **Shooting** methods simulate the problem forwards from the initial conditions for a given input. This is repeated until an input is found that produces a feasible trajectory satisfying some metric for optimality. In *single shooting*, the whole trajectory is simulated over a single time interval. This is a viable approach for simple systems, and it has been applied to the optimization of space flight [17], but the relationship between the input and result quickly becomes intractable as the complexity of the problem increases [143]. A more feasible option for legged locomotion problems is *multiple shooting*, where the trajectory is simulated in shorter time intervals linked by *defect constraints* ensuring continuity of the state variables.

- **Collocation** methods extend the concept of multiple shooting to its logical extreme by shrinking the simulation interval to the timestep of the discretized trajectory. The defect constraints relating the current state to the previous one now also perform the numerical simulation of the trajectory, leading to a single *simultaneous* process rather than a process of alternating simulation and evaluation stages. The complete optimization problem is thus *transcribed* to a constrained nonlinear programming problem (CNLP), with the model, task and numerical integration specified by the constraints.

Both multiple shooting and collocation methods have been extensively applied to legged locomotion problems. A strong argument can be made for either method, but our interest is in trajectory optimization via collocation, as the wide availability of generalized CNLP solvers and introductory tutorials [145, 143] make this likely the most intuitive method for new users.

Collocation methods can be categorized further, depending on the specifics of the numerical integration method selected. The direct collocation tutorial by Kelly [143] provides a helpful introduction to many of the widely-applied methods. Collocation methods will be discussed more extensively in the following chapter, so we will restrict this section to a brief outline of the properties we consider.

Methods used in trajectory optimization are typically *implicit*, meaning that the next state is calculated as a function of both the current and future state [143]. The alternative is *explicit* integration, where the next state is only a function of the current state. Implicit methods have favourable numerical properties that typically support more efficient solving [146]. In most cases, the discretization is applied at the level of the state variables in the form of a *Runge-Kutta* rule, but it is also possible to incorporate the discretization into the formulation of the dynamics, resulting in a model-specific *variational* integrator. A contact-implicit version of this *discrete mechanics* approach has been proposed for robotics applications by Manchester et al. [147].

The *order* of the method refers to the order (or, degree) of the polynomial that is used to approximate the state trajectory over each time interval. The accuracy of these methods can be improved by either increasing the number of intervals used to discretize the trajectory (referred to as an *h-type* approach) or by increasing the order of the polynomials (a *p-type* approach). Low-order, h-type methods are most common in the legged locomotion literature, particularly in CIO: descendants of the Stewart and Trinkle [129] time-stepping scheme, such as the framework by Posa et al. [12], tend to retain the first-order integration method. Although the order of the variational CIO method proposed by Manchester et al. [147] can theoretically be extended arbitrarily, its performance has yet to be demonstrated and quantified for higher-order problems.

4.1.5 Objective

The objective of a trajectory optimization problem functions as a *motion selection principle* [29] that lets a single solution emerge out of the space of feasible possibilities. The value of the objective (or *cost*) might be determined by the state variables at the boundaries (the *endpoint* or *Meyer* cost, J_E) or accumulated over the complete time interval $t \in [0, T]$ (the *running* or *Lagrange* cost, J_R). When combined, these give a general form known as the *Bolza* cost function [143, 17]:

$$J\left(t, \mathbf{x}, \mathbf{u}\right) = J_E(0, T, \mathbf{x}(0), \mathbf{x}(T)) + \int_0^T J_R(t, \mathbf{x}(t), \mathbf{u}(t))dt \tag{4.6}$$

Penalty Terms

It is necessary to differentiate between the *objective* and the *cost function*: the objective sets the high-level priority guiding the choice of motion, but it may only be one component of the cost. Other components might be included to regulate the trajectories in some way, as in the optimization of running by Schultz and Mombaur [2], where torque variation was penalized to produce smoother motions. Kelly [145] claims that the addition of regularization terms can also improve solver performance by separating otherwise-equivalent solutions, even if they are scaled to values many orders of magnitude lower than the objective to avoid conflict. A cost function is therefore still likely to be specified even when the primary goal is synthesizing feasible motion, as when trajectory optimization was used to generate jumping motions for the MIT Cheetah 3 robot [1]. Here, the cost penalized deviation from a reference pose and the squared sum of the joint torques, guiding the solver away from solutions with noncontributing limb motion and noisy actuation [1].

Difficult dynamic constraints might also be relaxed using penalties minimized in the cost function: this is one suggested method for making the complementarity constraints associated with CIO easier to handle [132]. The CIO approach by Mordatch et al. [114] takes this even further by relaxing the model dynamics in this way, too.

Common Objectives

Objectives in legged locomotion problems typically fall under the wide umbrella of *efficiency*. Some variations of this theme include:

- **Mechanical Cost of Transport:** The mechanical cost of transport (COT) measures the average mechanical work required to travel a unit of distance, normalized by the weight of the system. This is a widely-used objective in locomotion optimization studies across both robotics and biomechanics. Consequently, it is often chosen in studies seeking generalized insights into legged locomotion, such as the investigation into the task-optimality of limit cycles by Hubicki et al. [30], or the study by Fisher et al. [31] on optimal gait transitions.
- **Torque Squared:** The sum of squared actuator torques is a similarly widely-used metric of economical locomotion. Although this is often a component of the mechanical COT function – the aforementioned studies [30, 31] use it to represent the mechanical work – it should not necessarily be thought of as a simplification of this objective. Koch et al. [65] compared walking gaits produced by minimizing torque squared and mechanical COT (among other objectives), and found that torque squared tended to produce smoother motion, while mechanical COT lead to higher speeds.
- **Metabolic Cost:** For biological systems, additional factors beyond mechanical work might be taken into account to give a more complete approximation of the total metabolic cost of locomotion. A comprehensive overview of these objectives is provided by Srinivasan [148], who compares several variations of the metabolic cost for gait optimization on a simple bipedal model.

- **Motor Cost:** The analogue to the metabolic cost for legged robots is to explicitly consider the electrical motors driving the motion. This is particularly prevalent in studies where the purpose of the optimization is to guide the design of the robot: actuator selection is one goal of the bilevel optimization scheme used by Fadini et al. [54], and they account for non-ideal behaviour in the modelled motors by including losses due to friction and Joule effects in the cost function. Smit-Anseeuw et al. [57] minimized the *electrical* cost of transport – that is, the electrical work done by the motors per unit distance according to a known torque-speed gradient – in their study comparing energy-efficient gaits with forward- and backward-facing knees on the bipedal robot RAMone.

There have also been varied attempts to optimize the robustness of locomotion. This can be done directly by optimizing the stability of properties of the limit cycle (using the smooth criterion proposed by Diehl et al. [34], for example) or by minimizing the expected error due to an uncertain parameter, as in the work by Drnach et al. [3] on locomotion over uncertain terrain. Alternatively, researchers might attempt to guide the solver towards stable motion through the use of proxy objectives such as minimal angular displacement [120], or minimal displacement of the centre of pressure from some reference [65].

Outside of the case studies from our work discussed in the previous chapter, there are few examples of rapid performance objectives applied in the literature. Celik and Piazza [7] and Felis et al. [120] find time-optimal motions using a multiple-shooting approach, but this objective should be implemented carefully in collocation problems. In contact-implicit cases such as Steenkamp et al. [49], Fisher et al. [59] and Raw et al. [60], where the sum of variable timesteps is minimized, the variability of the timestep, (and therefore, the flexibility with which contact modes can change) effectively decreases with the cost, so the lower bound must be set far enough below the achievable minimum to retain the benefit of the variable step.

While efficiency and stability objectives have a regulating effect, performance objectives require the system to be thoroughly constrained to obtain realistic results. This was observed by Koch et al. [65], who included maximum average velocity among the options tested in their comparative study of optimal walking under different objectives. They found that additional bounds on the foot impact forces would be required to produce implementable results, as the trajectories generated exceeded the safety limits of the associated robot.

4.2 Solution

The advantage of collocation methods is that the CNLP format makes it possible to exploit prior work on nonlinear programming, which has a vast range of applications outside the trajectory optimization niche. Many open-source and commercial algorithms and programming environments have been developed for this purpose [18], allowing users without extensive knowledge of nonlinear optimization to implement these techniques.

Besides the choice of solver, there is also the question of the solution process. The solver must be initialized with a *seed* vector, which ideally should be something close to a favorable solution. Consequently, techniques for improving the tractability of problems frequently involve *warm-starting* the solver from previous attempts, usually increasing the complexity of the model or tightening the relaxation of problematic constraints with each iteration [132]. This is especially important in the context of unscheduled contact, as the additional variables and nonlinear constraints make these problems more susceptible to getting trapped in local minima [48].

There has been little research into effective methods initializing contact-implicit problems, but we can identify some possible avenues for exploration:

- **Warm-starting from a simplified solution:** Marcucci et al. [149] demonstrate that a simplified initial stage is a promising option, finding that problems solved more efficiently when warm-started from a simplified version using a quasistatic model, relaxed contact constraints and a coarser discretization.
- **Integration of a fast, high-level contact planner:** TrajectoTree – a motion-planning framework for grasping proposed by Chen et al. [150] – guides CIO using a tree-search algorithm acting as a high-level contact sequence planner. It is possible that a fast global footstep planner, such as the recent scheme proposed by Norby and Johnson [151], could be adapted to play a similar role in the CIO of legged locomotion.
- **Synthesis of favorable seed trajectories:** Mansard et al. [152] use a neural network trained on previous results of offline trajectory optimization (a 'Memory of Motion') to synthesize initial seeds for online predictive control. While this application is very different from ours, the general idea of initializing the solver using generated motions resembling successful results could potentially translate to trajectory

optimization.

4.3 Forward

The collocation approach to trajectory optimization gives the user complete control over the modeling and simulation of the system, environment and task. This creates an intimidating amount of room for design decisions, even if you never go "under the hood" of the solver. The remainder of this project aims to illuminate some of these decisions by looking into key aspects of the problem formulation with our legged maneuverability application in mind. Based on this overview of trajectory optimization methods, we have identified the following areas to investigate:

- **Collocation:** The numerical integration method is the foundation of the transcribed problem. Selecting one involves two main challenges: adapting the method for nonsmooth dynamics with unscheduled mode sequences, and navigating the trade-off between solver performance and accuracy.
- **Coordinate System:** The choice of coordinate system used to describe the dynamic model has seldom been discussed in the trajectory optimization literature, with minimal, joint-space representations being nearly ubiquitous in robotics applications [106]. The problem is that these produce dense, lengthy equations of motion for serial chains of rigid bodies, as the location of each body must be referenced through all its predecessors. Referencing coordinates absolutely leads to simpler, sparser equations, but increases the number of variables and constraints needed to model the system.
- **Implicit Contacts:** Contact-implicit schemes have seldom been implemented with high-order numerical integration. There is also room to expand them model a wider range of contact behaviors, namely partly-elastic collisions and more sophisticated representations of friction.
- **Seed Generation:** With repetitive randomized solving being the cornerstone of our approach to studying maneuverability, we require a method of generating random seeds that allow a diverse range of behaviors to be explored, but remain tractable to the solver.

Chapter 5

Contact-Implicit Orthogonal Collocation

The selection of the numerical method used to transcribe the trajectory optimization problem is a critical design decision. It forms the foundation around which all variables and constraints are structured, and determines the sparsity of the problem, and the accuracy of the solution with respect to the equations of motion.

Orthogonal collocation is an especially versatile transcription method, as the placement of the collocation points, number of finite elements (timesteps) and order of the approximation can all be varied. In this chapter, we extend the first-order contact-implicit trajectory optimization scheme devised by Posa et al. [12] to work with orthogonal collocation of arbitrary order, creating the novel high-order contact-implicit formulation published in our paper, *Contact-Implicit Trajectory Optimization Using Orthogonal Collocation* [20]. We then expand on our work in this paper by investigating the trade-off between accuracy and performance for three orthogonal collocation methods, and approximating polynomials of increasing order.

5.1 Orthogonal Collocation

Collocation methods approximate the solution of an ordinary differential equation (ODE) as a weighted sum of *trial* or *basis* functions (typically polynomials). Let $x_i(t)$ denote the trajectory of the i^{th} state variable, while $\mathbf{x}(t)$ and $\mathbf{u}(t)$ denote the state and input vectors at time t. For an ODE of the form

$$\dot{x}_i(t) = f_i(t, \mathbf{x}(t), \mathbf{u}(t)) \tag{5.1}$$

the approximate derivative $\dot{x}_i{}^P$ consists of P trial functions $\psi_p(t)$

$$\dot{x}_i(t) \approx \dot{x}_i{}^P(t) = \sum_{p=1}^{P} a_p \psi_p(t) \tag{5.2}$$

The weights a_p assigned to each trial function are adjustable parameters that depend on the specific method being implemented. The approximate solution $x_i^P(t)$ is then obtained by integrating $\dot{x}_i{}^P(t)$. This result solves the ODE exactly ($\dot{x}_i{}^P(t) = f_i(t, \mathbf{x}^P(t), \mathbf{u}(t))$ at a selected set of *collocation points* ($t = t_1, t_2 ... t_P$) in the domain.

The Lagrange interpolating polynomials provide an intuitive set of trial functions. They are generated from the collocation points as follows:

$$l_p(t) = \prod_{\substack{j=1 \\ j \neq p}}^{P} \frac{t - t_j}{t_p - t_j} \tag{5.3}$$

The key advantage of these polynomials is that they are *orthonormal* with respect to multiplication at the collocation points, so the weights are simply the calculated values of the derivative at these points:

$$\dot{x}_i{}^P(t) = \sum_{p=1}^{P} f_i(t, \mathbf{x}^P(t_p), \mathbf{u}(t_p)) l_p(t) = \sum_{p=1}^{P} \dot{x}_i{}^P(t_p) l_p(t) \tag{5.4}$$

The approximate derivative will be a $(P-1)$th-order polynomial, leading to a Pth-order approximation of $x_i(t)$. The approximation of some $\dot{x}(t)$ using fourth-order Lagrange polynomials and Legendre-Gauss collocation points is illustrated in Figure 5.1.

The trajectory of each input variable $u_i(t)$ is also effectively modelled as a $(P-1)^{\text{th}}$-order spline:

$$u_i^P(t) = \sum_{p=1}^{P} u_i(t_p) l_p(t) \tag{5.5}$$

For longer trajectories, greater accuracy can be achieved by linking N polynomial splines into a continuous or piecewise-continuous approximation. These subdivisions of the trajectory are called *finite elements*.

Orthogonal collocation refers to methods that place the collocation points at the roots of orthogonal polynomials, such as the Legendre or Chebychev

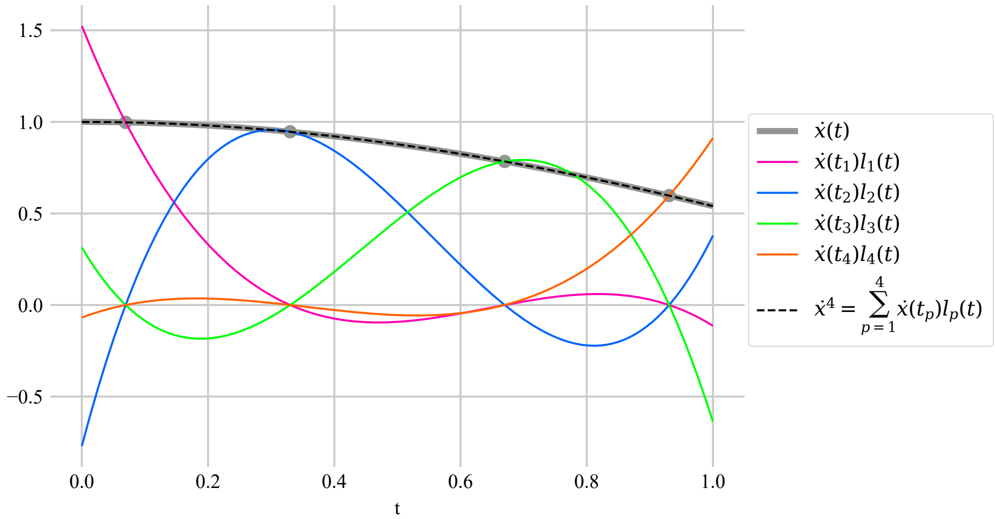

Figure 5.1: Approximation of derivative $\dot{x}(t)$ using fourth-order Lagrange polynomials and Legendre-Gauss collocation points.

polynomials. The benefit of using these points is that they support a highly accurate numerical integration method called Gaussian Quadrature.

Let τ_p represent the p^{th} root of an orthogonal polynomial defined in terms of a time variable $\tau \in [-1, 1]$. (The roots are also called *nodes* or *abscissas* in this context.) For each root, there is a corresponding scalar weight w_p that allows the definite integral of some function $g(t)$ to be approximated as a weighted sum of the function values at the nodes:

$$\int_{-1}^{1} g(\tau)d\tau \approx \sum_{p=1}^{P} w_p g(\tau_p) \tag{5.6}$$

This *quadrature rule* will be exact for polynomial functions up to some order, with the most accurate being Gauss-Legendre quadrature. This gives exact results for polynomials of order $2P-1$ or less. Using the transformation

$$t = \frac{h}{2}\tau + \frac{h}{2} + t_0, \tag{5.7}$$

the rule can be adapted to act over an arbitrary interval $t = [t_0, t_0 + h]$:

$$\int_{t_0}^{t_0+h} g(t)dt \approx \frac{h}{2} \sum_{p=1}^{P} w_p g(t_p) \tag{5.8}$$

where $t_p = \frac{h}{2}\tau_p + \frac{h}{2} + t_0$.

Placing the collocation points at the nodes of a Gaussian quadrature scheme allows the corresponding quadrature rule to be used to integrate $\dot{x}_i{}^P(t)$. Incorporating this rule into the fundamental theorem of calculus gives the following equation for the final value of $x_i^P(t)$ given an initial condition $x_i(t_0)$:

$$x_i^P(t_0 + h) = x_i(t_0) + \frac{h}{2}\sum_{p=1}^{P} w_p \dot{x}_i{}^P(t_p) \tag{5.9}$$

Because $\dot{x}_i{}^P(t)$ is a $(P-1)^{\text{th}}$-order polynomial, the quadrature rule will integrate it exactly, so the accuracy of the method depends only on how similar $\dot{x}_i{}^P(t)$ is to the true derivative. The accuracy can be improved by dividing the trajectory into a larger number of finite elements (*h-method*) or by increasing the order of the approximating polynomials (*p-method*).

Implementing this equation also requires integrating for $x_i^P(t_p)$ at each of the collocation points. Substituting (5.4) into the fundamental theorem of calculus gives

$$x_i^P(t_p) = x_i(t_0) + \int_{t_0}^{t_p} \dot{x}_i{}^P(t)dt \approx x_i(t_0) + \int_{t_0}^{t_p} \sum_{j=1}^{P} \dot{x}_i{}^P(t_j) l_j(t)dt \tag{5.10}$$

which can be rearranged as follows due to the linearity of integration:

$$x_i^P(t_p) \approx x_i(t_0) + \sum_{j=1}^{P} \dot{x}_i{}^P(t_j) \int_{t_0}^{t_p} l_j(t)dt \tag{5.11}$$

Let $L_j(\tau)$ be the j^{th} Lagrange polynomial in a set generated by applying (5.3) to the orthogonal polynomial roots $\tau_1, \tau_2 ... \tau_P$. The trial function $l_j(t)$ is just a mapping of this polynomial to an arbitrary time interval via the transformation 5.7, so the integral of $l_j(t)$ over the interval $t \in [t_0, t_p]$ can be calculated in terms of the integral of $L_j(\tau)$ over the interval $\tau \in [-1, \tau_p]$ as follows:

$$\int_{t_0}^{t_p} l_j(t)dt = \frac{h}{2}\int_{-1}^{\tau_p} L_j(\tau)d\tau \tag{5.12}$$

The integral of $L_j(\tau)$ is determined only by the locations of the collocation points, so the value

$$\Omega_j(\tau_p) = \frac{1}{2}\int_{-1}^{\tau_p} L_j(\tau)d\tau \tag{5.13}$$

can be considered a constant parameter. This gives a set of collocation equations

$$x_i^P(t_p) \approx x_i(t_0) + h \sum_{j=1}^{P} \dot{x_i}^P(t_j)\Omega_j(\tau_p) \tag{5.14}$$

which constitute an *implicit Runge Kutta* scheme when combined with the interpolation equation (5.9).

5.1.1 The Transcribed Problem

When using orthogonal collocation, it is convenient to assign each variable two indices: n for the finite element, and p for the collocation point. The roots of an orthogonal polynomial will not necessarily correspond to the endpoints of the interval ($\tau = -1$ and $\tau = 1$), so a Pth-order transcription will require up to $P + 2$ points to be defined on each finite element: the P collocation points, and two *mesh* points. For generality of notation across different collocation methods, we will index the mesh points separately, using $p = 0$ for the initial point and $p = P+1$ for the final one, even if they happen to coincide with collocation points.

Allowing a variable timestep $h[n]$ and assuming the all state variables are continuous across finite elements, the discrete trajectory is defined by the following constraints:

$$\mathbf{x}[n, p] = \mathbf{x}[n, 0] + h[n] \sum_{j=1}^{P} \Omega_j(\tau_p)\dot{\mathbf{x}}[n, j] \qquad \forall n = 1, 2...N, \forall p = 1, 2, ...P \tag{5.15a}$$

$$\mathbf{x}[n, P + 1] = \mathbf{x}[n, 0] + h[n] \sum_{j=1}^{P} w_p\dot{\mathbf{x}}[n, j] \quad \forall n = 1, 2, ...N \tag{5.15b}$$

$$\mathbf{x}[n, 0] = \mathbf{x}[n - 1, P + 1] \qquad \forall n = 2, 3, ...N \tag{5.15c}$$

Note that the derivative only needs to be calculated at the collocation points, so the equations of motion (EOM) are only evaluated at these points.

5.1.2 Orthogonal Collocation Methods

We will be comparing three orthogonal collocation methods based on the Legendre polynomials, or manipulations thereof:

1. the Legendre-Gauss (LG) method
2. the Legendre-Gauss-Radau (LGR) method
3. the Legendre-Gauss-Lobatto (LGL) method

The Pth-order Legendre polynomial, $\mathcal{L}_P(\tau)$, is defined by the formula

$$\mathcal{L}_P(\tau) = \frac{1}{2^P P!} \frac{d}{d\tau^P} \left[(\tau^2 - 1)^P \right] \tag{5.16}$$

The **Legendre-Gauss** method is based on the Legendre polynomials themselves, with the collocation points being the roots of $\mathcal{L}_P(\tau)$. The quadrature weights are given by the formula:

$$w_p = \frac{2}{(1 - \tau_p^2)[\dot{\mathcal{L}}_P(\tau_p)]^2} \tag{5.17}$$

LG quadrature is exact for polynomials up to $2P - 1$. Neither endpoint coincides with a collocation point, so transcription with this method requires two explicitly-defined mesh points.

The **Legendre-Gauss-Radau** points are an asymmetrical set that includes the initial endpoint, but not the final one. For our purpose, it is more useful to include the final endpoint, so we will consider the flipped set of LGR points. These can be calculated by finding the roots of the polynomial

$$\mathcal{L}_P^R(\tau) = \mathcal{L}_P - \mathcal{L}_{P-1} \tag{5.18}$$

The quadrature weights are not required, because the collocation point at the boundary matches the final mesh point. The integration constraint (5.15b) is therefore not required, while the continuity constraint (5.15c) effectively becomes

$$\mathbf{x}[n, 0] = \mathbf{x}[n - 1, P],\ \forall n = 2, 3..., N \tag{5.19}$$

Subjecting a collocation point to a continuity constraint reduces the degrees of freedom of the approximation, so the LGR quadrature method is only exact for polynomials up to order $2P - 2$.

The **Legendre-Gauss-Lobatto** method places collocation points on both boundaries, allowing it to be implemented with only P defined points per element, but constraining the approximation further. LGL quadrature is therefore accurate for polynomials up to an order of just $2P - 3$. The LGL points are the roots of the polynomial

$$\mathcal{L}_P^L(\tau) = (1 - \tau^2)\dot{\mathcal{L}}_{P-1}(\tau) \tag{5.20}$$

Due to the correspondence of the initial mesh point and collocation point, the continuity constraint is effectively

$$\mathbf{x}[n,1] = \mathbf{x}[n-1,P],\ \forall n = 2,3,...,N \tag{5.21}$$

while the interpolation constraint (5.15a only needs to be implemented for points $p = 2, ..., P$. Because the continuity constraint applies between two collocation points, the evaluation of the EOM at $p = 1$ can also be replaced by a continuity constraint on the derivative:

$$\dot{\mathbf{x}}[n,1] = \dot{\mathbf{x}}[n-1,P],\ \forall n = 2,3,...,N \tag{5.22}$$

This forces the derivative to be continuous between finite elements, unlike the other collocation methods. It also makes LGL the most computationally-efficient method of a given order, as one fewer ODE evaluation is required per element. For this reason, Hermite-Simpson collocation method, which is identical to the third-order LGL scheme, has become a popular option for trajectory optimization [143].

We will subsequently use the abbreviation of the name followed by a number to denote a particular method and order, for example, "LGR2" for the second-order Legendre-Gauss-Radau method. Table 5.1 gives the collocation points for all second- to fifth-order methods, adjusted to the interval $t = [0, 1]$. Figure 5.2 compares the locations of the collocation points for all three fifth-order methods. Note the asymmetry of the LGR points, and lack of points on the boundary in the LG method.

5.2 Contact Model

The primary aim of the work documented in this chapter is extending the contact-implicit scheme proposed by Posa et al. [12] to work with an orthogonal collocation method of any order. In this section, we will explain how contact is modelled for a single point of interaction between two rigid bodies.

We assume that a well-defined tangent plane exits between the contacting bodies, and we will use the coordinate y to represent the normal distance between them, while x and z describe the tangent plane. The reaction force $\mathbf{r}$ can be decomposed into normal ($\mathbf{r_n}$) and tangential ($\mathbf{r_t}$) components. Likewise, $\mathbf{v_n}$ and $\mathbf{v_t}$ denote the normal and tangential components of the relative velocity, $\mathbf{v}$.

As discussed in the preceding chapter, the premise of contact-implicit optimization (CIO) is that the reaction forces are calculated implicitly to enforce

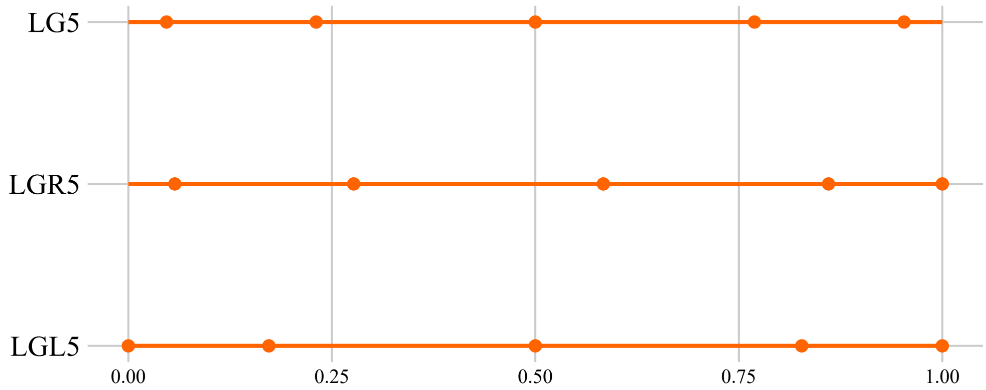

Figure 5.2: Collocation points for fifth-order Legendre-Gauss (LG5), Legendre-Gauss-Radau (LGR5) and Legendre-Gauss-Lobatto (LGL5) collocation methods, adjusted to the time interval $t = [0, 1]$.

Table 5.1: Collocation Points

Method	2	3	4	5
LG	0.2113 0.7887	0.1127 0.5000 0.8873	0.0694 0.3300 0.6700 0.9306	0.0469 0.2308 0.5000 0.7692 0.9531
LGR	0.3333 1.0000	0.1551 0.6449 1.0000	0.0886 0.4095 0.7877 1.0000	0.0571 0.2768 0.5836 0.8602 1.0000
LGL	0.0000 1.0000	0.0000 0.5000 1.0000	0.0000 0.2764 0.7236 1.0000	0.0000 0.1727 0.5000 0.8273 1.0000

mode-dependent path constraints. These can be described in the form of *complementarity* constraints between the contact state and reaction force variables, allowing them to be imposed without a predefined mode sequence. We will use the notation $A \perp B$ to represent a complementarity relationship between variables A and B. The zero product constraint is relaxed using a penalty variable ε, which is minimized in the cost function [132]. We consider a solution to be feasible if $\varepsilon \leq 1e-4$ for all contact interactions.

Anecdotally, we have found that including the complementarity penalties as an additional, scaled term in the cost function to be a less reliable method of solving the problem than solving it in two stages: a *feasibility* stage where minimizing these penalties is the only objective, followed by an *optimizing* stage where they are assigned an upper bound of $1e-5$ and only the true objective is minimized. Unless otherwise stated, this method is used for all non-smooth problems.

5.2.1 Frictionless Impact

It is intuitively obvious that $\mathbf{r_n}$ and y_n cannot both be nonzero at the same time, but it is not sufficient to complement $\mathbf{r_n}[n,p] \perp y[n,p]$, as this would allow the contact mode to change anywhere within the finite element. The polynomial approximation of the system assumes that the dynamics are smooth over each element, so it is advisable to restrict these non-smooth changes to the boundaries – that is, only allow contacts to close or open between elements. We do this by complementing the normal force with the contact distance at all points:

$$\mathbf{r_n}[n,p] \perp \sum_{j=0}^{P+1} y[n,j] \tag{5.23}$$

Figure 5.3 shows the position, velocity, and reaction force trajectories for a one-dimensional, frictionless collision. The details of this falling mass simulation are provided in section 6.2.3 of the next chapter. In this smooth approximation of impact, the reaction force begins acting prior to the bodies coming into contact, and decelerates the mass to rest over the course of the touchdown finite element ($n = 9$ here).

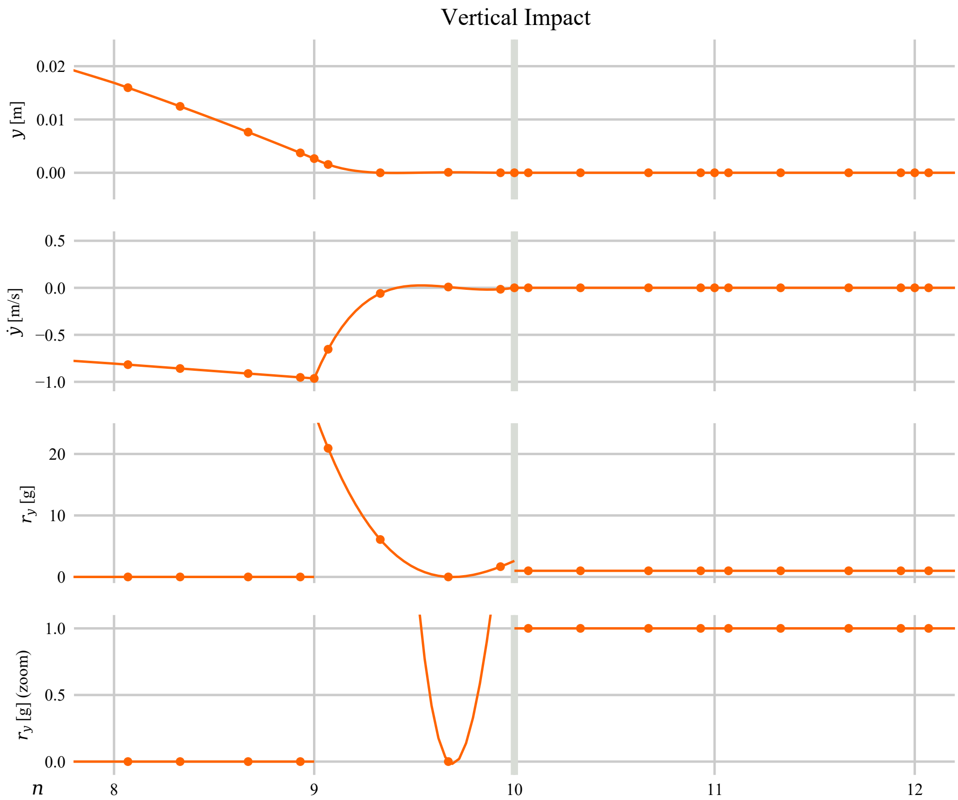

Figure 5.3: Caption

5.2.2 Contact with Friction

Friction Model

A bewildering variety of friction simulation models spanning a wide spectrum of complexity levels are used across different engineering and scientific disciplines [153]. To select a suitable friction model to complete the contact formulation, we must narrow down this broad taxonomy to those with favorable characteristics for trajectory optimization. Models from the dynamics and control field are the most likely candidates, as they are devised with rigid-body simulation in mind, and prioritize simplicity and computational efficiency. Reviews focusing on just this category include those by Marques et al. [154], Pennestri et al. [155] and Brown and McPhee [156]. Despite this restricted scope, there are still many approaches to consider.

One way to categorize the available options is based on their relationship to the state variables of the dynamic system:

- **Static** friction models calculate the friction force based on existing state variables, with Coulomb's Law being a simple example.
- **Dynamic** friction models add new state variables to capture additional properties, most of which relate to the transition between sticking and sliding modes. These include the Dahl model [157], and bristle models [158].

Static friction models are preferable for transcribed trajectory optimization problems, as they typically allow the friction force at a given collocation point to be calculated using only variables at the same point. This maximizes the sparsity of the friction constraint Jacobian, making these models more computationally tractable. Consequently, we will exclude dynamic friction models from our analysis, and any static models that include memory.

This leaves velocity-dependent friction models of the form

$$\mathbf{r_t} = -\text{sign}(\mathbf{v_t})|\mathbf{r_n}|\mu(|\mathbf{v_t}|)\hat{\mathbf{v}}_\mathbf{t} \tag{5.24}$$

where $\hat{\mathbf{v}}_\mathbf{t}$ is a unit vector in the same direction as $\mathbf{v_t}$, and $\text{sign}(\mathbf{v_t})$ is the set-valued function

$$\text{sign}(\mathbf{v_t}) \in \begin{cases} 1, & |\mathbf{v_t}| > 0 \\ [-1, 1], & |\mathbf{v_t}| = 0 \end{cases} \tag{5.25}$$

We will refer to models of this type as *Coulomb friction* models, as the simplest example with constant μ corresponds to Coulomb's Law.

Friction Cone Model

In the described friction models, the set of possible friction forces in a two-dimensional contact plane forms a disc with a radius of μ around the contact point. We refer to the combination of this disc with the set of possible normal forces as the *friction cone*. The reaction force acting at a stationary contact can fall anywhere in the interior of the cone, while the force at a sliding contact must lie on the boundary of the cone in the opposite direction to the relative tangential velocity, $\mathbf{v_t}$.

It is possible to simulate reaction forces falling anywhere on the friction cone [159, 160], but these techniques involve a nonlinear transformation of $\mathbf{v_t}$ and $\mathbf{r_t}$ into a polar representation of the contact plane. A more computationally-efficient option is to work with a polyhedral approximation of the friction cone, where the disc of possible tangential forces is replaced by a polygon [129]. This polygon is the convex hull of k evenly-spaced direction vectors of length μ. For planar problems, $k = 2$. The minimum number of vectors for a spatial problem is usually $k = 4$, giving a set of direction vectors that coincides with the positive and negative directions of the x and z axes describing the contact plane. We can write the friction force as

$$\mathbf{r_t} = \mu r_n \mathbf{d^k} \alpha^{\mathbf{k}} \tag{5.26}$$

using a set of k unit vectors, $\mathbf{d^k}$. The vector $\alpha^{\mathbf{k}}$ consists of k activation variables, each having a value between zero and one.

Unless the direction of $\mathbf{v_t}$ falls precisely between two of the direction vectors, only one element of $\alpha^{\mathbf{k}}$ should be nonzero at any point where sliding occurs. If $\mathbf{v_t} = \mathbf{0}$, $\alpha^{\mathbf{k}}$ must take on the values required to oppose the net tangential force acting on the bodies. To manage these activation variables, we introduce an auxiliary variable $\gamma \geq 0$ and relate it to each activation variable α_i^k with the complementarity constraint:

$$\alpha_i^k[n,p] \perp \gamma[n,p] - \mathbf{d_i^k} \mathbf{v_t}^T[n,p], \quad \forall i = 1, 2...k \tag{5.27}$$

The requirement that $\gamma - \mathbf{d_i^k} \mathbf{v_t}^T \geq 0$ means that γ will equal the magnitude of the largest projection of $\mathbf{v_t}$ onto one of the unit vectors in $\mathbf{d^k}$ – that is, the magnitude of the projection of $\mathbf{v_t}$ onto the unit vector that best matches its direction. The right-hand side of (5.27) can only be zero for the constraint corresponding to this nearest direction vector, so only the associated activation variable can have a nonzero value.

To ensure that its value will be one if $|\mathbf{v_t}| \geq 0$ (and hence, if $\gamma \geq 0$), we complement

$$\sum_{p=0}^{P+1} \gamma[n,p] \perp 1 - \sum_{i=1}^{k} \alpha_i^k \tag{5.28}$$

We sum γ over all points to prevent changes between sticking and sliding modes from happening mid-element.

5.2.3 Variable Timestep

Because the contact constraints prevent mode changes from occurring mid-element, a variable timestep is required to allow them to happen with more flexibility. This increases the computational complexity of the problem, as it causes the interpolation (5.15b) and integration (5.15b) constraints to become nonlinear, and reduces the sparsity of the problem. To minimize the effect of this change on solver performance, it is advisable to limit the range of variation to within one order of magnitude. We code $h[n]$ as a scalar on a maximum timestep, h_m, and unless otherwise stated, it is assigned the range $h[n] \in [0.1, 1]$. The constraint

$$h_m \sum_{n=1}^{N} h[n] = T \tag{5.29}$$

is used to give the complete trajectory a set duration T where required.

5.3 Method Comparison

When designing a collocation scheme for trajectory optimization, there is a trade-off between accuracy and performance. Major improvements in accuracy can ultimately only be achieved by adding points to the simulation, either through increasing the number of collocation points in each finite element, or by partitioning the trajectory into more elements.

For complicated systems with many degrees of freedom, the largest contributor to the computational cost is likely to be the number of EOM evaluations per element. Even if this is constant, the trade-off is evident in the details of the different collocation methods: LG is ostensibly the most accurate Pth-order method, but it requires additional, explicitly-defined meshpoints, and associated integration constraints that become nonlinear once the variable timestep is introduced. The LGL method is the least accurate, but because it requires one fewer EOM calculation, it appears to offer a $(P+1)^{\text{th}}$-order approximation at comparable performance cost to the Pth-order LG method.

If the number of ODE evaluations over the whole trajectory is considered, there are also different, approximately computationally-equivalent mesh configurations for the same method, as the same total number of points can either be divided into more low-order finite elements, or fewer high-order ones. It is not obvious which of these approaches is likely to be preferable: whether the solution on some interval can be better approximated by a single high-order polynomial or several low-order ones depends on the specific behaviour within that interval. For contact-implicit problems, high-order polynomials reduce the number of possible contact mode changes, but this could either be an advantageous simplification or disadvantageous restriction depending on the problem.

In this section, we conduct an empirical comparison of the different methods for test problems resembling our intended application.

5.3.1 Error Metrics

To evaluate which methods and mesh configurations are likely to give the best balance of accuracy and performance, we must first define suitable metrics to compare them. While the solve time serves as a straightforward metric of performance, measuring accuracy is more complex, as error can be quantified in various ways:

Global Error

The **global error** is the difference between the true solution and the approximation. It can only be obtained precisely for systems with a closed-form solution, but if the problem cannot be solved analytically, the true solution can be substituted with a better numerical approximation. This can be obtained by simulating the system using a more accurate method, or much finer mesh, using input values sampled from the piecewise-continuous input spline $\mathbf{u}^P(t)$.

For systems with contact, mode transition events in the true solution are unlikely to coincide with those in the approximate trajectory, making the approximation of the reaction forces inconsistent. This creates two possible versions of the "true" solution:

1. the true *smooth* solution i.e. the accurate smooth dynamic simulation of the system under the action of the approximate input and reaction force splines, which may violate the contact constraints, or
2. the true *non-smooth* solution i.e. the accurate non-smooth dynamic

simulation of the system under the action of the approximate input splines.

The error with respect to the true non-smooth solution is the more physically meaningful metric, and gives the most direct indication of the distance between simulation and reality for the approximation. This is an intuitive interpretation of "accuracy", but arguably not a good metric of whether the collocation method performs as intended, as it disregards the assumption of smooth dynamics that these methods are devised under. Even in the smooth case, the global error on latter finite elements tends to measure the compounding "butterfly effect" of preceding error more than the ability of the collocation polynomial to fit the local behaviour. The mean global error is therefore sensitive to the timing of errors within the trajectory, with early errors being exaggerated.

Discretization Error

The discretization error compares the approximate solution to the true smooth solution over individual finite elements. As with the global error, the "true" solution ($\hat{x}_i$) is typically obtained by simulating the system's progress under the action of the approximate input, but rather than taking error accumulated over previous intervals into account, the initial condition of each finite element is assumed to be correct:

$$\hat{x}_i(t) = x_i(t_0) + \int_{t_0}^{t} f_i(t, \hat{\mathbf{x}}(\lambda), \mathbf{u}^P(\lambda))d\lambda \tag{5.30}$$

$$\eta_i(t) = \hat{x}_i(t) - x_i^P(t) \tag{5.31}$$

To calculate the discretization error, we simulate each finite element forward from its initial point using LG5 collocation with a finer mesh ($N = 20P$, where P is the number of collocation points in the original approximation). For each state variable, we take the mean of the error at the collocation and endpoints, and normalize this by dividing by the largest magnitude of the variable on that element. These are then aggregated to give the mean discretization error across all finite elements and state variables.

Local Error

While the discretization error measures whether the approximate solution matches the true solution on a given finite element, the local error measures the extent to which the approximate solution solves the ODE between collocation points. The constraints of the problem ensure that the ODE must be solved at the collocation points, but generally, the approximate derivative $\dot{x}_i^P(t)$ will not be equal to the derivative calculated using the approximate solution and input $f_i(t, \mathbf{x}^P(t), \mathbf{u}^P(t))$. The accumulation of this difference produces the **local error**:

$$\epsilon_i(t) = \dot{x}_i^P(t) - f_i(t, \mathbf{x}^P(t), \mathbf{u}^P(t)) \tag{5.32}$$

$$\eta_i(t) = \left| \int_{t_0}^{t} |\epsilon_i(\lambda)| d\lambda \right| \tag{5.33}$$

This is related to the discretization error in that the true solution solves the ODE at all points, so perfect discretization accuracy implies perfect local accuracy and vice versa, but the local error does not directly indicate how closely the P^{th}-order spline approximation matches the true solution – it indicates whether this approximation is internally consistent. This distinction becomes especially apparent when the form of the true solution is very different from that of the polynomial.

The advantage of this metric it does not require the calculation of a "true" solution that might not adhere to the contact constraints. For this reason, Kelly [143] recommends refining the collocation mesh based on the local accuracy in his collocation tutorial, and likewise, this is the primary metric we will use to compare methods.

The local error at all points on a given finite element is calculated as follows:

1. Expressions for the derivative polynomials are obtained using (5.4).
2. Expressions for the state polynomials are obtained by integrating the derivative polynomials (analytically) from the initial state of the element.
3. For each collocation point τ_p, the approximate state and derivative are calculated at five quadrature points spaced between the start of the element and τ_p. The ODE error ϵ is then evaluated at each of the quadrature points.
4. The ODE error values are combined in a weighted sum according to a fifth-order Gaussian quadrature rule to give the local error. This rule is exact for polynomials of orders exceeding the approximating polynomials of all collocation methods trialed.

As with the discretization error, the local error for each variable on each finite element is normalized by the largest local value of the variable, and then aggregated into a mean for all variables and elements.

5.3.2 Experiments

Candidate Methods

We will compare the accuracy and performance of nine direct collocation schemes for different smooth and non-smooth problems. This includes the second- and fourth-order implementations of all three methods. We also compare the second-order methods to LGL3 because, in terms of ODE evaluations per element, LGL3 is equivalent to LG2 or LGR2. Likewise, we will compare the fourth-order methods to LGR5. For brevity, will refer to the grouping of the second-order methods and LGL3 as "low-order methods", and the fourth-order and LGL5 as "high order methods". For benchmarking purposes, we will also include the established first-order (FO) contact-implicit method devised by Posa, et al [12], as this is the most direct predecessor to our proposed CIO scheme in the literature.

Trials

We will conduct the following tests to compare the collocation schemes:

1. **Spring-Mass-Damper Simulation:** The spring-mass-damper (SMD) is a useful test system to use as a benchmark, because it is analytically solvable, and its parameters can be adjusted to vary the approximate order of the solution on each time interval without changing the basic form of the response. The purpose of this preliminary test is to provide a baseline comparison of each accuracy metric for the collocation methods with respect to the order of the approximate solution.

2. **Underactuated Pendulum Optimization:** Swinging an underactuated pendulum into an inverted position is a well-known test problem in optimal control. This can be converted to a non-smooth dynamic problem by limiting the range of motion of the joints using hard stops. This test will compare the performance of the collocation methods for an otherwise-identical problem with and without contact.

3. **Legged Locomotion Optimization:** This test compares the accuracy and performance of the collocation methods during their intended application: the trajectory optimization of legged locomotion tasks. Three tasks are included, encompassing systems with a similar number of degrees of freedom, but a varying number of contact constraints.

5.3.3 Spring-Mass-Damper Simulation

Method

The SMD model is a one-dimensional system consisting of a spring and damper in series and a mass sliding on a frictionless surface. The only input is an exciting force acting on the mass. The spring and damping coefficients are selected to give a damping ratio of $\zeta = 0.1$ and damped frequencies within the range $\omega_d \in [0.1\pi, 16\pi]$ with unit mass. The exciting force is constant, and scaled so the response will always have unit final value.

We generated 50 SMD simulations each of duration $T = 3$ seconds. The mesh for the low-order methods consists of $N = 30$ finite elements ($h = 0.1$ seconds), the high-order methods have $N = 15$ elements ($h = 0.2$ seconds), and the FO method was assigned $N = 60$ timesteps ($h = 0.05$ seconds) to give either 60 total points or 60 ODE evaluations for all methods.

The damped frequency varies randomly over the stated range. These values were selected so the approximate order of the true solution for a single timestep varies between 1 and 3 for the low-order mesh, and between 1 and 4 for the high-order mesh. We calculated the approximate order $\hat{\mathcal{O}}$ based on the number of stationary points typically occurring within a timestep, for a sinusoid at the damped frequency. Each period has two stationary points, so the number per element is taken to be twice the number of $\frac{2\pi}{\omega_d}$ second periods fitting within h, rounded to the nearest integer. This gives an order of roughly

$$\hat{\mathcal{O}} = 1 + \text{nint}\left(\frac{h\omega_d}{\pi}\right) \tag{5.34}$$

A convenient feature of the sinusoidal response is that all derivatives will have the same frequency, so both the velocity and acceleration components of the derivative polynomial $\dot{\mathbf{x}}^{\mathbf{P}}$ will be approximating trajectories of the same order. This creates a clear distinction between responses within the favourable order range for each method ($\hat{\mathcal{O}} \leq P - 1$) and those exceeding it.

Results

Figure 5.4 compares the global, discretization and local error for all methods. The global and discretization error values were obtained through comparison to the analytical solution at each collocation point. The global error for each variable is normalized by the largest magnitude of the variable over the whole trajectory, and these are then aggregated into the mean.

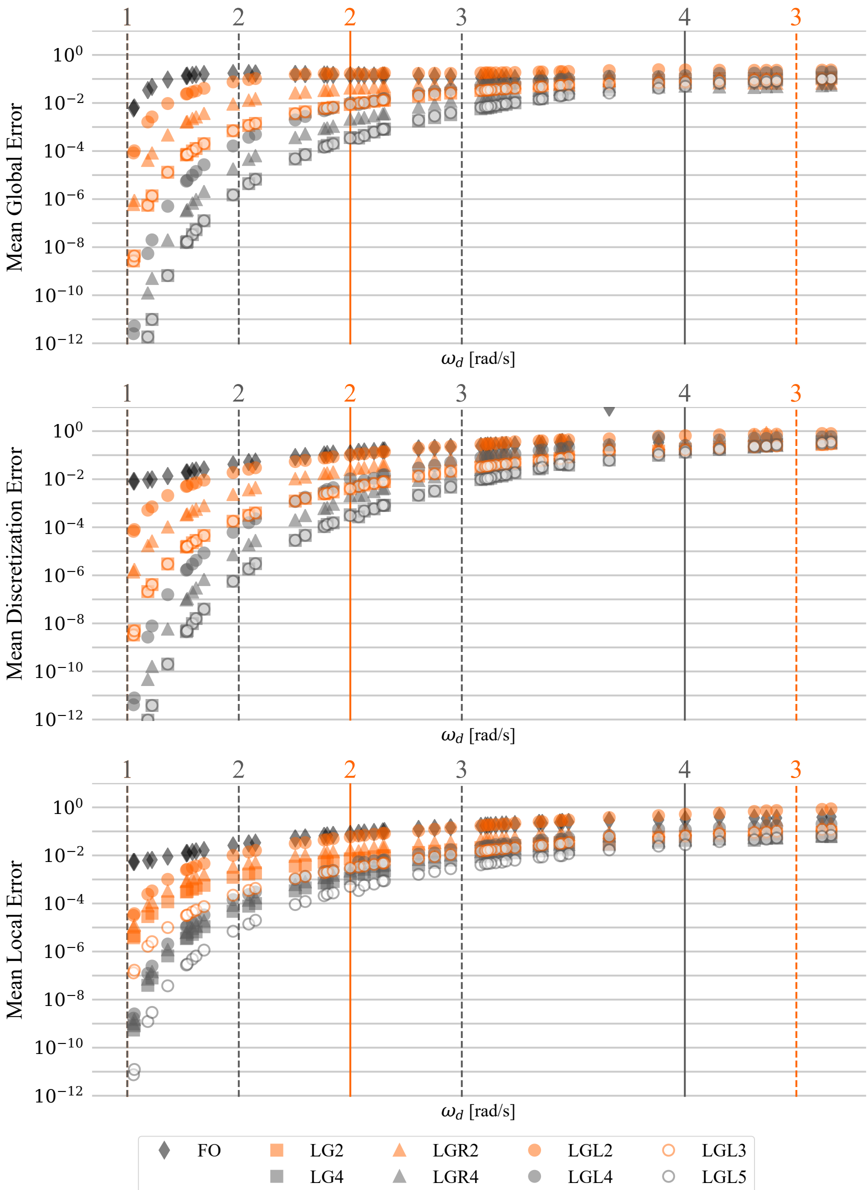

Figure 5.4: Mean global, discretization and local error for different collocation methods when simulating spring-mass-damper systems with different damped frequencies (ω_d). The vertical lines indicate the approximate order of the true derivative for the datasets plotted in the same color, with solid lines showing where it begins to exceed the order of the approximating polynomial for second- (orange) and fourth-order (gray) methods.

All three metrics show consistent trends and similar magnitudes of error for each method. As would be expected, higher-order methods achieve greater accuracy, but their advantage decreases as the approximate order of the solution approaches the theoretical limit for the method. The same is true of the accuracy advantage of less-constrained methods (LG) over more-constrained ones (LGL) of the same order.

The LGL($P+1$) and LGP methods are the most accurate options in each grouping of P ODE evaluations per element, achieving near-identical global and discretization error results. Interestingly, the LGL($P+1$) method tends to be more accurate with respect to local error within the approximate order limit of each method, while the LGP method gains a slight advantage once this limit is exceeded.

5.3.4 Underactuated Pendulum Optimization

Model

We use a three-link planar pendulum with all links modelled as uniform thin rods having unit mass and length. The coordinates of the system are the angles of the links anticlockwise from the vertical axis of the inertial frame [23]. The base link is actuated by an ideal, unbounded torque, while all other links are unactuated.

For the smooth version of the problem, the relative angles of the links are not bounded. For the non-smooth version, all joints except at the base are restricted so the subsequent link may not exceed an angle of $\frac{\pi}{2}$ radians in either direction with respect to the axis of the parent link. These limits are enforced by a pair of hard stops, modelled as frictionless contacts. These contacts are defined by modifying the impact constraint (5.23) such that the normal distance is replaced by the difference between the current relative position of the joint and the limit, and the normal reaction becomes a reaction torque opposing further motion [12].

Trajectory Optimization Task

To complete the swing-up motion successfully, the pendulum must move from hanging at rest, to inverted rest. This is described by the initial and final conditions:

$$\theta_i[1, p_i] = 0 \qquad \forall i = 1...3 \tag{5.35a}$$
$$\dot{\theta}_i[1, p_i] = 0 \qquad \forall i = 1...3 \tag{5.35b}$$
$$\theta_i[N, p_f] = \pi \quad \forall i = 1...3 \tag{5.35c}$$
$$\dot{\theta}_i[N, p_f] = 0 \quad \forall i = 1...3 \tag{5.35d}$$

The simulation time is fixed to $T = 2$ seconds. For the smooth problem, the timesteps are fixed to $h = \frac{T}{N}$, while the maximum timesteps for the non-smooth problem are set to $h_m = \frac{1.2T}{N}$. Meshes for the low-order, high-order and FO methods were assigned $N = 100$, 50 and 200 finite elements, respectively.

The objective of the task is to minimize the actuator effort. This is defined as the sum of squared actuator torques at all points:

$$\min_{\mathbf{X}} \sum_{n=1}^{N} \sum_{p=p_i}^{p_f} \alpha[n, p]^2 \tag{5.36}$$

Procedure

The smooth problem was solved in a single optimization stage, while the non-smooth problem was solved in two stages. Each problem was solved 50 times per method, starting from a seed generated by uniformly randomizing the state variables at all points within the range [-0.1,0.1].

Results

The mean local error and solve times for the pendulum problems are plotted in Figure 5.5. Unsurprisingly, the solve times are much longer for the problem with contacts. The higher-order versions of each method consistently solve this problem faster, indicating that allowing fewer opportunities for contact state changes does simplify the problem rather than over-constraining it. As expected, methods requiring more defined points per element do tend to solve slower, but this difference becomes relatively minor once contacts are introduced.

There are two key features of the non-smooth problem that potentially affect the accuracy of the collocation method:

1. the approximation of discontinuous velocity changes during collision events, and

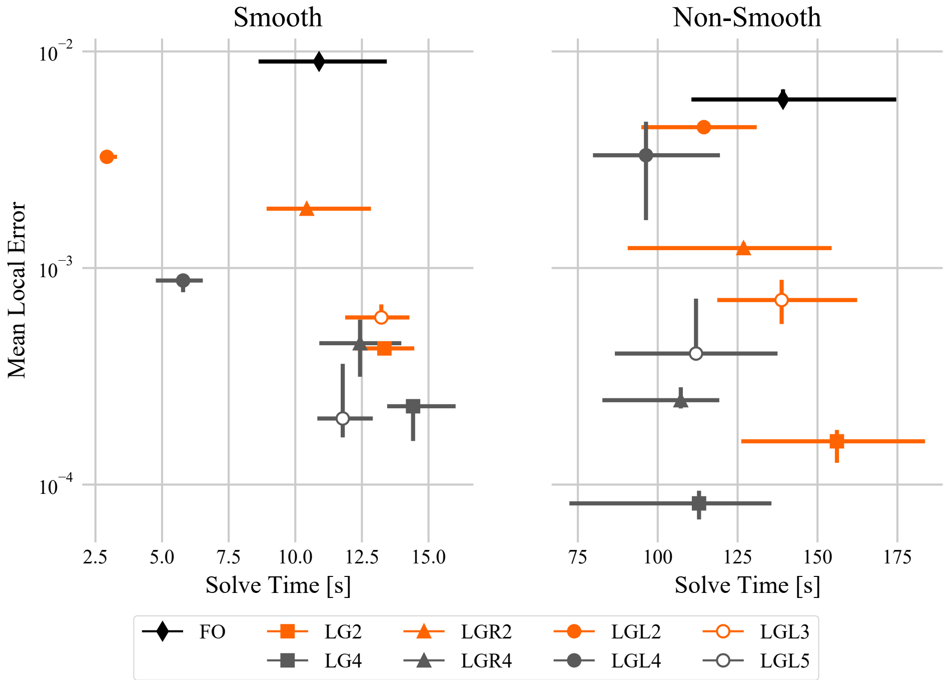

Figure 5.5: Mean local error and solve times for the smooth and non-smooth pendulum swing-up task. The marker gives the median value, while the bar indicates the interquartile range.

2. the variable timestep.

For this problem, the fixed simulation time means that the average duration of the finite elements is the same for the smooth and non-smooth problems, so differences in accuracy between the two problems are more likely to be a reflection of how effectively each method approximates the collision behaviour.

These events do not appear to be especially problematic for any particular method, with accuracy for most remaining within the same order of magnitude for both problems, or often even improving for the non-smooth problem. The comparative accuracy of the methods tends to follow the trends established in the SMD test, with the overlap between the less accurate high-order and more accurate low-order methods resembling the region of solutions with approximate orders exceeding the theoretical limit of the second-order methods. The exception is LGL($P + 1$), which falls behind compared to other methods for non-smooth problems.

Comparing the local and discretization accuracy of the methods (see: Figure 5.6) gives further insight into how the collision events affect the collocation. Because the discretization error is a direct measure of the accuracy with respect to the true solution, large values compared to the local error would indicate that the collocation polynomials are a poor fit for the contact transitions. The discretization error is far more varied when contact is involved, but the median values are still consistent with (i.e. within the same order of magnitude as) the local error and smooth problem results for almost all methods. Notably, the discretization error is smaller than the local error for the most accurate second-order methods, suggesting that collisions are better approximated using shorter, lower-order finite elements. This has a limit, however, as the first-order method is the only one revealed to be far less accurate in terms of the discretization error.

5.3.5 Legged Locomotion Optimization

Models

Two models are used in this test: a planar biped with arms, and a spatial monopod. These are illustrated in Figure 5.7. Both models effectively have nine degrees of freedom (DOF): although the monopod is modelled using 10 generalized coordinates to produce more tractable equations of motion [23], the leg is constrained so it cannot yaw relative to the body. Each model includes contacts between the feet and ground, subject to friction modelled using Coulomb's Law with friction coefficient $\mu = 0.6$.

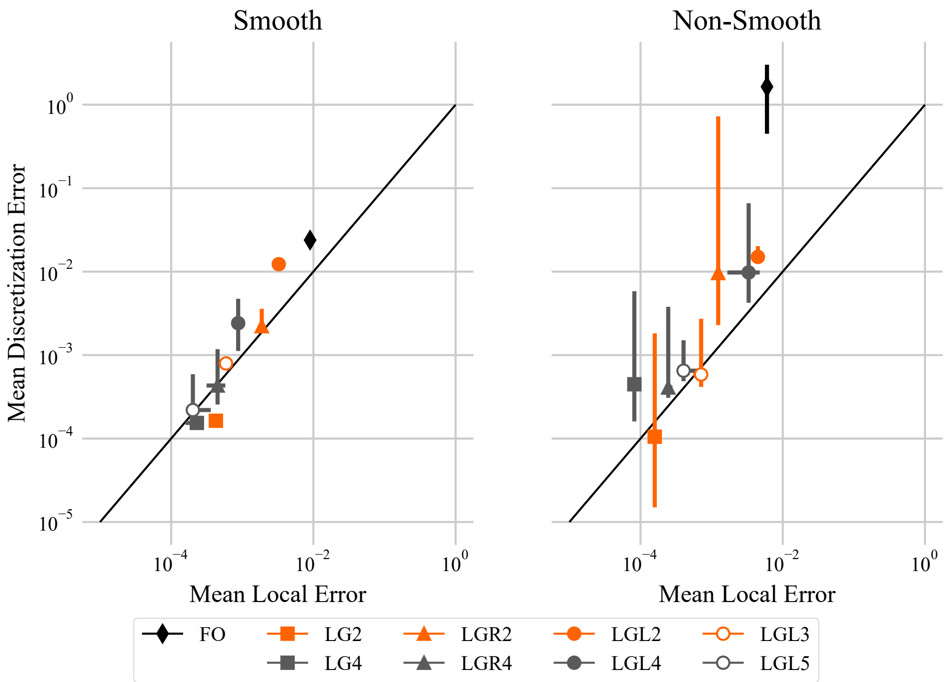

Figure 5.6: Mean discretization and local error for the smooth and non-smooth pendulum swing-up task. The marker gives the median value, while the bar indicates the interquartile range. The line of one-to-one correlation is indicated in black.

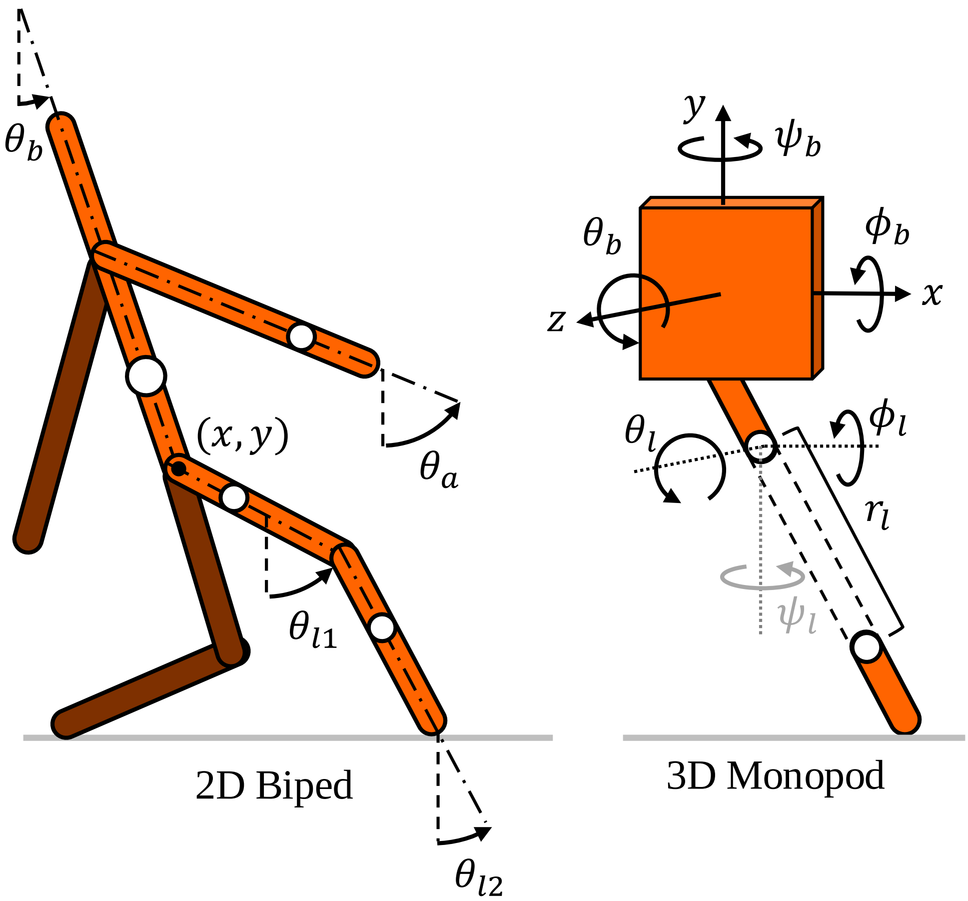

Figure 5.7: Planar biped and spatial monopod model used in trajectory optimization experiments.

Trajectory Optimization Problems

Bipedal Stopping: To simulate a stopping maneuver, the initial condition for the biped was sampled from the midstance phase of a sprinting trajectory. The final condition required grounded feet, no forward translational or rotational velocity (that is, $\dot{x} \leq 0$ and $\dot{\theta}_b \geq 0$), and all velocities to have magnitudes within five percent of their initial values.

The stopping distance was minimized by creating a variable upper bound

$$x[n,p] \leq x_m \quad \forall n = 1...N, \forall p = p_i...p_f \tag{5.37}$$

and then minimizing x_m.

Monopedal Turning: The monopod was required to start at rest in an upright position, and travel 2.5 m in the x direction without exceeding $z \leq 0$, followed by 2.5 m in the z direction. The final state was not specified beyond the requirements that $x = z = 2.5$ m, and that the yaw of the body (ψ_b) be displaced 90 degrees from its starting point.

We minimized the sum of the squared actuator forces and torques over all joints and collocation points:

$$\min_{\mathbf{X}} \sum_{n=1}^{N} \sum_{p=p_i}^{p_f} \alpha_h[n,p]^2 + \alpha_k[n,p]^2 \tag{5.38}$$

Mesh parameters: All described tasks were assigned the same mesh parameters. The duration of the trajectory was not fixed, but a maximum time of 2.5 seconds was selected by setting the maximum timesteps according to $h_m = \frac{2.5}{N}$. The low-order, high-order and FO methods were assigned $N = 50$, 25 and 100 finite elements, respectively.

Procedure

Each of the two problems was solved 50 times for each collocation method, starting from a seed generated by uniformly randomizing the state variables at all points within the range [-0.1,0.1].

5.3.6 Results and Discussion

The solve time and mean local error results for the legged locomotion tests are displayed in Figure 5.8. The first-order method is excluded for clarity, as its error values were several orders of magnitude larger in both tests.

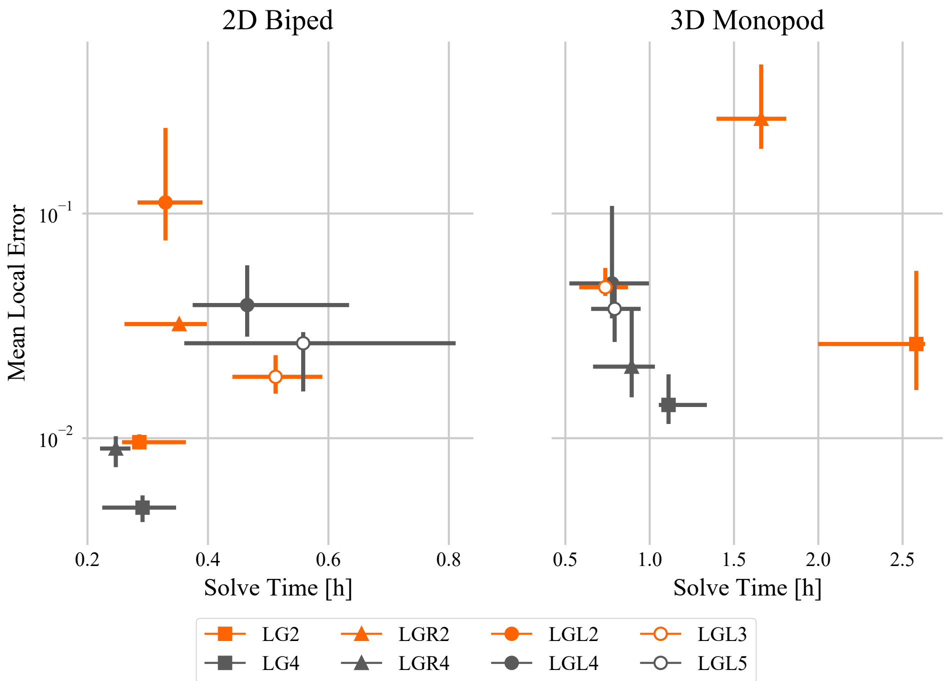

Figure 5.8: Mean local error and solve times for the planar biped stop and spatial monopod turn. The marker gives the median value, while the bar indicates the interquartile range.

The results mostly follow the trends observed in the non-smooth pendulum test, with higher-order methods mostly solving faster and LG4 proving most accurate.

The performance discrepancy between the higher- and lower-order methods is especially exaggerated in the monopod problem, with LG2 taking more than twice as long to solve as LG4, though it should be noted that all orders of LGL are the fastest methods for this problem. By contrast, both high-order LGL methods solve more slowly than their low-order counterparts in the biped test, confirming that the results of these trials are somewhat problem-dependent.

5.4 Conclusions

This chapter extended contact-implicit collocation [12] to high-order approximations, and compared different collocation methods over a range of smooth and non-smooth trajectory optimization problems. The effectiveness of some methods was problem-dependent, but there were some consistent trends:

- As would be expected, less-constrained methods are more accurate than more-constrained methods of the same order, with LG consistently achieving the lowest mean error for each order group.
- Higher-order methods are faster for non-smooth problems than lower-order methods.
- Less-constrained methods are generally slower than more-constrained methods of the same order due to the additional simulation points, but this difference is minor compared to the difference between high- and low-order methods.

Based on these observations, LG4 appears to offer the most consistently favorable combination of accuracy and computational efficiency.

Chapter 6

Impulsive Collisions

The orthogonal collocation formulation described in the previous chapter assumes that the trajectory of all state variables is continuous. In this chapter, we modify it to use piecewise-continuous velocities, so finite discontinuities can occur between finite elements. This allows collisions to be modeled as impulsive events, which expands the behaviors the formulation can handle to include partially-elastic impacts and impacts without collision - the established resolution of the Painlevé paradox.

The first section establishes context by expanding on the motivation for allowing velocity discontinuities, and giving the mathematical background. We then explain how the contact model is altered for impulsive impact. The final section compares the performance of the discontinuous and continuous fourth-order Legendre-Gauss formulations for the same legged locomotion test problems used in the previous chapter. The work in this chapter has been published in [26].

6.1 Background

6.1.1 Motivation

The dynamics of contact, as modelled in the previous chapter, can be considered "non-smooth" in that the system transitions between distinct dynamic modes where it is subject to different path constraints. This model is, however, still "smooth" in the sense that the trajectories of all state variables remain continuous over these mode transitions. This is not true to the way collisions are conceptualized for rigid-body systems. For the assumption of

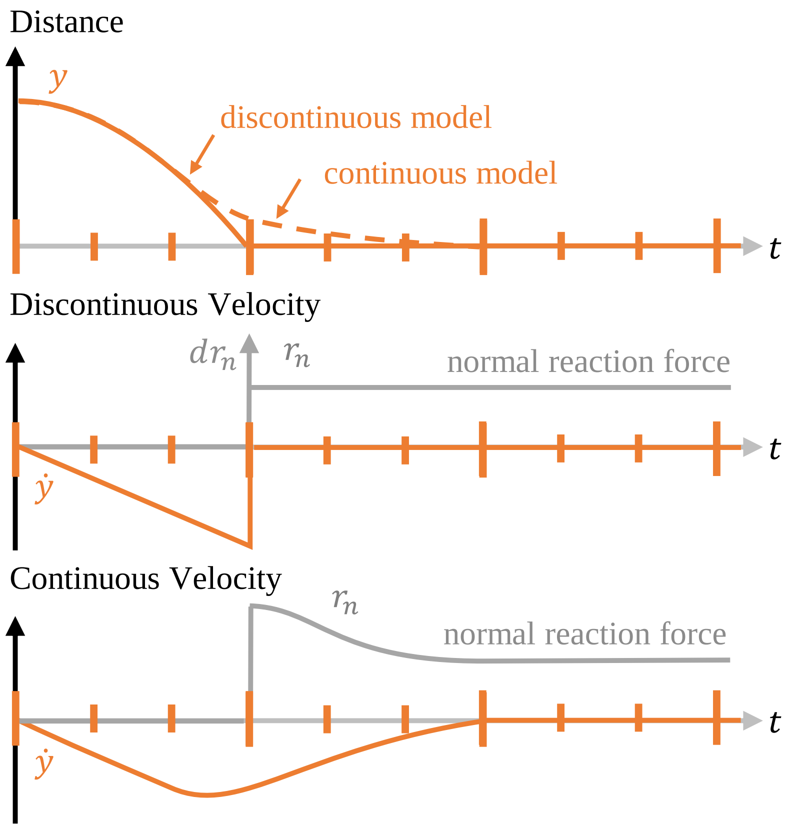

Figure 6.1: Conceptual comparison between implicit direct collocation with discontinuous, and continuous velocity states. In the continuous formulation, the velocity must transition to zero over a full timestep under the action of finite forces. This causes the instantaneous impact velocity to be smaller than its true value, making it impossible to model partially elastic collisions accurately.

rigidity to hold, collisions must be impulsive: relative motion must cease at the instant of contact, under the action of infinitely-large forces. A comparison between the impulsive model of impact and its continuous approximation is shown in Figure 6.1.

One reason to favour the impulsive version is that it allows partially-elastic behaviour to be modelled using an implicit numerical integration scheme. If the continuous impact formulation is combined with implicit integration, deceleration has already occurred at the first simulation point where the distance between bodies reaches zero, so the true velocity at the instant of collision is not captured. This makes it impossible to model partially-elastic collisions reliably, as the restitution law is applied at the velocity level. These problems can be avoided by using partly-implicit integration instead [161], but that sacrifices the accuracy and stability for which implicit integration is typically favoured.

Another reason is that there are some problems where the combination of a continuous impact model and Coulomb friction makes it impossible to find a solution. Consider the problem of a two-link pendulum resting on a conveyor belt, as illustrated in Figure 6.2. When the belt moves backwards relative to the pendulum, friction (r_t) is related to the normal force (r_n) by $r_t = \mu r_n$. If its coefficient of friction (μ) is sufficiently large, the torque on the lower link produced by friction will be larger than the torque produced by the normal force, creating an angular acceleration that directs the end of the link down into the conveyor belt. Of course, it is not possible for the link to move in this direction, so the problem appears to have no solution.

This apparent conflict between the interpenetration constraint and Coulomb friction is called the Painlevé paradox, and has been an important topic of discussion and driver of theoretical development in rigid-body dynamics since its description near the turn of the 20th Century [162]. Its history, consequences and the ongoing questions it raises are well-documented in a review by Champneys and Várkonyi [163]. Besides the inconsistent case described, the paradox might also result in an indeterminate case, where multiple solutions are possible [163].

Stewart [164] points to the assumption of finite reaction forces (and, by implication, the assumption of a time-continuous velocity state) as a key flaw leading to inconsistency: "In particular, it rules out the possibility that the horizontal component of the velocity (v_t) could be brought to zero *instantaneously* by impulsive contact force". If v_t is immediately brought to zero, it is no longer required that $r_t = \pm\mu r_n$, so a solution becomes possible. This demands that contact models allow not only impulsive forces during collisions,

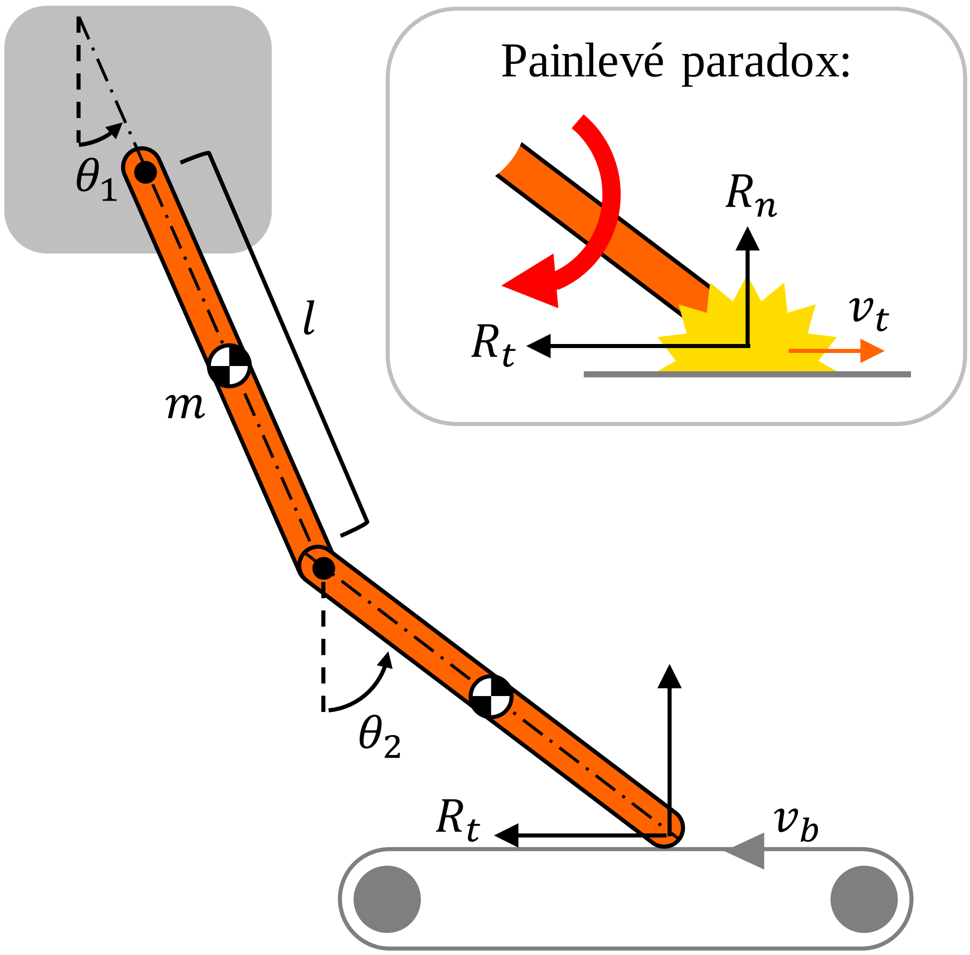

Figure 6.2: Two-link pendulum resting on a conveyor belt. The Painlevé paradox will occur if the coefficient of friction between the belt and pendulum is sufficiently large.

but *impacts without collision* (IWCs) – instantaneous jumps in the tangential velocity occurring without a change in the contact state. Experiments by Zhao et al. [165] using an apparatus similar to the pendulum and conveyor belt problem in Figure 6.2 indicate that IWCs are not just a convenient *patch* for a *bug* in the mathematics, but a representation of a real physical phenomenon, as tangential shocks were observed when the apparatus was arranged in paradoxical configurations.

It may be tempting to dismiss the paradoxes as niche cases happening only at unrealistically high coefficients of friction, but with unfortunate contact geometry or mass distribution of the bodies involved, they can come about under more typical conditions [166]. Analysis of two widely-used passive dynamic walking models suggests that they are far from unlikely in legged locomotion [167], for example. A further contribution of the model we describe in this paper is that it is the first higher-order collocation scheme to allow IWC resolution of frictional paradoxes.

6.1.2 Mathematical Background

The discontinuous nature of rigid-body systems with contact constraints can be conceptualized by changing the equations of motion from ODEs of the form,

$$\dot{x}_i(t) = f_i(t, \mathbf{x}(t), \mathbf{u}(t)), \tag{6.1}$$

to *measure differential inclusions* (MDIs). The MDI is a generalization that allows the right-hand side of the differential equation to be a combination of continuous and impulsive parts:

$$\dot{x}_i(t) = f_i(t, \mathbf{x}(t), \mathbf{u}(t)) + \sum_{j \in \mathbb{N}} \eta_j \delta(t - t_j) \tag{6.2}$$

Here, $\delta(t - t_j)$ is a unit impulse occurring at the instant t_j and η_j is the magnitude. Although $\delta(t)$ is often referred to as the Dirac δ-*function*, it is not really a function of time at all, but a *measure* – a function that acts on a set, which may be thought of as something closer to a distribution. The important assumptions we are making about this solution are that there are countably many discontinuities, and that $\dot{x}_i(t)$ has *bounded variation* over the trajectory (the difference between the left and right values of $\dot{x}_i(t)$ at each discontinuity can be assigned a finite value, η_j).

MDIs are the cornerstone of the mathematical framework developed by Moreau to handle a class of unilaterally-constrained mechanical problems he termed *sweeping processes* [168, 139, 169]. The immediate ancestors of our model are time-stepping methods based on Moreau's theory [161, 129, 170]. For further reference on these ideas, Stewart [161] gives a concise and accessible introduction, Brogliato et al. [166] gives a broader review of numerical simulation methods, and textbooks by Acary and Brogliato [171] or Leine and Nijmeijer [172] provide a more comprehensive text.

6.2 Impulsive Contact Model

6.2.1 Piecewise-Continuous Direct Collocation

In the previously-described continuous formulation, the value of each state variable at the initial boundary of one finite element was equated to the value the previous final boundary by a continuity constraint (5.15c). This is still applied to the position variables in the piecewise-continuous version, but it is modified for the velocity variables to allow finite jumps. This produces the updated continuity constraints $\forall n = 2, ...N$:

$$x_i[n, p_i] = x_i[n-1, p_f] \qquad \forall x_i \in \mathbf{q} \tag{6.3a}$$

$$x_i[n, p_i] = x_i[n-1, p_f] + \eta_i[n] \quad \forall x_i \in \dot{\mathbf{q}} \tag{6.3b}$$

where η_i represents the jump in the i^{th} coordinate. The resulting integration scheme is similar to a previous adaptation of arbitrary-order orthogonal collocation to hybrid dynamic problems [125].

The instantaneous velocity change is brought about by an impulsive contact reaction, **dr**. In our transcription, the velocity jump η and contact impulse are represented by the acceleration ($\ddot{\mathbf{q}}$) and reaction ($\mathbf{r}$) variables at $[n, 0]$. They are related by the impulsive equations of motion:

$$\mathbf{M}(\mathbf{q}[n, 0])\ddot{\mathbf{q}}[\mathbf{n}, \mathbf{0}] = \mathbf{J_c}^T(\mathbf{q}[n, 0])\mathbf{r}[n, 0] \tag{6.4}$$

As before, $\mathbf{M}$ is the inertial matrix, and $\mathbf{J_c}$ is the contact Jacobian.

6.2.2 Contact Constraints

As in the previous chapter, we will describe the constraints defining contact at a single point using the variables y for the relative normal distance between the involved bodies, x and z for the relative position in the tangential contact plane, $\mathbf{r}$ for the reaction force, and $\mathbf{v}$ for the relative velocity. The subscript n indicates the normal component of these vectors, while t indicates the tangential component.

The role of the impulsive reaction component is to produce a jump in the normal velocity (represented here in scalar form as $\dot{y}$) that satisfies $\dot{y}^+ = -e\dot{y}^-$, where e is the coefficient of restitution. We cannot complement $\mathbf{r_n}[n,0] \perp \dot{y}[n,0] - e\dot{y}[n-1,P+1]$ directly, however, as the right-hand side of this expression will not always be positive when the contact is inactive. We therefore implement the contact complementarity at the initial mesh point using positive auxiliary variables a^+ and a^- as follows:

$$a^+[n] - a^-[n] = \dot{y}[n,0] + e\dot{y}[n-1,P+1] \tag{6.5a}$$

$$\mathbf{r_n}[n,0] \perp y[n,0] + a^+[n] + a^-[n] \tag{6.5b}$$

$$a^+[n] \geq 0 \qquad a^-[n] \geq 0 \tag{6.5c}$$

The role of the finite reaction component is to prevent interpenetration when bodies are in contact for longer than an infinitesimal instant. They are governed by the same impact constraint defined in the previous chapter.

The friction constraints are also the same as those described for the continuous contact model, however, they are now additionally applied to the tangential reaction component defined at the initial mesh point.

6.2.3 Example: Falling Point Mass

To demonstrate impact, we simulated a one-dimensional point mass experiencing (A) an inelastic collision ($e = 0$) and (B) a partially elastic collision ($e = 0.5$) with the ground. $N = 20$ and $h_m = 0.02$, and the initial height $y[1,0] = 0.1$m. The only objective was minimizing the complementarity penalties. The inelastic collision was also simulated using the equivalent continuous-velocity formulation, to illustrate the differences between the two approaches. The partially elastic case is inadmissible for the continuous formulation, which is a key advantage of the proposed approach.

Figure 6.3 shows the resulting position, velocity and ground reaction force trajectories. The simulation of the inelastic collision using the continuous-velocity approximation was shown in isolation in Figure 5.3 in the previous chapter. The models behave as expected: the velocity in the impulsive problem jumps upon contact with the ground in accordance with the coefficient of restitution. A normal impulse occurs at the moment of impact, following which the normal force prevents the mass from falling through the ground in the inelastic case. The continuous approximation of touchdown takes place over a single finite element, with the mass not quite grounded when the normal force begins to decelerate it.

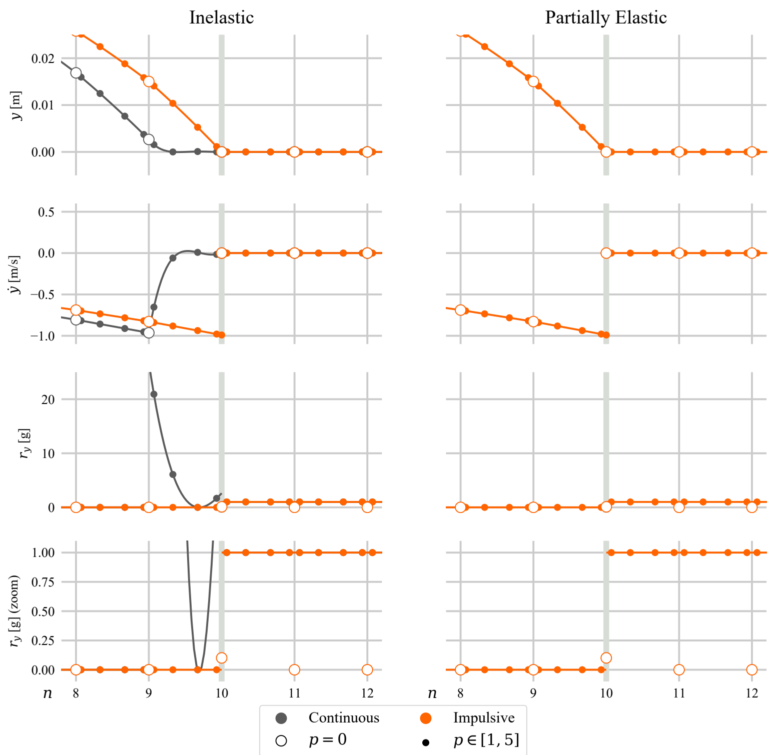

Figure 6.3: Trajectories of inelastic and partially elastic ($e = 0.5$) collisions between a point mass and the ground. The normal impulse dr_y is depicted at $p = 0$. A continuous velocity model is included for comparison on the inelastic case, showing the change in velocity spread out over an entire finite element.

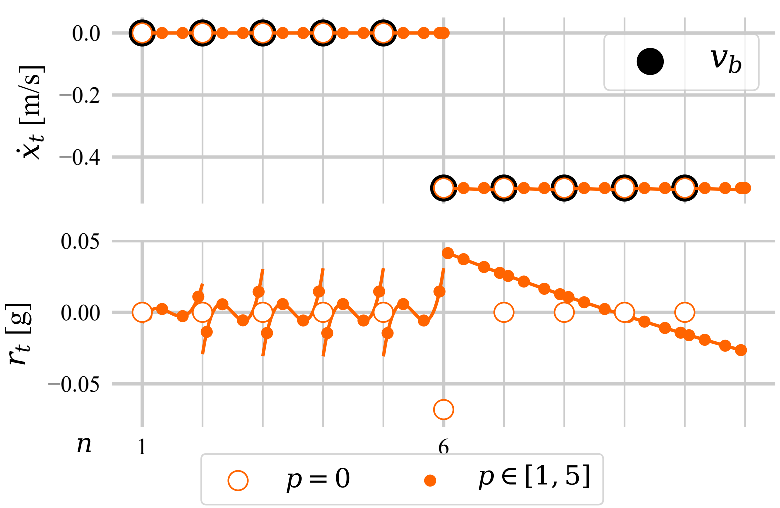

Figure 6.4: Solution to the pendulum and conveyor belt problem shown in Figure 6.2, demonstrating resolution of the Painlevé paradox via tangential impact at the start of the sixth finite element.

6.2.4 Example: Tangential Impact

Because the proposed formulation enables impulsive reaction forces to act at the boundaries of any finite elements where contact is active, it permits the resolution of the Painlevé paradox through a tangential impact without collision (IWC). To demonstrate this, we simulated a paradoxical situation in a planar system based on the two-link manipulator and conveyor belt apparatus that Zhao et al. used to investigate IWCs experimentally.

The model is shown in Figure 6.2. We assigned the links unit mass and a length of 0.5 metres, and assumed uniform mass distribution. Collisions were assumed to be perfectly inelastic. The height of the top link was selected such that the end of the double pendulum rests on the conveyor belt at initial angles of $\theta_1[1,0] = 0$ and $\theta_2[1,0] = 0.25\pi$ rad. The initial velocity was fixed to zero. The belt was initially stationary, but its velocity was abruptly stepped up to $v_b = 0.5$ m/s (that is, -0.5 m/s relative to the pendulum) at point [6,0]. We selected a very high coefficient of friction, $\mu = 2$, so Painlevé's paradox would be induced at this instant. The timing parameters were $N = 10$, $h_m = 0.02$s and $T \geq 0.1$s.

The results of the optimization are displayed in Figure 6.4. While the contact state never changes, an impulse occurs that instantly increases the tangential

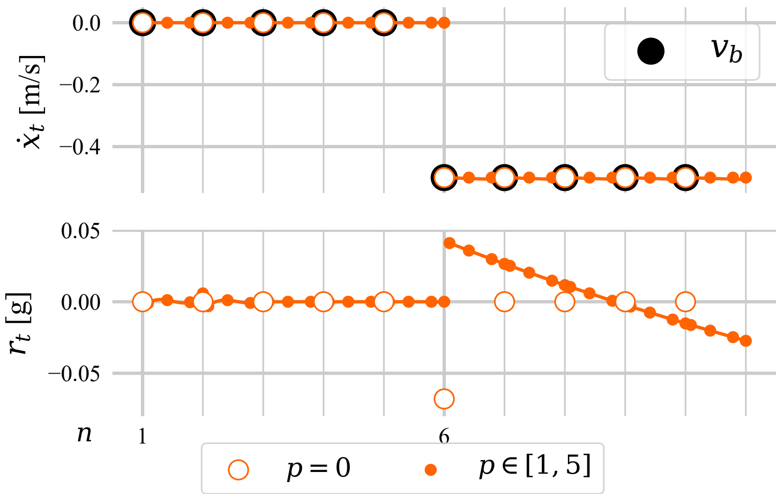

Figure 6.5: Solution to the pendulum and conveyor belt problem produced using Legendre-Gauss-Radau collocation. The ringing effect observed in Figure 6.4 is less apparent when this more constrained collocation scheme is used.

velocity of the pendulum's end to match the velocity of the belt. The end of the link therefore remains stationary relative to the belt throughout, so the magnitude of the friction force is allowed to be $< \mu r_n$, and a downward acceleration is not created at the contact point. This is precisely the resolution of the Painlevé paradox via IWC described before [165].

The tangential force oscillates over the finite elements in the stationary portion of the simulation. This *ringing* effect is able to occur due to the lack of an implied continuity constraint at the acceleration level in the Legendre-Gauss collocation scheme. (As the tangential reaction force is the only contributor to acceleration here, its form directly reflects the horizontal acceleration of the pendulum end.) In a scheme where collocation points are shared between adjacent finite elements such as Legendre-Gauss-Radau or -Lobatto, the acceleration spline – and therefore, the constraint force spline – would be more restricted, likely reducing this phenomenon. This can be observed in Figure 6.5, which shows the result of repeating the same experiment using Legendre-Gauss-Radau collocation. Unfortunately, there is not a clear way to eliminate constraint force ringing in high-order or unconstrained schemes without effectively reducing the order or degrees-of-freedom of the collocation.

6.3 Performance Comparison

The impulsive formulation has a clear benefit in trajectory optimization problems involving partially elastic contact and IWC, as shown above, but we are also interested whether it has any advantage over the continuous formulation when perfectly-inelastic contact is assumed, as is typical in legged locomotion tasks. It is possible that the ability to resolve frictional paradoxes could allow it to explore the solution space more effectively, and discover solutions that would be infeasible for other formulations, but this could be outweighed by an increased computational load similar to adding another collocation point to each element.

To evaluate its effect on solver performance and solution quality, we compare continuous and impulsive versions of the fourth-order Legendre-Gauss scheme (LG4C and LG4I, respectively) over two trajectory optimization tasks. We again include the first-order formulation by Posa. et [12] as a benchmark. Because this is a first-order method, it is also effectively discontinuous, and can therefore model tangential impacts, though it would require a semi-implicit formulation (as in the parent method by Stewart and Trinkle [129]) to evaluate partially-elastic collisions.

6.3.1 Method

The same two legged locomotion test problems used in the previous chapter (5.3.5) are used here to compare the continuous and impulsive collocation formulations. The trajectory was assigned a maximum duration of 2.5 seconds and discretized into $N = 25$ finite elements for the fourth-order methods, and $N = 100$ elements for the FO method. We ran each experiment with two coefficients of friction: $\mu = 0.6$, and $\mu = 1.6$ – a high value that is more likely to produce Painlevé paradoxes. As before, we gave the same random seed to all models in each test, and repeated this 100 times for each friction coefficient.

6.3.2 Results

The LG4I formulation produced solutions slightly slower than the LG4C formulation did in both biped tests, and substantially slower in both monopod tests, as shown in Figure 6.6. This suggests that the complementarity constraints are the primary contributor to the computation time, as the increase in their number between the LG4C and LG4I formulations is much larger for the 3D problem.

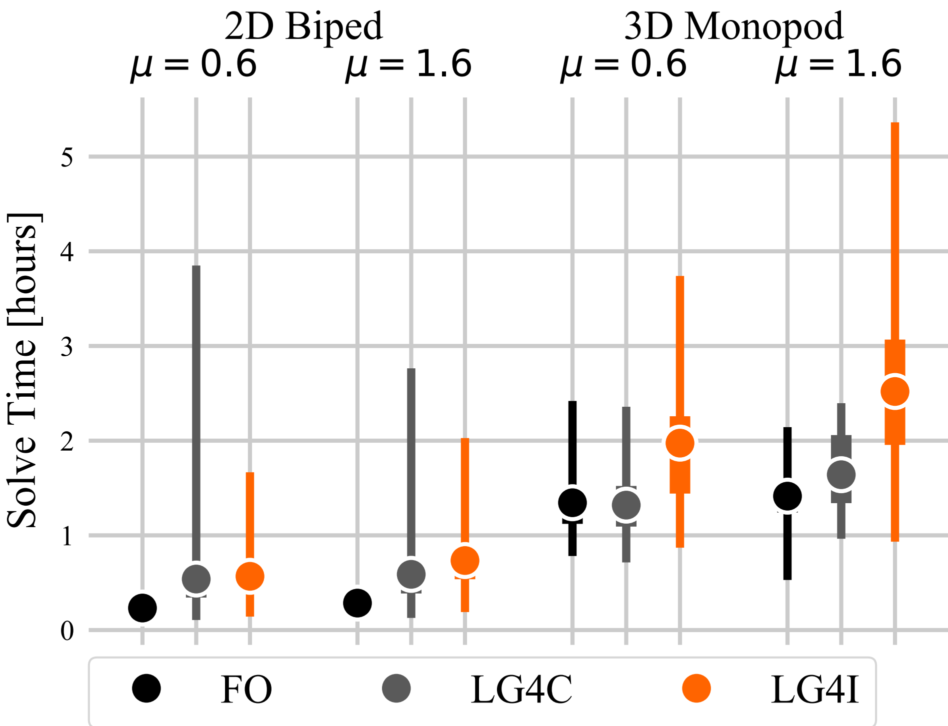

Figure 6.6: Combined solve times for the feasible and optimal stages on the 2D biped and 3D monopod using: First Order (FO) [12], Legendre-Gauss 4^{th} order continuous contact (LG4C), and impulsive contacts (LG4I). The circle indicates the mean, the narrow line is the range, and the wider line is the interquartile range.

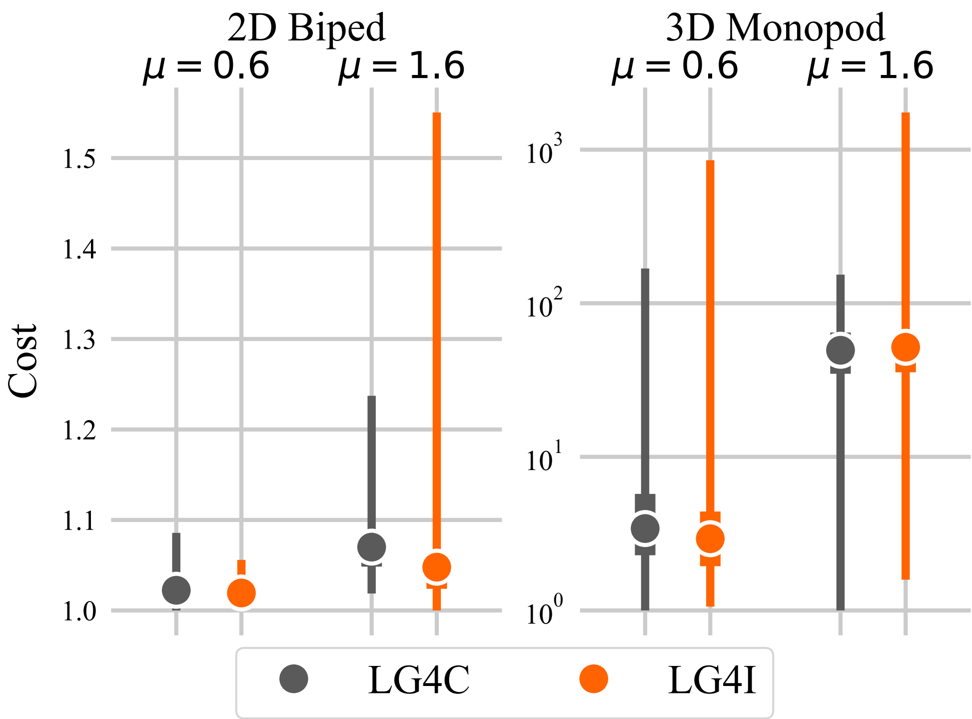

Figure 6.7: Cost values for optimal solutions, normalized to the minimum achieved by either model in each test. The circle indicates the mean, the narrow line is the range, and the wider line is the interquartile range.

Figure 6.7 compares the costs obtained, normalized to the lowest value achieved by either model in each case. (We exclude the FO solutions from the cost comparison, as they cannot be meaningfully compared to the fourth-order results due to the differences in accuracy.) The median costs were near-identical in all tests, but the spread of the LG4I results tended to be wider. While its median results were usually slightly better, it also generated the worst solution in most tests. This suggests that it might be able to identify superior strategies that are infeasible for the continuous version, but also that the slightly more cumbersome formulation could be more prone to getting trapped in bad local minima.

Overall, there does not appear to be a clear advantage to using the impulsive configuration for problems with exclusively inelastic collisions. While the accuracy was slightly improved over the continuous formulations, the computation time and consistency were worse.

6.3.3 Conclusions

The piecewise-continuous collocation formulation introduced in this chapter combines the advantages of hybrid-dynamic and contact-implicit approaches. By accommodating finite discontinuities in the velocity state, it can capture behaviour that continuous-velocity formulations cannot, such as partially elastic collisions and tangential "impacts without collision" – the established resolution of the frictional paradoxes identified by Painlevé. The more computationally cumbersome formulation leads to longer solve times for 3D or especially contact-heavy problems, however, so the approach is currently best suited to problems that require elasticity or impacts without collisions. Consequently, it has limited applicability to the study of legged locomotion.

Chapter 7

Coordinate System

Computational efficiency is the primary obstacle that must be overcome to make the contact-implicit trajectory optimization of whole-body models a practical technique. An aspect of the problem formulation that has received little attention regarding performance improvement is the coordinate system selected to describe the system model.

The different coordinate representations for articulated rigid-body systems can be organized along a spectrum from "more relative" to "more absolute". On the relative extreme, there is the minimal coordinate approach typically used in robotics [106] where each body is referenced to the frame of its predecessor. On paper, this produces the smallest problem, as it has the fewest coordinate variables and implicitly defines the constrained motion of the bodies, but the recursively-described link positions result in long, cumbersome equations of motion. On the absolute extreme, the position and orientation of all bodies can be referenced to the inertial (or *world*) frame, necessitating a much larger number of comparatively simple equations.

In this chapter, we compare these two approaches, and an option between them that combines relative translational and absolute orientation coordinates. We begin by motivating the importance of the coordinate system to the design of the trajectory optimization problem, and describing the coordinate schemes we will be comparing in the context of trajectory optimization. We then compare them through a series of experiments focused on two key factors: long serial chains of links, and contact.

The preliminary experiments, comparing relative and absolute orientation representations, were published in a co-authored paper [23]. Only my contributions to that paper are reproduced in this chapter. We subsequently extended that study by updating some experiments using a superior collocation scheme, and taking the direction of its findings to their logical conclusion by including maximal coordinates in the comparison. A follow-up paper on this work is under review.

7.1 Motivation

In robotics, a tree-like system of interconnected rigid bodies is typically described in the *joint space*: one body, designated the *floating base*, is referenced to the world frame, while all others are referenced to the preceding (parent) body [106]. This has the advantage of producing a minimal set of coordinate variables and ensuring that the bodies in the system remain connected without the need for explicit positional constraints, but because the position of a given link is defined as a recursive combination of all preceding joint positions and the position of the base, the length and complexity of the expressions describing the position, velocity and acceleration of the outermost bodies rapidly increase as more bodies are connected in series.

Because contact points are typically found at the extremities of the system (for example, at the feet in a legged robot or at the gripper in a manipulator arm), this directly affects the tractability of the contact constraints in contact-implicit problems. It also affects the equations of motion (EOM), with the Coriolis terms being especially prone to becoming unwieldly for long chains described in this way. While it has been observed that removing these problematic terms can allow the model to solve faster [109], neglecting them becomes detrimental to the accuracy of the simulation as the movement becomes more rapid and dynamic [23]. Besides being longer, these expressions are also more computationally cumbersome as they are less sparse: a constraint applied to a specific body involves not just the coordinates directly related to that body, but the coordinates of all preceding bodies in the same chain.

By contrast, coordinate representations that reference positions to the inertial frame, rather than relative to preceding links, result in simple, sparse expressions describing the system dynamics and contact model, at the cost of many more coordinate variables and explicit connection constraints.

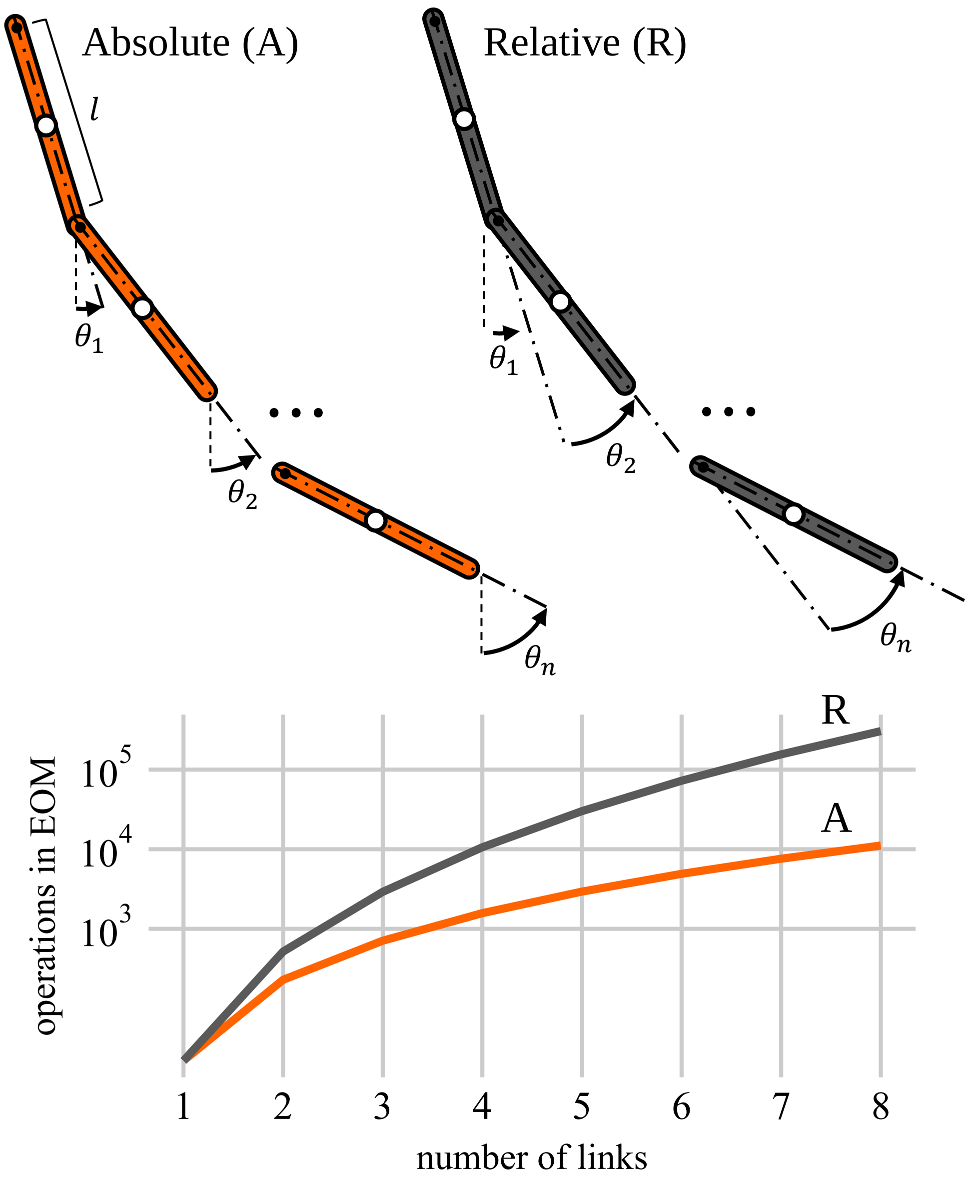

Figure 7.1: Diagram of a 2D n-link pendulum, contrasting the absolute (A) and relative (R) angle formulations. The plot compares the number of operations in the symbolic equations of motion (EOM) as the number of links increases.

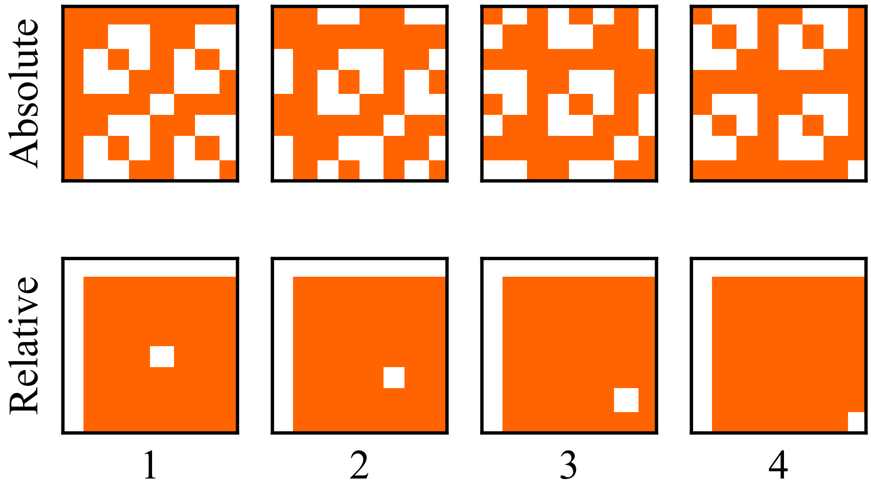

Figure 7.2: Sparsity of the Hessians of the Coriolis terms for each link of a planar 4-link pendulum arising from relative and absolute orientation formulations.

7.1.1 Example: Absolute vs. Relative Angles for a Planar Pendulum

To demonstrate the significance of these differences, consider the two representations of a planar pendulum model shown in Figure 7.1. The plot compares the number of operations in the EOM as the length of the pendulum increases. Referencing the angles of the links to the world frame rather than in the joint space is a seemingly minor change that does not alter the number of coordinate variables, but results in far fewer operations being added to the EOM with each subsequent link.

This is a qualitative change as well as a quantitative one: Figure 7.2 shows the sparsity pattern for the Hessian of the Coriolis term ($\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}}$ in (4.2)) of each link of a 4-link pendulum, illustrating that these equations also become far less dense when formulated using absolute angles. If this model represented the leg of a robot rather than an isolated pendulum, these improvements would also translate to more compact, sparse equations defining the contact state.

7.2 Coordinate Systems

We will be comparing three coordinate systems:

1. **Minimal:** the coordinates correspond to the DOFs of the system. The system is typically described in the *joint space*, with only the *floating base* referenced to the world frame, and all other bodies' positions described in terms of preceding bodies and joint positions [106].

2. **Relative Translation, Absolute Orientation (RTAO):** the orientation of all bodies is referenced to the world frame, while the translational positions are described in terms of preceding bodies. The number of coordinates matches the number of DOFs in a planar system, but may exceed it for spatial systems, so some additional motion constraints might be required.

3. **Maximal:** the position of all bodies is referenced to the world frame, leading to the maximum possible number of coordinates. Connection constraints, supported by constraint forces and torques, must be added to ensure bodies move together as required [173].

These systems are illustrated in Figure 7.3, which shows how each of them describes a 2-link pendulum. Besides these different ways of referencing the coordinates, there are also various methods of representing orientation [106], but we will only consider Euler angles.

7.2.1 Formulating Joints

The differences between the coordinate systems with respect to the formulation of the trajectory optimization problem amount to different methods of describing joints. In this section, we will briefly outline how joints are formulated in each coordinate system, using some common joint types as illustrative examples.

A *joint* is a relationship between two rigid bodies that determines the relative motion that can occur between them. We refer to the body that is closer to the base (with respect to the number of preceding joints in the shortest chain connecting it to the base) as the *parent* body, while the other is the *child*. The connection between two bodies can be summarized by the *joint equation:*

$$\mathbf{p_{i+1}} = \mathbf{p_i} + \mathbf{{}^{i+1}J_i}\,\mathbf{{}^{i+1}\Phi_i} \tag{7.1}$$

Here, $\mathbf{p_i}$ is a position vector giving the location of the parent's centre of mass (COM) and orientation with respect to the world frame. Likewise,

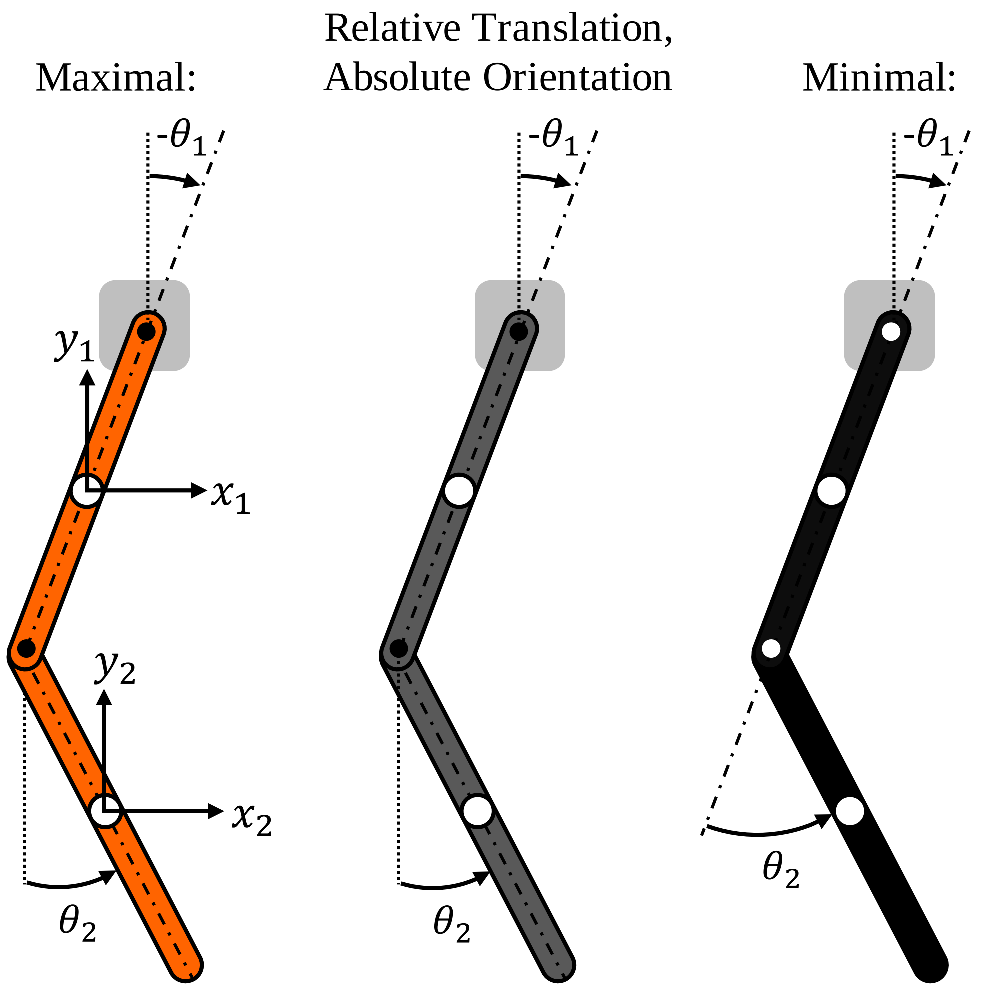

Figure 7.3: Two-link planar pendulums as described using maximal coordinates, relative translation absolute orientation (RTAO) coordinates, and minimal coordinates.

$\mathbf{p_{i+1}}$ gives the world-frame position of the child. The relative position vector ${}^{i+1}\Phi_i$ describes the position of the child with respect to a frame attached to the parent. It contains the m degrees of freedom (DOFs) of the joint, which are mapped to the world frame via the Jacobian, $\mathbf{{}^{i+1}J_i}$. An m-DOF joint removes $(6-m)$ DOFs of relative motion, referred to as the *constrained* DOFs of the joint.

A joint is modelled as follows in each coordinate system:

- **Minimal Coordinates:** The joint is described in terms of ${}^{i+1}\Phi_i$. The constrained DOFs do not need to be included in the coordinates, as they are implicitly removed when the EOM are derived.
- **RTAO Coordinates:** The joint is described in terms of the tangential components of the relative position. ${}^{i+1}\Phi_i^T$, and the orientation components of the absolute position vectors, $\mathbf{p_i^O}$ and $\mathbf{p_{i+1}^O}$. Constrained tangential DOFs are implicitly removed, as in the minimal formulation, but constrained orientation DOFs must be removed explicitly through restrictions defined in terms of $\mathbf{p_i^O}$ and $\mathbf{p_{i+1}^O}$.
- **Maximal Coordinates:** The joint is described in terms of $\mathbf{p_i}$ and $\mathbf{p_{i+1}}$. Constrained DOFs are removed through explicit restrictions defined for these vectors.

A complete model of a joint typically requires bounds on the range of motion, and on the maximum force and power output of the actuators. These are defined in the joint space, so the RTAO and maximal formulations still require all relative DOFs, and their velocities, to be included in the problem as auxiliary variables.

Connection Constraints

The explicit removal of DOFs in the non-minimal coordinate systems is accomplished using *connection constraints*. These are defined as follows at the position level [174]:

- **Translational restrictions** remove translational DOFs, and have the form,

$$(\rho_{\mathbf{i+1}} - \rho_{\mathbf{i}}) \cdot \hat{\mathbf{d}} = 0 \tag{7.2}$$

  where $\rho_{\mathbf{i}}$ and $\rho_{\mathbf{i+1}}$ are reference points on each of the bodies that the joint is considered to act between, and $\hat{\mathbf{d}}$ is a unit vector giving the forbidden direction of motion.

- **Orientation restrictions** remove angular DOFs, and have the form,

$$\hat{\mathbf{d}}_{\mathbf{i}} \cdot \hat{\mathbf{d}}_{\mathbf{i+1}} = 0 \tag{7.3}$$

  where $\hat{\mathbf{d}}_{\mathbf{i}}$ and $\hat{\mathbf{d}}_{\mathbf{i+1}}$ are unit vectors in the frames of the corresponding bodies that must remain perpendicular.

The derivatives of these equations give the connection constraints at the velocity and acceleration levels.

Ideally, all levels of constraint should be satisfied at all points in the finite element, but this is not possible in practice, as the error inherent to numerical integration means that the change in the value of a constraint expression $c(\mathbf{p_i}, \mathbf{p_{i+1}})$ as calculated at the start and end of some time interval $t = [t_0, t_0 + h]$ – that is,

$$\frac{c\left(\mathbf{p_i}(t_0 + h), \mathbf{p_{i+1}}(t_0 + h)\right) - c(\mathbf{p_i}(t_0), \mathbf{p_{i+1}}(t_0))}{h}, \tag{7.4}$$

will not necessarily correspond to the analytically-calculated derivative,

$$\dot{c}\left(\mathbf{p_i}(t_0 + h), \mathbf{p_{i+1}}(t_0 + h), \dot{\mathbf{p}}_{\mathbf{i}}(t_0 + h), \dot{\mathbf{p}}_{\mathbf{i+1}}(t_0 + h)\right). \tag{7.5}$$

For this reason, we apply only the acceleration constraints at the collocation points, as this is necessary for an accurate model of the full system's dynamics. The position and velocity constraints are applied at the initial mesh point, limiting *constraint drift* to what can occur over a single timestep.

Example 1: Rotational Joints

Rotational joints prevent translation between rigid bodies, but allow one or more DOFs of relative rotation.

In the maximal coordinate formulation, constraints must be applied to all translational DOFs so that the points $\rho_{\mathbf{i}}$ and $\rho_{\mathbf{i+1}}$ coincide. This can be thought of as applying the restriction (7.2) using $\hat{\mathbf{x}}_{\mathbf{0}}$, $\hat{\mathbf{y}}_{\mathbf{0}}$, and $\hat{\mathbf{z}}_{\mathbf{0}}$ – the unit vectors representing the principal axes of the world frame. These constraints can only be satisfied if the translational DOFs of the child are allowed to be piecewise-continuous – that is, if the continuity constraints (5.15c) are not applied to them. This forces the bodies be reconnected at the start of each finite element.

Unless the joint is *spherical* (allowing rotation in 3 DOFs), non-minimal formulations will also require orientation restrictions to confine relative rotation to the desired DOFs. For example, a 1-DOF *rotary joint* that restricts the

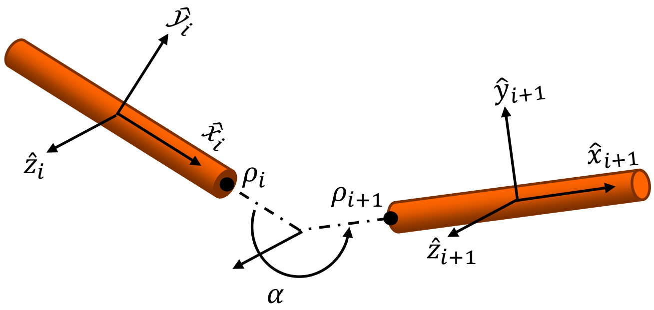

Figure 7.4: Diagram of rotary joint.

parent and child such that $\mathbf{\hat{z}_i} \parallel \mathbf{\hat{z}_{i+1}}$, (see: Figure 7.4) can be modeled using the orientation restrictions,

$$\mathbf{\hat{z}_i} \cdot \mathbf{\hat{x}_{i+1}} = 0, \quad \mathbf{\hat{z}_i} \cdot \mathbf{\hat{y}_{i+1}} = 0. \tag{7.6}$$

The constraint defining the relative angle α of the joint as a function of the absolute coordinates has a similar form to the orientation restrictions:

$$\cos(\alpha) = \mathbf{\hat{x}_i} \cdot \mathbf{\hat{x}_{i+1}}. \tag{7.7}$$

Example 2: Prismatic Joints

A prismatic joint prevents relative rotation between bodies, but allows translation along one axis of the parent frame. Using minimal and RTAO coordinates, a prismatic joint is modelled with a single axial coordinate, r.

The maximal formulation requires the rotational DOFs to be constrained explicitly. Because no relative rotation is possible, the orientation restrictions defined in (7.3) can typically be simplified to constraints equating the orientation DOFs. As with the redundant translational DOFs in the case of the rotational joint, the continuity constraints should then be deactivated for the orientation coordinates of the child.

To obtain the prismatic joint shown in Figure 7.5, which allows translation along the shared z axis of the bodies, translational restrictions are applied to the x and y axes as follows:

$$(\rho_{\mathbf{i+1}} - \rho_{\mathbf{i}}) \cdot \mathbf{\hat{y}_i} = 0 \tag{7.8a}$$
$$(\rho_{\mathbf{i+1}} - \rho_{\mathbf{i}}) \cdot \mathbf{\hat{z}_i} = 0 \tag{7.8b}$$

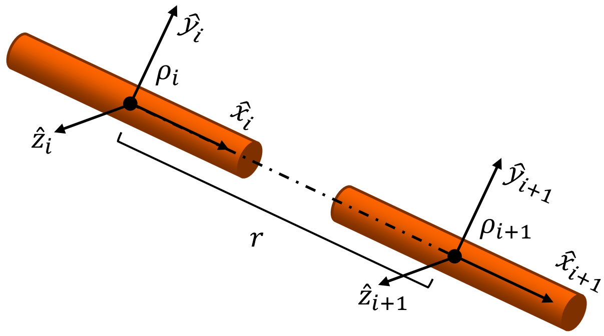

Figure 7.5: Diagram of prismatic joint.

A modified version of this restriction, applied to the active axis, can be used to calculate r in terms of the absolute coordinates:

$$(\rho_{\mathbf{i+1}} - \rho_{\mathbf{i}}) \cdot \hat{\mathbf{x}}_{\mathbf{i}} = r \tag{7.9}$$

7.3 Experiments

The performance of the three coordinate formulations is compared over three sets of trials, selected to highlight especially challenging aspects of modelling articulated systems:

1. **Pendulum Tests:** As illustrated in the Motivation section, the ability to model long serial chains efficiently is ultimately what separates the performance of the various coordinate systems. The pendulum is a useful model that tests this ability without the added complication of nonsmooth dynamics.

2. **Monopod Test – Minimal vs. RTAO:** This preliminary test was conducted as part of our first study into the performance of different coordinate systems [23]. The aim is to compare the minimal and RTAO formulations for a relatively simple planar problem involving contacts, to justify the elimination of the minimal coordinate system from subsequent tests on more complicated problems.

3. **Legged Locomotion Tests – RTAO vs. Maximal:** These updated tests compare the RTAO and maximal formulations for planar and spatial contact-implicit problems.

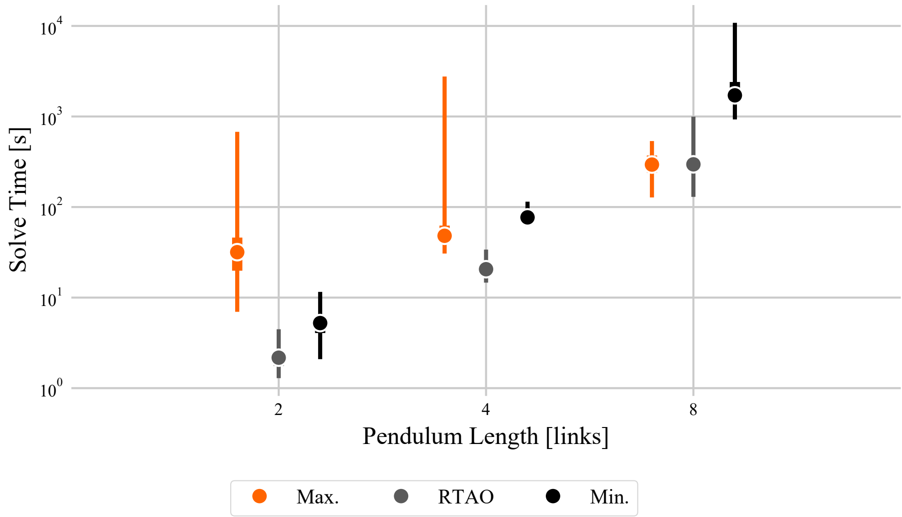

Figure 7.6: Solve times for swing-up problem on planar pendulums of different lengths modelled using minimal (Min.), maximal (Max.) and RTAO coordinates. The marker indicates the median value, the wider line is the interquartile range, and the narrow line is the range.

7.3.1 Pendulum Tests

Planar Pendulum Swing-Up

The pendulum was modelled as a chain of uniform rigid rods of unit mass, and a length of 0.5 meters. These links were connected by ideal rotary joints, with the topmost joint unactuated. The task required the pendulum to swing from rest in a hanging position, to rest in an inverted position. We assigned a total time of 2 seconds to the trajectory, discretized into $N = 50$ finite elements of fixed duration using a fourth-order Legendre-Gauss collocation scheme (LG4). The objective was to minimize the actuator effort (the sum of the squared actuator torques at all collocation points). The test was repeated for pendulums consisting of 2, 4 and 8 links. Each problem was initialized from 50 random seeds, which assigned small values to the position variables. This test is the only one to include all three coordinate schemes. The solve times achieved by the different coordinate formulations are plotted in Figure 7.6. In a planar system, the RTAO formulation does not require the burdensome restrictions on angular motion that might be called for in spatial problems, so it is not as encumbered by motion constraints as the maximal version, or by lengthy EOM as the minimal one. Unsurprisingly, it solves in a shorter time than the minimal coordinate version in all trials, but the balance shifts in favor of the maximal formulation as more links are added to the pendulum. Once the chain reaches eight links, the median solving time for the maximal formulation is marginally shorter (294 seconds vs. 296 seconds).

A similar test was performed in the preliminary study [23] comparing minimal and RTAO coordinates. This applied a two-stage solving approach, using a first-order method followed by third-order Legendre-Gauss-Radau (LGR3) collocation. The results of the updated version of this test are consistent with these initial results, confirming that the advantage of the RTAO system holds for different collocation schemes.

Spatial Pendulum Swing-Up

The model, task, timing parameters and timing parameters for this test are the same as for the planar pendulum version, but the links of the pendulum are now modelled as uniform cylinders with a radius of 2.5 centimeters, connected by ideal spherical joints. In 3D, the coordinate representations of the hanging and inverted positions are not unique, so it should be noted that the initial hanging condition was specified by setting the orientations of all links to zero with respect to the world frame, while the final inverted condition was specified by rotating all links $\frac{\pi}{2}$ radians about the world-frame x axis. Only a two-link pendulum model was trialed for this test. The minimal coordinate formulation was also excluded, as the factors that disadvantaged it in the planar test are exacerbated in the spatial case.

We found the convergence rate for this trial to be extremely low when initialized using the small random scatter approach applied in the planar test, so we instead seeded it using 50 *guided scatter* seeds [21] generated by applying a small random perturbation to a successful position trajectory. This trajectory was obtained by solving the problem with the default null guess.

The solve times for the 3D pendulum are given in Figure 7.7. In the spatial case, the additional rotations added by each subsequent link make the EOM for the RTAO model much more onerous, so the maximal formulation is able to solve faster even when there are only two links in the chain.

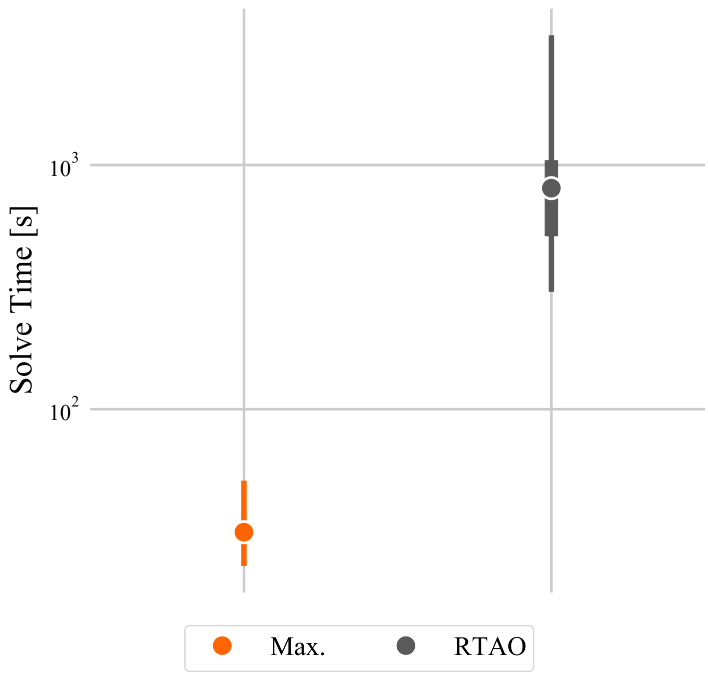

Figure 7.7: Solve times for swing-up problem on spatial pendulums of different lengths modeled using maximal (Max.) and RTAO coordinates. The marker indicates the median value, the wider line is the interquartile range, and the narrow line is the range.

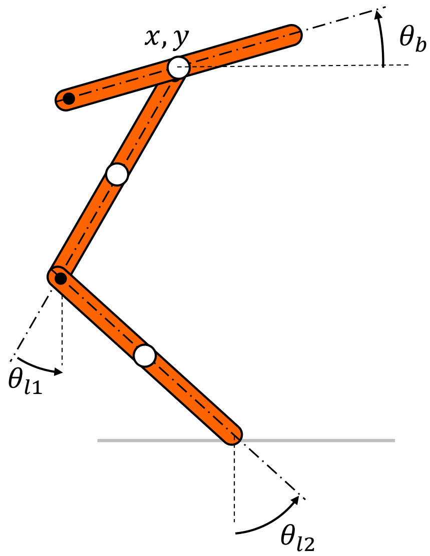

Figure 7.8: RTAO version of planar monopod model used in preliminary experiment comparing minimal and RTAO coordinate systems.

7.3.2 Monopod Test – Minimal vs. RTAO Coordinates

This test uses a planar monopod model with a backward-facing rotary knee (Figure 7.8). It was required to perform a 5 meter *missing the boat* [30] sprint from rest, minimizing the sum of the squared actuator torques over all collocation points. The initial and final poses were not specified beyond the requirement that it start at $x = 0m$ and finish with $x = 5m$. A total time of $T = 2$ seconds, discretized into $N = 100$ elements, was allocated to perform the maneuver. The problem was solved using the same two-stage approach as in the preliminary version of the planar pendulum test [23]. The first-order stage was initialized using 50 random seeds, assigning small values to the position variables.

The solving times for each angle configuration are shown in Figure 7.9. As in both versions of the planar pendulum test, the RTAO coordinate system performs better than the minimal one, with the more interesting point being *how much* better it does: compared to the relatively minor difference in the case of pendulum models having similar serial length, the improvement is greater. This confirms that the contact constraints specifically are made more tractable by the change.

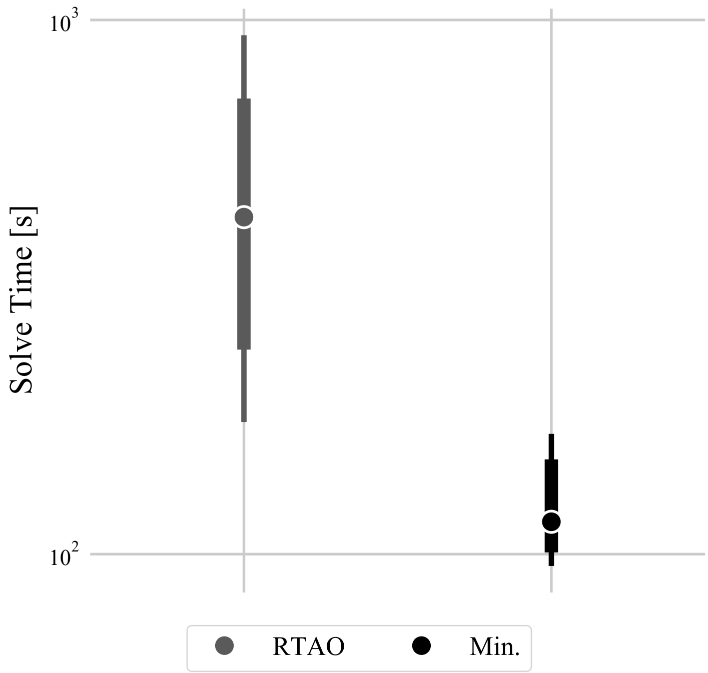

Figure 7.9: Solve times for planar monopod sprinting problem modelled using minimal (Min.) and RTAO coordinates. The marker indicates the median value, the wider line is the interquartile range, and the narrow line is the range.

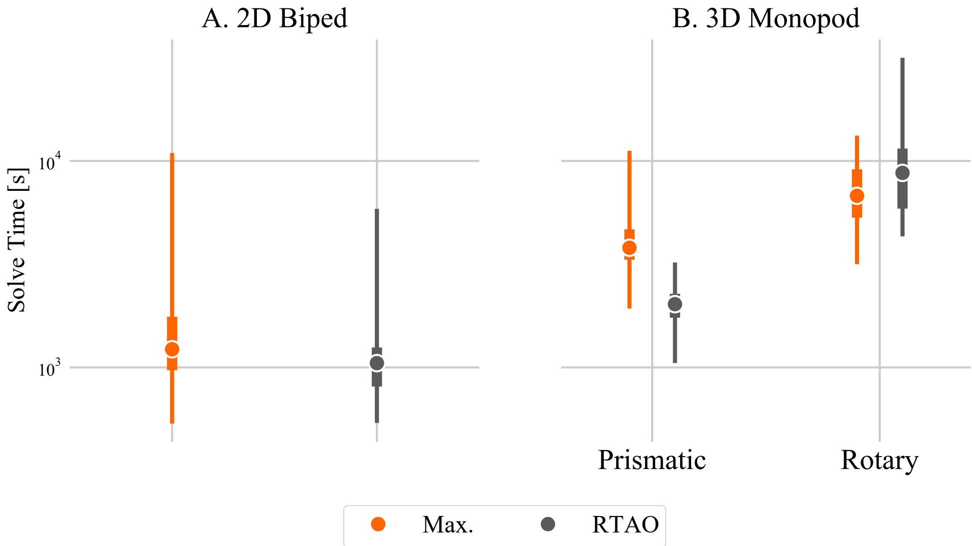

Figure 7.10: Solve times for planar biped stopping (A) and spatial monopod turning (B) problems using maximal (Max.) and RTAO coordinates. The marker indicates the median value, the wider line is the interquartile range, and the narrow line is the range.

Combined with the results of the remaining tests from this study, which confirm the advantage of the RTAO coordinate system over the minimal one for spatial legged locomotion problems of increasing complexity [23], this outcome provides sufficient justification for eliminating the minimal coordinate system from subsequent comparisons.

7.3.3 Legged Locomotion Tests – RTAO vs. Maximal Coordinates

Planar Biped Stop

The model, task parameters and solving procedure for this test are the same as for the minimum-distance bipedal stopping trials in Chapters 5 and 6. The results are plotted in Figure 7.10.A. As in the other planar examples, the comparatively simple connections between rigid bodies in the 2D RTAO formulation allow it to outperform the maximal one, despite the latter producing shorter, sparser expressions for the contact variables.

Spatial Monopod Turn:

The two configurations of the monopod model used in this test are show in Figure 7.10.B. Both are configured to have unit total mass, with 10 percent of the mass made up by the leg. The body is modelled as a cube, while each leg segment is a uniform cylinder. A spherical hip joint connects the body and leg. We included both the prismatic and rotary knee joints, as each is likely to advantage a different coordinate formulation: the RTAO system represents the prismatic joint using a single axial coordinate, which avoids the need for difficult motion restrictions and coordinate conversions. The rotary joint involves cumbersome orientation restrictions in both cases, but the maximal system avoids the added burden of longer EOM caused by referencing the position of the lower leg segment to two preceding rigid bodies.

As in the spatial monopod examples in previous chapters, the motion task can be described as a *missing the boat* sprint [30] with a right-angle turn in the middle. The initial condition is set as grounded rest with zero rotation of the body, and the leg fully extended, while the final condition was not specified beyond requiring it the body to have travelled 2.5 meters in both the x and z directions, and rotated $\frac{\pi}{2}$ radians about the y axis. An additional task constraint required the body to have position $x = 2.5, z = 0$ meters at the point $[\frac{N}{2}, 0]$, forcing the model to travel in a straight line in the x direction before turning to fulfil the final condition. The trajectory was discretized into $N = 50$ finite elements, and given a maximum total duration of 2.5 seconds. No objective was assigned beyond minimizing the contact penalties, so the purpose was just to find a feasible trajectory executing the turning maneuver.

The results for each monopod configuration are plotted in Figure **??**.D. The better-performing coordinate system depends on the knee joint type: the RTAO formulation solved the problem with the prismatic knee more quickly, due to its simple, single-coordinate model of this joint, while the maximal formulation was faster for the rotary knee, as this challenging joint model was not compounded by lengthy, relatively-referenced expressions for positions on the lower leg segment.

7.3.4 Discussion

Unlike the preliminary comparison between the minimal and RTAO coordinate systems [23], the comparison between RTAO and maximal coordinates did not reveal a decisive victor. Which coordinate system produces the most efficient problem formulation depends on the lengths of the serial chains involved, and the joint composition of the system, but we can make the following suggestions based on these empirical results:

- RTAO coordinates are likely to be the best option for planar problems, as the burden of the additional motion constraints and associated variables outweighs the relatively small improvement in the simplicity of the EOM and contact model unless the system includes very long serial chains.
- Maximal coordinates have the advantage in spatial problems, particularly if all joints are rotational.

In the planar biped test, and the prismatic 3D monopod test, the maximal coordinate system produced more complicated models of the connections between segments, but simpler expressions for the contact variables. It solved more slowly in both cases, indicating that the tractability of the joint models has a greater effect on performance than the tractability of the contact model.

The results of this study suggest that neither RTAO nor the maximal coordinates are an ideal option for modeling spatial motion. The examples where each one underperformed were fairly predictable based on the complexity of the joint models, so a better approach might be combining aspects of the two coordinate systems such that each of their weaknesses are avoided.

This "bespoke" coordinate scheme could use the translational restrictions from the maximal formulation to connect subsystems of bodies involving connections that can be modeled more efficiently using minimal or RTAO coordinates – for example, the leg of the 3D monopod could be a subsystem where the knee is modeled using a single coordinate, which is joined to the body as it is in the maximal formulation. This approach would break up long serial chains, while reducing the need for difficult relative motion restrictions. We plan to test this idea in future work.

Another avenue for further development is the use of unit quaternions to represent spatial orientation, rather than Euler angles. Quaternions have the well-known advantage of avoiding the problem of singular configurations referred to as *gimbal lock* [106], but implementation challenges include maintaining the unit norm, and imposing limits such as range of motion bounds that are more intuitively posed using angles. Maximal coordinates simplify the application of quaternions, as the process of deriving the dynamic equations does not have to be adapted to the same extent as would be required for other coordinate systems [175]. Maximal coordinate formulations using quaternions have been demonstrated with variational integration [**?**, 176], and our future work will focus on incorporating them into the maximal coordinate adaptation of orthogonal collocation described in this chapter.

7.4 Conclusions

The studies described in this chapter investigated whether absolute coordinates tend to convey a performance advantage in trajectory optimization problems by reducing the complexity and density of the equations of motion and contact model. Although the combination of relative translational and absolute orientation (RTAO) coordinates still tended to solve fastest for planar motion problems, the maximal formulation did perform better in most spatial motion tests. The exceptions to these general observations suggest that the optimal coordinate scheme for trajectory optimization might be a hybrid of these two approaches, which divides the model into relatively-referenced subsystems so long serial chains and complicated motion restrictions can both be avoided.

Chapter 8

Problem Initialization

While simple template models can provide some useful insights regarding the fundamental dynamics of legged maneuverability, investigating more detailed, specific questions on how these actions should be performed also requires the use of whole-body models. The downside of these models is that they lead to computationally cumbersome constrained nonlinear problems (CNLPs), especially when the additional demand of unscheduled contact sequences is taken into account. These are more likely to produce poor, locally-minimal solutions, as they inhibit the solver's ability to explore the solution space effectively.

In Chapter 3, we attempted to reframe this drawback as an advantage, arguing that poor-quality solutions can also provide valuable information about successful locomotion strategies when contrasted with better solutions. This is only an effective approach under the assumption that there is sufficient diversity in the pool of solutions to reveal the features separating successful strategies from unsuccessful ones. There is also the underlying assumption that it is possible to find a feasible solution at all, regardless of quality.

Both these assumptions relate to the *seed* vector used to initialize the solving process. This aspect of formulating the trajectory optimization problem has yet to be rigorously interrogated, however. In this chapter, we motivate the importance of the seed by demonstrating its effect on the solvability of a challenging skateboarding problem. We then present a preliminary investigation into different seeding approaches, focusing on how this affects solver performance and the diversity of the resulting dataset. This study also introduces a novel smooth-random seed generation technique, which we previously published in the paper 'On the Effectiveness of Silly Walks as Initial Guesses for

Optimal Legged Locomotion Problems' [21].

8.1 Motivation: Solving the Ollie

An implicit assumption throughout this dissertation has been that, if an NLP describes a physically feasible problem, it will be possible for the solver to find a feasible solution to that problem. In this section, we will use the example of a skateboarding trick called the *Ollie* to show that this is not always the case without the correct seeding strategy.

Although the achievement of *sick air* is not currently regarded as an important priority in robotics research, the Ollie is interesting as an example of a challenging object manipulation problem requiring an intricate sequence of diverse contact interactions to complete successfully. We have used the Ollie to explore trajectory optimization with varied contacts before [22], but required a combination of scheduled and complementarity-based contact schemes to generate the motion. The velocity-discontinuous impact model presented in Chapter 6 allows a fully contact-implicit formulation of the problem [26].

The objective of the Ollie is to get all four wheels of the skateboard off the ground. The rider stamps on the tail of the board while jumping up, so it bounces off the ground and propels the board into the air. Once airborne, the feet manipulate the board to execute further aerial tricks, or just position it for a safe landing.

Our model of a skateboard and humanoid rider is shown in Figure 8.1. We also attempted the test using the simplified model on the right of this figure, which isolates the contact problem by replacing the rider with a pair of point masses actuated by external forces. Three different types of contact are present in the system:

1. Partially elastic ($e = 0.6$), frictionless contact between the tail of the skateboard and the ground.
2. Inelastic, high-friction ($\mu = 1.6$) contact between the feet and skateboard. The position of the contact point with respect to the board is variable, so the feet can connect anywhere along the deck.
3. Inelastic, frictionless contact between the wheels of the skateboard and the ground. (The wheels are modeled as simple contact points offset below the deck, as they are only required to support the board in this example.)

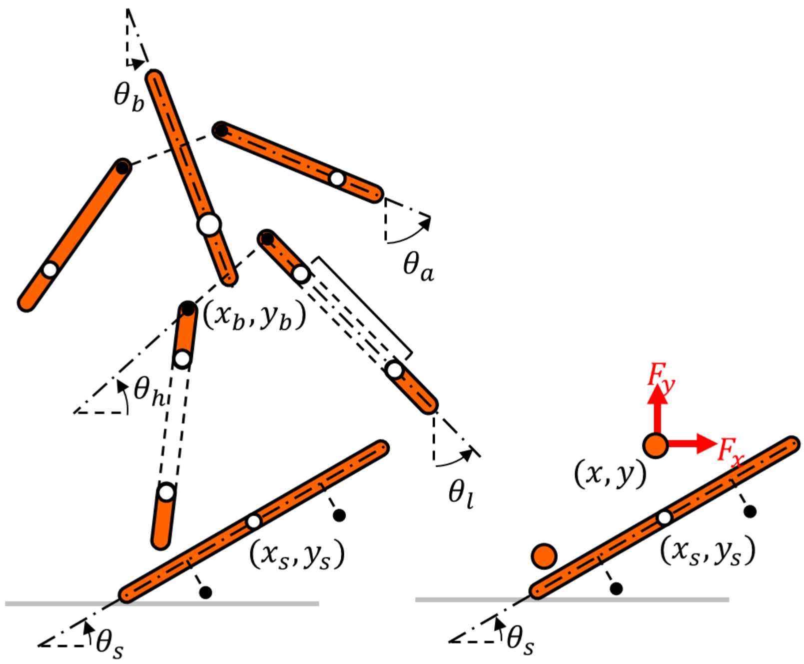

Figure 8.1: (Left) Planar model of skateboard and bipedal rider. (Right) Simplified system using two point masses instead of a full-body rider model.

We formulated the problem using Legendre-Gauss collocation with $P = 2$ collocation points and $N = 40$ finite elements, with maximum duration $h_m = 0.02$. Although we have previously used $P = 4$ in most examples, we selected shorter, lower-order elements for this problem to allow more opportunities for contact state changes than an equivalently-sized fourth-order formulation would. The initial and final conditions have the humanoid standing upright on the board with both wheels grounded, and the system initially at rest. An additional *air* condition requires both wheels to be more than 0.2 m above the ground at point $[n, p] = [20, 0]$.

We tried four seeding approaches to initialize the problem:

1. **null seed:** the default initial vector.
2. **perturbed null seed:** the position and contact variables are assigned small, random values.
3. **perturbed null seed with hint:** same as previous, but the height of the board tail is fixed to zero at the point $[10, 0]$.
4. **perturbed solution:** a previous successful result, perturbed by small random values.

Both models reliably generated the trick when given a *hint* specifying the initial tail contact, or a perturbed solution as a seed (strategies 3 and 4). The average solving time for the successful attempts on the full-body model was around 43 minutes. Unfortunately, neither model produced a feasible solution from the null seed, or from 20 randomized null seeds. This highlights a key challenge of contact-implicit trajectory optimization: in many cases, the desired result lies within a small basin of attraction that is exceedingly difficult to discover without some pre-existing knowledge of the contact sequence.

Although seeding with perturbed solutions is not a practical method, as it requires the problem to be solved at least once before, the success of this approach shows that the discovery of complex contact sequences is possible, given a seed of sufficient quality. This demonstrates that initialization is a critical aspect of formulating a successful contact-implicit trajectory optimization problem, but further research is needed to determine precisely what "sufficient quality" means in this context, and how these seeds can be obtained for truly unknown contact sequences.

The two-stage trajectory optimization strategy proposed by Marcucci et al. [149] demonstrates that the solution to a simplified version of the problem provides an effective seed, indicating that reduced dynamic models, such as centroidal [104] or quasistatic [109] approximations, could be useful for this purpose. These scaled-down problems must still be initialized themselves, however, so they provide an intermediate step between random initialization and the complete problem rather than a replacement for random initialization. Additionally, the failure of the skateboard-only model in this example shows that decreasing the dynamic complexity without improving the initial seed is still unlikely to yield a result in particularly challenging contact-rich cases.

Even for problems that can be solved from the null seed, it is not a foregone conclusion that the result will be a good solution. It is necessary for our broader goal of generating a set of diverse solutions to start the solving process from a variety of points, so the solution space can be more fully explored. The remainder of this chapter will compare different random initialization techniques.

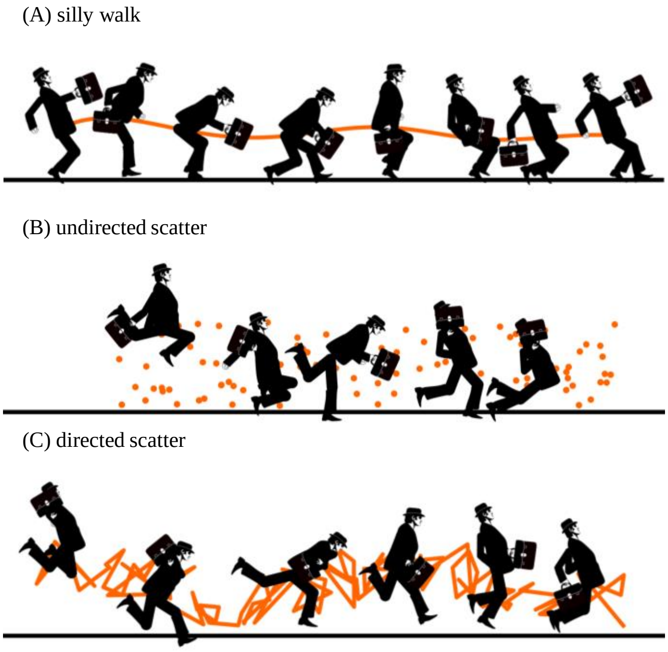

Figure 8.2: Different types of random initial seed compared in this paper.

8.2 Random Initialization Methods

The most straightforward method of generating a random seed for a problem is simply assigning random values to all variables (as we have done in most examples throughout this dissertation) but this contradicts the traditional wisdom regarding problem initialization. Ideally, the seed should resemble the intended solution as closely as possible, but this is not practical if the target motion is not well-specified. It is a reasonable assumption that any predetermined legged locomotion trajectory would be closer to a feasible solution than the null guess, but this approach risks biasing the results: if a gallop is given as the input to a problem, and the solution is also a gallop, it is less likely to be because the gallop is genuinely the optimal gait for that scenario so much as a special case of "GIGO" (gallop in – gallop out).

The bridge between these opposing goals would seem to be a method of generating random locomotion trajectories. The method we propose to accomplish this is **s**mooth-random **i**nterpolation **l**inking an **i**rregular sequence of foot **i**mpacts – SILII for short, leading to the descriptive name we have given to the resulting trajectories: *silly walks*. Our aim is to compare the effectiveness of silly walks to two forms of randomly-sampled seed, illustrated in Figure 8.2:

1. **Undirected scatter:** generated by randomly sampling the variables from the range $[-v, v]$, or $[0, v]$ if the variable is positive-valued, where v is a constant termed the *variation parameter*. (Figure 8.2B)

2. **Directed scatter:** these are generated in the same way as the undirected scatter, but some variables are randomly varied about simple guide trajectories so the model's motion loosely follows the expected path. For example, if a variable x represents the horizontal centre of mass (COM) coordinate of a robot travelling five metres, an example of a guide trajectory might be a straight line from zero to five. A random value with magnitude $\leq v$ is then added to the guide at each point to give the value assigned to x. (Figure 8.2C)

The efficacy of these methods is also likely to be affected by the selection of variables that are assigned non-null values, and the range of possible variation. To test this, we trialed the scatter techniques with five variation parameters evenly distributed between $\frac{\pi}{10}$ and $\frac{\pi}{2}$, and defined three levels of variable incorporation for all methods:

1. **Level 1:** only the position variables are included.

2. **Level 2:** position and ground reaction force variables are included.

3. **Level 3:** all variables are included.

For the first two levels, the uninitialized variables were reset to the default values (*None* for the modelling language used).

The experiments in this chapter use a two-stage solving method [149]: the simplified first stage reduces the collocation to a first-order (FO) method, while the final stage uses third-order Legendre-Gauss-Radau (LGR3) collocation. The same number of finite elements is used for both stages, with the FO problem formed by deactivating collocation points.

8.2.1 Generating Silly Walks

The SILII method can be divided into four steps:

1. **Trunk position:** a smooth trajectory for the base link (usually the body) is generated.

2. **Stance leg positions:** a Markov chain is used to generate a foot contact sequence, and leg positions are calculated accordingly.

3. **Swing leg positions:** smooth trajectories are generated linking the leg positions between liftoff and touchdown events.

4. **Other variables:** with the positions completely defined, all other variables are calculated to satisfy at least some of the constraints.

Trunk Position

Two types of smooth functions are used in the generation of the trunk motions. Random waves are created from k sinusoidal components, where the magnitude A, frequency ω and phase ϕ of each are sampled from a random distribution over a specified range:

$$w(n) = \sum_{i=1}^{k} A_i cos(w_i n - \phi_i) \tag{8.1}$$

A decaying variation, including an exponential with randomly-selected time constant τ, is also used:

$$d(n) = \sum_{i=1}^{k} A_i e^{-\tau n} cos(w_i n - \phi_i) \tag{8.2}$$

For the documented experiments, the number of components per function, and the ranges from which the random parameters were selected were coarsely tuned to values that tended to produce plausible values of the variables in question. Likewise, the choice of whether a wave or decaying wave was used was made by inspection, based on previously-generated trajectories.

Locomotion tasks are typically posed as boundary-value problems, with specific values imposed on the positions and/or velocities at the initial and final simulation points. We generate smooth trajectories that satisfy these constraints by combining manipulated versions of the random basis functions into a velocity profile that has the required endpoint values, and results in the required final position when integrated from the initial point:

1. a function $a_1(n)$ is generated, and then scaled and shifted such that its value at $n = 1$ is the desired initial velocity, and its value at $n = N$ is zero. Another function $a_N(n)$ with $a_N(1) = 0$ and $a_N(N)$ matching the final velocity is generated and added to $a_1(n)$ to give $a(n)$.
2. Assuming all timesteps to have duration h_m, $a(n)$ is integrated using the implicit euler method. A function $b(n)$ with initial and final values of zero is generated and scaled such that its integral, when added to the integral of $a(n)$, gives the difference between the initial and final positions.

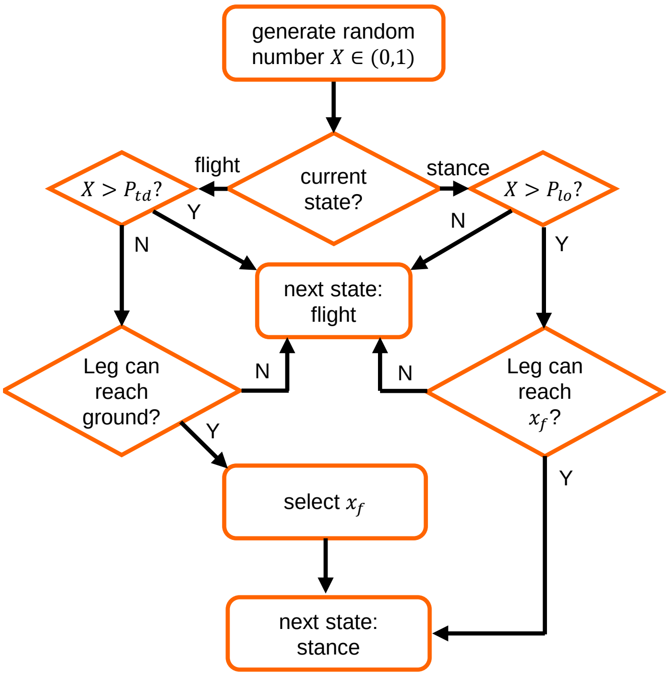

Figure 8.3: Process for randomly specifying the foot contact mode at each finite element.

3. The sum of $a(n)$ and $b(n)$ gives $\dot{x}_0(n)$, the guess trajectory for the velocity. This is then integrated to give the position guess $x_0(n)$.

   If the position and velocity states at the initial and final points are not fully defined, the missing values are randomly selected.

If the resulting $x_0(n)$ includes values that exceed the expected range of the variable, it is discarded and a new guess trajectory is generated.

Stance Leg Positions

The flow chart in Figure 8.2.1 summarizes how the contact state is specified at each finite element. A random contact sequence for each foot is generated based on a Markov chain with transition probabilities P_{td} for touchdown and P_{lo} for liftoff. Starting from the contact states at the initial position, a random number between zero and one is generated and the state at the next element is assigned accordingly. For the described experiments, we selected $P_{td} = P_{lo} = 0.1$.

The current position of the model can force or prevent a transition event: the foot contact is assumed to be static, and liftoff will occur if the leg can no longer reach the foot contact position. Similarly, touchdown cannot occur if the leg cannot reach the ground, so the model will remain in flight. If a touchdown is feasible, a contact position will be randomly chosen within the forward-reaching range of the leg.

Swing Leg Positions

Quadratic splines are used to link the foot positions from each liftoff event to the next touchdown. The basic spline is defined as a function of the horizontal position of the foot x_f:

$$s(x_f) = -(x_f - x_{lo})(x_f - x_{td}) \tag{8.3}$$

where x_{lo} and x_{td} are the positions at liftoff and touchdown. The spline is then scaled so its maximum height is a random value between zero and the half of the minimum value of the body height over that swing phase.

The leg positions are then calculated so the foot moves along the spline. If a position is not reachable, the leg is assigned its last feasible position.

Other Variables

With the position trajectory completely defined, the velocities and accelerations can be calculated to satisfy the FO integration constraints.

If a foot is on the ground at a point, the vertical ground reaction force acting on that foot is assumed to be equal to the body weight, and the instantaneous coefficient of friction is assumed equal to mu_s, with the direction of friction depending on whether the model is performing an acceleration or deceleration task.

The actuator torques were assigned random magnitudes between zero and their maximum output, with directions corresponding to the velocity of the joint at that point. If the model includes hard joint stops, the associated rebound torques are assumed to be zero.

Finally, all auxiliary variables are calculated based on the values that have already been defined.

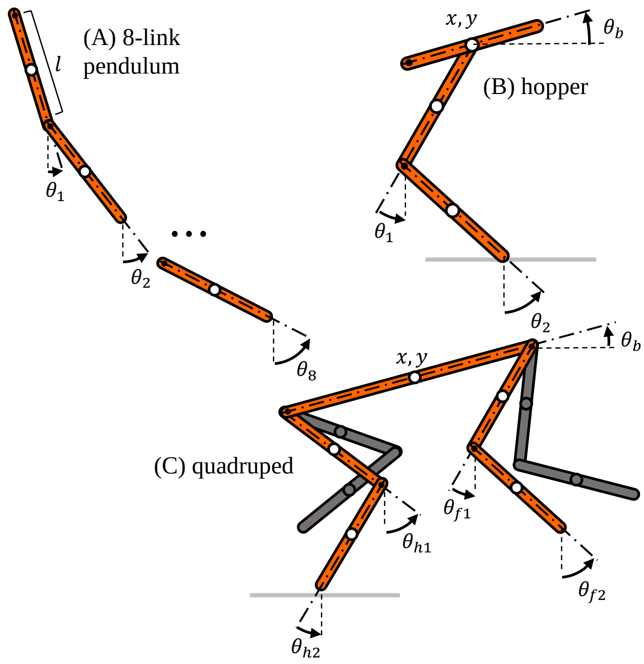

Figure 8.4: Models used to compare the initialization methods.

8.3 Comparison between Random Initialization Methods

Through tests on three different models, we evaluate their effects on solving time, robustness and the diversity of the solutions produced, and also examine how the performance of each method is affected if different variables are initialized, or if the range of the values assigned to the variables is changed.

8.3.1 Models

For this initial proof-of-concept experiment, we elected to test the methods using only planar models. We therefore replaced the spatial monopod used in previous chapters with a planar monopod. The prismatic knee was changed to a rotary joint so the same SILII algorithm could be used for both models with limited modification. We also replaced the biped model used before with a quadruped, to allow a wider variety of possible foot contact sequences. Besides these two, an eight-link pendulum was included to provide a comparison without the discrete dynamic elements, so the impact of the contact model on the performance of the initialization methods can be isolated. These models are shown in Figure 8.3.1. All contact-implicit problems used the continuous contact model described in Chapter 4.

The tasks each model was required to perform are as follows:

1. **Pendulum:** an underactuated swing-up minimizing actuator effort (sum of torques squared).
2. **Monopod:** a 10-metre sprint from rest minimizing actuator effort.
3. **Quadruped:** a 10-metre sprint from rest minimizing actuator effort, and gait termination from a fast gallop minimizing stopping distance.

8.3.2 Performance Metrics

The performance of the methods was compared across three categories:

1. **reliability**: the percentage of guesses that successfully converged to a solution out of the total attempted.
2. **solve time**: the solving time for the pre-solve stage, which is directly initialized by the random guess, and the total solving time were both compared.
3. **diversity**: to quantify the diversity, 20 solutions were selected at random for each method and the position trajectories for all coordinates were concatenated to give a single vector for each solution. The distance between every combination of vectors was then calculated as the 2-norm of the distance between them, and then the median of the distances was returned. In tests where an objective function was applied over and above minimizing the complementarity penalties, the resulting costs were compared as an additional diversity metric.

8.3.3 Pendulum Test

As before, the swing-up motion was defined by two fully-specified boundary conditions: initial rest hanging with all angles at zero, and final rest standing upright with all angles at π. We allocated $T = 2$ seconds for the movement, and the trajectory consisted of $N = 100$ finite elements. Actuator effort was minimized using the cost function

$$J = \sum_{n=1}^{N} \sum_{i=1}^{8} \tau_i^2[n] \tag{8.4}$$

where $\tau_i[n]$ is the torque acting at the top of the ith link at the nth element.

As an analogue for the silly walk, a smooth-random swing was created using the random wave function described in Section 8.2.1 to create a smooth trajectory from zero to π for each link. The same wave function created the guides for the directed scatter approach.

The results of the swing-up test are shown in Figure 8.5. Contrary to what was expected, the undirected scatter method produced faster, more reliable convergence than either of the more directed methods, however, the silly walk equivalent produced the most diverse solutions. A comparison of the cost values, shown in Figure 8.6, supports this: the range of costs achieved was wider for the smooth-random guess than it typically was for the scatter methods.

It must be noted that a low median cost should not be interpreted as a benefit of a method without further tests. The specific cost function used is likely to reward trajectories with little movement, so guesses where the position does not vary widely – especially between adjacent points –, would put the solver in a good region. This is sufficient explanation for why the undirected scatter - especially with a low variation parameter - tends to produce the lowest costs, and why the median cost of the smooth-random swing results is lower than that of the similarly-diverse directed scatter set. A different objective, such as minimum time, might produce very different relative magnitudes.

8.3.4 Monopod Test

To complete the sprint task, the monopod was required to start at rest in any pose with $x = 0$, and finish in any position or velocity state with $x = 10$. It was allowed $T = 3$ seconds to perform the motion and the trajectory was divided into $N = 100$ finite elements. Two cost functions were used: a feasibility objective where the only goal was minimizing the complementarity penalties, and a minimum-effort objective equivalent to 8.4 from the pendulum test, but with the addition of the penalty terms.

The directed scatter used a linear function from zero to ten as a guide for x and a constant as a guide for y.

The results for the feasibility objective are shown in Figure 8.7. As with the smooth-random guesses in the pendulum test, initializing all variables with a silly walk yields the most diverse solutions. When a higher variation

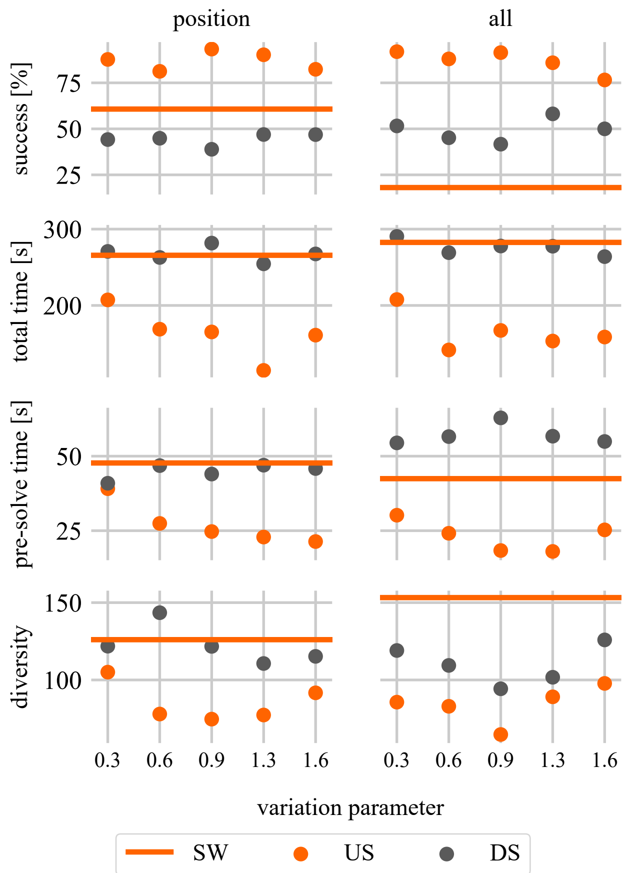

Figure 8.5: Success rate, average solve time and diversity for the pendulum swing-up for the silly walk equivalent (SW), undirected (US) and directed scatter (DS).

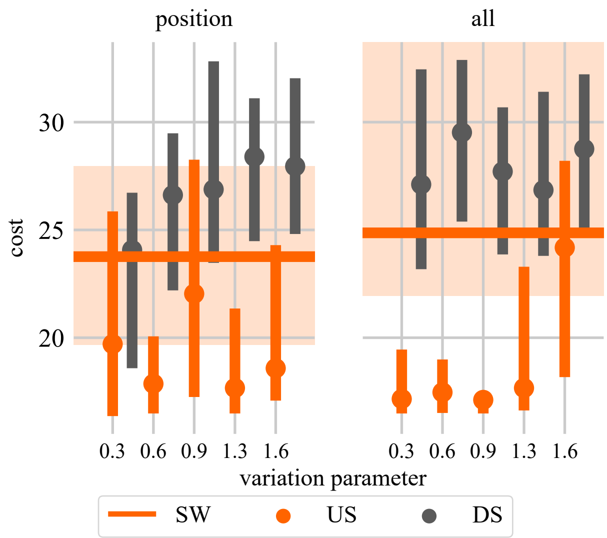

Figure 8.6: Cost values for the pendulum swing-up. The markers indicated in the legend indicate the median values, while the interquartile ranges are represented by bars for the undirected (US) and directed scatter (DS) and by the orange area for the silly walk equivalent (SW).

parameter is applied, the directed scatter achieves similar performance in both diversity and solving time, but its success rate is much lower.

The minimum-effort trial only compared the silly walk to the directed scatter with a variation parameter $v = 0.3\pi$. In both cases, all variables were initialized. The reliability of each remained consistent with the feasibility trial: 68 percent of silly walks converged, while only 20.4 percent of the scattered guesses were successful. The results for solve time and diversity are shown in Figure 8.8. While the total times are much longer than those achieved for the feasibility objective, the relative performance between the two methods is consistent with this test, as are both the distance norm and cost diversity metrics.

8.3.5 Quadruped Test

The sprint task for the quadruped was defined in the same way as for the monopod. For the rapid stopping task, its initial condition was sampled from a high-speed gallop, while the final condition required the horizontal velocity $\dot{x} \leq 0$, and all feet to be grounded. Both consisted of $N = 100$ finite elements, and the sprint was completed in $T = 5$ seconds, while the

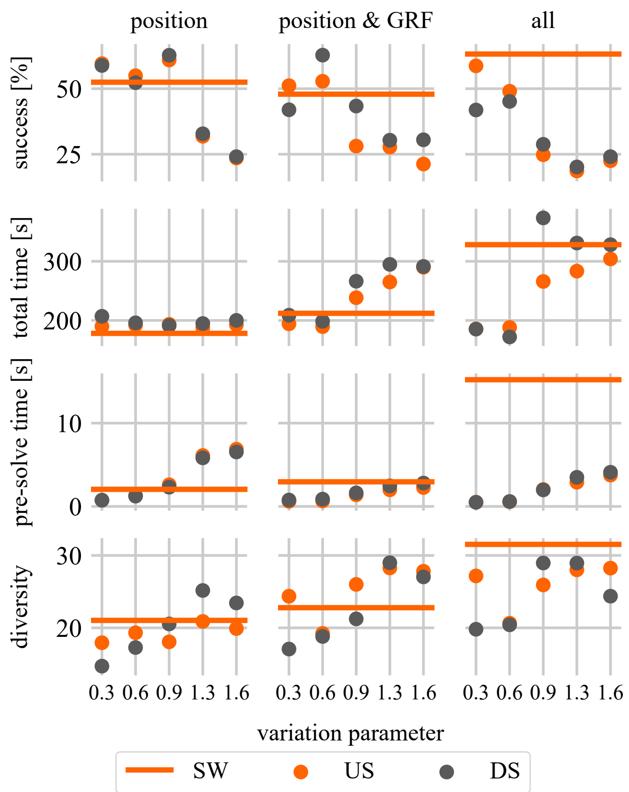

Figure 8.7: Success rate, solving time and diversity for the feasible sprint on the hopper for the silly walk (SW), undirected (US) and directed scatter (DS).

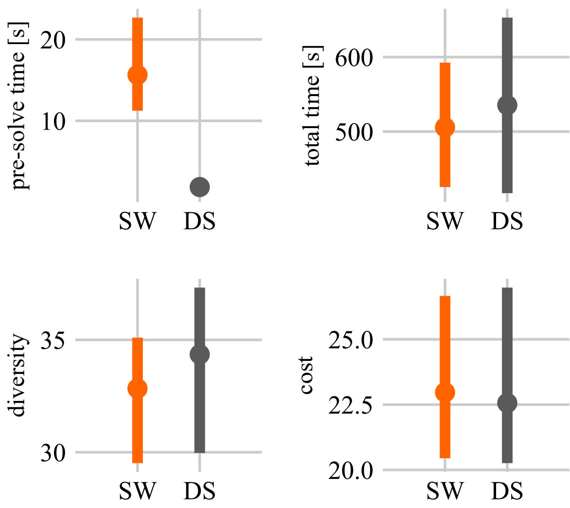

Figure 8.8: Solving time, diversity and cost comparison for minimum-effort sprint on the hopper model initialized with a silly walk (SW) vs. a directed scatter (DS). Bars indicate the interquartile range, while the point indicates the median value.

stop was allocated 8 seconds. Only a feasibility objective was tested for each task.

The guess trajectories were generated in the same way as for the monopod. The final x position for the directed scatter in the stop case was set to the *box benchmark* for distance – that is, the distance a rigid body with the same mass would take to stop from the specified initial velocity sliding on a surface with the same coefficient of friction. Only the fully-initialized versions of the silly walk and directed scatter with $v = 0.3\pi$ were compared.

For both trials, the silly walk solved much more reliably: its success rates were 80.6 and 25 percent for the sprint and stop respectively, compared to 59 and 9.4 percent for the guided scatter. The other metrics for both trials are shown in 8.9. While the diversity results were roughly consistent between the two tasks, the total solving times for the silly walks tended to be shorter than the scatter's for the sprint test and longer for the stop test. Despite this, the silly walk is still likely to produce a larger number of solutions over time due to its superior robustness.

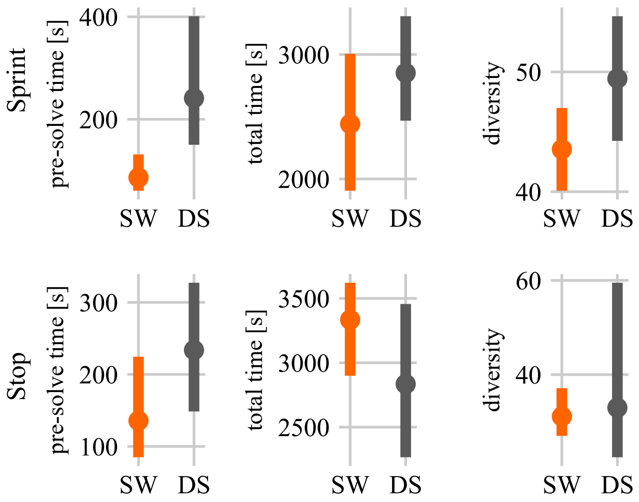

Figure 8.9: Solving time, diversity and cost comparison for sprint and rapid stop tests on the quadruped model initialized with a silly walk (SW) vs. a directed scatter (DS). Bars indicate the interquartile range, while the point indicates the median value.

8.4 Discussion

Throughout all tests, the silly walks consistently produced highly diverse solutions. A scatter can match this performance if the values are varied sufficiently widely, but the reliability and speed of convergence tend to decline as the variance is increased. Although the time performance of the silly walk tended to be highly task- and model-dependent, it was consistently far more robust for problems involving contacts, which would likely allow it to produce more solutions in a given time span over successive attempts.

Including more variables in the guess trajectory did not appear to have a strong effect on the convergence metrics, but fully-initializing the problem did appear to improve the diversity of the solutions.

This is only a proof-of-concept, and the SILII method used in these experiments is still rudimentary, but overall, the results presented demonstrate that even this unrefined way of generating semi-feasible smooth-random motions is superior to simply feeding random vectors into the solver, as it allows the solution space to be traversed widely without compromising convergence.

The effectiveness of the SILII method could potentially be improved with some simple modifications:

- Excluding infeasible leg positions that say, put the mid-leg joint below the ground or over-extend the upper joint's range of motion.
- Distributing the body weight over all grounded feet, rather than assigning the full body weight at each.
- Calculating the actuator profiles based on the equations of motion instead of randomizing them according to the direction of motion.
- Taking the possibility of sliding into account when generating the contact pattern.

8.5 Conclusions

This chapter compared three different random initialization methods for the trajectory optimization of legged locomotion problems. Silly walks - randomly-generated gaits that satisfy some of the constraints - were found to generate diverse solutions without compromising convergence to the extent that widely-varied scatter trajectories did.

The superior performance of the silly walk in the monopod and quadruped trials over the pendulum swing-up suggests that its success is tied to the near-feasible resolution of the contact constraints - a particularly challenging aspect of modeling legged motion. This type of guess could therefore also be a favorable option for optimal motion planning of manipulation tasks, as these share the problem of unscheduled, discrete impact events.

The SILII method for generating random gaits presented here is still primitive, but these tests indicate that it is worth applying and developing further.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

Legged systems – biological or mechanical – are intractably complex and redundant, with every action representing a choice made from an infinite variety of alternatives. Optimization is vital to understanding legged locomotion, as it gives us a means of explaining the order that emerges from this chaos of possibilities. Consequently, trajectory optimization is an especially powerful tool for investigating legged locomotion problems.

The first aim of this project was to develop a framework for using trajectory optimization to study the largely unexplored field of rapid, high-speed maneuverability. In Chapter 2, we collected examples of how it has been applied to legged locomotion research thus far, finding it to have demonstrated usefulness in the following roles:

1. **Motion Synthesis:** Trajectory optimization can generate feasible solutions to largely undefined tasks. This is essential to studying maneuverability, as motion patterns for transient maneuvers are not as well-established as constant-speed gaits.
2. **Morphological Design:** System parameters can be included in the optimization problem, allowing morphology and motion to be designed simultaneously. Further, trajectory optimization provides an objective method to compare the performance of different system configurations. This lets us study the contributions of particular morphological features

to maneuverability.

3. **Inverse Optimization:** Trajectory optimization can also be used "in reverse" to discover the objectives that a (typically biological) system is prioritizes during maneuvers.

We selected the direct collocation approach, where the trajectory optimization problem is converted to a constrained nonlinear program (CNLP) that can be solved using existing large-scale algorithms. While trajectory optimization of minimum-order *template* models confers the most general findings, it is necessary to *anchor* these templates in more detailed, specific models to discover the mechanisms that specific systems use to achieve rapid maneuverability [5]. Furthermore, it is necessary to use *contact-implicit* methods [20] so the most favorable foot contact sequences can be discovered. The use of higher-order, contact-implicit models exacerbates noted drawbacks of trajectory optimization, however: it makes the problems more computationally cumbersome, and therefore more likely to produce inferior, locally-minimal solutions.

For this reason, we chose to adapt the "Monte Carlo" approach proposed by Haberland and Kim [42, 4]. Rather than solving each problem for many randomized models, we solve it many times from randomized initial seeds for a single model. This produces a large dataset of solutions of varying quality, which can be analyzed to identify the features contributing to more successful maneuvers.

In Chapter 3, we demonstrated this approach for two test cases concerning the rapid termination of high-speed gaits:

1. Termination of human sprinting with and without arms – a morphological design problem demonstrating how free limbs are able to improve maneuverability by improving stability.

2. Termination of rotatory galloping in quadrupeds – a motion synthesis problem, where the described framework was enhanced by an iterative optimization technique that produced families of incrementally-improving stopping motions from each random seed.

Having established how to apply trajectory optimization to questions of rapid maneuverability, the remainder of the project focused on improving the tractability of the associated CNLPs to better facilitate the generation of these large collections of trajectories. We limited the scope to the problem formulation, rather than addressing it at the solver level. The primary technical contributions described in these chapters are as follows:

- **Chapter 5** adapted a complementarity-based time-stepping scheme for nonsmooth dynamic systems [129, 12] to work with orthogonal collocation of arbitrary order. Higher-order collocation gives a more favorable trade-off between problem size, and the accuracy of the simulation.
- **Chapter 6** modified the contact-implicit orthogonal collocation scheme described in the preceding chapter to include discontinuous transitions between contact phases. This extended the contact behaviors it is able to model to include partially-elastic collisions, and impacts without collision.
- **Chapter 7** proposed two coordinate formulations for rigid-body systems that improve the tractability of trajectory optimization problems when compared to the widely-used minimal coordinate formulation [106]: Relative Translation, Absolute Orientation (RTAO) coordinates reference the orientation of all bodies to the inertial frame, leading to more computationally-efficient planar problems, while maximal coordinates reference all coordinates to the inertial frame and were the best-performing option for spatial problems.
- **Chapter 8** presented a preliminary method of generating random, gait-like motions termed "Silly Walks", which were found to offer a more favorable balance of tractability and solution variety than other forms of random seed.

9.2 Perspective

In the conclusion of the second case study in Chapter 3, we declare that "the primary goal of the study was to determine whether a large set of incrementally-improving solutions produced by trajectory optimization would yield identifiable trends." Of course, that is not entirely true: the primary goal of the study was clearly to find out how a fast-moving quadruped should bring itself to a sudden halt. Particularly at the time that study was conducted, when our fledgling research group did not yet have a robot to its name, nor the subsequently-developed technology to reconstruct the motion of filmed cheetahs in 3D [177], the standard of proof associated with such a goal seemed too lofty for us to satisfy.

This reticence points to the fundamental limitation of this thesis: the lack of an ultimate validation step. While we have described our framework for investigating unknown maneuvers with trajectory optimization, and the ways we have adapted the formulation of these problems to support it, we have yet to show that the results it gives are grounded in reality.

There is good reason to be skeptical of trajectory optimization. In a popular textbook on Convex Optimization [178], local optimization of nonlinear problems is deemed to be "more art than technology". Even assuming that the model itself is a reasonable representation of reality, there is always a question of whether a superior feasible trajectory exists, but is more difficult to extract from the solution space, or whether a subtle shift in some parameter or bound might produce a radically different outcome. (In this regard, it might be helpful to introduce a degree of randomness into the system models in our future work.)

When interpreting the results of these studies, we must always keep the distinction between *optimal* and *optimized* locomotion in mind [32]. A mean value of some characteristic extracted from a collection of *optimized* trajectories might be an indication of how that characteristic contributes to successful motion, or it might just indicate which solutions a given numerical process produces most frequently.

To truly complete this thesis, the first priority of our future work – my first act as *Dr.* Shield, as it were – should be to pursue physical validation. One way to do this would be to extend the work to its logical conclusion of *optimization-inspired control*, and confirm that the strategies obtained are, indeed, viable ways to execute aggressive maneuvers on hardware. Another would be to (somewhat literally) *walk before we try to run*, by applying our approach to a more well-trodden legged locomotion problem such as walking initiation or termination in humans, and evaluating whether the conclusions match established results.

9.3 Opportunities

9.3.1 "Grey Box" Motion Planning

In recent years, many of the most spectacular advances in robotic motion planning have used model-free, *black box* methods – for example, the use of reinforcement learning in simulated environments to teach the quadruped ANYmal agile and robust motion skills [15, 16]. While advances in robust trajectory optimization [36, 40, 3] have improved the adaptability of solutions to uncertainty in the model or environment, it is still a decidedly *white box* (model-based) approach, with flexibility and computational efficiency remaining challenging in online applications. For complicated but well-specified tasks such as the MIT Cheetah 3's desk jump [1], however, trajectory optimization is a more effective method of generating a feasible solution *from scratch*.

Given the complementary nature of trajectory optimization and machine learning, there is potential for the two methods to enhance each other when combined in a *grey box* approach. An immediately obvious option is the possible use of the large datasets of local minima our proposed framework generates as training data for synthesizing successful maneuvers. Prior work has demonstrated the ability of trajectory optimization data to teach robots motion primitives for walking [179], and improve the efficiency of high-dimensional policy search algorithms for complex tasks including legged locomotion [105, 180].

Considering the computational cost of contact-implicit trajectory optimization, and the difficulty of finding viable contact sequences for complex tasks without a good initial seed [26], there is a clear case for bilevel approaches that offset the contact discovery problem to gradient-free methods that might handle it more efficiently. TrajectoTree [150] is a recent example of a combination along these lines.

Conversely, black box approaches could guide trajectory optimization by providing a rapid means of synthesizing favorable seed trajectories. 'Memory of motion' methods [152, 181] have shown that neural networks trained on the results of offline trajectory optimization can generate effective seeds for optimal control – an approach that extends logically to trajectory optimization.

9.3.2 Method Advancements

It is likely that major performance advances in trajectory optimization are only possible through developments at the solver level, as the general-purpose solvers currently in use are not optimized for the specific numerical challenges arising in these problems. Current work towards this goal includes that of Howell et al. [182].

There is, however, still room for improvement in the formulation and initialization of these problems. Some openings for future work we have identified over the course of the research discussed in this dissertation include:

- the incorporation of recent bilevel and convex optimization approaches to handle the contact problem more effectively [138, 116, 117, 136].
- a trajectory-optimization-specific "minimum rotation" coordinate formulation that combines the most efficient joint models from the RTAO

and maximal coordinate formulations.

- improved problem initialization, be that through further development of the "Silly Walk" method [21], an incremental process incorporating reduced-order versions of the problem [149], the aforementioned 'memory of motion'-inspired approach [152], or some combination thereof.

Bibliography

[1] Q. Nguyen, M. J. Powell, B. Katz, J. Di Carlo, and S. Kim, "Optimized jumping on the mit cheetah 3 robot," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7448–7454.

[2] G. Schultz and K. Mombaur, "Modeling and optimal control of human-like running," *IEEE/ASME Transactions on mechatronics*, vol. 15, no. 5, pp. 783–792, 2009.

[3] L. Drnach and Y. Zhao, "Robust trajectory optimization over uncertain terrain with stochastic complementarity," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1168–1175, 2021.

[4] M. Haberland and S. Kim, "On extracting design principles from biology: Ii. case study—the effect of knee direction on bipedal robot running efficiency," *Bioinspiration & biomimetics*, vol. 10, no. 1, p. 016011, 2015.

[5] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of experimental biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[6] S.-H. Lee and A. Goswami, "Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots," in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 4667–4672.

[7] H. Celik and S. J. Piazza, "Simulation of aperiodic bipedal sprinting," *Journal of biomechanical engineering*, vol. 135, no. 8, 2013.

[8] C. Paul and J. C. Bongard, "The road less travelled: Morphology in the optimization of biped robot locomotion," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 226–232.

[9] A. Falisse, G. Serrancolí, C. L. Dembia, J. Gillis, I. Jonkers, and F. De Groote, "Rapid predictive simulations with complex musculoskeletal models suggest that diverse healthy and pathological human gaits can emerge from similar control strategies," *Journal of The Royal Society Interface*, vol. 16, no. 157, p. 20190402, 2019.

[10] F. De Groote and A. Falisse, "Perspective on musculoskeletal modelling and predictive simulations of human movement to assess the neuromechanics of gait," *Proceedings of the Royal Society B*, vol. 288, no. 1946, p. 20202432, 2021.

[11] S. Shield and A. Patel, "Investigating rapid gait termination with synthetic data," 2019, presented at the Symposium on Comparative Biomechanics across Organizational Scales (Tissues to Whole Body Dynamics) at ISB/ASB 2019.

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

[13] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *Journal of Optimization Theory and Applications*, vol. 152, no. 2, pp. 271–306, 2012.

[14] D. Gong, J. Yan, and G. Zuo, "A review of gait optimization based on evolutionary computation," *Applied Computational Intelligence and Soft Computing*, vol. 2010, 2010.

[15] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[16] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[17] A. Shirazi, J. Ceberio, and J. A. Lozano, "Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions," *Progress in Aerospace Sciences*, vol. 102, pp. 76–98, 2018.

[18] D. Malyuta, Y. Yu, P. Elango, and B. Açıkmeşe, "Advances in trajectory optimization for space vehicle control," *Annual Reviews in Control*, vol. 52, pp. 282–315, 2021.

[19] S. Shield and A. Patel, "Balancing stability and maneuverability during rapid gait termination in fast biped robots," in *2017 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 4523–4530.

[20] A. Patel, S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler, "Contact-implicit trajectory optimization using orthogonal collocation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2242–2249, 2019.

[21] S. Shield and A. Patel, "On the effectiveness of silly walks as initial guesses for optimal legged locomotion problems," in *2020 International SAUPEC/RobMech/PRASA Conference*. IEEE, 2020, pp. 1–7.

[22] N. Anderson, S. Shield, and A. Patel, "The ollie: A case study in trajectory optimization with varied contacts," in *International SAUPEC/RobMech/PRASA Conference*. IEEE, 2020, pp. 1–7.

[23] A. Knemeyer, S. Shield, and A. Patel, "Minor change, major gains: the effect of orientation formulation on solving time for multi-body trajectory optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5331–5338, 2020.

[24] S. Shield and A. Patel, "Waste not, want not: Lessons in rapid quadrupedal gait termination from thousands of suboptimal solutions," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4012–4019.

[25] S. Shield, R. Jericevich, A. Patel, and A. Jusufi, "Tails, flails, and sails: How appendages improve terrestrial maneuverability by improving stability," *Integrative and comparative biology*, vol. 61, no. 2, pp. 506–520, 2021.

[26] S. Shield, A. M. Johnson, and A. Patel, "Contact-implicit direct collocation with a discontinuous velocity state," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5779–5786, 2022.

[27] M. Haberland and S. Kim, "On extracting design principles from biology: I. method–general answers to high-level design questions for bioinspired robots," *Bioinspiration & biomimetics*, vol. 10, no. 1, p. 016010, 2015.

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

[29] J.-P. Laumond, N. Mansard, and J. B. Lasserre, "Optimization as motion selection principle in robot action," *Communications of the ACM*, vol. 58, no. 5, pp. 64–74, 2015.

[30] C. Hubicki, M. Jones, M. Daley, and J. Hurst, "Do limit cycles matter in the long run? stable orbits and sliding-mass dynamics emerge in task-optimal locomotion," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5113–5120.

[31] C. Fisher, C. Hubicki, and A. Patel, "Do intermediate gaits matter when rapidly accelerating?" *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3418–3424, 2019.

[32] J.-P. Laumond, N. Mansard, and J.-B. Lasserre, "Optimality in robot motion: optimal versus optimized motion," *Communications of the ACM*, vol. 57, no. 9, pp. 82–89, 2014.

[33] K. D. Mombaur, H. G. Bock, J. Schloder, and R. W. Longman, "Open-loop stability-a new paradigm for periodic optimal control and analysis of walking mechanisms," in *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, vol. 2. IEEE, 2004, pp. 704–709.

[34] M. Diehl, K. Mombaur, and D. Noll, "Stability optimization of hybrid periodic systems via a smooth criterion," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1875–1880, 2009.

[35] K. Mombaur, "Using optimization to create self-stable human-like running," *Robotica*, vol. 27, no. 3, pp. 321–330, 2009.

[36] H. Dai and R. Tedrake, "Optimizing robust limit cycles for legged locomotion on unknown terrain," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 1207–1213.

[37] C. D. Remy, K. Buffinton, and R. Siegwart, "A matlab framework for efficient gait creation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 190–196.

[38] A. Gazar, M. Khadiv, A. Del Prete, and L. Righetti, "Stochastic and robust mpc for bipedal locomotion: A comparative study on robustness and performance," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 61–68.

[39] Y. Shirai, X. Lin, Y. Tanaka, A. Mehta, and D. Hong, "Risk-aware motion planning for a limbed robot with stochastic gripping forces using nonlinear programming," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 4994–5001, 2020.

[40] L. Drnach, J. Z. Zhang, and Y. Zhao, "Mediating between contact feasibility and robustness of trajectory optimization through chance

complementarity constraints," *Frontiers in Robotics and AI*, vol. 8, 2021.

[41] I. Mordatch, K. Lowrey, and E. Todorov, "Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5307–5314.

[42] M. Haberland and S. Kim, "On extracting design principles from biology: I. method–general answers to high-level design questions for bioinspired robots," *Bioinspiration & biomimetics*, vol. 10, no. 1, p. 016010, 2015.

[43] Z. Gan, Y. Yesilevskiy, P. Zaytsev, and C. D. Remy, "All common bipedal gaits emerge from a single passive model," *Journal of The Royal Society Interface*, vol. 15, no. 146, p. 20180455, 2018.

[44] Z. Gan, Z. Jiao, and C. D. Remy, "On the dynamic similarity between bipeds and quadrupeds: a case study on bounding," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3614–3621, 2018.

[45] M. Srinivasan and A. Ruina, "Computer optimization of a minimal biped model discovers walking and running," *Nature*, vol. 439, no. 7072, pp. 72–75, 2006.

[46] J. R. Usherwood, "Inverted pendular running: a novel gait predicted by computer optimization is found between walk and run in birds," *Biology Letters*, vol. 6, no. 6, pp. 765–768, 2010.

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

[48] W. Xi, Y. Yesilevskiy, and C. D. Remy, "Selecting gaits for economical locomotion of legged robots," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1140–1154, 2016.

[49] N. F. Steenkamp and A. Patel, "Minimum time sprinting from rest in a planar quadruped," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3866–3871.

[50] G. Buondonno, J. Carpentier, G. Saurel, N. Mansard, A. De Luca, and J.-P. Laumond, "Actuator design of compliant walkers via optimal control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 705–711.

[51] G. Saurel, J. Carpentier, N. Mansard, and J.-P. Laumond, "A simulation framework for simultaneous design and control of passivity based walkers," in *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*. IEEE, 2016, pp. 104–110.

[52] Y. Yesilevskiy, Z. Gan, and C. David Remy, "Energy-optimal hopping in parallel and series elastic one-dimensional monopeds," *Journal of Mechanisms and Robotics*, vol. 10, no. 3, p. 031008, 2018.

[53] K. Wampler and Z. Popović, "Optimal gait and form for animal locomotion," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 1–8, 2009.

[54] G. Fadini, T. Flayols, A. Del Prete, N. Mansard, and P. Souères, "Computational design of energy-efficient legged robots: Optimizing for size and actuators," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9898–9904.

[55] G. Fadini, T. Flayols, A. del Prete, and P. Souères, "Simulation aided co-design for robust robot optimization," 2022.

[56] T. Luksch, K. Berns, K. Mombaur, and G. Schultz, "Using optimization techniques for the design and control of fast bipeds," in *Advances In Climbing And Walking Robots*. World Scientific, 2007, pp. 454–465.

[57] N. Smit-Anseeuw, R. Gleason, R. Vasudevan, and C. D. Remy, "The energetic benefit of robotic gait selection—a case study on the robot ramone," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1124–1131, 2017.

[58] C. Fisher, S. Shield, and A. Patel, "The effect of spine morphology on rapid acceleration in quadruped robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2121–2127.

[59] C. Fisher and A. Patel, "On the optimal spine morphology of rapidly accelerating quadrupeds," *SAIEE Africa Research Journal*, vol. 112, no. 3, pp. 126–133, 2021.

[60] L. Raw, C. Fisher, and A. Patel, "Effects of limb morphology on transient locomotion in quadruped robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3349–3356.

[61] C. L. Dembia, N. A. Bianco, A. Falisse, J. L. Hicks, and S. L. Delp, "Opensim moco: Musculoskeletal optimal control," *PLOS Computational Biology*, vol. 16, no. 12, p. e1008493, 2020.

[62] V. Q. Nguyen, R. T. Johnson, F. C. Sup, and B. R. Umberger, "Bilevel optimization for cost function determination in dynamic simulation of human gait," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 7, pp. 1426–1435, 2019.

[63] Y. Blum, H. R. Vejdani, A. V. Birn-Jeffery, C. M. Hubicki, J. W. Hurst, and M. A. Daley, "Swing-leg trajectory of running guinea fowl suggests task-level priority of force regulation rather than disturbance rejection," *PloS one*, vol. 9, no. 6, p. e100399, 2014.

[64] A. V. Birn-Jeffery, C. M. Hubicki, Y. Blum, D. Renjewski, J. W. Hurst, and M. A. Daley, "Don't break a leg: running birds from quail to ostrich prioritise leg safety and economy on uneven terrain," *Journal of Experimental Biology*, vol. 217, no. 21, pp. 3786–3796, 2014.

[65] K. H. Koch, K. D. Mombaur, and P. Souères, "Studying the effect of different optimization criteria on humanoid walking motions," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots.* Springer, 2012, pp. 221–236.

[66] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Autonomous robots*, vol. 28, no. 3, pp. 369–383, 2010.

[67] K. Mombaur, J.-P. Laumond, and A. Truong, "An inverse optimal control approach to human motion modeling," in *Robotics Research.* Springer, 2011, pp. 451–468.

[68] A. Patel and M. Braae, "Rapid turning at high-speed: Inspirations from the cheetah's tail," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2013, pp. 5506–5511.

[69] ——, "Rapid acceleration and braking: Inspirations from the cheetah's tail," in *2014 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2014, pp. 793–799.

[70] C. Fisher, A. Blom, and A. Patel, "Baleka: A bipedal robot for studying rapid maneuverability," *Frontiers in Mechanical Engineering*, vol. 6, p. 54, 2020.

[71] A. Blom and A. Patel, "Investigation of a bipedal platform for rapid acceleration and braking manoeuvres," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 426–432.

[72] Y. Jian, D. A. Winter, M. G. Ishac, and L. Gilchrist, "Trajectory of the body cog and cop during initiation and termination of gait," *Gait & Posture*, vol. 1, no. 1, pp. 9–22, 1993.

[73] K. Hase and R. Stein, "Analysis of rapid stopping during human walking," *Journal of neurophysiology*, vol. 80, no. 1, pp. 255–261, 1998.

[74] A. Oates, A. Patla, J. Frank, and M. Greig, "Control of dynamic stability during gait termination on a slippery surface," *Journal of Neurophysiology*, vol. 93, no. 1, pp. 64–70, 2005.

[75] M. Bishop, D. Brunt, N. Pathare, and B. Patel, "The effect of velocity on the strategies used during gait termination," *Gait & posture*, vol. 20, no. 2, pp. 134–139, 2004.

[76] P. Vanitchatchavan, "Termination of human gait," in *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2009, pp. 3169–3174.

[77] M. Qiao and D. L. Jindrich, "Leg joint function during walking acceleration and deceleration," *Journal of biomechanics*, vol. 49, no. 1, pp. 66–72, 2016.

[78] F. W. O'Kane, C. A. McGibbon, and D. E. Krebs, "Kinetic analysis of planned gait termination in healthy subjects and patients with balance disorders," *Gait & posture*, vol. 17, no. 2, pp. 170–179, 2003.

[79] W. Sparrow and O. Tirosh, "Gait termination: a review of experimental methods and the effects of ageing and gait pathologies," *Gait & posture*, vol. 22, no. 4, pp. 362–371, 2005.

[80] Y.-C. Pai and J. Patton, "Center of mass velocity-position predictions for balance control," *Journal of biomechanics*, vol. 30, no. 4, pp. 347–354, 1997.

[81] S. B. Williams, H. Tan, J. R. Usherwood, and A. M. Wilson, "Pitch then power: limitations to acceleration in quadrupeds," *Biology letters*, vol. 5, no. 5, pp. 610–613, 2009.

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

[83] A. Goswami and V. Kallem, "Rate of change of angular momentum and balance maintenance of biped robots," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4. IEEE, 2004, pp. 3785–3790.

[84] M. Vukobratović and B. Borovac, "Zero-moment point—thirty five years of its life," *International journal of humanoid robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[85] M. Pijnappels, I. Kingma, D. Wezenberg, G. Reurink, and J. H. Van Dieën, "Armed against falls: the contribution of arm movements to balance recovery after tripping," *Experimental brain research*, vol. 201, no. 4, pp. 689–699, 2010.

[86] D. S. Marigold, A. J. Bethune, and A. E. Patla, "Role of the unperturbed limb and arms in the reactive recovery response to an unexpected slip during locomotion," *Journal of neurophysiology*, vol. 89, no. 4, pp. 1727–1737, 2003.

[87] P. E. Roos, M. P. McGuigan, D. G. Kerwin, and G. Trewartha, "The role of arm movement in early trip recovery in younger and older adults," *Gait & posture*, vol. 27, no. 2, pp. 352–356, 2008.

[88] Z. Merrill, A. J. Chambers, and R. Cham, "Arm reactions in response to an unexpected slip—impact of aging," *Journal of biomechanics*, vol. 58, pp. 21–26, 2017.

[89] A. B. Shetty and B. R. Etnyre, "Contribution of arm movement to the force components of a maximum vertical jump," *Journal of Orthopaedic & Sports Physical Therapy*, vol. 11, no. 5, pp. 198–201, 1989.

[90] E. A. Harman, M. T. Rosenstein, P. N. Frykman, R. M. ROSenStein *et al.*, "The effects of arms and countermovement on vertical jumping," *Med Sci Sports Exerc*, vol. 22, no. 6, pp. 825–833, 1990.

[91] B. M. Ashby and S. L. Delp, "Optimal control simulations reveal mechanisms by which arm movement improves standing long jump performance," *Journal of biomechanics*, vol. 39, no. 9, pp. 1726–1734, 2006.

[92] K. B. Cheng, C.-H. Wang, H.-C. Chen, C.-D. Wu, and H.-T. Chiu, "The mechanisms that enable arm motion to enhance vertical jump performance—a simulation study," *Journal of biomechanics*, vol. 41, no. 9, pp. 1847–1854, 2008.

[93] Z. J. Domire and J. H. Challis, "An induced energy analysis to determine the mechanism for performance enhancement as a result of arm

swing during jumping," *Sports Biomechanics*, vol. 9, no. 1, pp. 38–46, 2010.

[94] P. De Leva, "Adjustments to zatsiorsky-seluyanov's segment inertia parameters," *Journal of biomechanics*, vol. 29, no. 9, pp. 1223–1230, 1996.

[95] R. V. Mann, "A kinetic analysis of sprinting." *Medicine and science in sports and exercise*, vol. 13, no. 5, pp. 325–328, 1981.

[96] M. D. Johnson and J. G. Buckley, "Muscle power patterns in the mid-acceleration phase of sprinting," *Journal of sports sciences*, vol. 19, no. 4, pp. 263–272, 2001.

[97] D. H. Perrin, R. J. Robertson, and R. L. Ray, "Bilateral isokinetic peak torque, torque acceleration energy, power, and work relationships in athletes and nonathletes," *Journal of Orthopaedic & Sports Physical Therapy*, vol. 9, no. 5, pp. 184–189, 1987.

[98] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, J. D. Siirola *et al.*, *Pyomo-optimization modeling in python*. Springer, 2017, vol. 67.

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

[100] A. HSL, "collection of fortran codes for large-scale scientific computation," *See http://www. hsl. rl. ac. uk*, 2007.

[101] B. M. Nigg and M. Yeadon, "Biomechanical aspects of playing surfaces," *Journal of sports sciences*, vol. 5, no. 2, pp. 117–145, 1987.

[102] D. J. Block, K. J. Åström, and M. W. Spong, "The reaction wheel pendulum," *Synthesis Lectures on Control and mechatronics*, vol. 1, no. 1, pp. 1–105, 2007.

[103] P. E. Hudson, S. A. Corr, and A. M. Wilson, "High speed galloping in the cheetah (acinonyx jubatus) and the racing greyhound (canis familiaris): spatio-temporal and kinetic characteristics," *Journal of Experimental Biology*, vol. 215, no. 14, pp. 2425–2434, 2012.

[104] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE International Conference on Humanoid Robots*, 2014, pp. 295–302.

[105] S. Levine and V. Koltun, "Learning complex neural network policies with trajectory optimization," in *International Conference on Machine Learning.* PMLR, 2014, pp. 829–837.

[106] B. Siciliano, O. Khatib, and T. Kröger, *Springer handbook of robotics.* Springer, 2008, vol. 200.

[107] J. Brüdigam and Z. Manchester, "Linear-time variational integrators in maximal coordinates," in *International Workshop on the Algorithmic Foundations of Robotics.* Springer, 2020, pp. 194–209.

[108] H. Ferrolho, V. Ivan, W. Merkt, I. Havoutis, and S. Vijayakumar, "Inverse dynamics vs. forward dynamics in direct transcription formulations for trajectory optimization," in *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2021, pp. 12 752–12 758.

[109] A. Hereid, O. Harib, R. Hartley, Y. Gong, and J. W. Grizzle, "Rapid trajectory optimization using c-frost with illustration on a cassie-series dynamic walking biped," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2019, pp. 4722–4729.

[110] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids).* IEEE, 2015, pp. 874–880.

[111] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres, and K. Mombaur, "Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids).* IEEE, 2015, pp. 684–689.

[112] B. Ponton, A. Herzog, A. Del Prete, S. Schaal, and L. Righetti, "On time optimization of centroidal momentum dynamics," in *2018 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2018, pp. 5776–5782.

[113] C. Li, Y. Ding, and H.-W. Park, "Centroidal-momentum-based trajectory generation for legged locomotion," *Mechatronics*, vol. 68, p. 102364, 2020.

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

[115] R. Budhiraja, J. Carpentier, and N. Mansard, "Dynamics consensus between centroidal and whole-body models for locomotion of legged robots," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6727–6733.

[116] Z. Zhou and Y. Zhao, "Accelerated admm based trajectory optimization for legged locomotion with coupled rigid body dynamics," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 5082–5089.

[117] Z. Zhou, B. Wingo, N. Boyd, S. Hutchinson, and Y. Zhao, "Momentum-aware trajectory optimization and control for agile quadrupedal locomotion," *arXiv preprint arXiv:2203.01548*, 2022.

[118] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2703–2710.

[119] B. Ponton, M. Khadiv, A. Meduri, and L. Righetti, "Efficient multicontact pattern generation with sequential convex approximations of the centroidal dynamics," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1661–1679, 2021.

[120] M. L. Felis and K. Mombaur, "Using optimal control methods to generate human walking motions," in *International Conference on Motion in Games*. Springer, 2012, pp. 197–207.

[121] A. Hereid, C. M. Hubicki, E. A. Cousineau, J. W. Hurst, and A. D. Ames, "Hybrid zero dynamics based multiple shooting optimization with applications to robotic walking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5734–5740.

[122] 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 *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1447–1454.

[123] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for hzd gait optimiza-

tion," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 370–387, 2018.

[124] D. Pardo, M. Neunert, A. W. Winkler, and J. Buchli, "Projection based whole body motion planning for legged robots," *arXiv preprint arXiv:1510.01625*, 2015.

[125] D. Pardo, M. Neunert, A. W. Winkler, R. Grandia, and J. Buchli, "Hybrid direct collocation and control in the constraint-consistent subspace for dynamic legged robot locomotion." in *Robotics: Science and Systems*, vol. 10, 2017.

[126] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[127] D. W. Marhefka and D. E. Orin, "A compliant contact model with nonlinear damping for simulation of robotic systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 29, no. 6, pp. 566–572, 1999.

[128] A. Ö. Önol, R. Corcodel, P. Long, and T. Padır, "Tuning-free contact-implicit trajectory optimization," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1183–1189.

[129] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[130] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 2012, pp. 137–144.

[131] I. Mordatch, J. M. Wang, E. Todorov, and V. Koltun, "Animating human lower limbs using contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, pp. 1–8, 2013.

[132] B. Baumrucker, J. G. Renfro, and L. T. Biegler, "MPEC problem formulations and solution strategies with chemical engineering applications," *Computers & Chemical Engineering*, vol. 32, no. 12, pp. 2903–2913, 2008.

[133] M. Neunert, F. Farshidian, and J. Buchli, "Efficient whole-body trajectory optimization using contact constraint relaxation," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 43–48.

[134] E. Todorov, "A convex, smooth and invertible contact model for trajectory optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1071–1076.

[135] T. Erez and E. Todorov, "Trajectory optimization for domains with contacts using inverse dynamics," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4914–4919.

[136] I. Chatzinikolaidis, Y. You, and Z. Li, "Contact-implicit trajectory optimization using an analytically solvable contact model for locomotion on variable ground," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6357–6364, 2020.

[137] J. J. Moreau, "On unilateral constraints, friction and plasticity," in *New variational techniques in mathematical physics*. Springer, 2011, pp. 171–322.

[138] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization with implicit hard contacts," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3316–3323, 2018.

[139] J. J. Moreau, "Unilateral contact and dry friction in finite freedom dynamics," in *Nonsmooth mechanics and Applications*. Springer, 1988, pp. 1–82.

[140] B. Landry, J. Lorenzetti, Z. Manchester, and M. Pavone, "Bilevel optimization for planning through contact: A semidirect method," *arXiv preprint arXiv:1906.04292*, 2019.

[141] Y. Zhu, Z. Pan, and K. Hauser, "Contact-implicit trajectory optimization with learned deformable contacts using bilevel optimization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9921–9927.

[142] T. Howell, S. Le Cleac'h, S. Singh, P. Florence, Z. Manchester, and V. Sindhwani, "Trajectory optimization with optimization-based dynamics," *IEEE Robotics and Automation Letters*, 2022.

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

[144] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[145] M. P. Kelly, "Transcription methods for trajectory optimization," *Tutorial, Cornell University, Feb*, 2015.

[146] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes.* SIAM, 2010.

[147] Z. Manchester, N. Doshi, R. J. Wood, and S. Kuindersma, "Contact-implicit trajectory optimization using variational integrators," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1463–1476, 2019.

[148] M. Srinivasan, "Fifteen observations on the structure of energy-minimizing gaits in many simple biped models," *Journal of The Royal Society Interface*, vol. 8, no. 54, pp. 74–98, 2011.

[149] T. Marcucci, M. Gabiccini, and A. Artoni, "A two-stage trajectory optimization strategy for articulated bodies with unscheduled contact sequences," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 104–111, 2016.

[150] C. Chen, P. Culbertson, M. Lepert, M. Schwager, and J. Bohg, "Trajectotree: Trajectory optimization meets tree search for planning multi-contact dexterous manipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2021, pp. 8262–8268.

[151] J. Norby and A. M. Johnson, "Fast global motion planning for dynamic legged robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2020, pp. 3829–3836.

[152] N. Mansard, A. DelPrete, M. Geisert, S. Tonneau, and O. Stasse, "Using a memory of motion to efficiently warm-start a nonlinear predictive controller," in *2018 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2018, pp. 2986–2993.

[153] E. Berger, "Friction modeling for dynamic system simulation," *Appl. Mech. Rev.*, vol. 55, no. 6, pp. 535–577, 2002.

[154] F. Marques, P. Flores, J. Pimenta Claro, and H. M. Lankarani, "A survey and comparison of several friction force models for dynamic analysis of multibody mechanical systems," *Nonlinear Dynamics*, vol. 86, no. 3, pp. 1407–1443, 2016.

[155] E. Pennestrì, V. Rossi, P. Salvini, and P. P. Valentini, "Review and comparison of dry friction force models," *Nonlinear dynamics*, vol. 83, no. 4, pp. 1785–1801, 2016.

[156] P. Brown and J. McPhee, "A continuous velocity-based friction model for dynamics and control with physically meaningful parameters," *Journal of Computational and Nonlinear Dynamics*, vol. 11, no. 5, 2016.

[157] P. R. Dahl, "A solid friction model," Aerospace Corp El Segundo Ca, Tech. Rep., 1968.

[158] D. A. Haessig Jr and B. Friedland, "On the modeling and simulation of friction," 1991.

[159] J.-S. Pang and D. E. Stewart, "A unified approach to discrete frictional contact problems," *International Journal of Engineering Science*, vol. 37, no. 13, pp. 1747–1768, 1999.

[160] D. Prestorius and C. Fisher, "A novel method for computing the 3d friction cone using complementary constraints," in *IEEE International Conference on Robotics and Automation*, 2021.

[161] D. E. Stewart, "Time-stepping methods and the mathematics of rigid body dynamics," *Impact and Friction of Solids, Structures and Machines*, vol. 1, 2000.

[162] P. Painlevé, "Leçons sur le frottement," *Comptes Rendu des Séances de l'Académie des Sciences*, vol. 121, pp. 112–115, 1895.

[163] A. R. Champneys and P. L. Várkonyi, "The painlevé paradox in contact mechanics," *IMA Journal of Applied Mathematics*, vol. 81, no. 3, pp. 538–588, 2016.

[164] D. E. Stewart, "Rigid-body dynamics with friction and impact," *SIAM review*, vol. 42, no. 1, pp. 3–39, 2000.

[165] Z. Zhao, C. Liu, W. Ma, and B. Chen, "Experimental investigation of the painlevé paradox in a robotic system," *Journal of Applied Mechanics*, vol. 75, no. 4, 2008.

[166] B. Brogliato, A. Ten Dam, L. Paoli, F. Génot, and M. Abadie, "Numerical simulation of finite dimensional multibody nonsmooth mechanical systems," *Appl. Mech. Rev.*, vol. 55, no. 2, pp. 107–150, 2002.

[167] Y. Or, "Painlevé's paradox and dynamic jamming in simple models of passive dynamic walking," *Regular and Chaotic Dynamics*, vol. 19, no. 1, pp. 64–80, 2014.

[168] J. J. Moreau, "Standard inelastic shocks and the dynamics of unilateral constraints," in *Unilateral Problems in Structural Analysis.* Springer, 1985, pp. 173–221.

[169] M. Kunze and M. D. M. Marques, "An introduction to Moreau's sweeping process," in *Impacts in Mechanical Systems.* Springer, 2000, pp. 1–60.

[170] M. Anitescu and F. A. Potra, "A time-stepping method for stiff multibody dynamics with contact and friction," *International Journal for Numerical Methods in Engineering*, vol. 55, no. 7, pp. 753–784, 2002.

[171] V. Acary and B. Brogliato, *Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics.* Springer Science & Business Media, 2008.

[172] R. I. Leine and H. Nijmeijer, *Dynamics and bifurcations of non-smooth mechanical systems.* Springer Science & Business Media, 2013, vol. 18.

[173] D. Baraff, "Linear-time dynamics using lagrange multipliers," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 137–146.

[174] C. Roithmayr, "Relating constrained motion to force through newton's second law," Ph.D. dissertation, Georgia Institute of Technology, 2007.

[175] F. E. Udwadia and A. D. Schutte, "An alternative derivation of the quaternion equations of motion for rigid-body rotational dynamics," 2010.

[176] J. Brüdigam and Z. Manchester, "Linear-time variational integrators in maximal coordinates," in *International Workshop on the Algorithmic Foundations of Robotics.* Springer, 2020, pp. 194–209.

[177] D. Joska, L. Clark, N. Muramatsu, R. Jericevich, F. Nicolls, A. Mathis, M. W. Mathis, and A. Patel, "Acinoset: a 3d pose estimation dataset and baseline models for cheetahs in the wild," in *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2021, pp. 13 901–13 908.

[178] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[179] K. H. Koch, D. Clever, K. Mombaur, and D. Endres, “Learning movement primitives from optimal and dynamically feasible trajectories for humanoid walking,” in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 866–873.

[180] I. Mordatch and E. Todorov, “Combining the benefits of function approximation and trajectory optimization.” in *Robotics: Science and Systems*, vol. 4, 2014.

[181] T. S. Lembono, C. Mastalli, P. Fernbach, N. Mansard, and S. Calinon, “Learning how to walk: Warm-starting optimal control solver with memory of motion,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1357–1363.

[182] T. A. Howell, B. E. Jackson, and Z. Manchester, “Altro: A fast solver for constrained trajectory optimization,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7674–7679.