Optimization-Based Control for Dynamic Legged Robots

Patrick M. Wensing[1], Michael Posa[2], Yue Hu[3], Adrien Escande[4], Nicolas Mansard[5], Andrea Del Prete[6]

arXiv:2211.11644v1 [cs.RO] 21 Nov 2022

Abstract—In a world designed for legs, quadrupeds, bipeds, and humanoids have the opportunity to impact emerging robotics applications from logistics, to agriculture, to home assistance. The goal of this survey is to cover the recent progress toward these applications that has been driven by model-based optimization for the real-time generation and control of movement. The majority of the research community has converged on the idea of generating locomotion control laws by solving an optimal control problem (OCP) in either a model-based or data-driven manner. However, solving the most general of these problems online remains intractable due to complexities from intermittent unidirectional contacts with the environment, and from the many degrees of freedom of legged robots. This survey covers methods that have been pursued to make these OCPs computationally tractable, with specific focus on how environmental contacts are treated, how the model can be simplified, and how these choices affect the numerical solution methods employed. The survey focuses on model-based optimization, covering its recent use in a stand alone fashion, and suggesting avenues for combination with learning-based formulations to further accelerate progress in this growing field.

Fig. 1. Optimization-based control has been a common enabler for legged systems to handle complex environments [3], [14], [39], [63]–[65].

I. INTRODUCTION

Over the past decade, we have witnessed rapid growth in the capabilities of legged robots, transitioning from a state of the art where only a few research groups had access to capable platforms, to one where robust locomotion is now common in industry and academic laboratories. This rapid progress has been enabled by a combination of advances across design and control, with optimization-based control strategies playing a central role in many of the milestone demonstrations during this period. Through these advances, current bipeds, quadrupeds, and humanoids can now walk reliably in nominal environments, and these improved capabilities have led to the first practical deployments of legged systems (e.g., the robot Spot from Boston Dynamics). In a world built for legs, the growing scope of these deployments offers a broad opportunity for impact on applications spanning logistics (e.g., delivery), agriculture, and home assistance, among many others.

Across these applications, optimization-based control strategies offer many advantages as a pathway to (or component of) capable autonomy. Two prevailing types of optimization-based control have been pursued: predictive and reactive. Predictive controllers consider an explicit system model to "reason" (in a strictly mathematical sense) about the consequences of their actions, iteratively devising and improving motion plans in

response to the situation at hand. Reactive controllers, by contrast, only consider their actions for the current instant. The execution of these controllers may depend on the system model, or may be model-free (e.g., the result of offline policy optimization). The benefits of the model-based approach are that it allows one to naturally consider safety constraints, which will be critical for many of the envisioned applications for legged robots. An additional motivation is that first-principles models generalize well for performance in unforeseen situations, which will be an asset for flexible autonomous deployments. The downsides of the model-based approach are that any safety certificates or performance guarantees are only as good as the models they are based on, and there is a fundamental challenge of incorporating perceptual streams into problem formulations. As such, while the past decade of model-based advances has accelerated the progress of the legged locomotion community, many recent results have motivated the benefits of learning-based strategies for gracefully handling hard-to-model and perception aspects. With this big picture in mind, a central goal of the review is to synthesize the past decade of model-based advances so that they can be further improved in the future and also incorporated with learning-based strategies—ultimately accelerating the practical deployment of legged robots.

A. Historical Context for the Survey

Much of the recent progress in optimization-based control for robots rests on decades of progress from the mathematical programming community and its early translation to our field. The idea of using optimization for the specification and control of robot movement traces back to decades before its current

[1]University of Notre Dame: `pwensing@nd.edu`
[2]University of Pennsylvania: `posa@seas.upenn.edu`
[3]University of Waterloo: `yue.hu@uwaterloo.ca`
[4]INRIA: `adrien.escande@inria.fr`
[5]LAAS-CNRS: `nmansard@laas.fr`
[6]University of Trento: `andrea.delprete@unitn.it`

popularity (e.g., [66]). The 1990s saw the use of quadratic programming for torque control of redundant manipulators [67], with the 2000s seeing generalizations of these ideas for application to the control of humanoids in simulation [68]–[70]. The advent of polynomial-time solutions to important classes of convex optimization problems in the mid-1990s [71] and subsequent commercially available solvers (e.g., [72]) has contributed to the rise in maturity and accessibility of optimization solutions for control. Building on these advances, work in the late 2000s and in the wave of research before the 2015 DARPA Robotics Challenge (DRC) focused on convex optimization for maintaining balance via reactive control. The role of optimization in these accomplishments has since led to a growth of recent work for pushing the boundaries of predictive control for quadrupeds and humanoids executing broader dynamic locomotion in more challenging environments.

B. Survey Goals

The overarching goal of this survey is to provide a synthesized entry point into this recent body of work. As such, our intention is for the survey to serve as the most effective resource for early-stage graduate students, while also providing new perspectives and targeted follow-on reading for established experts. Work since the DRC has led to many new methods for optimization through contacts, for real-time optimization using new simplified models, and through advances to numerical methods for handling ever more complicated robot models. These advances are the primary focus of the review.

Beyond covering this past work, an additional motivation for this survey is to provide the background for the next steps in legged locomotion research, enabling these systems to move beyond nominal environments and unlock mobility in unstructured terrains. A great deal of current research interest focuses on how to combine machine learning strategies with previously developed model-based ones. In this sense, the survey will provide valuable background for machine learning practitioners to come up to speed on existing model-based approaches. Other research continues to push forward numerical methods and algorithms for legged robots to make informed model-based decisions. The survey should serve as a valuable overview of recent work for these groups as well.

II. PROBLEM STATEMENT AND OVERVIEW

During the last decade, most of the research community working on legged robots (humanoids and quadrupeds in particular) has converged to the idea of generating motion based on the formulation of an Optimal Control Problem (OCP). In a high-level form, such an OCP can be written as:

$$\underset{x(\cdot),u(\cdot),\lambda(\cdot)}{\text{minimize}} \quad \text{Cost}(x(\cdot), u(\cdot), \lambda(\cdot)) \tag{1a}$$

$$\text{subject to} \quad M(q)\dot{\nu} + C(q,\nu)\nu + \tau_g(q) = S^T\tau + J(q)^T\lambda \tag{1b}$$

$$\text{ContactConstraints}(x(t), \lambda(t), u(t), \text{Env}) \tag{1c}$$

$$\text{KinematicsConstraints}(x(t)) \tag{1d}$$

$$\text{InputConstraints}(u(t), x(t)) \tag{1e}$$

$$\text{TaskConstraints}(x(t), u(t), \lambda(t)) \quad \forall t. \tag{1f}$$

The (infinite-dimensional) decision variables of the OCP are the *trajectories* of state $x = (q, \nu)$, control $u \triangleq \tau$ and contact forces λ exchanged between the robot and the environment Env. The robot configuration is represented as q, whereas its velocity as ν. Eq. (1a) is a user-defined cost functional representing a metric to minimize, such as energy consumption or distance to a desired target. Eq. (1b) represents the nonlinear whole-body robot dynamics [73], where $M(q)$ is the mass matrix, $C(q,\nu)\nu$ accounts for Coriolis and centrifugal forces, $\tau_q(g)$ contains the gravity forces, S is typically a matrix that selects the actuated degrees of freedom, and $J(q)$ is the contact Jacobian. Eq. (1c) encodes the contact-related constraints, such as non-penetration of rigid objects and friction-cone force constraints. Eq. (1d) includes state constraints resulting from the kinematics, such as joint position and velocity limits. Eq. (1e) typically includes the motor torque limits, but it could in general represent any input constraint. Finally, (1f) can be used to include any task-specific constraint in the OCP, such as fixed initial/final state, or field-of-view limits for a camera.

Clearly, this OCP is very general and therefore allows one to encode a wide variety of movements. For instance, point-to-point locomotion can be generated by specifying the initial and final robot states [34]. An object manipulation behavior could instead be computed by including the state of the object in x and specifying its desired value with an appropriate cost function [37], [74], [75]. While appreciating its versatility, we should acknowledge that some intrinsic limitations do exist in the OCP approach. For instance, it is hard to ensure stability [34] or robustness for the computed trajectories [45]. However, this review focuses on another limitation and how to address it: the computational complexity of the problem. Indeed, problem (1) hides several challenges, the main ones being: non-smoothness/stiffness of the dynamics, nonconvexity, and dimensionality. The rest of this section acts as an executive summary of the rest of the paper by briefly discussing different approaches to tackle these challenges and make problem (1) computationally tractable for offline trajectory optimization, or for online model predictive control (MPC). Each aspect is analyzed more thoroughly in Sections III-VI.

A. Contact Models

The first issue (non-smoothness/stiffness), which is discussed in detail in Section III, arises from physical contacts modeled in (1c). Contacts can be modeled as either *rigid* or *visco-elastic*. With appropriate choices, visco-elastic models ensure continuous dynamics, and smoothing techniques can be used to make them differentiable. The downside of visco-elastic models is that large stiffness values are necessary to generate realistic behaviors. This feature leads to stiff differential equations with corresponding numerical challenges for simulation and optimization.

Alternatively, contacts can be modeled as rigid (i.e., no penetration is allowed). In this case, the dynamics becomes *hybrid* because when two points make contact, their relative velocity must immediately become zero to avoid penetration. Therefore, the robot dynamics must be described by a mix of continuous-time and discrete-time equations, i.e., as a hybrid

dynamic system. The resulting OCP can then be tackled either as a Linear Complementarity Program (LCP) [14], [18], [76] or as a Mixed Integer Program (MIP) [26], [77], both of which require customized optimization techniques that are typically much less efficient than classic smooth optimization strategies.

A common way around the non-smoothness introduced by rigid contacts is to let the user fix the order in which contacts are made and broken. This makes the system dynamics time-varying (a special case of hybrid dynamics) and the OCP differentiable, therefore efficient smooth optimization can be used. The obvious downside is that it may be hard to guess the contact phases.

B. Dynamic Models

Besides the contact dynamics, the other major source of complexity is the robot multi-body dynamics (1b), which is high-dimensional and nonlinear. In turn, this makes the resulting optimization problem high-dimensional and non-convex. The high dimension is especially concerning in the context of online optimization, where fast computation times are mandatory. The non-convexity instead is always concerning, as it makes the solver sensitive to the initial guess that is used. As partial or total remedy to these issues, several *simplified models* have been proposed in the literature (also known as reduced-order models, or template models), and they are discussed in Section IV. These models should capture the most important part of the robot dynamics with a reduced state size. For example, for locomotion, the widely-used Linear Inverted Pendulum (LIP) model [53] considers only the Center of Mass (CoM) of the robot and the contact locations, neglecting the details of joint angles and velocities. Other models capture more features of the robot state, at the price of higher state size and nonlinearity in the dynamics. While these simplified models are key enablers for fast online computation, their simplifying assumptions (e.g., constant CoM height), or neglected constraints [78] (e.g., joint position and torque bounds) can severely limit the generated motions.

C. Optimal Control Solution Methods

After the OCP is mathematically formulated, there are a range of design choices that remain for solving it. General optimal control methods can be categorized as direct methods, indirect methods, or those based on dynamic programming. Within robotics, direct optimization methods prevail almost universally. However, even after choosing to use a direct method, the optimization formulation needs to be approximated as a finite-size nonlinear program to be solved by a numerical solver. This approximation process is called *transcription* and it crucially affects the results in terms of accuracy, numerical stability, and computational complexity.

Two families of transcription methods have been used in robotics (shooting and collocation) and we discuss them in Section V. A shooting method known as Differential Dynamic Programming (DDP) has been a topic of recent interest in the community due to its favorable computational properties and provision of a locally optimal feedback policy in addition to a locally optimal open-loop control. More generally, but

TABLE I
OVERVIEW OF THE STATE OF THE ART.

Papers	Contact model	Gait	Dynamics	Transcription
[1]–[9]	Soft/Smoothed	Optimized	Full	DDP
[10]	Soft/Smoothed	Optimized	Full	Collocation
[11]	Soft/Smoothed	Optimized	Simplified	Collocation
[12], [13]	Rigid	Optimized	Full	DDP
[14]–[20]	Rigid	Optimized	Full	Collocation
[21]–[27]	Rigid	Optimized	Simplified	Collocation
[28]–[32]	Rigid	Fixed	Full	DDP
[33]–[36]	Rigid	Fixed	Full	Multiple Shooting
[37]–[40]	Rigid	Fixed	Full	Collocation
[30], [41]	Rigid	Fixed	Simplified	Multiple Shooting
[42]–[44]	Rigid	Fixed	Simplified	DDP
[45]–[52]	Rigid	Fixed	Simplified	Collocation
[53]–[62]	Rigid	Fixed	Simplified	Single Shooting

specific to direct optimization for locomotion, we examine how contact modeling choices affect the choice of transcription tools and algorithms available, which becomes most critical when optimization is left to freely choose contact sequences.

Table I serves to categorize the choices that state-of-the-art papers have used in adopting different approaches for modeling the contacts and their sequencing, for simplifying the dynamics, and, ultimately, for numerically solving variants of (1). In practice, the highlighted approaches play a key role in control designers breaking problem (1) into many smaller subproblems that focus on building an overall solution in a hierarchical fashion (e.g., solving for footsteps first, then optimizing motions with fixed footsteps). Methods for architecting this decomposition remain an art more than a science, with a few potential approaches depicted in Fig. 2.

D. Trajectory Stabilization via Whole-Body Control

Depending on the complexity of the formulated OCP, the computation time may be too large for it to be solved inside a fast control loop in MPC style. In these cases, a reactive stabilizing controller is required to execute the computed motion on real hardware, or to provide control of aspects of the system that were ignored during trajectory optimization (e.g., due to modeling simplifications, as with the first two configurations in Fig. 2). Many techniques can be used to track a reference motion on a robotic system. However, in the last decade, the legged robotics community has converged to a certain class of reactive optimization-based whole-body control techniques, which mainly rely on the fast solution of small convex Quadratic Programs (QPs) to compute motor commands as a function of state feedback.

These QP techniques have represented an evolution of previous operational-space control paradigms that were originally developed for the control of manipulators. However, for legged systems, new problems related to contact constraints, impacts, self-collisions, etc., have motivated a broader perspective on this classical problem. This evolution has led to the consideration of tasks that include fully-specified task-space motions, as well as constraints characterized by task-space inequalities. In spite of these many extensions, the formulation of reactive control as an instantaneous control strategy has enabled the method to retain a convex formulation that is lost in more general transcription strategies. The convex-optimization perspective on reactive control has
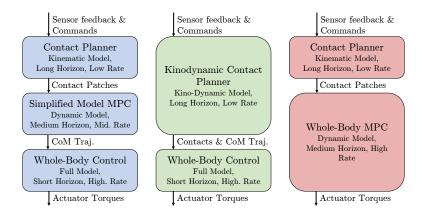
Fig. 2. Different potential strategies for breaking Problem 1 into smaller pieces to ease computational challenges.

also enabled other control paradigms, conventionally separate from operational-space control, to be incorporated as well. We describe these techniques and their connections in Section VI.

III. CONTACT

Computationally tractable treatment of physical, frictional contact with the environment, which leads to stiff and/or discontinuous equations of motion, is a fundamental challenge in optimization-based control for robotics and a primary distinction between (1) and standard optimal control problems. As a result, design decisions for (i) how to model the effects of contact and (ii) the sequencing or scheduling of contact events play a significant role in distinguishing different approaches to solving (1). Specifically, the details in this section center on how the choice of contact model affects parameterizations of the force λ and the form of the contact constraints in (1c).

This section is comprised of two parts. In Section III-A, we outline common techniques, and associated numerical challenges, for numerical models of contact dynamics. In Section III-B, we describe how the need to sequence contact events affects algorithmic choices and the modeling choices of Section III-A.

A. Modeling contact

Highly accurate mechanics models define contact as *visco-elastic*, representing local deformations of surfaces and the corresponding stress-strain relationships. While some approaches in robotics, particularly in soft robotics (e.g., [79]), do attempt to accurately represent deformation, where this deformation prevents interpenetration between bodies, rigid-body approximations to contact are most commonly used and will be the focus of this review. With this perspective, while bodies cannot deform, common visco-elastic contact models permit some penetration between contacting bodies, and the resulting contact forces are modeled as a function of penetration depth, relative normal and tangential velocities, and material properties. Noting critical definitions here, we write $\phi(q)$ to be the vector-valued signed distance function, for all possible contacts. $J_n(q)$ and $J_t(q)$ are the normal and tangential Jacobians, such that $J_n(q)\nu$ and $J_t(q)\nu$ are the contact frame velocities. Normal and tangential forces are similarly decomposed into λ_n and λ_t, and, where convenient, we use

$J(q)$ and λ to represent stacked Jacobians and forces. The contact force, therefore, can be expressed as some function
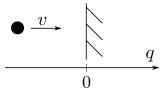
$$\lambda = F_{\text{contact}}(\phi, J\nu).$$

Common methods treat this function as a nonlinear spring and damper (e.g., [80], [81]), where unilateral constraints determine when the spring releases; typically, contact forces cannot "pull", constraining $\lambda_n \geq 0$ and so separation occurs if $\lambda_n < 0$. However, to accurately represent the interaction between rigid bodies, F_{contact} is numerically stiff [82], inheriting the mechanical stiffness of the material properties in the robot. When making and breaking contact ($\phi = 0$), F_{contact} may also be non-differentiable. Frictional effects exhibit similar issues: in dry Coulomb friction, λ_t depends on the sign of the tangential velocity, which is discontinuous.

Simply introducing these stiff or discontinuous dynamics to (1) leads to a poorly conditioned optimization problem and is generally avoided. Algorithmic approaches, therefore, tend to focus on one of two choices: (i) *infinitely* stiff representations, as hybrid models, or (ii) smoothing or softening F_{contact} to improve numerical performance. An illustrative example of a single contact is shown in Fig. 3, demonstrating the resulting state trajectory and, critical for optimization-based control, the sensitivity of the solution with respect to initial conditions.

1) Hybrid dynamics: In the limit of infinite stiffness, when contact is initiated, the resulting forces become *impulsive*; impacts, therefore, cause an instantaneous jump in velocity. While a full description of hybrid systems is outside the scope of this review (see [85] for an overview), we briefly note the principles here. Hybrid systems are defined by *modes*, *guards*, and *resets*. In this context, the mode is the contact state: identifying which objects are touching, and whether they are sticking or sliding. Guards determine when mode transitions occur, for instance making or breaking contact, and the reset map R specifies the result of the transition (e.g., impact):

$$x^+ = R(x^-).$$

Within a given mode, the dynamics are well-defined and differentiable; thus, this formulation compresses the complexity of contact to the guards and transition events. Referring to the root optimal control problem (1), a hybrid formulation typically explicitly splits the decision variables $x(\cdot), u(\cdot), \lambda(\cdot)$
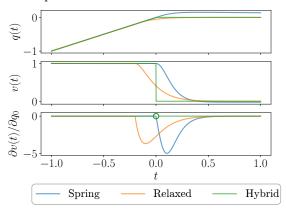
Fig. 3. In an illustrative example, a particle heads toward a wall, with inelastic impact when $q = 0$. Three common contact models are compared, showing (top-to-bottom) position, velocity, and sensitivity of velocity to initial conditions. (Blue) A Hunt-Crossley spring model [83] activates when contact occurs at $q(t) = 0$, allowing some penetration. The sensitivity $\partial v(t)/\partial q_0$ is well-defined, though not smooth (C^∞) at the impact. (Orange) a relaxed (softened) complementarity model, like MuJoCo, is smooth, with contact forces occurring prior to contact [84]. (Green) a hybrid model with inelastic impact captures rigid behavior with an instantaneous jump in velocity, noting that $\partial v(t)/\partial q_0$ is undefined at impact. All models have well-defined gradients *away* from the impact event, enabling effective differentiation in these regimes. For the hybrid and spring model, these gradients are zero prior to contact; for the relaxed model, as it is smooth, these gradients are non-zero everywhere.

by mode, introducing boundary value constraints at the resets when numerically transcribing the dynamics in Section V.

2) Complementarity models: A mathematically equivalent formulation of hybrid contact dynamics relies upon *complementarity constraints* [86], [87], where a relationship between λ and x implicitly defines the hybrid dynamics. For example, a non-penetration condition $\phi(q) \geq 0$, combined with $\lambda_n \geq 0$, and the property that forces can only be non-zero when in contact leads to the complementarity constraint

$$0 \leq \lambda_n \perp \phi(q) \geq 0\,,$$

where $\lambda_n \perp \phi(q)$ means that the vectors λ_n and $\phi(q)$ must be orthogonal, which is equivalent to $\lambda_n^T \phi(q) = 0$. This formulation can be expressed in time-stepping [76] or continuous [88] models, with a similar set of complementarity constraints used to capture Coulomb friction.

3) Pathologies: Hybrid formulations of multi-contact robotics, while effective for simulation and control, can exhibit certain pathologies where solutions do not exist or where infinite solutions are possible [86], [89], [90]. These challenges are the limiting case of the high sensitivity to initial conditions seen in the stiff differential equations. From the perspective of optimization-based control, this stiffness may be directly represented in the stiffness of the underlying optimization problem. Alternatively, for instance when using hybrid or complementarity models, planning algorithms that assume

uniqueness therefore implicitly (perhaps even unbeknownst to the algorithm designer) select from the set of possible solutions (a property, for instance, of [14]). In many scenarios, this selection is benign; in others, it indicates that the resulting trajectory is practically impossible to track, due to the non-uniqueness (equivalently, stiffness). Non-uniqueness can occur in a number of scenarios; most commonly, if $J(q)$ is rank deficient. Think, for example, of the static indeterminacy of a four-legged table or a flat robot foot on the ground, where the normal pressure distribution is indeterminate.

4) Hybrid differentiability: In the hybrid formulation, while both the resulting trajectories $x(t)$ and the equations of motion may be discontinuous, under certain circumstances it is possible to generate well-conditioned derivatives of $x(t)$ with respect to initial conditions and control actions [91], [92] (see Fig. 3 for an example). Typically, these derivatives exist if the sequence of modes is constant; the event times themselves might change, but the order does not and all transitions are inevitable. When deviations might change the mode sequence, resulting trajectories are often *not* differentiable with respect to initial conditions or parameterizations. For example, if a trajectory $x(t)$ does not make contact, then no amount of local differentiation provides insight into the effects of initiating contact; put more explicitly, solutions $x(t)$ are non-analytic in their initial conditions and control inputs. This property poses natural challenges for optimizers to discover new contacts through local information alone, and this motivates many relaxations or other tailored strategies for scheduling contacts.

To mitigate this issue, certain simulators ensure global differentiability or even smoothness (termed *differentiable simulation*, e.g., MuJoCo [84] or TDS [93]). While these methods do consistently generate local gradients, this inevitably ties the accuracy (stiffness) of the underlying model to stiffness in the resulting optimization problem. The precise nature of this trade-off and the relevance of the different modeling inaccuracies (see, e.g., [94]–[96]) remain unknown.

B. Scheduling contact

As the discussion of differentiability above implies, there are clear distinctions in computational tractability between versions of (1) with a fixed, known mode sequence and variations where the optimization problem must also determine the ordering. In many robotics problems, this sequence may be clear; for example, bipedal walking over flat terrain typically follows a "left foot, right foot" ordering with minimal deviation. In others, for instance, movements using whole-body contact or locomotion over varied surfaces with multiple potential foothold locations, the challenge of finding an optimal ordering may dominate the control problem.

1) Known modes sequences: In this setting, once discretized (see Section V), the hybrid optimal control problem is differentiable with respect to the transcribed decision variables. Within this framework, distinctions exist between methods in representation of the hybrid state space and the contact forces. In some settings, particularly when deploying simplified models, it is common to use *minimal coordinates* to represent the configuration of a robot in contact (e.g., [97], [98]). Contact

points are transformed into pin joints, removing degrees of freedom and simplifying the resulting optimal control problem. When contact modes change, however, the hybrid jump must also capture the change in dimension of the state space. The use of minimal coordinates is computationally efficient, although it is difficult to impose constraints (e.g., friction) on λ as inverse dynamics are necessary to reconstruct the constraint forces. Furthermore, in multi-contact settings, where the contact points generate a closed kinematic chain, global definitions of minimal coordinates may not exist.

The more general formulation utilizes the same maximal, or *floating-base coordinates* for all contact modes. The constrained dynamics then enforce that active contacts remain touching [31], [34]. Some approaches directly compute the force, e.g., $\lambda(x, u)$ [31]. This approach has the advantage of relative simplicity, though heuristics are necessary when the rigid-body force is non-unique. Alternate methods include $\lambda(t)$ as a decision variable (see (1)), along with corresponding constraints to ensure physical accuracy [14], [17]. While more variables are needed in this lifted formulation, it can be better conditioned numerically. It should also be noted, however, that imposing constraints *differentially* in this fashion comes at the risk of constraint drift due to integration error. Methods in this class can employ implicit integration schemes to eliminate this drift at some computational cost [38], [99].

2) Hybrid sequence optimization: As an immediate extension of the known-sequence scenario, one could jointly optimize over the discrete states of the hybrid sequence and the corresponding robot motion. This can naturally be expressed via *mixed-integer optimization* [22], [26], or as a bilevel optimization problem [21]. Mixed-integer programs (MIPs) have been used to compute contact sequences accounting for obstacle avoidance and step-to-step reachability [77] and approximated as computationally efficient L1-norm minimization (SL1M) [100]. Other strategies ensure "quasi-static" feasibility by limiting the search to "quasi-flat" contact surfaces (i.e., surfaces where the friction cone contains the gravity direction) [101]. In the worst case, these methods must explore every possible mode sequence, and so are most effective when there are relatively few potential sequences or when effective heuristics are available to guide the search.

Sampling-based techniques are a valid alternative to MIPs. For instance, rapidly-exploring random trees (RRTs) have been used to plan footsteps on flat ground, avoiding obstacles [102]. Similarly, probabilistic road maps (PRMs) have been used to plan a collision-free path for the robot's base, keeping it sufficiently close to the environment to allow for contact creation, while coupled with corresponding whole-body planning [103].

3) Contact-implicit planning: Contact-implicit (alternatively, contact-invariant) methods are variations on hybrid optimization that embed the relationship between state and force into a nonlinear program, implicitly representing the hybrid mode without discrete variables. Approaches here are most often based on either the complementarity formulation [14] or on smooth approximations of the contact dynamics [4], [11]. Prior to convergence, these methods typically violate strict complementarity, thus simultaneously exploring multiple contact sequences. Contact-implicit optimization, however, can

suffer from poor numerical conditioning and can require high-quality initial guesses. Contact-implicit methods have found applications in a number of robotic domains, including model predictive control [1] and gait optimization for microrobots [19]. Recent work in this area has focused on improving the numerical performance, for example by using simplified models [23], [25] or improving the accuracy of the numerical integration schemes [17], [18]. While these methods are typically limited to offline computation of motion plans, progress has been made in real-time scheduling of contact [7], [20]

These previous contact-implicit approaches explicitly account for the hybrid nature of planning through contact. Alternatively, gradient-based methods, like iLQR or DDP (see Section V-C), can be applied. This is most common when coupled with a differentiable contact model, enabling the optimization method to "discover" new modes (when it happens to bump into them, perhaps guided by the smoothing in the differentiable model) [1], [6], [9]. Other methods maintain a rigid model, and consider gradient analyses around the contact sequence found during forward simulation (e.g., [12], [13]), leveraging the almost everywhere differentiability discussed above. Broadly speaking, these methods are typically highly sensitive to their initializations and struggle to discover contact sequences that vary dramatically from that of the initial guess.

C. Summary

When the sequence of environmental contacts can be known *a priori*, then the choice of contact model should focus purely on physical realism, as nearly any transcription method can be applied without disruption. It is substantially more difficult, however, to simultaneously plan motion and contact schedule. While significant progress has been made over the last decade, contact planning remains the main challenge for generating arbitrary locomotion behaviors in complex environments. Ultimately, this challenge arises from our reliance on gradient-based optimization, which despite being the workhorse behind the progress discussed in this survey, is fundamentally unsuited for non-smooth contact-implicit problems.

1) Relationship to learning: While this review focuses on model-based optimization, there is an implicit requirement that such a model must first be identified or learned. Methods for learning and identification of contact align closely with the modeling choices detailed in this section. When contact modes are known or can be easily identified, hybrid approaches are commonly used [95]. Otherwise, machine learning can be directly applied to a smoothed or differentiable contact model (e.g., [104]). Alternatively, contact-implicit methods have led to data-efficient mechanisms for contact model learning [105]. Aside from identifying *models*, machine learning has also shown promise in identifying (or providing high-quality guesses for) potential mode sequences for control [106], [107], where an effective pre-solve for the mode schedule substantially reduces the computational difficulty of the subsequent trajectory optimization. We note, as well, that *model-free* reinforcement learning (RL), which has been widely applied to similar problems in control (e.g., legged locomotion [108], [109]), typically also leverages smoothed (or stochastic [110]) contact models for sim-to-real transfer.
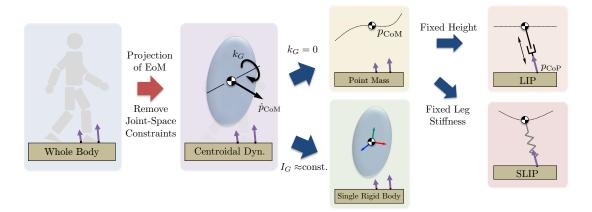
Fig. 4. The centroidal dynamics is a common modeling simplification that focuses on the CoM and net momentum. It is a special modeling simplification in that it exactly projects the whole-body dynamics, while relaxing joint-space constraints. Restrictions placed on this model then lead to a tree of other models.

IV. SIMPLIFIED MODELS

A second source of complexity for the efficient solution of (1) arises from the whole-body dynamics model (1b). As noted earlier, the dynamics of legged systems are high-dimensional and nonlinear, and these features correspondingly make the optimization problem high-dimensional and nonconvex. These challenges motivate simplified models in place of (1b) that capture its most salient features in a reduced set of differential equations. A central question is then how to select such a simplified model. We use the evolution in Fig. 4 to walk through the most common modeling simplifications adopted.

The role that contacts play within the whole-body dynamics (1b) has motivated the majority of simplified models employed to date. For a floating-base system, it is common to partition the generalized velocity as $\nu = (\nu_b, \ \nu_j)$ where $\nu_b = (\omega_b, \ v_b) \in \mathbb{R}^6$ gives the angular and linear velocity of the floating base and ν_j the generalized velocity of the joints. With this partitioning in ν, the dynamics are partitioned as

$$\begin{bmatrix} M_{bb} & M_{bj} \\ M_{jb} & M_{jj} \end{bmatrix} \begin{bmatrix} \dot\nu_b \\ \dot\nu_j \end{bmatrix} + C(q,\nu)\nu + \tau_g(q) = \begin{bmatrix} 0 \\ \tau \end{bmatrix} + J(q)^T \lambda \,. \quad (2)$$

Suppose now that the robot is given a desired motion $(q_d(t), \nu_d(t))$. Considering (2), the left-hand side of the equation is fixed along the desired motion, and so the motion is possible only when the equation can be satisfied through proper choice of joint torques τ and contact forces λ. If the robot has sufficiently strong actuators, then τ can be chosen to satisfy the bottom set of equations. However, since joint torques do not affect the first six rows, the main dynamic limitations on movement are determined by the contact force constraints (i.e., friction and unilaterality).

A. Centroidal Dynamics - Modeling and Optimization

It can be shown that the first six rows of (2) describe the evolution of the net linear and angular momentum of the system as a whole. This property follows from first principles of mechanics, and has been detailed within the context of floating-base robotics models in [111], [112]. It is common to consider the angular momentum about the CoM denoted $k_G \in \mathbb{R}^3$, as this quantity is both conserved during free flight and empirically

remains close to zero during human walking. Denoting the total linear momentum as $l_G \in \mathbb{R}^3$, these momenta compose the *centroidal momentum* $h_G = (k_G, \ l_G) \in \mathbb{R}^6$ [113]. This quantity is related to the generalized velocities via

$$h_G = A_G(q)\nu \,, \quad (3)$$

where $A_G(q)$ is called the centroidal momentum matrix [113]. Consider a case with n_c contact points at locations $\{p_i\}_{i=1}^{n_c}$. It can be shown that the top rows of (2) are equivalent to:

$$\dot h_G = \begin{bmatrix} \dot k_G \\ \dot l_G \end{bmatrix} = \begin{bmatrix} 0 \\ -Ma_g \end{bmatrix} + \sum_{i=1}^{n_c} \begin{bmatrix} (p_i - p_{\mathrm{CoM}}) \times \lambda_i \\ \lambda_i \end{bmatrix} \,, \quad (4)$$

where $p_{\mathrm{CoM}} = [c_x, c_y, c_z]^T$ gives the CoM position, M the total mass, and a_g the gravitational acceleration [111], [112]. It is important to note that $M\frac{\mathrm{d}}{\mathrm{d}t}p_{\mathrm{CoM}} = l_G$. As a result, these *centroidal dynamics* (4) motivate solving a reduced problem posed over trajectories for the CoM p_{CoM}, angular momentum k_G, contact locations p_i, and contact forces λ_i.

While the reduction to a centroidal dynamics model addresses the high-dimensionality of the original problem, it remains non-trivial to ensure that centroidal solutions are whole-body feasible, since they neglect the geometric constraints and actuation bounds. This relationship is depicted in Fig. 5 showing that only a subset of the motions that are feasible according to the centroidal dynamics are feasible for the whole-body model. This omission in the model has motivated planning with whole-body kinematics and centroidal dynamics [23]. Other strategies have separated these two, alternating between a centroidal dynamics and whole-body solve [30], [61]. Despite these advances, heuristic constraints on the CoM and footholds (e.g., to ensure reachability) remain prevalent, and simple bounds on the angular momentum (or fixing it to zero) are often used to simplify the problem.

Even with these simplifications, the centroidal dynamics (4) contains nonlinearities in the angular momentum equation due to bilinear terms from the cross product $(p_i - p_{\mathrm{CoM}}) \times \lambda_i$. Tailored methods to address these terms have been a topic of great recent focus. Dai et al. [47] considers fixing a-priori a polytopic or ellipsoidal region for the CoM, enabling optimization of an upper bound on the magnitude of the
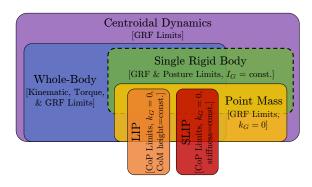
Fig. 5. Relationship between motions feasible according to different models. Parentheses [·] list constraints imposed, which may be physical (e.g., GRF limits) or artificial (e.g., CoM height for LIP). For example, the strict containment of the whole-body model within the centroidal dynamics indicates that there exists feasible centroidal momentum trajectories for any whole-body feasible motion, but not vice versa (due to missing constraints in the centroidal model). Since the point-mass model imposes an artificial restriction $k_G = 0$ but retains full GRF limits, its set is strictly inside the centroidal one. LIP and SLIP relax full GRF feasibility by focusing on the CoP only, and so are not strictly within the point-mass region. Note: The size of the areas pictured does not convey the range of feasible motions in any precise sense.

angular momentum. Valenzuela [114] considers relaxing the bilinear equalities with McCormick envelopes, which adds integer variables to the problem, but enables planning centroidal trajectories using mixed-integer convex optimization (see also, [27]). Fernbach et al. [52] presented an exciting development where the contact locations are fixed and the CoM trajectory is parameterized via a Bézier curve with one free knot point, which avoids nonlinear effects. Overall, there are many choices for optimizing centroidal trajectories, with natural tradeoffs between flexibility, accuracy, and computational efficiency.

It is possible to further simplify the optimization of contact forces by considering their net effect. Given frictional limitations, each individual contact force λ_i is constrained to a friction cone $\mathcal{C}_i$. Then, the net effect of these forces is a 6D cone known as the *Contact Wrench Cone* (CWC)

$$CWC = \left\{ \sum_{i=1}^{n_c} \begin{bmatrix} p_{i/\text{CoM}} \times \lambda_i \\ \lambda_i \end{bmatrix} \,\middle|\, \lambda_i \in \mathcal{C}_i, \ \forall i \in \{1 \ldots, n_c\} \right\}.$$

When each friction cone is replaced with a polygonal approximation (e.g., friction pyramid), then the CWC is a polyhedral convex cone, and many tools from computational geometry are available to support its use in planning (c.f., [115]). The CWC plays an integral role in checking for static stability in multicontact scenarios [116], [117] and more recently has seen applicability for multicontact CoM motion generation [48] and extension to address actuation limits [118]. Feasibility constraints based on the CWC are natural generalizations of the celebrated zero-moment point (ZMP) criteria [119] to multicontact. Since the computation of the CWC is itself costly, it is most useful when the CWC can be precomputed offline (e.g., when contact locations are known in advance).

B. Other Simplified Models

The centroidal dynamics (4) are a special modeling simplification in that they represent a projection of the equations of motion without adding artificial restrictions on possible movements (i.e., any whole-body feasible motion can be

projected to a feasible solution of the centroidal dynamics, as depicted in Fig. 5). Many other simple models follow by adopting artificial motion restrictions (or approximations) to further simplify analysis and optimization.

A common simple model for quadruped locomotion is the single-rigid-body (SRB) model (e.g., [62], [120], [121]). This simplification is motivated by the light leg designs of many quadruped robots, which makes the total rotational inertia of the system I_G about the CoM approximately invariant with configuration [122]. Thus, the SRB can be viewed as a restricted version of the centroidal momentum model in the case when the mass distribution takes a constant shape. Retaining a fixed inertia shape is not practically necessary, which makes this model an approximation in a way that simplified point-mass models are not. As a result, we denote this model slightly differently in Fig. 5 in recognition of this distinction. Variations on this model have been proposed to accommodate inertia shaping [121], [123], however, there is an important subtlety that must be kept in mind in this case. While the linear momentum of the system is related to the CoM velocity (i.e., the rate of change in some average position), the angular momentum cannot, in general, be equated to the rate of change in any meaningful average orientation of the system overall [124]. This property is intimately related to the fact that the conservation of angular momentum, in general, represents a non-holonomic constraint [111].

The most common simple model for humanoid gait planning considers the centroidal dynamics with $k_G = 0$ (i.e., thus removing consideration of orientation dynamics) and the additional restriction that the CoM height c_z takes a fixed value denoted h. Rather than considering a collection of contact points, when walking on level ground, the overall Center of Pressure (CoP) position $p_{\text{CoP}} = [p_x, p_y, p_z]^T$ can be considered with $p_z = 0$ an assumed ground height. These restrictions lead to the LIP Model [53] with dynamics: $\ddot{c}_{x,y} = \omega^2(c_{x,y} - p_{x,y})$ where $\omega = \sqrt{g/h}$ represents the natural frequency of the LIP. The beauty and power of the LIP resides in the fact that these dynamics are linear, enabling convex optimization for planning CoM and CoP trajectories [55] or linear systems tools (e.g., LQR) for trajectory tracking [49].

More recently, an area of interest has been discharging assumptions on the CoM height to enable more flexible motion plans [125]. One approach is to model height variations as a perturbation to the LIP [58] along with constraint tightening to ensure robust feasibility. Rather than treating height variations as a disturbance, they can be introduced into the dynamics model through [42], [125]. A common simple form is $\ddot{p}_{\text{CoM}} = a_g + k(p_{\text{CoM}} - p_{\text{CoP}})$ where k represents a variable stiffness-like parameter, constrained to be positive. Strategies for running (e.g., based upon the spring-loaded inverted pendulum model) [126], [127] can be seen as placing additional restrictions on the form of k based on assumptions of the virtual leg operating like a Hookean spring. While the SLIP dynamics are nonlinear, hybrid approaches can approximate SLIP physics via a combination of vertical spring dynamics and horizontal LIP dynamics [56], recovering linearity. Other hybrid approaches retain nonlinear dynamics [50], [128] by embracing more general NLP solvers.

C. Summary

Modeling simplifications are commonly employed to reduce the computational burden for dynamic planning in legged robots. The most successful of these models (e.g., the centroidal model, SRB model, and LIP model) all focus their modeling detail on addressing the motion limitations imposed by contact interactions. The centroidal model doesn't impose any restrictions on motions (i.e., it represents a projection of the whole-body dynamics), while the other simplified models can be viewed as resulting from adding motion restrictions. These simplifications can result in additional structure (e.g., linearity) that can be leveraged when posing motion optimization problems over them (e.g., using the methods in the next section). In all cases, then there are many whole-body details that remain to be planned (e.g., swing foot trajectories between contacts). This omission represents a main downside to simple model planning, and has motivated the development of advanced numerical methods, as covered in Section V, for the efficient optimization of whole-body plans.

1) Relationship with learning: There are many ways in which emerging learning strategies have been used to advance the application of simplified models for locomotion planning. With regards to the models themselves, this review has covered existing strategies based on considerations of physics and expert intuition. Moving forward, the automatic discovery of simple models remains an important open problem (c.f., [129]), potentially leveraging a range of dimensionality reduction tools from the learning literature. Other work has looked at how to close the gap between these simplified models and their whole-body counterparts, for example, to learn constraints on simplified models that address kinematics constraints at the whole-body level [78] or that address the gap via learning robust control strategies for MPC [130]. Yet other strategies have wrapped full RL pipelines around low-level control based on simple models [131] in an effort to increase the sample efficiency of learning. It should be acknowledged that even though RL strategies are often architected (e.g., via the autoencoder trick) to learn low-dimensional representations within their layers (c.f., [132]), these latent representations remain difficult to interpret. It is an open problem whether the existing simplified models may be used to increase the interpretability of learned controllers as well.

V. NUMERICAL METHODS FOR SOLVING OCPs

While the decomposition of our general problem formulation in Section II (e.g., via modeling simplifications) remains a bit of an art, the technical details of solving a trajectory optimization problem represents a field unto its own. This field of dynamic optimization has benefited from contributions across engineering applications ranging from chemical process control [133], to flight planning [134], and others. We begin this section by making a few simplifications to our general problem (1), so that we may review the most common methods for solving OCPs at large, which provides context for the most common approaches taken for robotics in particular. We then gradually discharge our original simplifications, and explain nuances that arise in OCPs for locomotion. The treatment

of these nuances has led to a rapid increase in whole-body optimization in recent years, where the need for fast online re-planning has led the community to develop fast structure-exploiting solvers tailored for robotics.

To begin, let us consider (1) in the case of a single contact phase (e.g., optimizing a trajectory for a humanoid doing an in-place dance on two-feet). For better alignment with the dynamic optimization literature, we rewrite the dynamics (1b) as a system of first-order ODEs $\dot{x}(t) = f(x(t), u(t), p(t))$ with time-varying parameters $p(t) = \lambda(t)$. The formulation can also be readily extended to consider time-invariant parameters. In the presence of contacts, the contact points must not move relative to the ground, which can be written as an algebraic constraint $g(t, x(t), u(t), p(t)) = 0$. These ODEs, combined with the algebraic constraints, lead to system dynamics that are more formally classified as a system of differential-algebraic equations (DAEs) [133]. A generic nonlinear optimal control problem for such a system can then be considered as:

$$\underset{x(\cdot),u(\cdot),p(\cdot)}{\text{minimize}} \quad \int_{t_0}^{t_f} \ell(x(t), u(t), p(t))dt + L(x(t_f)) \quad (5a)$$

$$\text{subject to} \quad \dot{x}(t) = f(x(t), u(t), p(t)) \quad (5b)$$

$$0 = g(x(t), u(t), p(t)) \quad (5c)$$

$$\forall t \in [t_0, t_f],$$

where the objective function, with running cost ℓ and terminal cost L, is minimized over the time interval $[t_0, t_f]$ subject to the system dynamics defined by the DAEs.

For locomotion problems, the formulation (5) requires extension to consider how the dynamics change in different contact modes. Here, we assume that the mode sequence is fixed a-priori, and consider the multi-phase problem:

$$\underset{\substack{x(\cdot),u(\cdot),p(\cdot), \\ s_1,...,s_{n_{\text{ph}}}}}{\text{minimize}} \quad \sum_{j=1}^{n_{\text{ph}}} \left[\int_{s_{j-1}}^{s_j} \ell_j(x(t), u(t), \lambda(t))dt + L_j(x(s_j)) \right]$$

$$(6a)$$

$$\text{subject to} \quad \dot{x}(t) = f_j(x(t), u(t), \lambda(t)) \quad (6b)$$

$$g_j(x(t), u(t), \lambda(t)) = 0 \quad (6c)$$

$$h_j(x(s_j^-)) = 0 \quad (6d)$$

$$x(s_j^+) = R_j(x(s_j^-)) \quad (6e)$$

$$\forall t \in [s_{j-1}, s_j], \ j = 1, ..., n_{\text{ph}} \quad (6f)$$

$$s_0 = t_0, s_{n_{\text{ph}}} = t_f, \quad (6g)$$

where $j = 1, ..., n_{\text{ph}}$ denotes the phase index, with the time interval of the j-the phase given by $[s_{j-1}, s_j]$, and n_{ph} the number of phases. At the mode switch, a guard constraint (6d) is typically used to ensure the active contact points are in contact with the environment. Depending on the contact model, the reset maps (6e) may correspond to the impact dynamics or to switching between coordinate representations.

Generally speaking, many methods for solving an OCP only apply directly to (5) in the ODE case, but often admit extensions to the DAE and multi-phase settings. These methods can be categorized in three groups: global methods based on the Hamilton-Jacobi-Bellman (HJB) equation (or Dynamic Programming in discrete-time), local indirect methods, and local direct methods (see Table II).

		Supporting Theory	Computation
Global		Hamilton-Jacobi-Bellman (HJB)	Numerical PDE Solvers or Dynamic Programming
Local	*Direct* (discretize then optimize)	Numerical Integration	**Direct Shooting/Collocation & NLP**
	Indirect (optimize then discretize)	Pontryagin Maximum Principle (PMP)	Indirect Shooting/Collocation & Root Find

TABLE II

HIGH-LEVEL RELATIONSHIP BETWEEN OPTIMAL CONTROL SOLUTION METHODS. THIS SURVEY FOCUSES MOSTLY ON DIRECT METHODS FOR CONTINUOUS-TIME PROBLEMS DUE TO THEIR PROMINENCE IN ROBOTICS APPLICATIONS.

Methods based on Dynamic Programming exploit Bellman's principle of optimality [135] to solve a discrete-time version of the OCP (5). They do so via finding the optimal cost-to-go $V(t_0, x_0)$ (also called the *Value function*). In continuous time, the Value function must satisfy a partial differential equation (PDE) known as the HJB. Its solution can be approximated by discretizing time and space and applying dynamic programming (DP). However, it is well-known that the complexity of this strategy increases exponentially with the number of states and controls. Therefore, it is not directly applicable to most legged robots, requiring clever approximate decomposition strategies for application to whole-body models [136].

Indirect methods transform the original OCP into a Boundary Value Problem (BVP) by using Pontryagin's Maximum Principle (PMP) [137] to formulate the so-called co-state equations. This approach enables pre-optimizing the control as a function of state and co-state if the dynamics is control affine—which is the case for most legged robots. Indirect methods turn the OCP into a root finding problem, which can provide fast and accurate solutions, but suffers from high sensitivity to the initialization. These methods are challenging to initialize, and it is also not possible to directly enforce state inequality constraints.

Direct methods, on the other hand, have been more broadly adopted by the legged robotics community for motion generation, and are generally compatible with systems of DAEs. In direct methods, the OCP is transcribed into a finite-dimensional Nonlinear Program (NLP) by discretizing the controls and states with respect to time. Therefore, direct methods directly find the minimum of (1a), where the NLP can be solved with well-established optimization techniques, e.g., Sequential Quadratic Programming (SQP) [138]. The difference between direct methods lies in the way the discretization is carried out. In the following, we focus exclusively on the three prominent direct methods in the community in recent years, namely *direct multiple shooting*, *direct collocation*, and *Differential Dynamic Programming (DDP)*. A comprehensive review of numerical methods for trajectory optimization (including discussion of indirect shooting and indirect collocation) can be found in [139].

A. Multiple shooting

In direct shooting methods, the controls $u(t)$ are discretized over the time horizon, while the state trajectories are obtained via forward integration. In contrast to *single shooting*, where a single integration is performed over the whole time horizon, in multiple shooting, the time horizon is discretized using a finite grid, then integration is performed on each of the
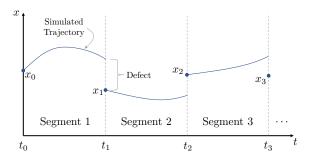

Fig. 6. Depiction of multiple shooting. Each shooting segment is associated with a numerical simulation (e.g., via a numerical integration method). Optimization variables include the states x_i at the beginning of each segment and variables describing the controls applied in each segment. Constraints enforce continuity, i.e., that the defect between segments is zero.

segments (Fig. 6), i.e., at each multiple shooting node. The initial state of each segment is added to the decision variables, and *continuity constraints* are introduced to guarantee that the final state of each segment i (computed with integration) matches the initial state of the next segment $i + 1$. In essence, as in Fig. 6, the initial optimization problem is divided into smaller optimization problems over the discretized grid, whose initial conditions can be set separately. By parameterizing the time horizon with a series of Initial Value Problems (IVP), the time interval over which integration is performed is shortened, reducing the high sensitivity issue of single shooting and preventing states from diverging. This makes multiple shooting a method that can deliver rather robust solutions as shown in [140], which motivated many researchers to use it in the context of motion generation for complex robots.

In particular, the use of multiple shooting has been popular for generating motions considering whole-body dynamics where contact modes are set a priori, i.e., predefined contact sequences. This is the case of [34], [33], and [35], where contacts are assumed to be rigid and impacts are instantaneous and inelastic. The resulting OCP is then a multi-phase problem with discontinuous phase transitions, as in (6). In [36], the authors used multiple shooting to optimize virtual constraints, coupled with Hybrid Zero Dynamics (HZD), for the gait generation of a compliant bipedal robot.

B. Collocation

In direct collocation (also known as direct transcription [139]) the original OCP is transformed into an NLP having control and state trajectories as decision variables. Controls and states are discretized over a time grid, where the intervals are called *finite elements* (see Fig. 7). Controls are approximated in each finite element by a finite dimensional representation, and the states are approximated by polynomials.
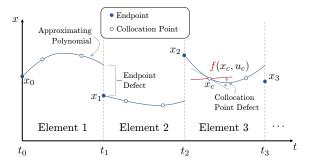
Fig. 7. Depiction of collocation. Each finite element is associated with a polynomial that approximates the trajectory. Optimization variables include the states x_i at each endpoint, and states/controls at each collocation point. Constraints enforce that the endpoint defect is zero, and that the slope of the approximating polynomial matches the evaluation of the dynamics function f at each collocation point.

The system dynamics is then enforced imposing *continuity constraints* to eliminate endpoint defects at the end of each element and though internal *collocation constraints* imposed at each of the *collocation points* (see Fig. 7). These collocation constraints enforce that the slope of the approximating polynomial matches the system dynamics $f(x(t), u(t), p(t))$ at each collocation point. The choice of the number and location of these collocation points has been thoroughly investigated [141] because it affects the accuracy of the method, namely how closely the NLP approximates the original OCP. More details on collocation methods can be found in [142]–[144].

Collocation has been a popular method to compute walking motions for legged robots using whole-body models and contact constraints. In [39], a periodic motion was generated for a whole-body bipedal robot, where the convergence of the periodic cycle is guaranteed through the implementation of HZD virtual constraints. In [40], the authors used collocation to generate motions for a quadruped robot, where the contact sequences were predefined. Collocation constraints were modified for solutions to respect contact constraints in [38], providing third-order integration accuracy with demonstrations on a full-body humanoid.

Most robotics methods to date have considered the above formulation where the system dynamics are transformed into a system of first-order ODEs. Vanilla collocation methods apply separate approximating polynomials for each component of the state (e.g., separate polynomials for q and $\dot{q}$), with their consistency only enforced at collocation points [145]. Recent work [145], [146] has shown the benefit of using a unified interpolating polynomial for the components of q and $\dot{q}$, which ensures their consistency across the trajectory, and improves the dynamic accuracy of the solution [145].

C. Differential Dynamic Programming (DDP)

In recent years a high interest has grown towards DDP and its variants. DDP [147] is an age-old method for solving discrete-time unconstrained OCPs. The method is based on Dynamic Programming, but it overcomes the curse of dimensionality by working on a local quadratic expansion of the so-called "Q function" (the sum of the current stage cost and the Value function at the next state). This makes DDP different from Dynamic Programming, because the method does not

attempt to exhaustively explore the state space of nonconvex problems. DDP roughly can be seen as an efficient iterative algorithm for solving the banded system of linear equations associated with the KKT conditions of an unconstrained OCP transcribed with collocation. Indeed, DDP closely resembles Netwon's method [148]; it has the same convergence rate (quadratic), but it is not completely equivalent to it. The main difference is that when solving the KKT system, DDP expresses the control inputs as linear functions of the state, but then, instead of computing control perturbations by using the linearized dynamics from the KKT system, it performs a forward simulation with the original nonlinear dynamics. This makes DDP a single shooting strategy—because the state is computed by integrating the dynamics—but it is better suited to handle unstable dynamics compared to vanilla shooting methods since the feedback form of the optimal control inputs helps to prevent divergence.

DDP requires the second derivatives of the dynamics to achieve quadratic convergence. While this is a nice property, it may be challenging to compute these terms for complex systems. So, in recent literature, variants of DDP that make use of Hessian approximations have been more popular, such as iLQR and iLQG [1], [29], [149], with other recent work focused on efficiently computing the full Hessian [150], [151].

The main limitation of DDP is that, in its original form, it cannot handle constraints besides the system dynamics. However, DDP has been extended in many ways to overcome this issue. Box DDP [152] is a simple extension that accounts for box constraints on the control inputs. Hierarchical DDP [29] allows for the optimization of a hierarchy of cost functions in lexicographic order. DDP extensions that can treat arbitrary nonlinear inequality state-control constraints have been proposed based either on interior-point techniques [153], active-set methods [154], Augmented Lagrangian [155], or relaxed barrier [44] approaches. Recently DDP has also been extended to handle implicit dynamics [6], hybrid dynamics [32], and multi-phase dynamics [31], all of which are useful for handling contacts. Even though DDP was born as a single-shooting method, it has been extended to multiple shooting [156]–[158] to further improve its handling of sensitive dynamics.

D. Contact Implicit Considerations

As a departure from the main assumptions of this section, contact-implicit (or contact-invariant) formulations (e.g., as first discussed in Section III-B2) instead seek to optimize the mode sequence. The choice of transcription strategy is highly coupled with the choice of contact model adopted. When using shooting methods, the key need is to be able to provide gradients that relate changes in states/controls to trajectory outcomes. For example, when adopting a relaxed contact model, the outcomes of simulation are always differentiable, enabling the use of shooting solvers and/or DDP [1], [6]. When adopting an event-driven hybrid model, these methods can still be used when paired with suitable hybrid sensitivity analysis, as in [13]. Likewise, discrete-time shooting can be applied with time-stepping LCP solvers [12], although both this case and the hybrid case exhibit pathologies that prevent differentiability in corner cases (c.f., Sec. III-A4).

When using direct collocation methods, existing contact-implicit methods rely on the inclusion of complementary constraints in the optimization. First-order accurate methods in [14] have been extended to allow for high-order schemes assuming the mode is not changing during an element [16]–[18]. These methods either require impacts at the element boundaries [18], [159] or allow for impacts during the element at the expense of integration accuracy [17], [18]. There is a clear trade-off here, since enforcing impacts at the element boundaries requires additional complementarity constraints, which are notoriously difficult to enforce. The alternative is to consider relaxations of the LCP constraints [7], which again present trade-offs for accuracy and computational complexity.

E. Summary

Within robotics, direct methods remain the most attractive numerical approaches for trajectory optimization. While both direct shooting and direct collocation can be used in cases when the mode sequence is fixed a priori, contact-implicit strategies require more careful consideration regarding the contact modeling choices adopted and how those affect the available transcription approach. Recent years have seen progress on accelerating the solution of shooting problems through the use of DDP, with recent sparse QP solvers (e.g., [160]) opening the door for similar accelerations to closely related collocation formulations.

1) Relationship to learning: There have been many motivating examples of learning being used to support trajectory optimization. A common strategy is to employ learned networks to provide either a policy or a trajectory to warm start optimization (see, e.g., [161], [162]). Another common way for learning to support trajectory optimization is by learning the value function, which enables online trajectory optimization over shorter horizons [163]. Other strategies simplify the optimization by learning footholds [164], e.g., akin to a higher-level contact planner.

VI. REALIZING MOTION PLANS

When using the tools of the previous section (posed over either simplified or whole-body models) for real-time predictive control, there often arises a need for separate reactive control components to realize the motion plans created. For example consider the setup depicted in Fig. 8. When performing trajectory optimization over simplified models (e.g., the SRB or a point-mass model), the OCP may be able to be solved quickly, but there still remain many details that need to be defined (e.g., swing leg motion). For trajectory optimization over whole-body models, the solver is often limited to running at a slower rate (though these rates are continually improving), requiring some additional high-rate closed-loop control. Optimal closed-loop feedback policies can sometimes be extracted from TO solutions, e.g., as with DDP [44]. In most other cases, some additional reactive control is required to realize optimized motion plans and handle disturbances.

Conceptually, when the time interval in the OCP (1) is reduced to a single instant, the OCP becomes a problem of *instantaneous control*, with the functional decision variables
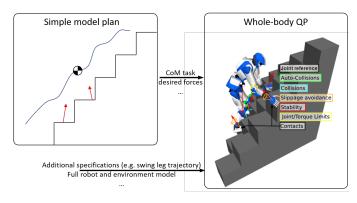

Fig. 8. A classical use of a Whole-Body QP in a pipeline to execute a simple-model plan obtained from the methods of the previous sections. A typical list of tasks and constraints is also depicted.

and their derivatives becoming simple vector variables, and objectives and constraints considering a single point in time. While it is conceptually interesting to consider instantaneous control schemes as limiting cases of the OCP, this is not how they were developed. Rather, the topic stands on its own, with a large body of literature and schemes that are widely used in practice on legged robots, either as a direct way to generate motion, or as a means to execute trajectories generated online with simplified models or offline by whole-body OCP.

The popularity of instantaneous reactive control schemes owes to their relative ease of implementation and the fact that, with a few assumptions and well-chosen objective/constraint formulations, the problem can be written as a Quadratic Program (QP), or a close variant, and solved very quickly (from hundreds of microseconds to a few milliseconds).

A. Formulation

In instantaneous settings, one common way to define the motion is through quantities e_i that we want to regulate to 0, or to keep above 0. For example, let us suppose that a walking gait has been generated using the LIP model, and we want the CoM of the robot to follow the LIP plan. For the robot CoM to track that trajectory, we want to achieve $e_i(q, t) = 0$ where $e_i \in \mathbb{R}^3$ gives the error between the target position from the LIP and the true position deduced from q by forward kinematics. Other tasks, such as collision avoidance, are naturally captured by inequality constraints $e_i(q) \geq 0$ with, for example, $e_i(q)$ being the distance between a robot body and an object. More generally, e_i can be a function of any subset of (q, ν, t). Such a definition is often called a *task error function* by reference to the task function [165] or operational-space [166] formalism, but can be also considered as an output of the system from a more controls-centric point of view.

A key point is that the dynamics (1b) are linear in $\dot{\nu}$, τ, and λ. This points at writing the regulation at the acceleration level where τ and λ can also be used directly to affect motions.

1) Task Dynamics: Consider a general task, in the special case where $\dot{q} = \nu$ and $e_i \in \mathbb{R}^m$. Differentiating $e_i(q, t)$ twice with respect to time gives

$$\ddot{e}_i(q, \nu, \dot{\nu}, t) = J_i(q)\dot{\nu} + \dot{J}_i(q, \nu)\nu + a_i(q, \nu, t), \quad (7)$$

where J_i is the Jacobian matrix of e_i and $a_i = \frac{\partial^2 e_i}{\partial t^2}$. This construction can also be generalized to the case where q and/or e_i do not lie in Euclidean space (e.g., for orientation tasks).

To bring or keep e_i to a desired value, one then writes $\ddot{e}_i = \ddot{e}_i^d$ or $\ddot{e}_i \geq \ddot{e}_i^{min}$ where $\ddot{e}_i^d/\ddot{e}_i^{min}$ is a function of any subset of (q, ν, t) that indicates how we want to regulate e_i (see VI-A3). This gives an equality/inequality that is linear in $\dot{\nu}$:

$$J_i(q)\dot{\nu} \gtreqqless \ddot{e}_i^d - \dot{J}_i(q,\nu)\nu - a_i(q,t). \tag{8}$$

If e_i depends on ν, a single differentiation leads to:

$$\frac{\partial e_i}{\partial \nu}\dot{\nu} \gtreqqless \dot{e}_i^d - J_i(q,\nu)\nu - \frac{\partial e_i}{\partial t}(q,\nu,t). \tag{9}$$

Both (8) and (9) can be written as $A_i(q,\nu)\dot{\nu} \gtreqqless b_i(q,\nu,t)$.

2) Task expressions: The most common tasks involve geometric features (e.g., body or CoM position) that are direct functions f_i of q. Typical equality tasks have the form $e_i(q,t) = f_i(q) - f_i^d(q,t)$, where f_i^d is the target value. For non-Euclidean tasks, this error can be generalized as $e_i(q,t) = f_i(q) \ominus f_i^d(q,t)$, where $\ominus$ denotes a meaningful difference in the task space [167]. This general definition can capture tasks for the CoM position, body positions, body orientations, posture, and others. An important use of position and orientation tasks is to describe the geometric part of a contact. Other equality tasks include a gaze task [168], visual servoing [169], or any subpart of one of the above [64], [170]. Geometric inequality tasks commonly include bounds, collision avoidance, balance, etc. [168], [171].

Few tasks depend directly on ν. This is obviously the case of joint speed limits $\underline{\nu} - \nu \geq 0$ and $\overline{\nu} - \nu \geq 0$, but also the centroidal momentum task [172]–[174] $e_i = A_G(q)\nu$ from (3).

We can also call tasks, by extension, constraints or motion requirements that are directly written on $\dot{\nu}$, τ and λ and do not need regulation. These include, for example, limits on $\dot{\nu}$, τ, as well as the friction constraints on the forces (to prevent sliding), that are usually approximated using a linear form $C\lambda \geq 0$ (e.g., as was also considered to simplify the CWC in Sec. IV-A). Direct references can also be given for these variables. For example $\lambda - \lambda^d = 0$ can be used for force control [175].

3) Regulation of Equality Tasks: The behavior the controller must achieve to keep a task error e_i at or above 0 can be described by a desired value for its derivative $e_i^{(l)}$ where l is the number of differentiations (0, 1 or 2) such that $e_i^{(l)}$ is an affine function of $\dot{\nu}$, τ or λ. For geometric equality tasks ($l = 2$), the most common regulator is a PD construction: $\ddot{e}_i^d = -K_d\dot{e}_i - K_p e_i$, where K_d and K_p are positive definite matrices, which are usually taken as diagonal or scalar matrices to get fully decoupled dynamics. K_d is typically set to $2K_p^{1/2}$ to obtain a critical damping. Constant gain matrices work well when tracking a trajectory or when e_i remains small, but incur either large acceleration or slow convergence when e_i starts large. Solutions to work around this issue include gains varying with the norm of e_i (e.g., main ideas in [70], [168]), or clamping/scaling $\ddot{e}_i^d$.

A common (and critical) equality task is the one related to maintaining contacts. Most often, hard contacts are assumed and $\ddot{e}_i^d$ is set to 0, even though it may be better to set a damping term to stabilize the contact in case of slippage [64]. As a related aside, it should be noted that around the initiation of contact events, this constraint can lead to a rapid change in the problem formulation and thus a rapid change in controls.

For velocity-level equality tasks, regulation is typically achieved with a simple linear feedback: $\dot{e}_i^d = K_p e_i$. For acceleration/force-level equality tasks, regulation is either omitted (e.g., $e_i^d = 0$), or implemented with Proportional-Integral schemes: $e_i^d = -K_p e_i - K_i \int e_i \, \mathrm{d}t$.

An alternative for making e_i converge to zero is to view the task error e_i as an output of the system and to design a control Lyapunov function V_i stabilizing it to 0. Then the inequality $\dot{V}_i \leq -\gamma_i V$, where γ_i sets the desired exponential convergence rate, forces the convergence of e_i to 0. This is the idea behind the CLF-QP [176], which brings convergence guarantees provided the inequalities are always feasible. However, ensuring persistent feasibility with practical robot models remains an open challenge.

4) Regulation of Inequality Tasks: Inequality tasks at the acceleration level, such as joint torque limits or force friction cones, do not require any regulation and can be directly imposed. Inequality tasks at the velocity level, such as joint velocity limits, can be easily regulated by computing $\dot{e}_i^d$ such that e_i remains positive at the next time step: $e_i^+ = e_i + \Delta t \dot{e}_i \geq 0$, where Δt is the time step of the controller. This approach works well for small Δt (which is usually the case), and implies setting $\dot{e}_i^d = -e_i/\Delta t$. Inequality tasks at the position level, such as joint position limits or obstacle avoidance, are much harder to enforce. The most common approach is to compute $\ddot{e}_i^d$ so that e_i^+ is positive, but this can easily lead to conflicts with the acceleration-level constraints [177]. This has motivated the development of several alternative approaches, ranging from the simple trick of using a larger value of Δt in the computation of e_i^+ [168], [177], to bounding velocities with a hand-tuned function of the distance to the position limit [178], to control barrier functions [179], [180]. However, none of these approaches ensure compatibility between position-level and acceleration-level inequalities. The only methods that can guarantee them are those that account for position-level and acceleration-level inequalities at once [181]–[183], even if they only work under the assumption of constant acceleration bounds, which is not the case in practice. Fundamentally, enforcing position constraints remains precarious with instantaneous control, since it is the evolution of the system dynamics over an extended time period that determines whether the position constraint can be respected in the future. This observation motivates the need for predictive strategies (e.g., repetitively solving trajectory optimization formulations) in order to address position inequality constraints via suitable lookahead.

B. Resolution

Instantaneous control schemes make the assumption that the state x is known and constant during the control period Δt. This is supported by the fact that Δt is small (typically between 1 and 10 ms), and relies on the ability to solve the

problem in that time. The contact state (i.e., which bodies are in contact) is assumed given and fixed at each control cycle.

Under these conditions, and if the friction cones are approximated as pyramids, aggregating all the desired behaviors of the tasks leads to a set of linear constraints:

$$M \quad \dot{\nu} - J^T \quad \lambda - S^T \quad \tau = -C\nu - \tau_g \tag{10a}$$

$$J_c \quad \dot{\nu} \qquad\qquad\quad = a_c \tag{10b}$$

$$A_{\dot{\nu},i}\dot{\nu} + A_{\lambda,i}\lambda + A_{\tau,i}\tau \gtreqless b_i \quad \text{for non-contact tasks } i, \tag{10c}$$

where (10a) is a reorganization of (1b), (10c) is a generalization of (VI-A1) to account for tasks directly written on τ or λ, and we have singled out the contact constraints in (10b).

Problem (10) represents what we would like the robot to achieve. The idea is to solve (10) for $\dot{\nu}$, τ, and λ as best as possible (for some definition of best given below). On position- or velocity-controlled robots, $\dot{\nu}$ is integrated twice or once and the resulting configuration [64] or velocity [169] is sent to the robot. For torque-controlled robot, τ is used as command.

The problem should give enough constraints on the motion to specify a single solution, and not let the numerical solver arbitrate and possibly induce discontinuities between the solutions for two consecutive states. A common remedy is to use regularization tasks (typically a posture task, $\lambda = 0$, or $\tau = 0$). In practice, the problem will be over-constrained, meaning that some tasks are in conflict, and the design of the controller needs to specify how to arbitrate the conflicts during the resolution (see Section VI-B2).

1) Pre-solving variants: Problem (10) is written over all 3 variables $\dot{\nu}$, τ and λ. This gives great flexibility to express new tasks/constraints and offers a clear vision of what we want to achieve. However, the specific structure, of (10a) and (10b) can be exploited to remove some variables and constraints, either to speed-up computation or for formulation-related reasons, giving rise to different families of controllers. This variable elimination can be regarded as partially pre-solving manually the problem before handling it to the solver.

The simple form of S (e.g., $\begin{bmatrix} 0 & I \end{bmatrix}$ for legged robot) allows to directly express τ as a linear function of $\dot{\nu}$ and λ and eliminate it from the problem. Only the non-actuated rows of (10a) are kept. This is the approach in [64], [174] and leads to an easy decrease of computation time. Alternatively, (10a) can be used to eliminate $\dot{\nu}$ (and removed), leveraging the positive definite characteristic of M for a cheap resolution based on the Cholesky decomposition [173], [184].

When $J_c = J^1$, (10b) can be used, in conjunction with (10a), to remove λ. This projection on the nullspace of the contact constraints is used in [185], [186] to extend the operational space control to underactuated robots [187]. More generally, one can use (10a) and (10b) to solve for $\dot{\nu}$ and λ, keeping τ as the only variable.

Some variable eliminations are rooted in the history of families of controllers. But when it comes to performance, some points have to be kept in mind. Solvers are internally performing operations akin to variable eliminations. Therefore performing one manually is only useful when some properties

of the matrices (e.g., sparsity or positive-definiteness) can be leveraged, that the solver cannot know. Solvers, on the other hand, can usually better handle numerical issues. The number of variables can be a good indicator of the computational cost, but the structure of the problem is also important [188]. Removing a variable can destroy the structure, preventing a solver to exploit it. The simplest example is when removing a variable transforms bound constraints into general constraints. Overall, the optimal pre-solving variant is highly coupled with structure exploiting capabilities of the numerical solver used.

2) Arbitrating conflict: For most task sets considered in practice, tasks requirements will conflict, meaning (10) has no solution. In this case, one can associate a violation measure to each constraint, based on the $L2$ norm, and specify how important it is to minimize each violation. Two main approaches exist that can be combined: weighing each violation (also known as *soft priorities*) or defining a hierarchy between them (*strict priorities*).

Numerous works use the so-called QP approach[2]: all inequality constraints and some equality constraints in (10) are kept as such in the QP, effectively being assigned the top priority. The weighted sum of the square norms of all other constraint violations forms the objective of the QP. For example, let us consider the problem of tracking a LIP-based reference trajectory from a simple model, and while trying to do so by using minimal joint torques within bounds. In this case, a QP could be formulated as:

$$\underset{\dot{\nu},\tau,\lambda}{\text{minimize}} \quad w_1\|J_G\dot{\nu} + \dot{J}_G\nu - (\ddot{p}_G^d + \ddot{e}_G^d)\|^2 + w_2\|\tau\|^2$$

$$\text{subject to } M\dot{\nu} + C\nu + \tau_g = S^T\tau + J_c^T\lambda \quad \text{(dynamics)}$$

$$J_c\dot{\nu} = a_c \quad \text{(fixed contacts)}$$

$$C\lambda \leq 0 \quad \text{(friction cones)}$$

$$\dots \text{(other top-priority constraints)} \dots$$

The use of QP formulations for reactive control is now nearly universal, but follows a long historical development. After an early work motivated by the inclusion of unilateral contact forces [189], the QP-based inverse dynamics approach really started with [68], [70] and closely related convex formulations [69], continuing with [64], [170], [190]–[192] among many others. An advantage of this approach is that it relies on off-the-shelf, mature solvers, largely available and often free (e.g., [193], [194]) Its main limitations are the inability to handle inequalities at lower priority levels and that the tuning of weights becomes difficult when many tasks exist.

One of the benefits of the QP formalism is its flexibility to incorporate other control paradigms into reactive whole-body control. For example [49], [195] embed optimal control for simple-model tasks via a cost function term designed to descend a task-space value function. In this regard, the instantaneous nature of the QP is mitigated by the long-term performance considerations embedded in the value function. Other work has explored the incorporation of passivity-based control approaches that tweak the above problem in subtle ways that improve robustness to unmodeled effects [196]. To improve robustness through other means, recent QP strategies

---

[1]this depends on the representation of the contact forces

[2]technically constrained linear least-squares

have explored methods for controls to be invariant to velocity jumps from impact events [197], which are hard to detect in practice. Alternative strategies that add the consideration of soft contact can also be incorporated into whole-body QPs [198], [199].

In contrast to using a weighting approach for tasks, a fully hierarchical approach lets the designer specify an explicit priority level for each task. Tasks at the same level can be combined with weights, and top priority constraints need not be feasible. The resulting problem is called *lexicographic least-squares* or *hierarchical QP* (HQP) and used in numerous works [168], [172], [174]. The cascade of QP of Kanoun [200] was the first solver able to tackle inequality constraints at any priority level, followed by variants improving its computation time [172], [174]. A few dedicated solvers [201], [202] have been introduced, solving the problem efficiently in one pass. The hierarchical formulation is a strict superset of the QP-based formulation, as well as the limit case for when the ratio between task weights goes to infinity [203]. It is shown in [202], contrary to the common belief, a HQP can be solved faster using a dedicated solver, both in theory and practice. However, the computational cost of the cascade solvers, and the complexity of handling singularities [204] have limited the use of the hierarchical approach.

C. Summary

This section has introduced common methods for reactive control that coordinate the selection of joint torques and contact forces at the current instant. These methods complement predictive control (i.e., over a horizon) as provided by methods in the previous section. By focusing on an instantaneous control problem, the formulation inherits desirable structure (e.g., nonlinear constraints become linear ones). This enables solutions at rapid rates which are amenable to computation using off-the-shelf QP solvers. There are many different variants to such whole-body problems that consider soft priorities vs. hard priorities between tasks, or that strategically pre-solve for some decision variables to accelerate the QP solution. While these methods have represented a natural outgrowth of operational-space control to legged robotics applications, the shift from solving least-square problems to solving QPs has enabled many other control paradigms to be considered with WBC, including CLFs, CBFs, passivity-based strategies, and others guided by value functions. Overall, while there remain opportunities to accelerate these solution methods and improve their robustness, the current state of the art represents a reliable technology for current and future applications.

VII. OUTLOOK AND PROSPECTS

We conclude this survey by reflecting on some of the main trends that were identified through the literature review, and also some of the notable gaps that remain to be addressed. While many of these observations represent desired further improvements to model-based approaches, we close with a few final remarks on the outlook for model-based approaches within the context of the parallel growing inertia of data-driven approaches (e.g., RL) that have also had impressive recent demonstrations (e.g., [132], [205]).

A. Main Trends

Main trends for optimization with contacts: The past five years have seen an increase in contact implicit approaches that avoid specifying gait/contact sequences a-priori. These methods have considered either complaint contact or MIP/LCP strategies. Solution speed remains a challenge for these methods, as does non-convexity of the problem. Strategies that combine planning and/or learning with contact implicit methods appear as a necessary next step. In the near term, fixing contact sequences/timing remains the most viable option for practical deployment in applications that require online optimization (e.g., in model-predictive control). In the end, while much more recent work has concentrated on addressing rigid contacts, other approaches end up being similar since rigid complementarity constraints are often relaxed and smoothed, leading to similar operations as with compliant models.

Main trends in the dynamic models adopted for model-based optimization: The past five years have observed an increase in model complexity used for MPC (with several examples using the full model). This advance has occurred from increased understanding of gradient-based numerical methods and the availability of fast open-source software libraries to compute robot's dynamics and its derivatives [206]. With Moore's law ending, subsequent advances may center on parallelization. However, even with increases in computational speed, the nonconvexity of problems is an unavoidable challenge with complex models. This observation suggests that simple models will likely remain pertinent for the near future. The review points that combination with learning strategies that include exploration (as in conventional RL) presents promise for optimization-based strategies to avoid poor local minima when adopting complex models.

Main trends for transcription: The past five years have included emphasis on DDP methods as well as extensions of collocation schemes to contact-constrained/implicit settings. Adoption of DDP has been motivated in part by its demonstrated scalability to high-DoF systems. However, recent work has also shown specialized Riccati-like solvers [160], which have the promise to bring many of the advantages of shooting via DDP to other transcription approaches. While the numerical methods are improving, convergence to infeasible solutions is a challenge for complex systems, and this holds with any of the transcription methods. The review points that addressing these challenges is an important open point for future work. Further, simultaneous methods (collocation and multiple shooting) will present additional opportunities for algorithm-level parallelism to take advantage of the shift in the how computational resources are expected to grow in the coming decades.

B. Overview of Open Problems

Formal Analysis: Despite a great deal of recent progress, there is a broad lack of formal analysis regarding the properties and operation of existing optimization-based robot control paradigms. Existing theoretical results regarding the stability and recursive feasibility of nonlinear MPC are mostly focused on the simplified problem of regulation to an equilibrium

or tracking a feasible reference state trajectory [207], which leaves a theoretical gap for the classes of problems considered in the locomotion community. Another barrier to meaningful analysis is a relative lack of work addressing considerations of robustness to bounded or stochastic uncertainty (e.g., [28], [58], [208]). Such a treatment would, for example, be critical to understand how our control frameworks interact with state estimation. While there are clear academic open questions, it remains an open question of its own as to how much additional performance such analysis would unlock.

Reducing the Expertise Required: Beyond the technical expertise required for model-based control, there remains a great deal of domain-specific expertise required for crafting cost functions and constraints. It remains a challenge to remove the human as an outer loop in the tuning of problem formulations. A part of the issue is that it is often not apparent how to write down cost functions for even simple tasks. For example, the problem of opening a door has binary success, and such cost structures are largely incompatible with current tools. Whether through automated methods, or through new tools that enable a broader set of cost designs, the best path for reducing required domain-specific expertise remains open.

Optimal Control Versus or With RL?: With the recent progress in RL for legged locomotion (e.g., [132], [205]) much debate has surrounded whether real-time optimization or reinforcement learned policies are the path forward. RL methods have some clear benefits, the main one probably being their generality. Indeed, being gradient free, they have no trouble with non-smooth contact-implicit problems, which are instead hard to solve with gradient-based optimization. An interesting explanation of why that is the case has been recently provided in [209], drawing a connection between *Policy Gradient* and the stochastic optimization technique *Randomized Smoothing*. Although our community has grown attached to gradient-based optimization, in order to overcome the challenges discussed in this survey it may be worth exploring gradient-free approaches, with RL being among the most promising. Another positive feature of RL methods (especially those based on Dynamic Programming [210]) is their exploration ability, which is fundamental to solve non-convex problems in cases where TO struggles.

Still, there remain exciting opportunities for how model-based optimization can help accelerate learning. Strategies such as the popular guided policy search method [211] and relatives (e.g., [212]) can leverage TO to cut down on needed exploration and guide learning. Another important opportunity for future work is to use the efficient constraint-handling nature of TO to help make learning *safe* [213]. Overall, since RL and OCP methods try to solve similar problems (especially in offline simulation settings where models are heavily used), the history of their synergy is likely still quite early.

C. Closing Remarks

Reflecting back on the trajectory of the field, we see a great deal of commonality on the transformation that is taking place now as occurred in the mid 2010s. In 2007, convex optimization strategies for reactive control were beginning to

appear, with no shortage of concern regarding both (a) their online computation requirements or (b) the abandonment of previously available closed-form control solutions. By 2015, these methods were common place.

From where we are at today, early demonstrations of whole-body predictive control on hardware have greatly benefited from the structural understanding of our physics-based models from previous reactive control developments, while also adopting reactive methods as low-level controllers. Yet, legitimate concerns remain regarding (a) online computational requirements and (b) the abandonment of convexity properties that would give solution assurances.

While we hope these methods will become technologies in the near future, we also look forward to their role in shaping the next-generation of learning-based controllers, where shared optimal control fundamentals will allow some of the insights in formulation and methodologies to inform and accelerate RL solutions. Concerns regarding (a) offline computation requirements and (b) a lack of generalization guarantees with RL seem to both be well addressed by structure exploiting methods from model-based optimization, and we hope this survey will play a small role in helping those solutions come to light. This evolutionary progress shows a steady maturation of the field toward applications in logistics, agriculture, construction, and yet others. With external commercial funding at unprecedented levels, we look forward to the next decade of progress from legged robots, which promises to be greater than the previous one.

REFERENCES

[1] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2012.

[2] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2015, pp. 3346–3351.

[3] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds," *IEEE Robot. and Autom. Lett.*, vol. 2, no. 3, 2017.

[4] M. Neunert, C. De Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *IEEE Int. Conf. on Robot. and Autom.*, 2016, pp. 1398–1404.

[5] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds," *IEEE Robot. and Autom. Lett.*, vol. 3, no. 3, pp. 1458–1465, 2018.

[6] I. Chatzinikolaidis and Z. Li, "Trajectory optimization of contact-rich motions using implicit differential dynamic programming," *IEEE Robot. and Autom. Lett.*, vol. 6, no. 2, pp. 2626–2633, 2021.

[7] S. L. Cleac'h, T. Howell, M. Schwager, and Z. Manchester, "Fast contact-implicit model-predictive control," *arXiv preprint arXiv:2107.05616*, 2021.

[8] T. A. Howell, S. Le Cleac'h, S. Singh, P. Florence, Z. Manchester, and V. Sindhwani, "Trajectory optimization with optimization-based dynamics," *IEEE Robot. and Autom. Lett.*, vol. 7, no. 3, pp. 6750–6757, 2022.

[9] A. Ö. Önol, R. Corcodel, P. Long, and T. Padır, "Tuning-free contact-implicit trajectory optimization," in *IEEE Int. Conf. on Robot. and Autom.*, 2020, pp. 1183–1189.

[10] Y. Zhu, Z. Pan, and K. Hauser, "Contact-implicit trajectory optimization with learned deformable contacts using bilevel optimization," in *IEEE Int. Conf. on Robot. and Autom.*, 2021, pp. 9921–9927.

[11] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. on Graphics*, 2012.

[12] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization with implicit hard contacts," *IEEE Robot. and Autom. Lett.*, 2018.

[13] N. J. Kong, G. Council, and A. M. Johnson, "iLQR for piecewise-smooth hybrid dynamical systems," in *IEEE Conf. on Decision and Control*, 2021, pp. 5374–5381.

[14] M. Posa, C. Cantu, and R. Tedrake, "A Direct Method for Trajectory Optimization of Rigid Bodies Through Contact," *Int. J. of Robot. Res.*, vol. 33, no. 1, 2014.

[15] W. Xi and C. D. Remy, "Optimal gaits and motions for legged robots," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2014, pp. 3259–3265.

[16] W. Xi, Y. Yesilevskiy, and C. D. Remy, "Selecting gaits for economical locomotion of legged robots," *Int. J. of Robot. Res.*, vol. 35, no. 9, pp. 1140–1154, 2016.

[17] Z. Manchester, N. Doshi, R. J. Wood, and S. Kuindersma, "Contact-implicit trajectory optimization using variational integrators," *Int. J. of Robot. Res.*, 2019.

[18] A. Patel, S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler, "Contact-implicit trajectory optimization using orthogonal collocation," *IEEE Robot. and Autom. Lett.*, 2019.

[19] N. Doshi, K. Jayaram, B. Goldberg, Z. Manchester, R. J. Wood, and S. Kuindersma, "Contact-Implicit Optimization of Locomotion Trajectories for a Quadrupedal Microrobot," 2019.

[20] A. Aydinoglu and M. Posa, "Real-time multi-contact model predictive control via admm," in *2022 Int. Conf. on Robot. and Autom.*, 2022, pp. 3414–3421.

[21] K. Wampler and Z. Popovi, "Optimal gait and form for animal locomotion," in *ACM Trans. on Graphics*, 2009.

[22] A. Ibanez, P. Bidaud, and V. Padois, "Emergence of humanoid walking behaviors from mixed-integer model predictive control," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2014.

[23] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE-RAS Int. Conf. on Humanoid Robots*, Madrid, Spain, 2014, pp. 295–302.

[24] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2016.

[25] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Syst. Through Phase-Based End-Effector Parameterization," *IEEE Robot. and Autom. Lett.*, 2018.

[26] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous Contact, Gait, and Motion Planning for Robust Multilegged Locomotion via Mixed-Integer Convex Optimization," *IEEE Robot. and Autom. Lett.*, 2018.

[27] Y. Ding, C. Li, and H. W. Park, "Single Leg Dynamic Motion Planning with Mixed-Integer Convex Optimization," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2018.

[28] J. Morimoto and C. G. Atkeson, "Minimax differential dynamic programming: An application to robust biped walking," *Advances in Neural Information Processing Syst.*, 2003.

[29] M. Geisert, A. Del Prete, N. Mansard, F. Romano, and F. Nori, "Regularized Hierarchical Differential Dynamic Programming," *IEEE Trans. on Robot.*, vol. 33, no. 4, 2017.

[30] R. Budhiraja, J. Carpentier, and N. Mansard, "Dynamics consensus between centroidal and whole-body models for locomotion of legged robots," in *IEEE Int. Conf. on Robot. and Autom.*, 2019.

[31] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential Dynamic Programming for Multi-Phase Rigid Contact Dynamics," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2019.

[32] H. Li and P. M. Wensing, "Hybrid Syst. differential dynamic programming for whole-body motion planning of legged robots," *IEEE Robot. and Autom. Lett.*, vol. 5, no. 4, pp. 5448–5455, 2020.

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

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

[35] K. H. Koch, K. Mombaur, O. Stasse, and P. Souères, "Optimization based exploitation of the ankle elasticity of HRP-2 for overstepping large obstacles," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014, pp. 733–740.

[36] 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 *IEEE Int. Conf. on Robot. and Autom.*, 2015, pp. 5734–5740.

[37] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *IEEE Int. Conf. on Robot. and Autom.*, 2009, pp. 489–494.

[38] M. Posa, S. Kuindersma, and R. Tedrake, "Optimization and stabilization of trajectories for constrained dynamical Syst." in *IEEE Int. Conf. on Robot. and Autom.*, 2016.

[39] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *IEEE Int. Conf. on Robot. and Autom.*, may 2016, pp. 1447–1454.

[40] 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: Sci. and Syst.*, 2017.

[41] J. Carpentier and N. Mansard, "Multicontact Locomotion of Legged Robots," *IEEE Trans. on Robot.*, 2018.

[42] S. Feng, X. Xinjilefu, W. Huang, and C. G. Atkeson, "3D walking based on online optimization," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2013.

[43] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *IEEE Int. Conf. on Robot. and Autom.*, 2017, pp. 93–100.

[44] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback MPC for Torque-Controlled Legged Robots," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2019.

[45] H. Dai and R. Tedrake, "Optimizing robust limit cycles for legged locomotion on unknown terrain," in *IEEE Conf. on Decision and Control*, 2012.

[46] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida, "Model preview control in multi-contact motion-application to a humanoid robot," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2014.

[47] H. Dai and R. Tedrake, "Planning robust walking motion on uneven terrain via convex optimization," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2016.

[48] S. Caron and A. Kheddar, "Multi-contact walking pattern generation based on model preview control of 3D COM accelerations," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2016.

[49] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, 2016.

[50] X. Xiong and A. D. Ames, "Dynamic and Versatile Humanoid Walking via Embedding 3D Actuated SLIP Model with Hybrid LIP Based Stepping," *IEEE Robot. and Autom. Lett.*, 2020.

[51] M. Khadiv, A. Herzog, S. A. A. Moosavian, L. Righetti, and L. Righetti, "Walking Control Based on Step Timing Adaptation," in *IEEE Trans. on Robot.*, 2020.

[52] P. Fernbach, S. Tonneau, O. Stasse, J. Carpentier, and M. Taïx, "C-CROC: Continuous and Convex Resolution of Centroidal Dynamic Trajectories for Legged Robots in Multicontact Scenarios," *IEEE Trans. on Robot.*, 2020.

[53] S. Kajita and F. Kanehiro, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE Int. Conf. on Robot. and Autom.*, 2003.

[54] P. B. Wieber, "Viability and predictive control for safe locomotion," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2008.

[55] A. Herdt, H. Diedam, P. B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Advanced Robot.*, 2010.

[56] I. Mordatch, M. D. Lasa, and A. Hertzmann, "Robust physics-based locomotion using low-dimensional planning," in *ACM SIGGRAPH*, 2010.

[57] A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Whole body motion controller with long-term balance constraints," in *IEEE-RAS Int. Conf. on Humanoid Robots*, nov 2014, pp. 444–450.

[58] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P. B. Wieber, "A robust linear MPC approach to online generation of 3D biped walking motion," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2015.

[59] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Compliant locomotion using whole-body control and Divergent Component of Motion tracking," in *IEEE Int. Conf. on Robot. and Autom.*, 2015.

[60] Y. Liu, P. M. Wensing, J. P. Schmiedeler, and D. E. Orin, "Terrain-Blind Humanoid Walking Based on a 3-D Actuated Dual-SLIP Model," *IEEE Robot. and Autom. Lett.*, 2016.

[61] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2016.

[62] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2018.

[63] A. Kheddar, S. Caron, P. Gergondet, A. Comport, A. Tanguy, C. Ott, B. Henze, G. Mesesan, J. Englsberger, M. A. Roa, *et al.*, "Humanoid robots in aircraft manufacturing: The airbus use cases," *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 30–45, 2019.

[64] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, E. Yoshida, S. Kajita, and F. Kanehiro, "Multi-contact vertical ladder climbing with an HRP-2 humanoid," *Autonomous Robots*, 2016.

[65] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *IEEE/RSJ Int. Conf. on Intel. Robots and Sys.*, 2018, pp. 2245–2252.

[66] J. E. Slotine and M. W. Spong, "Robust robot control with bounded input torques," *J. of Robotic Syst.*, 1985.

[67] F. T. Cheng, R. J. Sheu, and T. H. Chen, "The Improved Compact QP Method For Resolving Manipulator Redundancy," *IEEE Trans. on Syst., Man, and Cybernetics*, 1995.

[68] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle, "Dynamic Balance Control of Humanoids for Multiple Grasps and non Coplanar Frictional Contacts," in *Humanoids*, 2007.

[69] J. Park, J. Haan, and F. C. Park, "Convex optimization algorithms for active balancing of humanoid robots," *IEEE Trans. on Robot.*, vol. 23, no. 4, pp. 817–822, 2007.

[70] Y. Abe, M. Da, S. Jovan, and P. Popovi´c, "Multiobjective Control with Frictional Contacts," in *ACM SIGGRAPH/Eurographics Symp. on Computer animation*, 2007, pp. 249—-258.

[71] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.

[72] E. D. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Mathematical Programming, Series B*, 2003.

[73] R. Featherstone, *Rigid body dynamics algorithms*. Springer Berlin:, 2008, vol. 49.

[74] M. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable Physics and Stable Modes for Tool-Use and Manipulation Planning," *Robot. Sci. and Syst.*, pp. 1–9, 2018.

[75] F. R. Hogan, E. R. Grau, and A. Rodriguez, "Reactive Planar Manipulation with Convex Hybrid MPC," in *Int. Conf. on Robot. and Autom.*, 2018, pp. 247–253.

[76] D. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with Coulomb friction," *Int. J. for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[77] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *IEEE-RAS Int. Conf. on Humanoid Robots*, nov 2014, pp. 279–286.

[78] J. Carpentier, R. Budhiraja, and N. Mansard, "Learning Feasibility Constraints for Multicontact Locomotion of Legged Robots," in *Robotics, Sci. and Syst. (RSS)*, 2017.

[79] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *IEEE Int. Conf. on Robot. and Autom.*, 2013.

[80] D. W. Marhefka and D. E. Orin, "A compliant contact model with nonlinear damping for simulation of robotic Syst." *IEEE Trans. on Syst., Man, and Cybernetics Part A:Syst. and Humans.*, 1999.

[81] R. Tedrake and the Drake Team Development, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical Syst." 2016.

[82] U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, 1998.

[83] K. H. Hunt and F. R. E. Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *J. of Applied Mechanics*, 1975.

[84] E. Todorov, "Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in MuJoCo," in *IEEE Int. Conf. on Robot. and Autom.*, 2014.

[85] A. J. Van Der Schaft and J. M. Schumacher, *An introduction to hybrid dynamical Syst.* Springer London, 2000, vol. 251.

[86] D. E. Stewart, "Rigid-body dynamics with friction and impact," *SIAM Review*, 2000.

[87] B. Brogliato, *Nonsmooth mechanics : models, dynamics, and control.* Springer, 1999.

[88] M. Anitescu and F. A. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dynamics*, vol. 14, no. 3, pp. 231–247, 1997.

[89] C. D. Remy, "Ambiguous collision outcomes and sliding with infinite friction in models of legged Syst." *Int. J. of Robot. Res.*, 2017.

[90] M. Halm and M. Posa, "Modeling and Analysis of Non-unique Behaviors in Multiple Frictional Impacts," in *Robotics: Science and Systems (RSS)*, Freiburg im Breisgau, Germany, 2019.

[91] A. Saccon, N. Van De Wouw, and H. Nijmeijer, "Sensitivity analysis of hybrid Syst. with state jumps with application to trajectory tracking," in *IEEE Conf. on Decision and Control*, 2014.

[92] S. A. Burden, S. S. Sastry, D. E. Koditschek, and S. Revzen, "Event-selected vector field discontinuities yield piecewise-differentiable flows," *SIAM J. on Applied Dynamical Syst.*, 2015.

[93] E. Heiden, D. Millard, E. Coumans, Y. Sheng, and G. S. Sukhatme, "Neuralsim: Augmenting differentiable simulators with neural networks," in *IEEE Int. Conf. on Robot. and Autom.*, 2021, pp. 9474–9481.

[94] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX," in *IEEE Int. Conf. on Robot. and Autom.*, 2015.

[95] N. Fazeli, S. Zapolsky, E. Drumwright, and A. Rodriguez, "Fundamental Limitations in Performance and Interpretability of Common Planar Rigid-Body Contact Models," in *Int. Symp. of Robotic Res.*, 2017.

[96] B. Acosta, W. Yang, and M. Posa, "Validating robot. simulators on real-world impacts," *IEEE Robot. and Autom. Lett.*, vol. 7, no. 3, pp. 6471–6478, 2022.

[97] K. Byl and R. Tedrake, "Approximate optimal control of the compass gait on rough terrain," in *IEEE Int. Conf. on Robot. and Autom.*, 2008.

[98] P. A. Bhounsule, "Control of a compass gait walker based on energy regulation using ankle push-off and foot placement," 2015.

[99] R. Bordalba, T. Schoels, L. Ros, J. M. Porta, and M. Diehl, "Direct collocation methods for trajectory optimization in constrained robotic systems," *IEEE Transactions on Robotics*, 2022.

[100] D. Song, P. Fernbach, T. Flayols, A. D. Prete, N. Mansard, S. Tonneau, and Y. J. Kim, "Solving Footstep Planning as a Feasibility Problem Using L1-Norm Minimization," *IEEE Robot. and Autom. Lett.*, vol. 6, no. 3, pp. 5961–5968, 2021.

[101] S. Tonneau, D. Song, P. Fernbach, N. Mansard, M. Taix, and A. Del Prete, "SL1M: Sparse L1-norm Minimization for contact planning on uneven terrain," in *IEEE Int. Conf. on Robot. and Autom.*, 2020.

[102] N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux, and E. Yoshida, "Fast humanoid robot collision-free footstep planning using swept volume approximations," *IEEE Trans. on Robot.*, 2012.

[103] S. Tonneau, A. Del Prete, J. Pettre, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *IEEE Trans. on Robot.*, vol. 34, no. 3, 2018.

[104] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, "End-to-end differentiable physics for learning and control," *Advances in neural information processing systems*, vol. 31, 2018.

[105] S. Pfrommer, M. Halm, and M. Posa, "ContactNets: Learning of Discontinuous Contact Dynamics with Smooth, Implicit Representations," in *The Conf. on Robot Learning*, 2020.

[106] R. Deits, T. Koolen, and R. Tedrake, "Lvis: Learning from value function intervals for contact-aware robot controllers," in *2019 Int. Conf. on Robot. and Autom.*, 2019, pp. 7762–7768.

[107] A. Cauligi, P. Culbertson, E. Schmerling, M. Schwager, B. Stellato, and M. Pavone, "Coco: Online mixed-integer control via supervised learning," *IEEE Robot. and Autom. Lett.*, vol. 7, no. 2, pp. 1447–1454, 2021.

[108] J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-real learning of all common bipedal gaits via periodic reward composition," in *IEEE Int. Conf. on Robot. and Autom.*, 2021, pp. 7309–7315.

[109] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Sci. Robot.*, vol. 7, no. 62, p. eabk2822, 2022.

[110] H. J. T. Suh, T. Pang, and R. Tedrake, "Bundled gradients through contact via randomized smoothing," *IEEE Robot. and Autom. Lett.*, vol. 7, no. 2, pp. 4000–4007, 2022.

[111] P. B. Wieber, "Holonomy and nonholonomy in the dynamics of articulated motion," in *Lecture Notes in Control and Information Sci.s*, 2006.

[112] P. M. Wensing and D. E. Orin, "Improved Computation of the Humanoid Centroidal Dynamics and Application for Whole-Body Control," *Int. J. of Humanoid Robot.*, vol. 13, no. 01, p. 1550039, 2016.

[113] 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.

[114] A. K. Valenzuela, "Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[115] S. Caron, Q. C. Pham, and Y. Nakamura, "Leveraging cone double description for multi-contact stability of humanoids with applications to statics and dynamics," in *Robotics: Sci. and Syst.*, 2015.

[116] T. Bretl and S. Lall, "Testing static equilibrium for legged robots," *IEEE Trans. on Robot.*, 2008.

[117] H. Audren and A. Kheddar, "3-D Robust Stability Polyhedron in Multicontact," *IEEE Trans. on Robot.*, 2018.

[118] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. G. Caldwell, and C. Semini, "Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots," *IEEE Robot. and Autom. Lett.*, 2018.

[119] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A universal stability criterion of the foot contact of legged robots - Adios ZMP," in *IEEE Int. Conf. on Robot. and Autom.*, 2006.

[120] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Trans. on Robot.*, vol. 37, no. 4, pp. 1154–1171, 2021.

[121] 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.

[122] Y. Sim and J. Ramos, "Tello leg: The study of design principles and metrics for dynamic humanoid robots," *IEEE Robot. and Autom. Lett.*, vol. 7, no. 4, pp. 9318–9325, 2022.

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

[124] A. Saccon, S. Traversaro, F. Nori, and H. Nijmeijer, "On centroidal dynamics and integrability of average angular velocity," *IEEE Robot. and Autom. Lett.*, vol. 2, no. 2, pp. 943–950, 2017.

[125] T. Koolen, M. Posa, and R. Tedrake, "Balance control using center of mass height variation: Limitations imposed by unilateral contact," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2016.

[126] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3D-SLIP model," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2013, pp. 5134–5140.

[127] G. Piovan and K. Byl, "Approximation and Control of the SLIP Model Dynamics via Partial Feedback Linearization and Two-Element Leg Actuation Strategy," *IEEE Trans. on Robot.*, 2016.

[128] X. Xiong and A. D. Ames, "Motion Decoupling and Composition via Reduced Order Model optimization for Dynamic Humanoid Walking with CLF-QP based Active Force Control," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2019.

[129] Y. M. Chen and M. Posa, "Optimal Reduced-order Modeling of Bipedal Locomotion," in *IEEE Int. Conf. on Robot. and Autom.*, 2020.

[130] A. Pandala, R. T. Fawcett, U. Rosolia, A. D. Ames, and K. A. Hamed, "Robust predictive control for quadrupedal locomotion: Learning to close the gap between reduced-and full-order models," *IEEE Robot. and Autom. Lett.*, vol. 7, no. 3, pp. 6622–6629, 2022.

[131] Z. Xie, X. Da, B. Babich, A. Garg, and M. van de Panne, "Glide: Generalizable quadrupedal locomotion in diverse environments with a centroidal model," *arXiv preprint arXiv:2104.09771*, 2021.

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

[133] L. T. Biegler and V. M. Zavala, "Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization," *Computers & Chemical Engineering*, vol. 33, no. 3, pp. 575–582, 2009.

[134] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Açıkmeşe, "¡ convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently," *IEEE Control Syst. Magazine*, vol. 42, no. 5, pp. 40–113, 2022.

[135] R. Bellman, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503–515, 1954.

[136] E. C. Whitman and C. G. Atkeson, "Control of instantaneously coupled systems applied to humanoid walking," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2010, pp. 210–217.

[137] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton university press, 2011.

[138] S. Wright, J. Nocedal, and Others, "Numerical optimization," *Springer Sci.*, vol. 35, no. 67-68, p. 7, 1999.

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

[140] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[141] W. W. Hager, H. Hou, S. Mohapatra, and A. V. Rao, "Convergence Rate for an hp Collocation Method Applied to Unconstrained Optimal Control," *J. of Optimization Theory and Applications*, vol. 169, no. 3, pp. 801–824, 2016.

[142] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *J. of guidance, control, and dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[143] O. Von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of operations Res.*, vol. 37, no. 1, pp. 357–373, 1992.

[144] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, 2017.

[145] S. Moreno-Martin, L. Ros, and E. Celaya, "Collocation methods for second order system," in *Robotics: Sci. and Syst.*, 2022.

[146] N. Hoang, "Collocation runge–kutta–nyström methods for solving second-order initial value problems," *Int. J. of Computer Mathematics*, vol. 98, no. 12, pp. 2423–2444, 2021.

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

[148] D. M. Murray and S. J. Yakowitz, "Differential dynamic programming and Newton's method for discrete optimal control problems," *J. of Optimization Theory and Applications*, vol. 43, no. 3, pp. 395–414, jul 1984.

[149] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic Syst." in *American Control Conference*, 2005.

[150] J. N. Nganga and P. M. Wensing, "Accelerating second-order differential dynamic programming for rigid-body systems," *IEEE Robot. and Autom. Lett.*, vol. 6, no. 4, pp. 7659–7666, 2021.

[151] S. Singh, R. P. Russell, and P. M. Wensing, "Analytical second-order partial derivatives of rigid-body inverse dynamics," in *2022 IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2022.

[152] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *IEEE Int. Conf. on Robot. and Autom.*, 2014.

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

[154] Z. Xie, C. K. Liu, and K. Hauser, "Differential dynamic programming with nonlinear constraints," *IEEE Int. Conf. on Robot. and Autom.*, pp. 695–702, 2017.

[155] T. A. Howell, B. E. Jackson, and Z. Manchester, "ALTRO: A Fast Solver for Constrained Trajectory Optimization," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2019.

[156] E. Pellegrini and R. P. Russell, "A Multiple-Shooting Differential Dynamic Programming Algorithm," *AAS/AIAA Space Flight Mechanics Meeting*, no. March, pp. 759–778, 2017.

[157] B. Plancher and S. Kuindersma, "A performance analysis of parallel differential dynamic programming on a gpu," in *Int. Workshop on the Algorithmic Foundations of Robot.* Springer, 2018, pp. 656–672.

[158] M. Giftthaler, M. Neunert, M. Stäuble, J. Buchli, and M. Diehl, "A family of iterative gauss-newton shooting methods for nonlinear optimal control," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2018, pp. 1–9.

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

[160] G. Frison and M. Diehl, "Hpipm: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[161] O. Melon, R. Orsolino, D. Surovik, M. Geisert, I. Havoutis, and M. Fallon, "Receding-horizon perceptive trajectory optimization for dynamic legged locomotion with learned initialization," in *IEEE Int. Conf. on Robot. and Autom.*, 2021, pp. 9805–9811.

[162] 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 *IEEE Int. Conf. on Robot. and Autom.*, 2018, pp. 2986–2993.

[163] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, "Plan online, learn offline: Efficient learning and exploration via model-based control," in *Int. Conf. on Learning Representations*, 2018.

[164] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control," *IEEE Trans. on Robot.*, 2022.

[165] C. Samson, B. Espiau, and M. L. Borgne, *Robot control: the task function approach*. Oxford University Press, Inc., 1991.

[166] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. on Robot. and Autom.*, vol. 3, no. 1, pp. 43–53, 1987.

[167] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

[168] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J. Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Trans. on Robot.*, 2013.

[169] D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette, "Visual Servoing in an Optimization Framework for the Whole-Body Control of Humanoid Robots," *IEEE Robot. and Autom. Lett.*, 2017.

[170] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based full body control for the DARPA Robot. Challenge," *J. of Field Robot.*, 2015.

[171] H. Audren, A. Kheddar, and P. Gergondet, "Stability polygons reshaping and morphing for smooth multi-contact transitions and force control of humanoid robots," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2016.

[172] M. de Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Trans. on Graphics*, 2010.

[173] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *IEEE Int. Conf. on Robot. and Autom.*, 2013.

[174] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, 2016.

[175] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic Programming for Multirobot and Task-Space Force Control," *IEEE Trans. on Robot.*, 2019.

[176] A. D. Ames and M. Powell, "Towards the Unification of Locomotion and Manipulation through Control Lyapunov Functions and Quadratic Programs," in *Lecture Notes in Control and Information Sci.s*, 2013.

[177] K. C. Park, P. H. Chang, and S. H. Kim, "The Enhanced Compact QP Mathod for Redundant Manipulators Using Practical Inequality Constraints," in *IEEE Int. Conf. on Robot. and Autom.*, 1998.

[178] F. Kanehiro, M. Morisawa, W. Suleiman, K. Kaneko, and E. Yoshida, "Integrating geometric constraints into reactive leg motion generation," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2010.

[179] Q. Nguyen and K. Sreenath, "Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, 2016.

[180] M. Rauscher, M. Kimmel, and S. Hirche, "Constrained Robot Control Using Control Barrier Functions," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2016.

[181] W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter, "Extending iTaSC to support inequality constraints and non-instantaneous task specification," in *IEEE Int. Conf. on Robot. and Autom.*, 2009.

[182] S. Rubrecht, V. Padois, P. Bidaud, M. Broissia, and M. Da Silva Simoes, "Motion safety and constraints compatibility for multibody robots," *Autonomous Robots*, vol. 32, no. 3, pp. 333–349, 2012.

[183] A. Del Prete, "Joint Position and Velocity Bounds in Discrete-Time Acceleration / Torque Control of Robot Manipulators," *IEEE Robot. and Autom. Lett.*, vol. 3, no. 1, 2018.

[184] N. Mansard, "A dedicated solver for fast operational-space inverse dynamics," in *IEEE Int. Conf. on Robot. and Autom.*, 2012.

[185] J. Park and O. Khatib, "Contact consistent control framework for humanoid robots," in *IEEE Int. Conf. on Robot. and Autom.*, 2006.

[186] L. Sentis, J. Park, and O. Khatib, "Compliant control of multicontact and center-of-mass behaviors in humanoid robots," *IEEE Trans. on Robot.*, 2010.

[187] L. Righetti, J. Buchli, M. Mistry, and S. Schaal, "Inverse dynamics control of floating-base robots with external constraints: A unified view," in *2011 IEEE international conference on robotics and automation*, 2011, pp. 1085–1090.

[188] D. Dimitrov, P.-B. Wieber, and A. Escande, "Multi-Objective Control of Robots," *J. of the Robot. Society of Japan*, 2014.

[189] Fujimoto Y., Obata S., and Kawamura A., "Robust Biped Walking with Active Interaction Control between Foot and Ground," in *IEEE Int. Conf. on Robot. and Autom.*, 1998.

[190] K. Bouyarmane and A. Kheddar, "Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations," in *IEEE Int. Conf. on Intell. Robots and Syst.*, 2011.

[191] S. H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Autonomous Robots*, 2012.

[192] T. Koolen, et. al, "Summary of Team IHMC's Virtual Robot. Challenge Entry," *IEEE-RAS Int. Conf. on Humanoid Robots*, 2013.

[193] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[194] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[195] M. Chignoli and P. M. Wensing, "Variational-based optimal control of underactuated balancing for dynamic quadrupeds," *IEEE Access*, vol. 8, pp. 49 785–49 797, 2020.

[196] J. Englsberger, A. Dietrich, G. Mesesan, G. Garofalo, C. Ott, and A. Albu-Schaeffer, "MPTC – Modular Passive Tracking Controller for stack of tasks based control frameworks," 2020.

[197] W. Yang and M. Posa, "Impact invariant control with applications to bipedal locomotion," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2021, pp. 5151–5158.

[198] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini, "Stance: Locomotion adaptation over soft terrain," *IEEE Trans. on Robot.*, vol. 36, no. 2, pp. 443–457, 2020.

[199] T. Flayols, A. Del Prete, M. Khadiv, N. Mansard, and L. Righetti, "Reactive Balance Control for Legged Robots under Visco-Elastic Contacts," *Applied Sci.s*, vol. 11, no. 353, 2021.

[200] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality Syst.: Application to local motion planning for redundant robots," in *IEEE Int. Conf. on Robot. and Autom.*, 2009.

[201] A. Escande, N. Mansard, and P. B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *Int. J. of Robot. Res.*, 2014.

[202] D. Dimitrov, A. Sherikov, and P.-B. Wieber, "Efficient resolution of potentially conflicting linear constraints in robotics," Aug. 2015.

[203] C. Van Loan, "On the Method of Weighting for Equality-Constrained Least-Squares Problems," *SIAM J. on Numerical Analysis*, 1985.

[204] K. Pfeiffer, A. Escande, and A. Kheddar, "Singularity Resolution in Equality and Inequality Constrained Hierarchical Task-Space Control by Adaptive Nonlinear Least Squares," *IEEE Robot. and Autom. Lett.*, 2018.

[205] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[206] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," *IEEE/SICE Int. Symp. on System Integration*, pp. 614–619, 2019.

[207] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[208] 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 *IEEE-RAS Int. Conf. on Humanoid Robots*, 2021, pp. 61–68.

[209] Q. L. Lidec, L. Montaut, C. Schmid, I. Laptev, and J. Carpentier, "Augmenting differentiable physics with randomized smoothing," *arXiv preprint arXiv:2206.11884*, 2022.

[210] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Foundations and Trends in Machine Learning*, 2015.

[211] S. Levine and V. Koltun, "Guided policy search," in *Int. conference on machine learning*. PMLR, 2013, pp. 1–9.

[212] Z. Zhao, S. Zuo, T. Zhao, and Y. Zhao, "Adversarially regularized policy learning guided by trajectory optimization," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 844–857.

[213] M. Zanon and S. Gros, "Safe reinforcement learning using robust mpc," *IEEE Trans. on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2020.