CAFE-MPC: A Cascaded-Fidelity Model Predictive Control Framework with Tuning-Free Whole-Body Control

He Li and Patrick M. Wensing

***Abstract*—This work introduces an optimization- ning and control framework for real-time synthesi body motions for legged robots. At the core of th framework is a cascaded-fidelity model predictiv (CAFE-MPC). CAFE-MPC strategically relaxes the pla lem along the prediction horizon (i.e., with descen fidelity, increasingly coarse time steps, and relaxed for computational and performance gains. This numerically solved with an efficient customized multi iLQR (MS-iLQR) solver that is tailored for hybr The action-value function from CAFE-MPC is th the basis for a new value-function-based whole-b (VWBC) technique that avoids additional tuning. In the proposed framework unifies whole-body MPC conventional whole-body quadratic programming (have been treated as separate components in previou study the effects of the cascaded relaxations in CAFE- tracking performance and required computation ti show that CAFE-MPC , if configured appropriately, a performance of whole-body MPC without necessaril computational cost. Further, we show the superior p of VWBC over a conventional Riccati feedback c terms of constraint handling. The proposed framew accomplishing a gymnastic-style running barrel roll for the first time on quadruped hardware, where CAFE-MPC runs at 50 Hz, and the solver spends on average 5.3 ms per iteration. Results are incorporated in the accompanying video [1].**

***Index Terms*—Legged Locomotion, Model Predictive Control, Whole-Body Control, Differential Dynamic Programming**

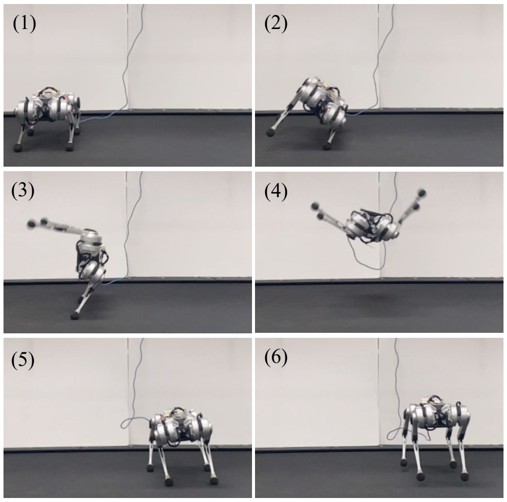


Fig. 1: In-place barrel roll on MIT Mini Cheetah accomplished with the proposed control framework. The robot performs an in-place barrel roll, followed by a hopping step, and a pacing gait. All transitions are fully synthesized online. The results section includes more challenging tasks where the robot synthesized a barrel roll in the middle of running.

I. INTRODUCTION

UNLOCKING biological-level mobility on legged robots is helpful to understand and discover their full potential for applications. Significant progress has been made over the past decades on both very capable hardware platforms and advanced control techniques. The MIT Cheetah-series robots demonstrated robust stair climbing skills [1], jumping over obstacles [2], and back-flipping maneuvers [3]. The ETH ANYmal-series robots showed remarkable capabilities with stepping stones [4], and traversing extremely unstructured environments [5]. Researchers from KAIST enabled quadruped robots to walk on vertical walls [6] and deformable soft terrains [7]. The hydraulic-actuated HyQ-series robots were shown to drag a 3-ton airplane [8], and with terrain adaptation capability [9].

Despite the rapid progress, achieving biological levels of mobility on robots remains difficult. Well-trained human professionals can perform parkour and gymnastics that involve significant body rotations (barrel roll, flip, etc.) in the middle of running. This level of mobility is a challenge for robots, and has not been shown on any quadruped platforms in the literature. The challenges are two fold. First, gymnastic-style motions require careful coordination of the whole body. When using a model-based approach, the whole-body dynamics poses a challenge to satisfy real-time computation constraints, due to its non-convexity and high dimension. Second, the controller should be sufficiently flexible and robust to smoothly synthesize different motions (e.g., running, in-air body rotations, etc.), and the transitions between them.

Prior studies have made important progress toward attaining dynamic acrobatic maneuvers on quadruped robots. Offline trajectory optimization (TO) with a detailed dynamics model including the motors was used in [3] to achieve a back flip with the MIT Mini Cheetah. Due to its offline nature, the robot needed to be sufficiently close to a proper initial condition for successful execution. Tuning of the tracking controller was further treated as a fully separate component. Online TO with single-rigid body (SRB) models has been studied [10], [11] to accomplish an in-place barrel roll. Since leg momentum is ignored, the SRB inertia needs to be tuned as a proxy for the leg inertia in order to obtain sufficient take-off velocity. In addition to the model-based method, Li *et al.* [12] sought to learn a back-flipping policy from partial demonstration. Regardless, the acrobatic maneuvers attained in all previous works require intermediate static starting and landing poses.

This work aims to further push the mobility of quadruped

*This work was supported in part by a research gift from Google and through NSF Grant CMMI-2220924 and ONR Award N0001420WX01278 via subawards to the University of Notre Dame.

He Li and Patrick M. Wensing are with the Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (`hli.notredame@gmail.com, pwensing@nd.edu`)

[1] https://youtu.be/v-yOewSXbwA

robots. We propose an optimization-based control framework that takes as its input a reference trajectory obtained from a motion library, and then outputs commands that are directly executable on the robot hardware. Apart from diverse regular locomotion skills such as pacing and bounding, this framework unlocks on-the-fly synthesis of gymnastic running barrel rolls. An in-place barrel roll reference is used, and the controller is capable of adapting it online to several running gaits with different initial contact configurations. The proposed framework makes it easy to create diverse behaviors by simply specifying the references. Figure 1 highlights one of the main results achieved in this work.

The proposed framework is built upon model predictive control (MPC), a method of producing control inputs by predicting the future behavior of the robot as part of solving a TO problem. MPC accounts for model uncertainties and external disturbances by frequently adapting the predicted plan to the most recent state. Whole-body MPC produces a high-fidelity plan that is coherent with the full robot dynamics. However, it is known to be subject to large computation times, leading to a policy delay and prohibiting the use of a long prediction horizon or fast update frequency. For this reason, the capability of this approach to control highly dynamic motions on robot hardware has not yet been demonstrated [13]–[15]. A local feedback Riccati controller [13] that runs at a much higher frequency can be used to account for the policy delay and the slow MPC update. While this local feedback controller admits an analytical solution in the unconstrained case, it is not guaranteed to respect the constraints from the whole-body MPC. A notable strategy to alleviate the computational burden is to instead use a simplified template model, thus template MPC. A low-level controller that converts the low-dimensional plan to whole-body commands is needed, often an inverse-dynamics quadratic program (QP). This strategy is thus far the most popular MPC approach, and has demonstrated great success for legged locomotion [4], [9], [16]–[20]. However, the omitted model details and the possibly added artificial constraints may result in infeasible or over-conservative target plans [21]. As a result, nontrivial tuning of the whole-body QP is often needed. The proposed framework introduces a novel approach that accelerates the whole-body MPC, and a novel low-level controller that integrates the Riccati controller and the whole-body QP.

A. Contributions

The contributions of this work are summarized as follows. The first contribution is a cascaded-fidelity model predictive controller (CAFE-MPC). CAFE-MPC generalizes our previous MPC formulation ideas [22] beyond model cascades, exploring the use of increasingly coarse integration time steps and progressively relaxed constraints along the prediction horizon. The second contribution is to extend multiple-shooting iLQR [23] to a class of hybrid systems whose phase sequence and timings are fixed. Specifically, we focus on how value function approximations can be back-propagated and how shooting nodes are updated across the switching surfaces. The third contribution is a value-function-based whole-body controller (VWBC). The VWBC employs a local action-value function as its minimization objective, ensuring a direct link between MPC and the WBC objectives. This makes it tuning-free, as opposed to the non-trivial additional cost design needed for conventional WBCs. The resulting scheme of CAFE-MPC + VWBC provides a structure that unifies whole-body MPC and whole-body QPs, which were conventionally implemented as separate designs.

The last contribution is the overall optimization-based control framework. Beyond regular locomotion skills, this framework unlocks real-time synthesis of gymnastic-style motions, by simply specifying an input reference motion without further tuning of the system parameters. With the developed framework, we show a quadruped robot can achieve a barrel roll in the middle of running. To the best of our knowledge, this is the first time a running barrel roll has been accomplished on quadruped hardware. Figure 1 highlights the result of an in-place barrel roll on the MIT Mini Cheetah, while more challenging tasks are described in Section VII-C. Our code is open-sourced [2]. We hope it can be a helpful companion for readers and a useful resource for the broader community.

B. System Overview and Outline

An overview of the system architecture is shown in figure 2. The overall control framework takes as input a reference trajectory, and outputs the whole-body commands (joint torques, angles, and velocities) which are directly executable on the robot. For regular locomotion skills like trotting and bounding, simple heuristic reference trajectories are sufficient. For combined motions with the barrel roll, a long-horizon TO is solved offline to provide a more detailed motion sketch that can then be generalized to new situations. For example, the in-place barrel roll reference can be combined with a pacing reference, with the online synthesis of a running barrel roll then left to our framework. The CAFE-MPC runs at 33-50 Hz, and the VWBC refines the MPC command at 500 Hz. For both offline TO and online MPC, the customized MS-iLQR is employed as the underlying numerical solver.

The rest of the paper is structured as follows. In Section II, we summarize related works that are associated with the core components of figure 2. Section III first reviews our previous work on MS-iLQR, and discusses its extension to constrained multi-phase optimal control problems. In Section IV, we present the detailed formulation of CAFE-MPC. Section V proposes the novel VWBC, and discusses its relation with conventional controllers. In Section VI, we present the heuristic reference generation, offline TO for the barrel roll, and other details used to complete the blocks in figure 2. Section VII discusses the simulation and hardware results. Section VIII discusses limitations, and concludes the paper with suggestions for future work.

II. RELATED WORK

A. MPC for Legged Robots

MPC provides a means of controlling robotic systems by predicting future behaviors. The expected behaviors (e.g.,

[2] https://github.com/ROAM-Lab-ND/CAFE-MPC

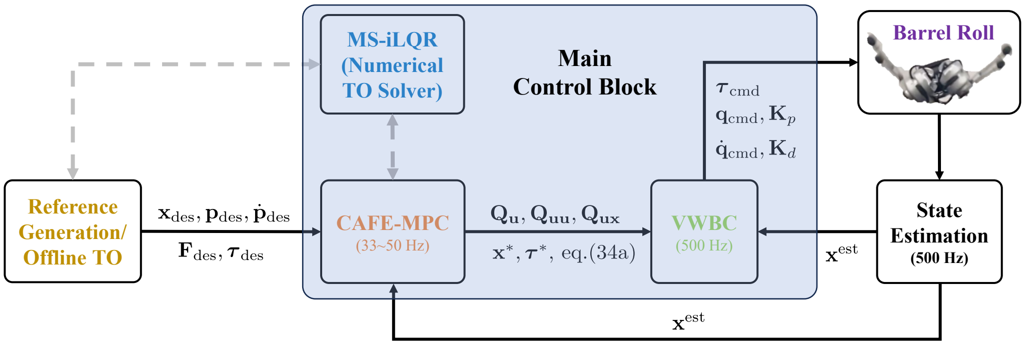

MS-iLQR
(Numerical
TO Solver)
Main
Control Block
Barrel Roll
$\boldsymbol{\tau}_{\mathrm{cmd}}$
$\mathbf{q}_{\mathrm{cmd}}, \mathbf{K}_p$
$\dot{\mathbf{q}}_{\mathrm{cmd}}, \mathbf{K}_d$
Reference
Generation/
Offline TO
$\mathbf{x}_{\mathrm{des}}, \mathbf{p}_{\mathrm{des}}, \dot{\mathbf{p}}_{\mathrm{des}}$
$\mathbf{F}_{\mathrm{des}}, \boldsymbol{\tau}_{\mathrm{des}}$
CAFE-MPC
(33~50 Hz)
$\mathbf{Q}_{\mathbf{u}}, \mathbf{Q}_{\mathbf{uu}}, \mathbf{Q}_{\mathbf{ux}}$
$\mathbf{x}^*, \boldsymbol{\tau}^*$, eq.(34a)
VWBC
(500 Hz)
$\mathbf{x}^{\mathrm{est}}$
State
Estimation
(500 Hz)
$\mathbf{x}^{\mathrm{est}}$

Fig. 2: An overview of the system architecture. The proposed control framework takes a reference trajectory as input, and outputs commands that are directly executable on the robot. The main control block consists of the CAFE-MPC, the customized MS-iLQR solver for numerical optimization, and the VWBC. The MS-iLQR solver is used for offline TO as well. The MPC shares the same cost function for all tasks, and the VWBC is tuning-free.

maintaining balance) are achieved via minimizing an objective function subject to some modeled dynamics and constraints, which are often nonlinear. The capability of MPC to cope with nonlinear dynamics and constraints makes it well-suited and increasingly popular for the control of legged robots, as seen by a growing body of literature [4], [9], [16], [24]–[28].

The classical template-MPC approach generates plans of low dimensions. Therefore, a low-level controller is needed to (1) produce the whole-body commands (2) provide fast feedback control for stabilization. One such notable controller is the inverse-dynamics QP [29]. Popular template models include the Linear Inverted Pendulum Model (LIP) [24], the Spring-Loaded Inverted Pendulum Model (SLIP) [30], Single-Rigid-Body Model [16], [31], [32], and the centroidal model [4], [33]–[36]. Template MPC has the advantage of fast computation due to the relatively small and possibly convex optimization problems. One limitation of this approach, however, is that the operational envelope of the resulting motions either underestimates or overestimates the set of whole-body feasible motions. As a result, the resulting target plan may not be tractable by the low-level whole-body controller. A recent survey paper provides a more detailed overview of this perspective [21].

The whole-body MPC approach does not have the problem of producing infeasible trajectories of the template MPC. This approach, however, is notoriously known for its computational burden, due to the increased dimensions, non-linearity, and non-convexity. For these reasons, numerous efforts have been made to accelerate the whole-body MPC by developing efficient structure-exploiting solvers [14], [37]–[40], fast analytical dynamics [41], [42], and by using inverse dynamics [13]. Several works have shown the success of this approach on robot hardware with regular locomotion skills [13]–[15]. However, highly dynamic behaviors have yet been demonstrated on any hardware platform.

Our prior work on Model Hierarchy Predictive Control (MHPC) [22] combines the benefits of whole-body MPC and template MPC, by placing a whole-body model in the near term and a template model in the distant term. This similar idea was further explored in several follow-up works [43]–[45] with different models that were tailored towards humanoid robots. While the effect of model schedule on disturbance rejection is studied in [22], Kahzoom *et al.* [46] proposed a method to optimize the model schedule. Following a related but distinct idea, Norby *et al.* [47] fixed the prediction horizon while adapting the model fidelity based on the task complexity. The CAFE-MPC proposed in this work further relaxes the template plan using coarse timesteps and relaxed constraints. The motivation is that the template model has slower dynamics (e.g., due to ignoring fast swinging appendages), and the constraints in the distant end are less critical to the current decisions. Similar ideas have also been studied beyond the legged robot community, such as for chemical process control [48], with simple mobile robots [49], and with autonomous vehicles [50].

Contact-implicit MPC is another line of compelling research for legged locomotion, which enables simultaneous generation of the contact modes and the whole-body motions. Prior studies that leverage differentiable contact models have shown the promise of this approach of generating complex multi-contact behaviors [25], [51]–[53]. This approach, however, is known to have numerical problems such as the difficulty to obtain good initial guesses, and computational burden. Our work is complementary to the contact-implicit approach, in the sense that the contact-implicit planning could be considered as a low-frequency top layer that informs the CAFE-MPC with contact sequences, and leverages CAFE-MPC for fast online synthesis.

B. Numerical Optimization for MPC

One critical factor for the success of MPC is to reliably and efficiently solve the underlying TO problems. Conventional numerical optimization methods for TO take one of the three approaches [54]: *dynamic programming*, *indirect methods*, and *direct methods*. *Direct methods* are the most widespread in robotics, and they proceed by transcribing the TO problems to a Nonlinear Programming (NLP) problem, which could be effectively solved using well-developed off-the-shelf NLP solvers, for instance, SNOPT [55] or IPOPT [56]. Most NLP solvers proceed by taking successive linearization of the first-order necessary conditions for optimality (i.e., the KKT conditions), resulting in complexity of $O(N^3)$ (in the worst case) where N is the problem size. Consequently, this approach is prohibitive for online use with robots where N is large. Fortunately, some *direct methods* such as multiple shooting and direct collocation result in sparse NLPs whose computation complexity could be reduced to $O(N)$ [57]. For this reason and thanks to the development of computation hardware, *direct methods* have been employed by some research groups to successfully solve nonlinear MPC for legged robots [4], [26], [27], [40]. It was shown in [4] that the nonlinear MPC can run up to 100 Hz with a real-time iteration scheme [58] and an interior-point-method (IPM) based QP solver [59].

Differential Dynamic Programming (DDP) [60] (also known as iLQR when adopting a Gauss-Newton Hessian approximation [25]) is a powerful tool for nonlinear optimal control and has gained increasing attention in the past decade in the robotics community. It successively solves a sequence of small sub-problems leveraging Bellman's equation. Similar to the structure-exploiting sparse NLP solvers [39], [40], [59], DDP has a linear computational cost relative to the prediction horizon. However, it comes with a value function approximation and a local feedback policy for free as intermediate results, which can be used for higher-rate lower-level control [13], [61]. These properties make DDP/iLQR well-suited for MPC of legged robots.

Many research groups have made efforts toward this direction. Tassa *et al.* achieved complex motion control with iLQR on a simulated humanoid robot, with slow simulation speed [25]. Koenemann *et al.* implemented the same framework on the humanoid robot HRP-2 [62]. Though the task was simple (balanced standing), it was the first time whole-body MPC was achieved on humanoid robot hardware. Neunert *et al.* accomplished locomotion control on two high-performance quadruped robots, ANYmal and HyQ, in simulation [63] and on hardware [37], where iLQR served as the underlying solver. Though promising, there are several aspects where opportunities for improvement are clearly seen. Effective constraint handling and sensitivity to initial guesses have been the major bottlenecks. A large amount of recent works thus have contributed to addressing these problems. To deal with constraints, DDP/iLQR is combined with many common algorithms in the numerical optimization community, for instance, the Projected Newton method [64], penalty method [65], barrier method [61] and interior-point method [66], Augmented Lagrangian (AL) method (e.g., [67]–[69]) and primal-dual AL [15]. To deal with the sensitivity problem, DDP /iLQR is extended with multiple-shooting formulations that permit an infeasible warm start with a state trajectory. [15], [23], [38], [67], [70].

C. Whole-Body Control

The whole-body controller is an important component in many MPC-based control frameworks. It operates at a higher loop frequency and produces commands that are directly executable on the robot hardware. Multiple WBC techniques have been developed in the literature. A classical approach is to formulate WBC as a QP problem [4], [28], [29], [71], where the objective function incorporates multiple tasks related to operational space control [72], thus denoted as OSC-QP. These tasks are designed to track certain references at the acceleration level in the least-square sense, such as the center of mass (CoM) and swing foot accelerations, etc. It was shown in [73] that the dynamic capability of a quadruped can be significantly improved by incorporating the ground reaction force (GRF) in the objective. The OSC-QP reasons about the whole-body dynamics, and enforces constraints such as torque limits, non-slipping constraints, etc. As such, it ensures physical feasibility of the control actions. The OSC-QP requires non-trivial tuning, which arises from three aspects. First, the relative importance between tasks is based on heuristics. Second, the tasks are often designed using simplified models, which are inconsistent with the whole-body dynamics in the OSC-QP. Third, the OCS-QP is designed for a short term (one control step). The long-term stability of the OSC-QP relies on the quality of the reference, which is regulated by a PD controller, and requires further tuning.

The Riccati controller is a less-known but powerful WBC technique. The feedback gains can be computed either via DDP/iLQR [13], [61], [74], or via multiple-shooting based numerical optimization [4], though the ideas are similar where the difference Riccati equations are computed. The Riccati controller can either be directly used if the MPC reasons about whole-body dynamics [13], or an additional layer is needed to produce the whole-body actuation if the MPC reasons about simplified models [61], [74]. A known problem of this approach is that the feedback control does not necessarily satisfy constraints, which can interfere with robot stability, for instance, when the friction cone constraints are violated.

The VWBC proposed in this work overcomes the issues of the conventional OSC-QP and the Riccati controller. The VWBC is not significantly different from OSC-QP in terms of formulation. The novelty is that it employs an objective function that considers the long-term cost-to-go (value function). This value function is a proxy of the long-term stability, and is readily available from the top-level CAFE-MPC, thus leaving the VWBC free of additional tuning. The VWBC generalizes the Riccati controller to include constraints, thereby unifying whole-body MPC with conventional WBC.

The works most relevant to VWBC in the literature are [71], [75]. The value functions in these works are approximated based on simplified models, which reflect the long-term balance only in a low-dimensional state space. Consequently, additional tracking terms and regularization terms are needed to produce whole-body coordination. The value function used by the VWBC in this work, however, reasons about the whole-body model, which does not require additional cost terms.

D. RL for Legged Robots

Reinforcement Learning (RL) is yet another powerful technique for the control of legged robots. Instead of performing heavy online optimization as in MPC, RL trains a control policy offline in simulated environments. The control policy can be made very robust (either in the sense of sim-to-real transfer or uncertain environments) by injecting noise during training into the robot dynamics, the environments, and the sensory information (encode measurements, depth images, etc). As such, great success has been shown with RL that enables quadruped robots to traverse extremely unstructured environments [5], [76], imitate animal locomotion behaviors [77], run on slippery ground [78], and walk over deformable terrains [7].

Despite the remarkable progress of RL on robust quadruped locomotion, it remains challenging to achieve highly dynamic animal-like behaviors on hardware. Many researchers thus have shifted gears towards agile quadruped locomotion. Recent works [79], [80] accomplished parkour-like maneuvers with RL on quadruped robots. Though impressive versatility and agility are achieved, the gap remains in terms of dynamic capabilities when compared to the jumping-over-obstacle behaviors previously achieved with the optimization-based approach [2], not to mention biological-level mobility. Other works show fast locomotion [81] and continuously dynamic jumping [82], which focus on regular locomotion gaits, while versatility and agility remain to be explored.

A recent work [83] shows the promise of advancing RL using MPC. Though a pure RL policy enables robust locomotion, precision (such as foot placements) has been a problem. It was shown in [83] that both robustness and precision can be achieved by guiding RL with MPC-planned foot placements. Since our work advances MPC, it can potentially be used to advance learning-based frameworks such as [83] as well.

III. Multiple-Shooting Differential Dynamic Programming

Cafe-Mpc considers multiple dynamics phases along the planning horizon due to the cascaded model fidelity and the change of contact status, resulting in a hybrid system. We assume this hybrid system has a fixed phase (dynamics) sequence and timings. Optimal control over such hybrid systems can be modeled by a multi-phase TO problem [68]. In this section, we present an efficient customized multiple-shooting DDP (MS-DDP) solver for generic multi-phase TO problems. We then discuss how to construct a multi-phase TO problem from Cafe-Mpc in Section IV.

Let n_p denote the number of phases, and i denote the phase index. We model such systems in discrete time as below [84]

$$\begin{cases} \mathbf{x}_{k+1}^{[i]} &= \mathbf{f}_i(\mathbf{x}_k^{[i]}, \mathbf{u}_k^{[i]}) \\ \mathbf{x}_0^{[i+1]} &= \mathbf{P}_i(\mathbf{x}_{N_i}^{[i]}) \end{cases}, \tag{1}$$

where $\mathbf{x}$ denotes the state variable, $\mathbf{u}$ the control variable, $\mathbf{f}_i(\cdot,\cdot)$ the phase dynamics, and $\mathbf{P}_i(\cdot)$ is the reset map from the current phase to the next phase, N_i denotes the number of time steps of the i^{th} phase. Note that the dimensions of $\mathbf{x}$ and $\mathbf{u}$ may vary between phases. A constraint is applied at the end of each phase

$$\mathbf{g}_i(\mathbf{x}_{N_i}^{[i]}) = 0. \tag{2}$$

The switching constraint (2) (also known as phase terminal constraint [68]) encodes state-based switching. It requires that the trigger of the reset map is conditioned on satisfying the constraint (2). For legged robots, for instance, the impact dynamics is triggered only when a foot touches the ground (foot height is zero). Without Eq. (2), the system (1) represents a time-based switched system. Details can be found in [84].

Giftthaler *et al.* [70] and Mastalli *et al.* [38] introduced two seminal implementations of MS-DDP. Our previous work [23] provides a framework that unifies these previous methods and offers a few more advancements. Nevertheless, these past implementations were either presented for single-phase systems [70], or were not designed to cope with state-based switching constraints. In this section, we advance the MS-DDP for TO of hybrid systems (with fixed phase sequence and time), specifically dealing with the reset maps and the constraints. We first review the background on MS-DDP for single-phase TO. We then discuss how to incorporate the reset maps in MS-DDP for unconstrained multi-phase TO, and finally introduce methods to deal with the constraints.

A. MS-DDP for Single-Phase TO

A single-phase TO problem has the form

$$\min_{\mathbf{U},\mathbf{X}} \quad J(\mathbf{X},\mathbf{U}) = \sum_{k=0}^{N-1} \ell_k(\mathbf{x}_k,\mathbf{u}_k) + \phi(\mathbf{x}_N) \tag{3a}$$

$$\text{subject to} \quad \mathbf{f}(\mathbf{x}_k,\mathbf{u}_k) - \mathbf{x}_{k+1} = 0 \tag{3b}$$

where k ($0 \leq k \leq N$) denotes the time index with N the length of the predicted trajectory, $\mathbf{u}_k \in \mathbb{R}^m$ is the control variable, $\mathbf{x}_k \in \mathbb{R}^n$ denotes the state variable, $\mathbf{U} = \{\mathbf{u}_k\}_{k=0}^{N-1}$ and $\mathbf{X} = \{\mathbf{x}_k\}_{k=0}^{N}$ respectively, stack the controls and states along the trajectory, $\ell_k(\mathbf{x}_k,\mathbf{u}_k)$ and $\phi(\mathbf{x}_N)$ respectively are the running cost and the terminal cost, and $\mathbf{f}(\mathbf{x}_k,\mathbf{u}_k)$ is the system dynamics. In Eq. (3), we omit the dependency on the phase index i for clarity as it is a single-phase problem.

In the multiple-shooting setting, both $\mathbf{X}$ and $\mathbf{U}$ are decision variables, whereas in the single-shooting setting, $\mathbf{U}$ is the decision variable, and $\mathbf{X}$ depends on $\mathbf{U}$ via system dynamics. Denote $\bar{\mathbf{X}}$, $\bar{\mathbf{U}}$ the current estimates for the optimal solution (also known as the nominal values) of $\mathbf{X}$ and $\mathbf{U}$. The simulated state $\mathbf{f}(\bar{\mathbf{x}}_k,\bar{\mathbf{u}}_k)$ and the estimate $\bar{\mathbf{x}}_{k+1}$ are likely different. This difference is also known as the defect [38], [70], and is defined as

$$\bar{\mathbf{d}}_{k+1} := \mathbf{f}(\bar{\mathbf{x}}_k,\bar{\mathbf{u}}_k) - \bar{\mathbf{x}}_{k+1}. \tag{4}$$

Giftthaler [70] proposed a flexible formulation that allows a subset of $\mathbf{X}$ to be independent while keeping the rest dependent variables. These independent variables are called shooting states, the indices of which are collected to a set $\mathbb{M}$. These dependent variables are called roll-out states, the indices of which are collected to the complementary set $\bar{\mathbb{M}}$. If $k \in \mathbb{M}$, then $\bar{\mathbf{x}}_k$ is seeded from an initial guess. Otherwise, it is overwritten by the simulated state $\mathbf{f}(\bar{\mathbf{x}}_{k-1},\bar{\mathbf{u}}_{k-1})$. MS-DDP iteratively improves $(\bar{\mathbf{X}},\bar{\mathbf{U}})$ until the defects are sufficiently small and the cost function is minimized. At each iteration, it performs a backward sweep followed by a forward sweep within a line search process.

1) Backward Sweep: Leveraging Bellman's principle of optimality for discrete-time systems, the backward sweep performs a one-step optimization at every time step along the nominal trajectory, which produces a local optimal control policy. Denote $(\delta\mathbf{x},\delta\mathbf{u})$ small perturbations to $(\bar{\mathbf{x}},\bar{\mathbf{u}})$, and $\boldsymbol{\pi}$

the local optimal control policy. The one-step optimization performs

$$\boldsymbol{\pi}_k(\bar{\mathbf{x}}_k + \delta\mathbf{x}_k) = \arg\min_{\delta\mathbf{u}_k} \big(\underbrace{\delta\ell_k(\delta\mathbf{x}_k, \delta\mathbf{u}_k) + v_{k+1}(\delta\mathbf{x}_{k+1})}_{Q_k(\delta\mathbf{x}_k, \delta\mathbf{u}_k)}\big) \tag{5}$$

where $\delta\ell_k(\cdot,\cdot)$ is the variation of $\ell_k(\cdot,\cdot)$ in the neighborhood of $(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k)$ due to $(\delta\mathbf{x}_k, \delta\mathbf{u}_k)$, v_{k+1} is the local value function (optimal cost-to-go) approximation for the perturbed state at time $k+1$, and $Q_k(\cdot,\cdot)$ denotes the local action-value function at time k. The function $Q_k(\cdot,\cdot)$ cannot generally be represented in closed form, and thus is approximated to the second order. Temporarily omitting the subscript k, we have

$$Q(\delta\mathbf{x}, \delta\mathbf{u}) \approx$$
$$\frac{1}{2}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\end{bmatrix}^\top \begin{bmatrix}\mathbf{Q}_{\mathbf{xx}} & \mathbf{Q}_{\mathbf{ux}}{}^\top\\ \mathbf{Q}_{\mathbf{ux}} & \mathbf{Q}_{\mathbf{uu}}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\end{bmatrix} + \mathbf{Q}_{\mathbf{x}}{}^\top\delta\mathbf{x} + \mathbf{Q}_{\mathbf{u}}{}^\top\delta\mathbf{u} \tag{6}$$

where $\mathbf{Q}_{\mathbf{uu}}$ and $\mathbf{Q}_{\mathbf{ux}}$ are second-order partials, and $\mathbf{Q}_{\mathbf{x}}$ and $\mathbf{Q}_{\mathbf{u}}$ are first-order partials. Performing the minimization in Eq. (5) with Eq. (6) results in the locally optimal control policy

$$\boldsymbol{\pi}_k(\mathbf{x}_k) = \underbrace{-\mathbf{Q}_{\mathbf{uu},k}^{-1}\mathbf{Q}_{\mathbf{u},k}}_{\delta\tilde{\mathbf{u}}_k}\underbrace{-\mathbf{Q}_{\mathbf{uu},k}^{-1}\mathbf{Q}_{\mathbf{ux},k}}_{\mathbf{K}_k}(\mathbf{x}_k - \bar{\mathbf{x}}_k), \tag{7}$$

where $\delta\tilde{\mathbf{u}}$ and $\mathbf{K}$ are the locally-optimal feed-forward control and the locally-optimal feedback gain, respectively. At convergence, $\delta\tilde{\mathbf{u}} \approx 0$ and the resulting policy is the Riccati feedback controller [13]. Let $\mathbf{A} = \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\big|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}$, $\mathbf{B} = \frac{\partial\mathbf{f}}{\partial\mathbf{u}}\big|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}$. Denote $\mathbf{f}_{\mathbf{xx}}, \mathbf{f}_{\mathbf{uu}}, \mathbf{f}_{\mathbf{ux}}$ the tensors that represent the second-order partials of $\mathbf{f}$. Denote $\mathbf{q}_k$ and $\mathbf{r}_k$ the gradients of ℓ_k w.r.t. $\mathbf{x}$ and $\mathbf{u}$ respectively, $\mathbf{Q}_k$, $\mathbf{R}_k$, and $\mathbf{P}_k$ the second-order partials of ℓ_k. The derivatives of $Q_k(\cdot,\cdot)$ along the trajectory are then calculated recursively using

$$\hat{\mathbf{s}}_{k+1} = \mathbf{s}_{k+1} + \mathbf{S}_{k+1}\bar{\mathbf{d}}_{k+1} \tag{8a}$$
$$\mathbf{Q}_{\mathbf{x},k} = \mathbf{q}_k + \mathbf{A}_k^\top\hat{\mathbf{s}}_{k+1} \tag{8b}$$
$$\mathbf{Q}_{\mathbf{u},k} = \mathbf{r}_k + \mathbf{B}_k^\top\hat{\mathbf{s}}_{k+1} \tag{8c}$$
$$\mathbf{Q}_{\mathbf{xx},k} = \mathbf{Q}_k + \mathbf{A}_k^\top\mathbf{S}_{k+1}\mathbf{A}_k + \mathbf{s}_{k+1}\cdot\mathbf{f}_{\mathbf{xx},k} \tag{8d}$$
$$\mathbf{Q}_{\mathbf{uu},k} = \mathbf{R}_k + \mathbf{B}_k^\top\mathbf{S}_{k+1}\mathbf{B}_k + \mathbf{s}_{k+1}\cdot\mathbf{f}_{\mathbf{uu},k} \tag{8e}$$
$$\mathbf{Q}_{\mathbf{ux},k} = \mathbf{P}_k + \mathbf{B}_k^\top\mathbf{S}_{k+1}\mathbf{A}_k + \mathbf{s}_{k+1}\cdot\mathbf{f}_{\mathbf{ux},k} \tag{8f}$$

where $\mathbf{S}_{k+1}$, $\mathbf{s}_{k+1}$, and s_{k+1} are the Hessian, gradient, and drift terms that quadratically approximate v_{k+1} (Eq. (5)) at the shooting state, and $\hat{\mathbf{s}}_{k+1}$ is an intermediate variable. The operator $\cdot$ denotes the vector-tensor multiplication. The value function approximations are recursively calculated using

$$\mathbf{S}_k = \mathbf{Q}_{\mathbf{xx},k} - \mathbf{Q}_{\mathbf{ux},k}^T\mathbf{Q}_{\mathbf{uu},k}^{-1}\mathbf{Q}_{\mathbf{ux},k} \tag{9a}$$
$$\mathbf{s}_k = \mathbf{Q}_{\mathbf{x},k} - \mathbf{Q}_{\mathbf{ux},k}^T\mathbf{Q}_{\mathbf{uu},k}^{-1}\mathbf{Q}_{\mathbf{u},k} \tag{9b}$$
$$s_k = s_{k+1} - \frac{1}{2}\mathbf{Q}_{\mathbf{u},k}^T\mathbf{Q}_{\mathbf{uu},k}^{-1}\mathbf{Q}_{\mathbf{u},k} \tag{9c}$$

where the boundary conditions for Eqs. (8) and (9) are

$$\mathbf{S}_N = \mathbf{Q}_N, \mathbf{s}_N = \mathbf{q}_N, s_N = 0. \tag{10}$$

The Eqs. (8) and (9) become the standard DDP backward sweep equations [25], [60] when the defect is identically zero, and become multiple-shooting iLQR [38], [70] when the last terms of Eqs. (8d)-(8f) are removed. Our previous work [23] enables unifying the backward sweep equations of all previous formulations [25], [38], [60], [70].

Remark 1. *The action-value function* (6) *measures the effect of a control action on the long-term cost-to-go given the current state. In the context of legged robots, the use of MPC not only provides an optimal open-loop control, but also imbues viability as a side effect [85] with suitable cost design. Minimizing* (6) *allows local adjustment of the optimal open-loop control based on the current state. Therefore, if* (6) *can be incorporated somehow in the whole-body QP, then a high-frequency feedback control may be generated for stabilization. With MS-iLQR, the function* (6) *naturally results from the solution process, providing a direct synergy between* CAFE-MPC *and the ultimate lower-level VWBC in Section V.*

2) Forward Sweep: MS-DDP employs a hybrid forward roll-out [70] to update the nominal trajectory $(\bar{\mathbf{X}}, \bar{\mathbf{U}})$ by applying the control policy

$$\mathbf{u}_k' = \bar{\mathbf{u}}_k + \alpha\delta\tilde{\mathbf{u}}_k + \mathbf{K}_k(\mathbf{x}_k' - \bar{\mathbf{x}}_k). \tag{11}$$

where the superscript $'$ indicate variables after a trial step of size $\alpha \in (0, 1]$, which is determined by a backtracking line search. This control policy is first used to update $\bar{\mathbf{X}}$ for $0 \leq k \leq N-1$ via linearized dynamics

$$\mathbf{x}_{k+1}' = \bar{\mathbf{x}}_k + \mathbf{A}_k(\mathbf{x}_k' - \bar{\mathbf{x}}_k) + \alpha\mathbf{B}_k\delta\tilde{\mathbf{u}}_k + \bar{\mathbf{d}}_{k+1}. \tag{12}$$

Then the following equation is executed to sequentially overwrite the roll-out states, i.e., for $k+1 \in \tilde{\mathbb{M}}$

$$\mathbf{x}_{k+1}' = \mathbf{f}(\mathbf{x}_k', \mathbf{u}_k'). \tag{13}$$

The defect variable and the cost function are then evaluated along the trial trajectory $(\mathbf{X}', \mathbf{U}')$. In practice, Eq. (13) is evaluated in parallel, i.e., we simulate the dynamics starting from a shooting node until the next subsequent shooting node.

3) Line Search and Regularization: An adaptive merit function that automatically balances the defect and cost is used for backtracking line search. An exact expected cost change in the sense of linearized dynamics and quadratic cost approximation with the Armijo condition is used for the acceptance condition. Details on this regard are found in [23]. A similar regularization method as in [25] is used to ensure the positive definiteness of $\mathbf{Q}_{\mathbf{uu}}$ in Eq. (8e). The result of a successful line search is that the nominal trajectory $\bar{\mathbf{X}}$, $\bar{\mathbf{U}}$ is updated, with the backward and forward sweeps then repeated to convergence.

B. MS-DDP for Multi-Phase TO

This section discusses the extension of MS-DDP to account for the hybrid system effects (1) and (2), which represents a contribution compared to our previous work [23]. We first discuss the case without considering the switching constraint.

1) Unconstrained Multi-Phase TO: An unconstrained multi-phase TO problem is formulated as

$$\min_{\mathbf{X}^{[\cdot]}, \mathbf{U}^{[\cdot]}} \quad \sum_{i=1}^{n_p} J^{[i]}(\mathbf{X}^{[i]}, \mathbf{U}^{[i]}) \tag{14a}$$
$$\text{subject to} \;\; (1) \tag{14b}$$

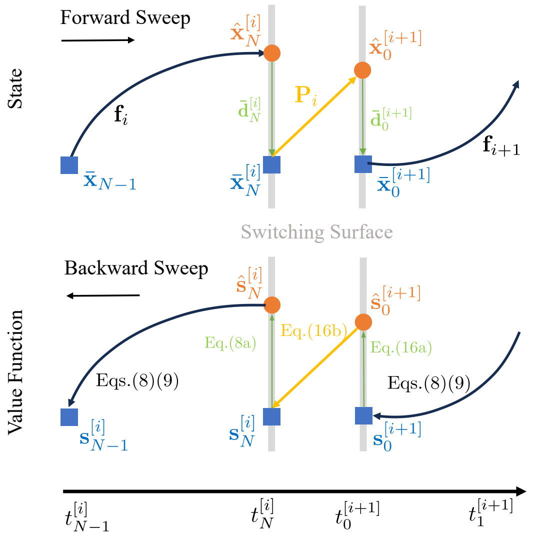


Fig. 3: Conceptual illustration of a forward sweep and backward sweep of MS-DDP for hybrid systems TO.

where the variables with the superscript $^{[i]}$ or subscript i indicate they are phase-dependent. The remaining variables share the same definitions as in problem (3) and Eq. (1).

The reset map (Eq. (1)) instantaneously changes the state, potentially to a lower-dimensional space, involving a discontinuous jump. While the forward sweep and the backward sweep remain the same as in a single-phase problem until the end of a phase, care must be taken when performing the forward sweep and backward sweep across the reset map. We define the defect after the reset map as

$$\bar{\mathbf{d}}_0^{[i+1]} = \mathbf{P}_i(\bar{\mathbf{x}}_{N_i}^{[i]}) - \bar{\mathbf{x}}_0^{[i+1]} \tag{15}$$

which is evaluated in the forward sweep once the state trajectory $\mathbf{X}^{[i]}$ is initialized for each phase. In the backward sweep, we update the value functions across the reset map using

$$\hat{\mathbf{s}}_0^{[i+1]} = \mathbf{s}_0^{[i+1]} + \mathbf{S}_0^{[i+1]}\bar{\mathbf{d}}_0^{[i+1]} \tag{16a}$$

$$\mathbf{s}_N^{[i]} = \mathbf{q}_N^{[i]} + \frac{\partial \mathbf{P}_i^\top}{\partial \mathbf{x}}\hat{\mathbf{s}}_0^{[i+1]} \tag{16b}$$

$$\mathbf{S}_N^{[i]} = \mathbf{Q}_N^{[i]} + \frac{\partial \mathbf{P}_i^\top}{\partial \mathbf{x}}\mathbf{S}_0^{[i+1]}\frac{\partial \mathbf{P}_i}{\partial \mathbf{x}} + \mathbf{s}_0^{[i+1]} \cdot \mathbf{P}_{\mathbf{xx},i} \tag{16c}$$

where $\mathbf{P}_{\mathbf{xx},i}$ is a tensor representing the second-order partials of $\mathbf{P}$. Same as Eq. (8), omitting the last term in Eq. (16c) results in iLQR/MS-iLQR. The Eq. (16) is an impact-a backward step. It is analogous to HS-DDP in our pre work [84], but is extended to incorporate defects due infeasible warm start. Figure 3 graphically illustrates a for sweep and a backward sweep across two consecutive ph

2) Constrained Multi-Phase TO: We now conside constrained multi-phase TO problems, which are formu as

$$\begin{aligned} \min_{\mathbf{X}^{[\cdot]},\mathbf{U}^{[\cdot]})} \quad & \sum_{i=1}^{n_p} J^{[i]}(\mathbf{X}^{[i]}, \mathbf{U}^{[i]}) \\ \text{subject to} \quad & (1), (2) \\ & \mathbf{h}_i(\mathbf{x}_k^{[i]}, \mathbf{u}_k^{[i]}) \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{h}_i(\cdot) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_h}$ represents general inequality constraints.

We follow the previous work [84] to take a bi-level approach to solve the problem (17) with MS-DDP. The terminal constraint (2) is handled with Augmented Lagrangian (AL) methods [86], and the inequality constraint (17c) is dealt with using the Relaxed Barrier (ReB) method [87]. Though similar in spirit to the previous work [84], the inner-loop optimization here is based upon the multiple-shooting formulation. We can now provide a reasonable initial guess of $\mathbf{X}$ so that the violation of the terminal constraint is not as bad as in the previous single-shooting formulation, facilitating faster convergence. Note that we omit the second-order terms in Eqs. (8) and (16) for real-time implementation, thus resulting in MS-iLQR, which we use in the rest of the paper. Work on incorporating second-order analytical derivatives [88] is in progress.

IV. Cascaded-Fidelity Model Predictive Control

This section describes in detail the CAFE-MPC formulation. Our previous work [22] proposed an MPC formulation that schedules a sequence of models with descending fidelity along the prediction horizon in each planning problem. The leading part of the plan (figure 4) reasons about higher-fidelity models that are more consistent with the physics of the real robot. The trailing part of the plan reasons about less-expressive models to gain computational efficiency. The trailing part provides an approximation of the long-term cost to guide the leading part of the plan. With the same motivation, CAFE-MPC generalizes this idea of cascaded fidelity beyond dynamics modeling. Specifically, it employs a finer integration time step in the near term and a coarse time step in the long run. Further, it considers the full set of constraints in the leading plan and removes certain constraints in the trailing plan.

An overview of the CAFE-MPC formulation is shown in figure 4, illustrated for a quadruped robot. We use the whole-body dynamics for the leading plan, and the SRB dynamics for the trailing plan. The durations of the two plans are denoted by T_w and T_s, respectively. As opposed to planar models in previous work [22], these 3D representations enable more behaviors in the 3D space, for instance, a barrel roll. We use multi-resolution integration time steps, $dt_\text{w} = 10$ ms for whole-body dynamics, and $dt_\text{s} = 50$ ms for the SRB dynamics. In Section VII-A, we study the effect of different dt_s on

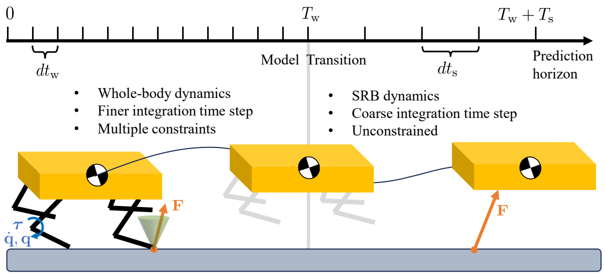


Fig. 4: Illustration of the sequentially cascaded-fidelity plans along the prediction horizon.

of constraints in the leading whole-body plan, such as torque limits, friction constraints, etc, and leave the the trailing SRB plan unconstrained. The rest of this section describes how each planning problem is constructed, and how they are connected and are cast into a multi-phase TO problem.

A. Whole-Body Plan Formulation

The leading high-fidelity whole-body plan employs whole-body dynamics with hard contacts. We use Euler angles $\boldsymbol{\theta} \in \mathbb{R}^3$ to represent the orientation of the floating base. Let $\mathbf{c} \in \mathbb{R}^3$ be the xyz position of the floating base, $\mathbf{q}_J \in \mathbb{R}^{12}$ be the joint angles, $\boldsymbol{\tau} \in \mathbb{R}^{12}$ be the actuation torques, and $\mathbf{q} = [\boldsymbol{\theta}^\top, \mathbf{c}^\top, {\mathbf{q}_J}^\top]^\top \in \mathbb{R}^{18}$ be the generalized coordinates, and $\mathbf{p} \in \mathbb{R}^{12}$ be foot locations. Let $\mathbf{x} = [\mathbf{q}^\top, \dot{\mathbf{q}}^\top]^\top$ be the state variable, and $\mathbf{u} = \boldsymbol{\tau}$ be the control variable.

1) Cost Function: The running cost in this work consists of three terms (1) tracking of a state reference (2) tracking of swing foot positions and velocities and (3) torque minimization, i.e.,

$$l_k = \|\Delta\mathbf{q}\|^2_{\mathbf{W}_q} + \|\Delta\dot{\mathbf{q}}\|^2_{\mathbf{W}_{\dot{q}}} + \\ \|\mathbf{S}\Delta\mathbf{p}\|^2_{\mathbf{W}_p} + \|\mathbf{S}\Delta\dot{\mathbf{p}}\|^2_{\mathbf{W}_{\dot{p}}} + \|\mathbf{u}\|^2_{\mathbf{R}_\tau} \quad (18)$$

where $\Delta\cdot$ denotes the difference between a variable and its reference. Let $s_j \in \{0, 1\}$ denote the swing status of the j^{th} leg, then $\mathbf{S} \in \mathbb{R}^{12\times 12}$ is a diagonal matrix whose diagonal concatenates vectors $[s_j, s_j, s_j]$ of all four legs. The terminal cost is similarly defined but without the last term.

2) Dynamics: The dynamics of a quadruped robot in contact are given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}_\tau^\top\boldsymbol{\tau} + \mathbf{J}_c^\top\mathbf{F}_c \quad (19)$$

where $\mathbf{M}$ is the generalized mass matrix, $\mathbf{n}$ collects the Coriolis, centripetal, and gravity forces, $\mathbf{B}_\tau \in \mathbb{R}^{12\times 18}$ is a constant selection matrix due to the un-actuated floating base, $\mathbf{F}_c \in \mathbb{R}^{3n_c}$ concatenates the GRF $\mathbf{f}_{EE_j} \in \mathbb{R}^3$ of each contact foot, with n_c the number of active contacts, and $\mathbf{J}_c \in \mathbb{R}^{3n_c\times 18}$ is the contact Jacobian. In addition to the dynamics (19), the contact foot is often assumed to be static with the acceleration-level non-slip constraint

$$\mathbf{J}_c\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c\dot{\mathbf{q}} = -\alpha\mathbf{J}_c\dot{\mathbf{q}} \quad (20)$$

where we use the first-order Baumgarte stabilization [26], [89] to mitigate the violation of non-slip constraint at the velocity level due to numerical integration, and $\alpha > 0$ represents the first-order Baumgarte stabilization parameter. We use $\alpha = 10$ in this work for a stabilization time constant of $1/10$ second. Further, we address the constraint (20) at the dynamics level, resulting in the well-known KKT contact dynamics [90]

$$\begin{bmatrix}\mathbf{M} & \mathbf{J}_c^T \\ \mathbf{J}_c & \mathbf{0}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{q}} \\ -\mathbf{F}_c\end{bmatrix} = \begin{bmatrix}\mathbf{B}_\tau^T\boldsymbol{\tau} - \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) \\ -\dot{\mathbf{J}}_c\dot{\mathbf{q}} + \alpha\mathbf{J}_c\dot{\mathbf{q}}\end{bmatrix}. \quad (21)$$

To model a change in the contact mode, when a new foot touches down, the impact dynamics are similarly defined as

$$\begin{bmatrix}\mathbf{M} & \mathbf{J}_c^T \\ \mathbf{J}_c & \mathbf{0}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}^+ \\ -\boldsymbol{\Lambda}_c\end{bmatrix} = \begin{bmatrix}\mathbf{M}\dot{\mathbf{q}}^- \\ \mathbf{0}\end{bmatrix}. \quad (22)$$

where the superscripts $^+$ and $^-$ indicating post and pre-impact event, and $\boldsymbol{\Lambda}_c$ represents the impulse. The linear systems (21) and (22) can be efficiently solved with Cholesky decomposition when $\mathbf{J}_c$ has full rank, which is performed using Pinocchio [91] in this work. The contact dynamics (21) and the impact dynamics (22) vary based on the foot contact status, resulting in a hybrid system. Assuming a fixed contact sequence and timing throughout this work, we obtain the discrete-time state-space equation via explicit Euler integration

$$\mathbf{x}_{k+1}^{[i]} = \mathbf{x}_k^{[i]} + \begin{bmatrix}\dot{\mathbf{q}}_k^{[i]} \\ \ddot{\mathbf{q}}_k^{[i]}\end{bmatrix} dt_{\text{w}}, \quad (23)$$

where the superscript $^{[i]}$ follows the convention of Eq. (1) to indicate phase-dependent dynamics. It is emphasized here that dt_{w} is the integration time step for the whole-body plan. Similarly, the state-space reset map $\mathbf{P}_{\text{w}}$ is

$$\mathbf{x}_0^{[i+1]} = \left[(\mathbf{q}_{N_i}^{[i]})^\top, (\dot{\mathbf{q}}_0^{[i+1]})^\top\right]^\top := \mathbf{P}_{\text{w}}^{[i]}(\mathbf{x}_N^{[i]}). \quad (24)$$

To clarify the notation, the post-reset velocity $\dot{\mathbf{q}}_0^{[i+1]}$ is noted equivalently by $\dot{\mathbf{q}}^+$ in Eq. (22). Care must be taken here that the reset map (24) should also consider the case of taking off, where the mode-transition dynamics are trivial, with $\dot{\mathbf{q}}^+ = \dot{\mathbf{q}}^-$.

3) Constraints: Multiple constraints are considered in the whole-body plan. At the moment of touchdown, we require that the height of the touchdown foot be on the contact surface. Let $\mathbf{p}_{EE_j}$ be the foot position of the j^{th} foot in world frame. Then the touchdown constraint is

$$\begin{bmatrix}0, 0, 1\end{bmatrix}\mathbf{p}_{EE_j} - h_c = 0 \quad \forall j \text{ touch down} \quad (25)$$

where h_c is the height of the contact surface in the world frame, $\mathbf{p}_{EE_j}$ is calculated via the forward kinematics. For each contact foot, the contact force needs to satisfy the friction cone constraint. In this work, we use an inner pyramid approximation

$$|f_{EE_j}^{x,y}| \leq \mu f_{EE_j}^z \quad (26)$$

where $\mu > 0$ is the friction coefficient. Note that Eq. (26) implies $f_{EE_j}^z \geq 0$. In addition, the torque limit, joint limit, and joint speed limit are always enforced throughout the whole-body plan

$$\boldsymbol{\tau}_L \preceq \boldsymbol{\tau} \preceq \boldsymbol{\tau}_U, \quad (27a)$$

$$\mathbf{q}_{J_L} \preceq \mathbf{q}_J \preceq \mathbf{q}_{J_U}, \quad (27b)$$

$$\dot{\mathbf{q}}_{J_L} \preceq \dot{\mathbf{q}}_J \preceq \dot{\mathbf{q}}_{J_U}, \quad (27c)$$

where the subscripts L and U indicate lower bound and upper bound, and $\preceq$ denotes element-wise inequality.

B. SRB Plan Formulation

The trailing lower-fidelity SRB plan uses coarse integration time step $dt_{\text{s}} > dt_{\text{w}}$, and removes all constraints. For the SRB model, let $\mathbf{x} = [\boldsymbol{\theta}^\top, \mathbf{c}^\top, \dot{\boldsymbol{\omega}}^\top, \dot{\mathbf{c}}^\top]^\top$ be the state variable, $\mathbf{u} = [\mathbf{f}_{EE_1}^\top, \mathbf{f}_{EE_2}^\top, \mathbf{f}_{EE_3}^\top, \mathbf{f}_{EE_4}^\top]^\top$ be the control variable (for the ground forces at the feet). Additionally, let $\boldsymbol{\omega} \in \mathbb{R}^3$ be the angular velocity of the floating base in the body frame.

1) Cost Function: The running cost consists of a tracking cost and a GRF regularization term

$$l_k = \|[\Delta\boldsymbol{\theta}, \Delta\mathbf{c}]\|^2_{\mathbf{W}_{\theta,c}} + \left\|[\Delta\dot{\boldsymbol{\theta}}, \Delta\dot{\mathbf{c}}]\right\|^2_{\mathbf{W}_{\dot{\theta},\dot{c}}} + \left\|\tilde{\mathbf{S}}\Delta\mathbf{F}_{EE}\right\|^2_{\mathbf{R}_u} \tag{28}$$

where $\mathbf{W}_{\theta,c}$ are the components in $\mathbf{W}_q$ corresponding to θ and $\mathbf{c}$, and $\mathbf{W}_{\dot{\theta},\dot{c}}$ are the components in $\mathbf{W}_{\dot{q}}$ corresponding to $\dot{\boldsymbol{\theta}}$ and $\dot{\mathbf{c}}$, and $\tilde{\mathbf{S}} = \mathbf{1}_{12} - \mathbf{S}$ is the contact status matrix where $\mathbf{1}_{12} \in \mathbb{R}^{12\times 12}$ is an identity matrix and $\mathbf{S}$ denotes swing status as defined in Eq. (18). The terminal cost is similarly defined but excludes the regularization term.

2) Dynamics: The SRB dynamics is

$$\begin{aligned} \ddot{\mathbf{c}} &= \sum_{j}^{4} \tilde{s}_j \frac{\mathbf{f}_{EE_j}}{m} - \mathbf{g} \\ \dot{\boldsymbol{\omega}} &= \mathbf{I}^{-1}(-\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \mathbf{R}^\top \sum_{j=1}^{4} \tilde{s}_j(\mathbf{p}_{EE_j} - \mathbf{c}) \times \mathbf{f}_{EE_j}), \end{aligned} \tag{29}$$

where $\mathbf{I}$ is the rotational inertia of the body, $\mathbf{g}$ is the earth gravity, $\mathbf{R}$ denotes the body orientation w.r.t. the world frame, $\tilde{s}_j$ is a diagonal component of $\tilde{\mathbf{S}}$ indicating the contact status of the j^{th} foot. All quantities except for $\boldsymbol{\omega}$ are expressed in the world frame. Let $\mathbf{T}(\cdot)$ be the transformation matrix that converts angular velocity to the rate of change of Euler angles [92]. With $\dot{\boldsymbol{\theta}} = \mathbf{T}(\boldsymbol{\theta})\boldsymbol{\omega}$, its time derivative, and the second equation of (29), we can obtain $\ddot{\boldsymbol{\theta}}$ as a function as the SRB state. The discrete-time state-space equation of the SRB dynamics is then

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \begin{bmatrix} \dot{\boldsymbol{\theta}}^\top & \dot{\mathbf{c}}^\top & \ddot{\boldsymbol{\theta}}^\top & \ddot{\mathbf{c}}^\top \end{bmatrix}^\top dt_{\text{s}}, \tag{30}$$

where it is emphasized here that dt_{s} is the integration time step for the simplified model. Note that the foot location $\mathbf{p}_{EE_j}$ in the SRB model is assumed to be known from a reference trajectory according to Raibert heuristics [16], it is neither part of the state nor the control.

3) Constraints: The SRB planning is formulated as an unconstrained optimization problem. The idea of CAFE-MPC is to relax the problem constraints/cost/dynamics later in the horizon for computational efficiency over accuracy. In our case, we empirically find that removing the constraints on the tail is adequate for the highly dynamic motions we aim to produce.

C. Connecting WB Plan & SRB Plan

The WB plan and the SRB plan are not decoupled in CAFE-MPC , but rather are connected via a transition constraint. To differentiate their dimensionality, we use $\mathbf{x}_{\text{w}}$ to denote the whole-body state and $\mathbf{x}_{\text{s}}$ to denote the SRB state in this section. At the instance of model transition (figure 4), the whole-body plan and SRB plan are connected via

$$\mathbf{x}_{\text{s}}^{+} = \mathbf{T}_{w\to s}\mathbf{P}_{\text{w}}(\mathbf{x}_{\text{w}}^{-}) \tag{31}$$

where the superscripts $^+$ and $^-$ denote the moment immediately before and after the model transition, $\mathbf{P}_{\text{w}}(\cdot)$ is given by Eq. (24), and

$$\mathbf{T}_{w\to s} = \begin{bmatrix} \mathbf{1}_6 & \mathbf{0}^{6\times 12} & \mathbf{0}^{6\times 6} & \mathbf{0}^{6\times 12} \\ \mathbf{0}^{6\times 6} & \mathbf{0}^{6\times 12} & \mathbf{1}_6 & \mathbf{0}^{6\times 12} \end{bmatrix}_{12\times 36} \tag{32}$$

is the state projection matrix.

D. Cast To Multi-Phase TO

As discussed in Section III, a new phase is determined when there is a change in one of the following (1) system dynamics, (2) state or control, (3) cost function, or (4) constraint. A rough choice of phases would be to consider the full whole-body plan as one phase and the SRB plan as another. This choice of phases, however, is not sufficiently accurate, as the contact status of each foot can likely change along the whole-body plan. The dimension of the KKT matrix of the contact dynamics depends on the number of active contacts. Therefore, the whole-body plan is further divided into multiple phases depending on foot contact status. A whole-body phase is determined when there are any foot contact changes. Since the contact schedule is known a priori, the number of whole-body phases as well as the start and end time of each phase can be induced given the prediction horizon T_{w} of the whole-body plan. The SRB plan is considered as one single phase, since with the predetermined contact schedule, the variable $\tilde{s}$ is simply a time-varying parameter, and the foothold locations $\mathbf{p}_{EE_j}$ are specified from a reference trajectory [16].

Remark 2. *We emphasize two important features of the* CAFE-MPC. *Firstly, the trailing SRB plan provides an approximation (with low-rank Hessian) for the long-term cost to go of the whole-body model. Adding future costs to an optimal control problem empirically helps make the state more viable [85], provided the cost function is designed properly. An ideal case is perhaps to extend the whole-body plan with the same problem structure. This strategy, however, is computationally expensive. Relaxing the dynamics/cost/constraints later in the prediction saves computational effort, while still capturing salient features of the future plan. Secondly, the choice of the whole-body model and SRB model can be generalized to other sequences of models [22]. For instance, [43] employs a centroidal followed by a choice of convex model (such as linear inverted pendulum (LIP)) for humanoid walking, and [45] employs a centroidal model followed by an SRB model for humanoid balancing.*

Remark 3. *For readers already familiar with adaptive-complexity MPC [47], we note its major differences with* CAFE-MPC. CAFE-MPC *aims to robustify whole-body MPC by adding a low-rank future cost without significantly increasing computational efforts. The adaptive-complexity MPC, by contrast, focuses on adjusting the model expressiveness along the horizon based on the assessed need for motion complexity. In the worst case, it comes at the same computational cost as whole-body MPC. Naturally, these two ideas could be combined, for example, changing the prediction model in the tail as needed to further improve the long-term cost predictions in* CAFE-MPC.

V. VALUE-BASED WHOLE-BODY CONTROLLER

The whole-body plan of CAFE-MPC incorporates the actuation torque ($\boldsymbol{\tau}$) and the whole-body states ($\mathbf{q}$, $\dot{\mathbf{q}}$), which are executable on the robot. As a result, one strategy for

WBC is to directly apply the feed-forward torque with a PD controller that regulates toward the optimized $(\mathbf{q},\ \dot{\mathbf{q}})$. When gains are low, this approach is similar to open-loop MPC, and requires higher MPC update frequency to account for model uncertainties. The whole-body plan, however, includes a value function approximation and a local feedback policy, which are available for free as intermediate results of MS-iLQR. An alternative approach for WBC is thus to apply the closed-loop control

$$\boldsymbol{\pi}_k(\mathbf{x}_k) = \mathbf{u}_k^* + \mathbf{K}_k(\mathbf{x} - \mathbf{x}_k^*) \tag{33}$$

where $\mathbf{K} = -\mathbf{Q}_{\mathbf{uu}}^{-1}\mathbf{Q}_{\mathbf{ux}}$ as in Eq. (7). This approach is known as the Riccati feedback control. It smooths the actuation torque in between MPC time steps, enables improved stability, and allows for slower MPC updates [13], [61]. Despite these benefits, a well-known problem with (33) is that the closed-loop trajectory does not necessarily satisfy certain physical constraints. This problem can be critical to the robot stability, for instance, the robot can fall when friction cone constraints are violated.

A. Value-Function Based WBC

The VWBC proposed in this work embeds the Riccati feedback controller (33) within a conventional OSC-QP formulation, thereby unifying the conventional WBC and the Riccati feedback control. The VWBC produces a whole-body control command that is close to the Riccati controller, while at the same time satisfying all necessary physical constraints. The VWBC is formulated as

$$\min_{\boldsymbol{\tau},\ddot{\mathbf{q}},\boldsymbol{\lambda}} \ Q_k(\mathbf{x} - \mathbf{x}_k^*, \boldsymbol{\tau} - \boldsymbol{\tau}_k^*) \tag{34a}$$

$$\text{s.t. } \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}_\tau^\top \boldsymbol{\tau} + \mathbf{J}_c^\top \mathbf{F}_c \tag{34b}$$

$$\mathbf{J}_j\ddot{\mathbf{q}} + \dot{\mathbf{J}}_j\dot{\mathbf{q}} = -\alpha \mathbf{J}_j\dot{\mathbf{q}}, \quad \forall j \in \mathcal{C} \tag{34c}$$

$$\boldsymbol{\tau}_L \preceq \boldsymbol{\tau} \preceq \boldsymbol{\tau}_U, \tag{34d}$$

$$|f_{EE_j}^{x,y}| \leq \mu f_{EE_j}^{z}, \quad \forall j \in \mathcal{C} \tag{34e}$$

where $Q_k(\cdot,\cdot)$ is the action-value function approximation defined in Eq. (6), k is the time index in the whole-body plan of CAFE-MPC, Eq. (34b) is the whole-body contact dynamics as in Eq. (19), $\mathcal{C}$ denotes the set of active contacts. Eq. (34c) represents the acceleration-level non-slipping constraint with first-order Baumgarte stabilization. Eqs. (34d) - (34e), respectively, represent the torque limits, the unilateral constraint, and the linearized friction-cone constraint with an inner pyramid approximation.

The partial derivatives of $Q_k(\cdot,\cdot)$, the (sub)optimal state-control pair $(\mathbf{x}_k^*, \mathbf{u}_k^*)$, and the contact status $\mathcal{C}$ are obtained from the whole-body plan of CAFE-MPC for the time index k that is the closest to the current low-level control tick. Minimizing Q_k will encourage the resulting solution to stay close to the optimal control policy (7). Since $\mathbf{Q}_{\mathbf{uu}}$ in Eq. (6) is guaranteed (via regularization) to be positive definite, the resulting QP (34) is strictly convex.

The VWBC (34) is a generalization of the Riccati controller (7). This can be seen by removing all the constraints in (34). The problem (34) then becomes an unconstrained optimization problem, the solution of which is Eq. (7). Thus, the VWBC seeks a control signal that is the closest to the Riccati controller (7) but with all necessary constraints satisfied. More formally, one can show that minimizing the objective in Eq (34a) is equivalent to minimizing

$$\frac{1}{2}\left\|\boldsymbol{\tau} - \left(\boldsymbol{\tau}_k^* + \mathbf{K}_k(\mathbf{x} - \mathbf{x}_k^*)\right)\right\|_{\mathbf{Q}_{\mathbf{uu},k}}^2 . \tag{35}$$

In other words, the VWBC can be considered as the Riccati controller disguised in a QP, thus enjoying the benefit of feedback stabilization, while preventing the resulting solution from being too aggressive to violate the constraints. The VWBC is similar in spirit as [71], [75], except that the value functions in [71], [75] are obtained for simplified models.

The VWBC avoids additional cost tuning beyond CAFE-MPC. The action-value function $Q_k(\cdot,\cdot)$ marries the VWBC to CAFE-MPC. Thus, one can focus on the cost design for the CAFE-MPC, and leave CAFE-MPC to fully specify the VWBC cost. This is an advantage over the conventional OSC-QP, since conventional OSC-QP concatenates multiple tasks specified in the operational space, such as CoM tracking, swing feet tracking, torque regularization, etc. As a result, nontrivial tuning is often unavoidable to balance the relative importance of each task. Further, these tasks are often generated independently using separate planners that are based on simplified models. For instance, CoM trajectories can be generated using LIP model [24], [28], [71], [93], SRB model [1], [16], [73], and Centroidal dynamics [4], [26]. Swing foot trajectories are often generated by interpolating predicted foot placements using Bezier polynomials [1], [4], [16], where the foot placements are obtained with Raibert heuristics. The loss of whole-body information can potentially produce a plan that is not trackable by the low-level controller [94]. The VWBC, by contrast, overcomes the above issues, and avoids any additional meticulous tuning beyond the MPC.

Solving the QP (34) can be warm started using solutions of CAFE-MPC, and thus requiring fewer solver iterations. The whole-body plan of CAFE-MPC incorporates $\boldsymbol{\tau}_k^*, \mathbf{q}_k^*, \dot{\mathbf{q}}_k^*, \mathbf{F}_{c,k}^*$. We obtain $\ddot{\mathbf{q}}_k^* = (\dot{\mathbf{q}}_{k+1}^* - \dot{\mathbf{q}}_k^*)/dt_{\text{w}}$, and use $\boldsymbol{\tau}_k^*, \ddot{\mathbf{q}}_k^*, \mathbf{F}_{c,k}^*$ as an initial guess for the QP (34). Detailed evaluation and benchmark of the proposed VWBC and its performance are discussed in Section VII-B.

VI. IMPLEMENTATION DETAILS

In this section, we discuss a few implementation details, including reference generation, offline TO design for barrel rolls, motion composition of a running barrel roll, and other engineering details that are important to maintain fast computation and robust hardware execution. For regular locomotion skills such as bounding, pacing, etc, we use kinematic references that are based on integrating the twist and Raibert heuristics and use IK for reference joint angles. For the barrel roll and any composed motions that incorporate the barrel roll, the reference trajectories are obtained via offline TO that covers all degrees of freedom.

The contact patterns and timings are heuristically determined by observing the gaits of similar-sized quadruped animals. The contact patterns are represented by a switched system formulation, i.e., an ordered sequence of phases. Each phase is associated with a start time and change of phase is determined by a take a touch-down event of any legs. Let FR, represent the stance status of the front rigl right, and hind left legs, respectively. Le flight phase, and FS represent a full-stance contact sequence of one gait cycle of a represented by {HL-HR, FT, FL-FR, FT ciated timings $\{[t_s^{[1]}, t_e^{[1]}], [t_e^{[1]}, t_e^{[2]}], [t_e^{[2]}, t_e^{[3]}$ switched-system representation makes form phase TO problem (17) straightforward. W used here to determine the contact patter approach is to employ a contact planner, design [32], [95] or via online re-planning

A. Heuristic Locomotion Reference

The kinematic references for regular loc designed via simple heuristics, consisting of the CoM trajectory, the foot placements, and the swing foot trajectory. The users provide commands that include horizontal velocities v_x and v_y, height z, and yaw rate $\dot{\theta}_z$. The horizontal positions x, y, and the yaw angle θ_z are obtained via integrating the corresponding velocity components. The rest of CoM states are set to zero. The reference foot placements are obtained via Raibert heuristics [1], [16], but are clamped to be within a bounded box about the corresponding hip so that they are kinematically reachable. The foot placements are interpolated with cubic Bezier polynomials [1], [16], from which the swing foot positions and velocities are obtained, with reference joint angles then obtained via IK and reference joint velocities at zero. All components of the kinematic reference are used in Eq. (18) to design the cost of the whole-body plan. The CoM reference is used in Eq. (28) to design the cost of SRB plan, as well as normal GRF references that are obtained by averaging the total weight of the robot over the number of active contacts.

B. Offline Barrel Roll TO

While a heuristic kinematic reference is sufficient for regular locomotion skills, we empirically find that richer dynamic information addressing leg coordination is helpful to quickly synthesize complex motions online such as barrel roll given the real-time constraints. As such, we solve a TO problem offline to obtain a whole-body reference trajectory for an in-place barrel roll starting on all fours. We then let the Cafe-Mpc perform the fine-tuning online to account for the model mismatch or mismatch in initial motion or contact configuration. Details of this offline design process can be found in the Appendix VIII-B.

C. Engineering Details

Engineering implementations are important for successful and robust executions on robot hardware. In this section, we review some of these implementation details in terms of Cafe-Mpc configuration, warm-start strategy, policy-lag compensation, evaluation of dynamics and its partials, etc.

Cafe-Mpc is configured to use fixed prediction horizons (i.e., whole-body plan horizon T_w, and SRB plan horizon T_s in figure 4). As discussed in Section IV, the SRB plan is designed to have a single phase while the whole-body plan is constructed to span multiple phases. As the Cafe-Mpc shifts forward, the phases involved in the whole-body plan may vary as the most recent contact status moves out of the horizon, and the upcoming contact status moves in (as illustrated in figure 5). In other words, the Cafe-Mpc may entail dynamic removal of old phases and addition of new phases.

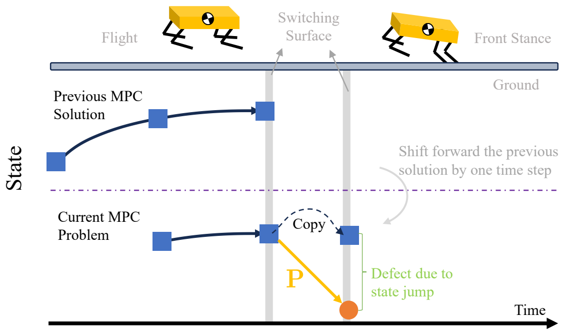


Fig. 5: Conceptual illustration of a large defect when a new phase involving state jumps is added to the current MPC problem. In warm-starting the current MPC problem, the previous MPC is shifted forward by one step. The last state of the previous MPC solution is duplicated to initialize the newly created state, causing a large defect due to the state jump. The illustration example shows when the MPC moves out of a flight phase, and adds in a new stance phase.

It is well accepted that for fixed-horizon MPC, the optimization problems between two control ticks are similar. Thus, a common approach to save computational effort is to warm start the current MPC problem using the solution from the previous MPC problem. In constructing the current MPC problem, the previous MPC solution is shifted forward by a certain amount of time steps, and new decision variables of proper sizes are appended to the shifted trajectory so that the prediction horizon remains invariant. With a multiple-shooting-based solver, the new state variables are initialized with the last state in the previous MPC solution. Care must be taken, however, on this warm-start strategy for multi-phase problems (17) subject to state jumps (e.g., impact (22)) as in Cafe-Mpc. When a new phase is added, this scheme can produce a large defect due to the state jump (illustrated in figure5), potentially requiring more iterations to converge. We propose an adaptive scheme to address this issue. If no new phases are created or the new phase does not create any state jumps (e.g., stance to flight), we keep the appended states as shooting states and use the same warm-start scheme. Otherwise, the new states are treated as roll-out states, and are computed via the transition map (32) with impact dynamics (22). As a result, the defect of the new states is identically zero. To remind the readers, a shooting state refers to an independent state that can be initialized by the user, whereas a roll-out state is a dependent variable on controls and previous states, as is detailed in Section III.

We solve the first Cafe-Mpc problem to convergence. For all subsequent Cafe-Mpc problems, the MS-iLQR solver is terminated with either maximum CPU time (18 ms) or a maximum number of iterations (4), whichever is reached first. We use Pinocchio [91] for the calculation of the whole-body dynamics and its analytical derivatives, and use CasADi [96] for SRB dynamics and its derivatives. All gradient information (dynamics, costs, and constraints) along the trajectory are computed in parallel with 4 threads using OpenMP.

While the VWBC runs on the robot hardware at 500 Hz, CAFE-MPC is executed on a separate computer at 33 to 50 Hz depending on the motion task. To account for the policy lag due to the MPC solve time and communication latency, the first six commands (states, value function, etc.) of the MPC solution are sent to the low-level VWBC controller along with their timestamps. The VWBC then finds the solution command with the timestamp closest to the current instant to formulate Eq. (34a). Since the VWBC plays a role of Riccati feedback controller and executes at a significantly higher rate than CAFE-MPC, it further minimizes the effect of policy lag on the system stability.

VII. RESULTS

The performance of the proposed motion control framework is benchmarked on MIT Mini Cheetah with several tasks: regular locomotion skills, a dynamic running jump, and highly dynamic barrel rolls. The benchmark is carried out both in simulation with a highly-fidelity dynamics simulator and on the robot hardware. For all regular locomotion skills, it is sufficient to run CAFE-MPC at 33 Hz. For the dynamic running jump and highly dynamic barrel roll, CAFE-MPC runs at 50 Hz. Results and analysis of CAFE-MPC are presented in Section VII-A. Results and discussions of VWBC are reported in Section VII-B. In addition, we compare CAFE-MPC with two other MPC schemes: an SRB MPC developed in previous work [74], and a whole-body MPC. The comparisons are to investigate their capabilities to accomplish a barrel roll. The comparison results are discussed in Section VII-C. A subset of the simulation results (e.g., for the dynamic running jump) are omitted herein but can be found in the accompanying video.

A. CAFE-MPC

We investigate the performance of CAFE-MPC under different model schedules and integration time steps. As a reminder, a model schedule $\mathcal{S} = (T_{\rm w}, T_{\rm s})$ specifies the whole-body prediction horizon $T_{\rm w}$ and the SRB prediction horizon $T_{\rm s}$. The performance is measured in terms of reference tracking and solve times. This investigation considers a bounding gait in simulation and a trotting gait on hardware. For the simulation study, three groups of experiments (denoted, G1, G2, and G3) are conducted, as summarized in Table I. Each experiment starts with a whole-body plan of 0.25 s, and differs in how an additional planning horizon T_+ is added to the formulation. The choice of 0.25 s is empirically determined as it is the shortest whole-body plan that enables bounding after some careful tuning of the terminal cost. For experiment groups G1 and G2, the additional horizon T_+ is an SRB plan, and the difference is in the integration time step. For G3, the additional plan is a whole-body plan, so CAFE-MPC becomes whole-body MPC. We increase T_+ from 0 s to 2 s, and examine the effect of the extended horizon on tracking performance and solve time.

TABLE I: Three groups of experiments for CAFE-MPC benchmark.

	$T_{\rm w}$	$T_{\rm s}$	$dt_{\rm w}$	$dt_{\rm s}$
G1	0.25 s	T_+	10 ms	10 ms
G2	0.25 s	T_+	10 ms	50 ms
G3	0.25 s + T_+	0	10 ms	50 ms

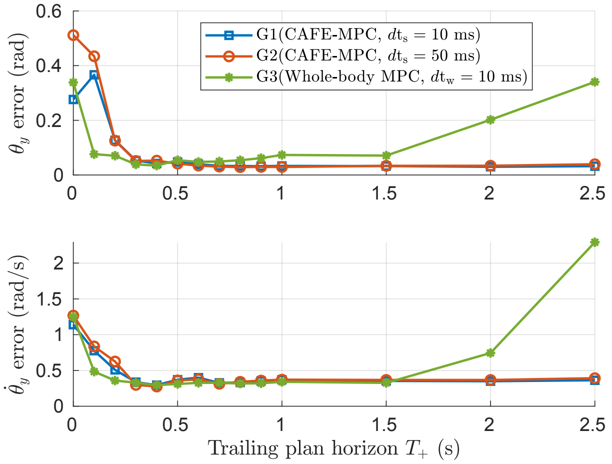


Fig. 6: Tracking performance of CAFE-MPC using model schedules and integration time steps in Table I. The data were collected with a bounding gait on a simulated MIT Mini Cheetah. Tracking performance is measured using RMS error on pitch θ_y (top) and pitch rate $\dot{\theta}_y$ (bottom).

1) Simulation Results: Figure 6 reports the tracking performance, which is measured using the root-mean-square (RMS) errors of the pitch angle θ_y and the pitch rate $\dot{\theta}_y$. For each group of experiments, CAFE-MPC attains the worst performance in the base case when $T_+ = 0$. As the length of horizon grows with increases to T_+, whether it is SRB plan or whole-body plan, the tracking errors are significantly reduced (roughly 90% reduction on θ_y and 73% reduction on $\dot{\theta}_y$) until $T_+ = 0.5$ s, after which the performance enhancement is minimal. The lack of further performance improvement is justifiable, which is likely a result of the extended horizon already providing a reasonable metric for the satisfaction of long-term goals. The extended horizon helps ensure that the current actions are appropriate to ensure longer-term balance.

The results of G1 and G2 reveal that even though the long-term objective is formulated using a simplified model, it helps improve the whole-body plan in the near term. This analysis is aligned with the results on disturbance rejection obtained in previous work [22], but is benchmarked in a high-fidelity simulator herein and using more practical real-time MPC implementation. In addition, this performance enhancement is observed for both $dt_{\rm s}$ = 10 ms and $dt_{\rm s}$ = 50 ms, meaning that using a fine integration timestep later in the prediction horizon is not necessary. The results of G3 shows that the tracking performance of whole-body MPC becomes worse when T_+ is beyond 1.5 s. This is likely because the MPC problem has become more difficult to solve due to the accumulation of high nonlinearity from the whole-body dynamics over long horizons, which makes it easier to reach bad local optima. This observation shows the advantage of CAFE-MPC over whole-body MPC for long-horizon problems.

Figure 7 depicts the solve time statistics, which are mea-

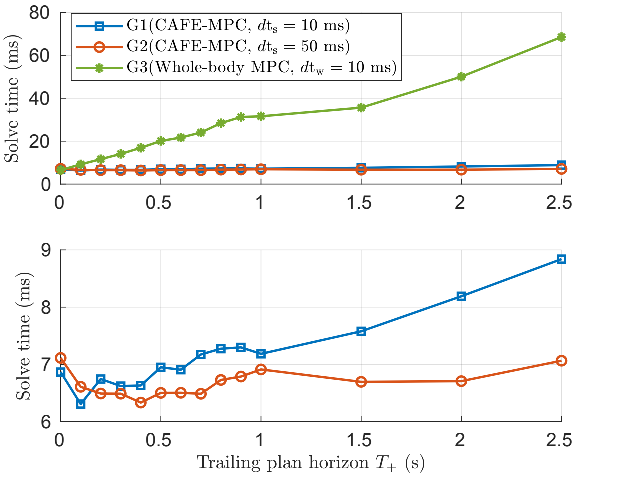


Fig. 7: Solve time statistics of CAFE-MPC using model schedules and integration time steps in Table I. Solve time here refers to the average computation time per MS-iLQR iteration. Top: solve times for all three groups in Table I. Bottom: solve times for groups G1 and G2.

sured per MS-iLQR iteration and averaged over all MPC control ticks for each motion. Comparing the results of G1 and G2, we found that the coarse integration timestep dt_s = 50 ms is more favorable than the fine integration timestep dt_s = 10 ms for CAFE-MPC . There are two reasons. First, the solve times associated with dt_s = 50 ms are in general less than those with dt_s = 10 ms, and this difference tends to increase as the planning horizon grows. Second, the difference in tracking performances between the two, as observed in figure 6, is negligible as the SRB plan horizon increases. In addition, a notable increase in solve time is observed with whole-body MPC (i.e., G3). With $T_+ = 2s$, the solve time of whole-body MPC is 8x slower than that of CAFE-MPC (G2). In summary, CAFE-MPC achieves better or on par tracking performance than whole-body MPC for regular locomotion skills with significantly less computation time, which can be further decreased with a coarse integration time step.

2) Hardware Results: To further examine the performance of CAFE-MPC as observed in figure 6, we conduct a series of similar experiments on MIT Mini Cheetah hardware with a trotting gait. Rather than testing all model schedules (Table I) as in the previous section, we use the configuration of G2 with T_s between 0 s and 0.4 s, since the most significant error reductions are observed in this range in figure 6. Note that we use dt_s = 50 ms for the SRB plan. Figure 8 depicts the tracking results in terms of forward velocity (v_x) and the body CoM height (z). The solid lines represent the actual states of the robot under different model schedules, and the dashed line represents the desired states. With $T_s = 0$ s, the robot gradually deviates from the desired trajectory, and eventually falls down after walking for a few steps. The robot can move longer with $T_s = 0.1$ s but still eventually falls down. By further elongating the SRB plan, stable trotting is achieved, and the tracking performance is improved until $T_s = 0.2$ s. The lack of further performance improvement is similarly observed in figure 6, and is likely because the trailing SRB plan already serves as a reasonable indication of the long-term goal.

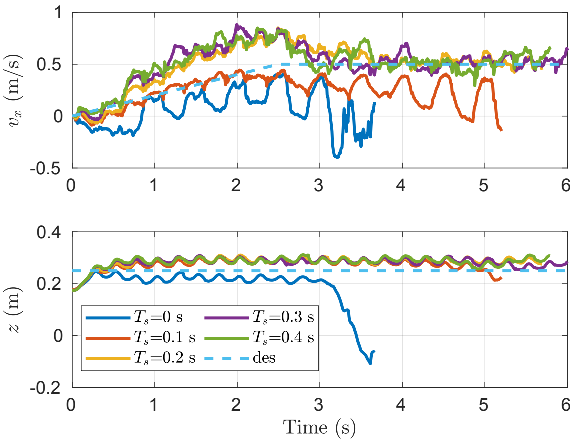


Fig. 8: Tracking performance of CAFE-MPC under model schedules of G2 (Table I) evaluated with a trotting gait on MIT Mini Cheetah hardware.

B. VWBC

In this section, we study the performance of the VWBC, and compare it with the performance of the Riccati controller. To remind the readers of the differences, the Riccati controller controls robots with the feedforward-feedback controller (7) resulting from the MS-iLQR solver. The VWBC solves a value-based QP (34) that embeds (7), but with all necessary constraints satisfied. Toward that regard, we benchmark the differences with a disturbance rejection test, and are mainly concerned with constraint satisfaction and disturbance recovery. We use qpOASES [97] as the QP solver. The whole-body plan of CAFE-MPC offers an initial guess for the valued-based QP without additional computational cost. We compare the QP solve times with and without using this initial guess.

1) Disturbance Recovery: The disturbance rejection test was conducted with the same bounding gait as in Section VII-A. We change the body velocity by a total of 1.5 m/s over 50 ms, which corresponds to an effective external force of 270 N over the period. Figure 9 depicts the joint torques of the front right and back right legs as well as their upper and lower limits. It demonstrates that the Riccati controller sometimes violates the torque constraints, especially after the push disturbance, whereas the VWBC consistently satisfies the torque constraints. One may argue that a simple clamping technique can work here, and question the need for the proposed technique. This argument is true in this case, and clamping torque is sufficient to prevent control saturation. However, the VWBC provides a systematic way of balancing optimality (via the local Q function) and constraint satisfaction. Further, some other constraints such as friction that cannot be clamped are important to robot stability. To demonstrate, figure 11 shows time-series snapshots of the robot after the push disturbance. With the Riccati controller, the robot has a non-trivial slipping that eventually disablizes the robot. With the VWBC, the slipping is slight, and the robot quickly recovers a stable contact in one gait cycle.

2) QP Solve Time: We compare the number of solver iterations[3] to solve the QP (34) with and without warm-

[3] In qpOASES, the concept of working number of recalculations (nWSR) is used. The number of iterations here is nWSR + 1.

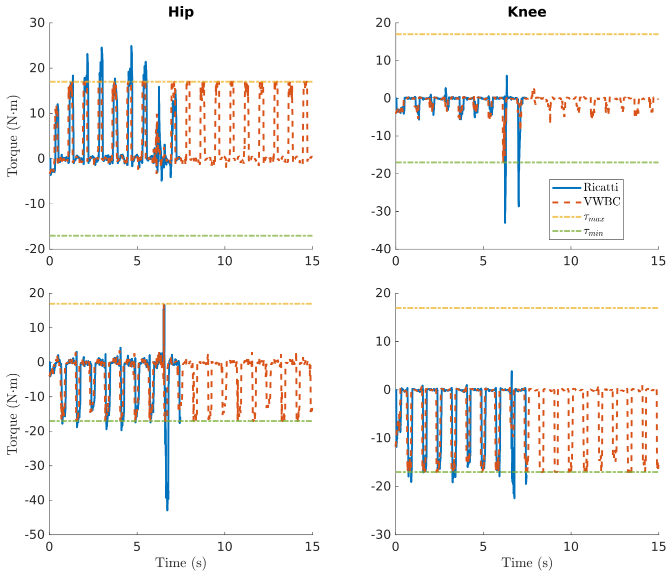


Fig. 9: Hip and knee torques of the front right leg (top) and the back right

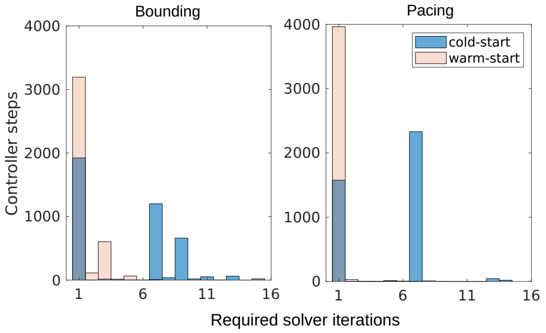


Fig. 10: Histogram for the number of required solver iterations needed to solve the QP (34) with and without warm start during a bounding gait (left) and a pacing gait (right).

start on a bounding gait and a pacing gait. The number of required solver iterations is collected at each controller step (500 controller steps per second from VWBC, figure 2), and the histogram of the collected data is shown in figure 10. For bounding, the VWBC with the proposed warm-start strategy only requires one iteration to solve the QP (34) for 79.85% of all controller steps, and no more than three iterations for 97.78% of all times. For pacing, 99.08% of the QPs are solved witl

proj

C. Barrel Roll

To investigate the capability of the CAFE-MPC+VWBC framework, three tasks are conducted that each involve a highly dynamic barrel roll. (1) In the first task, the robot is commanded to execute an in-place barrel roll followed by a pacing gait. We then compare the proposed approach with a conventional MPC approach in terms of their capabilities to accomplish this task. (2) In the second task, we make the motion a bit more challenging. The robot is commanded to perform a barrel roll in the middle of a running locomotion gait. An intermediate full-stance phase is used before the barrel roll to help gain stability. Two locomotion gaits (trotting and pacing) are tested. (3) In the third task, we make the robot imitate a human athlete performing a Fosbury flop. This task is the most difficult as no intermediate full stance is employed before the barrel roll. We repeat this task multiple times for the reliability test. We discuss the task (1) and (3) in detail in this manuscript, and encourage the readers to check the results of task (2) in the accompanying video. For all three tasks, CAFE-MPC is configured with T_w = 0.25 s, T_s= 0.5 s, dt_w = 10 ms, and dt_s = 50 ms.

References for the barrel roll are generated via the offline TO as discussed in Section VIII-B. All three tasks share the same barrel-roll reference, though it is designed assuming taking off in place. References for the locomotion gaits are obtained from a similarly constructed offline TO, but the objective is to track a kinematic trajectory as introduced in Section VI-A. The resulting references capture rich information in the joint space, enabling the CAFE-MPC to use one single cost function across all tasks. The composed reference trajectory for each task is obtained by simply connecting the two types of references in proper order without special care on motion transitions.

Figure 1 shows time-series snapshots of executing task (1) on the Mini Cheetah hardware. The hopping step after barrel roll and before pacing is introduced by design to mimic a gymnast who often takes a step to gain balance after large aerial rotations. The successful hardware execution demonstrates the capability of the proposed framework to refine references online for highly dynamic motions, effectively accounting for model uncertainties. By comparison, previous works [10], [11] on in-place barrel roll need nontrivial tuning of both cost function and dynamics, and the robot has to be locked to a pre-defined configuration for landing.

1) Comparison To Template MPC: To further investigate

conven-

lels. We

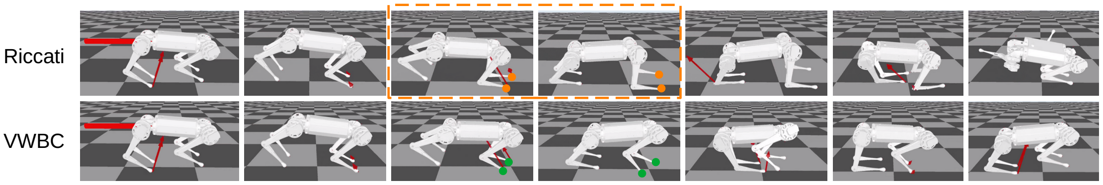


Fig. 11: Time-series snapshots of Mini Cheetah bounding under disturbances (1.5 m/s forward direction) in simulation using different whole-body control schemes. Top: Riccati feedback controller. The highlighted area in orange indicates foot slipping, leading to a fall. Bottom: proposed VWBC. The robot recovers from the initial slip (green markers), preventing a fall.

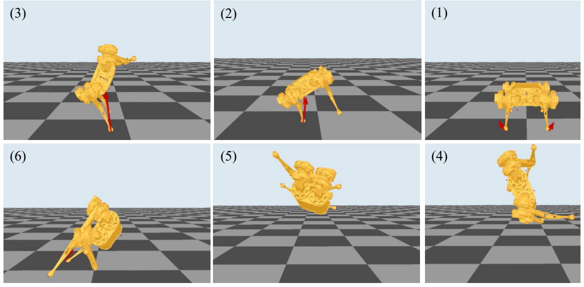


Fig. 12: Time-series snapshots of in-place barrel roll using HKD-M robot fails to accomplish the barrel-roll task because the swing-leg are not coordinated with the body motion.

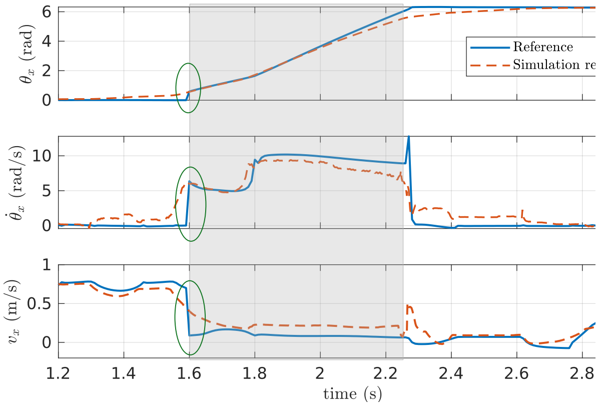


Fig. 13: Offline-composed running barrel roll reference (solid bl the executed trajectory (dashed red) on the Mini Cheetah in sin The circled areas represent discontinuities when composing the ba reference and the pacing reference. The grey area indicates the barre

use the Hybrid Kinodynamic MPC (HKD-MPC) developed in our previous work [74]. The HKD-MPC reasons about SRB dynamics, and employs contact-dependent kinematics (full joint kinematics for swing legs and prismatic foot for stance legs). The HKD-MPC computes the CoM trajectory (translation and orientation), foot placements, and GRFs. The GRFs are directly used for stance leg control. A separate swing controller is used to track swing trajectories that interpolate the optimized foot placements. The HKD-MPC has been shown to have robust performance for synthesizing running jumps, and strong disturbance rejection capability [74]. We utilize the already proficient HKD-MPC to follow the identical reference as CAFE-MPC in the task (1). Figure 12 shows the time-series snapshots of the resulting motion in simulation. The robot behaves reasonably until after taking off, where the swing legs cross over the body fast, reducing the body angular momentum and resulting in an unsafe landing configuration. Fundamentally, the failure arises from the fact that HKD-MPC does not coordinate the body angular momentum and the leg angular momentum. By comparison, CAFE-MPC explores the whole-body dynamics (though over a short prediction horizon) and accounts for the conservation of angular momentum implicitly via the whole-body dynamics for re-orienting the body for safe landing.

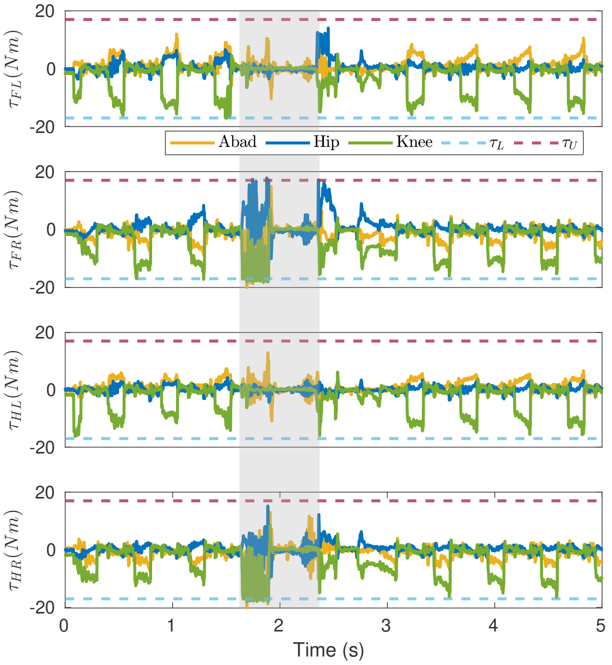


Fig. 15: Joint torques of the Mini Cheetah hardware during the mid-run barrel roll. Dashed lines indicate torque limits.

2) Fosbury Flop: The Fosbury flop is a challenging technique used by well-trained athletes in high jumps. We push the capability of the CAFE-MPC +VWBC framework to imitate this highly dynamic behavior on the robot hardware. The

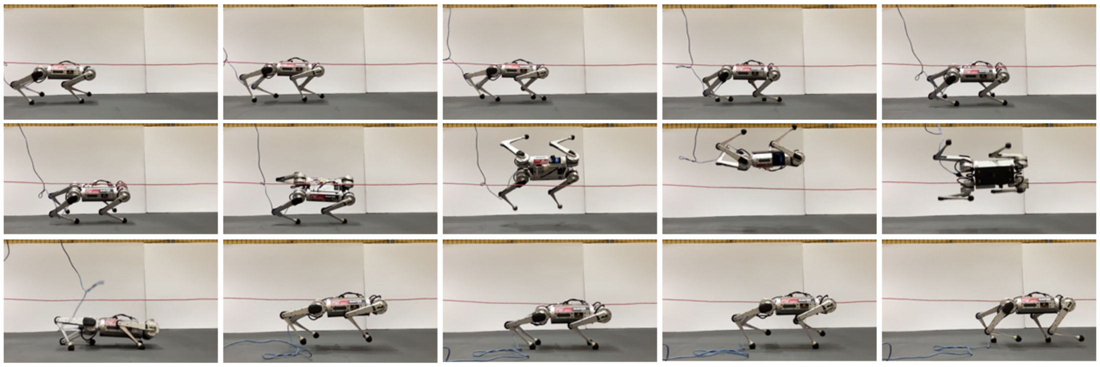


Fig. 14: Time-series snapshots of the MIT Mini Cheetah performing a Fosbury Flop. The robot runs at 0.8 m/s with a pacing gait, decelerates and performs a barrel roll over a 0.4 m cable, hops for one step, and accelerates to 0.8 m/s with the pacing gait. All motions and transitions are synthesized online in real time. The proposed controller is sufficiently powerful so that no intermediate full stance is needed before the barrel roll.

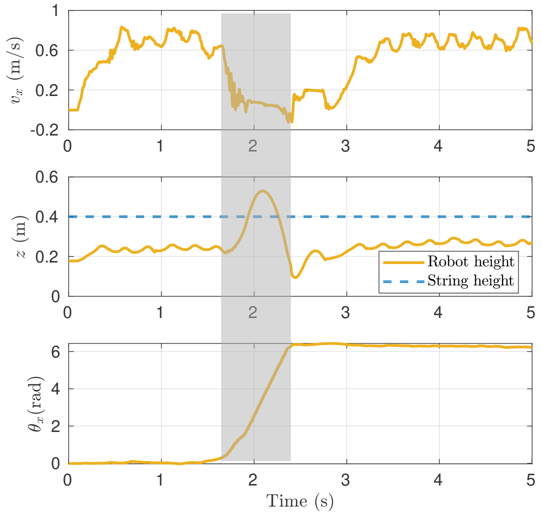


Fig. 16: Forward velocity, height, and roll angle of the Mini Cheetah hardware during the mid-run barrel roll. The dashed line indicates the height of the string over which the robot jumps.

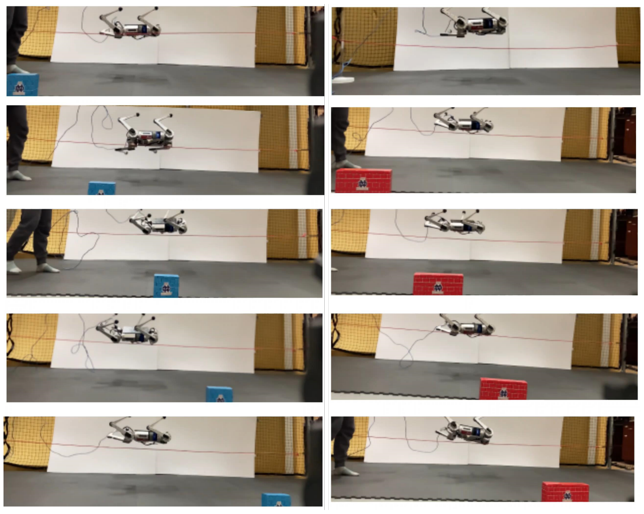


Fig. 17: Reliability test of performing a mid-run barrel roll on Mini Cheetah. 22 experiments were conducted with a success rate of 72.6%. 10 successful experiments are shown here, where the blue and red boxes at different locations indicate the experiments were performed independently.

robot alternates between left and right legs (pacing) at a certain speed, decelerates while holding the right legs, and suddenly makes a barrel roll. Pacing is resumed with the same speed after landing and a hopping step. Note that there is no intermediate full stance before the barrel roll. The composed trajectory is obtained by simply terminating the pacing reference with left legs at stance, starting the barrel roll reference with right legs at stance, and connecting them without additional effort. The solver takes roughly 5.3 ms per iteration and performs 3.6 iterations on average per MPC update. Note that CAFE-MPC is configured with 25 time steps for the whole-body plan and 10 time steps for the SRB plan. Figure 13 depicts the composed reference and the executed trajectory obtained in simulation. The green circle represents the discontinuity of the composed reference. CAFE-MPC is capable of synthesizing a smooth transition online to deviate from this discontinuity. The jump in the actual state at the end of the barrel roll arises from the impact at touchdown. Figure 14 shows time-series snapshots of the executed motion on the robot hardware. As shown in figure 16, the robot jumps up to 0.52 m, and clears the string at 0.4 m high. The attained clearance height matches that of the MIT Cheetah 2 [2], which is twice the size of the Mini Cheetah, though the jumping manners are different. The robot runs at up to 0.8 m/s before and after the barrel roll. Figure 15 depicts the torque measurements, indicating that the torque limits are almost always satisfied with small violations at a few moments. We note that there are relatively large torque oscillations on the right stance legs at taking off. These oscillations happen at about 50 Hz, close to the CAFE-MPC update frequency. A hypothesis to explain these oscillations thus is that the control bandwidth of CAFE-MPC is not sufficient to resolve the fast dynamics at take-off.

3) Reliability Test: We repeat the Fosbury-Flop experiment 22 times to evaluate the reliability of the proposed control framework. Sixteen experiments are successful, resulting in a success rate of 72.6%. Figure 17 depicts screenshots of 10 successful experiments, where the blue and red boxes at different locations indicate these experiments were conducted independently. While two failed tests are due to low battery, and one due to temporary loss of Ethernet connection, the rest of the failed experiments were mainly caused by the sliding of the stance feet immediately before the barrel roll. This sliding arises from the oscillation (similar as observed in figure 15) of the stance legs that break the friction constraints. We believe that once the oscillation problem is resolved, the reliability of the proposed control framework will be further improved.

D. Comparison to Whole-Body MPC

In a final investigation, we compare CAFE-MPC with whole-body MPC regarding their capability to achieve a Fosbury-Flop. In Section VIII-B, the barrel-roll reference for the Fosbury-Flop was obtained via offline TO, with a period of flight time 0.4 s. In this section, we extend the flight time by 0.1 s online without re-optimizing the reference offline. To obtain this new reference, the last state of the flight phase of the original barrel-roll reference is repeated until the length of the new plan is reached. This hand-crafted reference is harder to track since it breaks the dynamics feasibility. However, if it is trackable, it would make the reference design process easier, and enable online generalization to different the flight times.

Table II shows the model schedules of the participating MPC schemes used in the comparison study. The notation $\mathcal{S}$ represents model schedule, which specifies the lengths of the whole-body plan and the SRB plan. The model schedules $\mathcal{S}_0$, $\mathcal{S}_1$, and $\mathcal{S}_2$ have zero-length SRB plan, thus representing whole-body MPC. For $\mathcal{S}_0$, $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$, the simulation runs at full speed to mimic the real-time scenario, and the real-time constraints are imposed on the solver. For $\mathcal{S}_1$-slowsim, the real-time constraint is disabled, and the simulation runs at 0.5x. For each model schedule in Table II, we repeat the Forbury-Flop task for twenty trials. Each trial is slightly different from the other due to uncertainties from (1) measurement

TABLE II: Model schedules of CAFE-MPC for performance comparison of executing Fosbury flop and success rates.

Model schedule	$\mathcal{S}_0$	$\mathcal{S}_1$	$\mathcal{S}_1$-slowsim	$\mathcal{S}_2$	$\mathcal{S}_3$
T_w (seconds)	0.70	0.80	0.80	0.90	0.40
T_s (seconds)	0.00	0.00	0.00	0.00	0.50
Success rate	15%	35%	95%	30%	95%

noises (2) variations in controller frequency and (3) run-time variations of the solver. A trial is considered to fail if the robot cannot recover locomotion after landing.

The success rates achieved with each participating MPC scheme are reported in Table II. With the real-time constraints, CAFE-MPC ($\mathcal{S}_3$) shows a significantly better success rate than all the whole-body MPC controllers ($\mathcal{S}_0$, $\mathcal{S}_1$, and $\mathcal{S}_2$). Among the whole-body MPC controllers, better performance is achieved with $\mathcal{S}_1$ and $\mathcal{S}_2$ than $\mathcal{S}_0$, which is due to the extended planning horizon. Note that even though the whole-body MPC schemes $\mathcal{S}_1$ and $\mathcal{S}_2$ employ a similar-length overall prediction horizon as CAFE-MPC , they achieve lower success rates. This loss of performance is because the whole-body MPC problems are harder to solve and may not find feasible solutions given limited computation times. This argument is verified with $\mathcal{S}_1-$slowsim where the real-time constraints are removed, and the resulting success rate is comparable to CAFE-MPC . In a nutshell, for whole-body MPC to synthesize the running barrel roll, the planning horizon needs to be sufficiently long, which on the other hand, increases the computational difficulty, thus impeding the control performance. Speeding up computation time for whole-body MPC is a vibrant research area [13], [15]. We anticipate that whole-body MPC has the potential to execute a running barrel roll on hardware in the foreseeable future. Nonetheless, it's worth noting that our approach offers a straightforward yet powerful means of integrating whole-body planning into MPC and making the best use of available computational resources.

VIII. Conclusions and Future Work

A. Discussions and Future Works

The results in Section VII-A show the enhanced performance of CAFE-MPC with the increased horizon of the SRB plan. This improvement is explained as follows. Minimizing the value function obtained at the initial state of the SRB plan provides a proxy for ensuring viability and long-term tracking performance. When this value function is used as a terminal cost, the WB plan has a better idea of how the current decision would affect the long-term performance. This approach is similar to the finite-horizon LQR when the terminal cost is given by the value function of an infinite-horizon LQR.

There are a few limitations with the CAFE-MPC . The orientation is represented by Euler angles, which are notoriously known for the singularity problem. Future work would use quaternions [22], [98], and unlock more gymnastic maneuvers such as mid-run front flip. Self-collision is currently not considered, which can cause the robot to fall during leg crossing. Contact timing optimization is another line of future research to account for contact mismatch. Our previous work [84] and current research on timing-free optimization show promise in this direction. Further, the computational efficiency of CAFE-MPC can further be improved with the real-time iteration scheme [58].

The VWBC disguises the Riccati feedback controller in a QP problem. Results in Section VII-B demonstrate that the VWBC enjoys the benefits of Riccati feedback closing the control loop at a higher rate, while at the same time satisfying necessary constraints. This feature allows the high-level CAFE-MPC to be solved at coarse precision. One limitation of VWBC is that the foot contacts are currently specified using a known schedule, which can cause instability problems in the presence of a large contact mismatch. Future work would seed VWBC with contact detection [99].

B. Conclusion

The major contribution of this work is the control pipeline of CAFE-MPC + VWBC. CAFE-MPC enables flexible scheduling of multi-fidelity models, multi-resolution time steps, and relaxed constraints along the prediction horizon. When compared to the whole-body MPC on the regular locomotion tasks, CAFE-MPC was shown to achieve better tracking performance due to an appended template SRB plan that grows the prediction horizon. Further, this performance enhancement for CAFE-MPC does not necessarily increase the computational time with the appropriate design of the SRB plan, which is beneficial for on-board computing with limited computational resources. Thanks to the multiple-shooting iLQR, CAFE-MPC computes for free a local Q-value function for the current state of the whole-body model. This local Q-value function measures the long-horizon effects of the current control perturbations on the resulting cost. The VWBC minimizes this value function so that the resulting solution is encouraged to stay near the optimal path discovered by MPC. As a result, the VWBC is free of additional tuning, which is an advantage over conventional QP whole-body controllers.

The proposed method is sufficiently powerful that it can synthesize multiple highly dynamic and complex behaviors on the fly. This capability is demonstrated via performing common locomotion skills, unusual dynamic running jump, and extremely dynamic running barrel rolls on MIT Mini Cheetah. The reference for running barrel roll is obtained by simply connecting an in-place barrel roll trajectory to a pacing gait, without sophisticated processing. The running barrel roll experiment was repeated 22 times on the Mini Cheetah hardware, and a 72.6% success rate was achieved. Most failures were caused by the sliding of the stance feet. A template MPC that does not incorporate leg momentum is shown with failure to accomplish an in-place barrel roll in simulation, indicating the importance of incorporating whole-body dynamics for highly dynamic motions. Further, a whole-body MPC is shown with less success rates than CAFE-MPC due to the excessive computational cost. Beyond quadrupeds, the proposed control pipeline is very general and could be applied to many other robotic platforms. Further work will investigate its application to humanoid robots, in particular.

Ackowledgements

We would like to thank David Kelly, Shenggao Li, Nicolas Adrian, John Nganga, and Xuemin Liu for their help on the experimental setup and video shooting, Prof. Hai Lin and Zihao Song for lending the lab space, and Dr. Tingnan Zhang and Dr. Wenhao Yu for their helpful discussions. The Mini Cheetah is sponsored by the MIT Biomimetic Robotics Lab and NAVER LABS.

References

[1] 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 Intelligent Robots and Systems*, 2018, pp. 2245–2252.

[2] H.-W. Park, P. M. Wensing, and S. Kim, "High-speed bounding with the MIT cheetah 2: Control design and experiments," *Int. J. of Robotics Research*, vol. 36, pp. 167–192, 2017.

[3] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *Int. Conf. on Robotics and Automation*, 2019, pp. 6295–6301.

[4] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, 2023.

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

[6] S. Hong, Y. Um, J. Park, and H.-W. Park, "Agile and versatile climbing on ferromagnetic surfaces with a quadrupedal robot," *Science Robotics*, vol. 7, no. 73, p. eadd1017, 2022.

[7] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 8, no. 74, p. eade2256, 2023.

[8] E. Ackerman. (2022) Watch the hyqreal robot pull an airplane. [Online]. Available: https://spectrum.ieee.org/watch-iits-new-hyqreal-quadruped-robot-pull-an-airplane

[9] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini, "MPC-based controller with terrain insight for dynamic legged locomotion," in *IEEE Int. Conf. on Robotics and Automation*, 2020, pp. 2436–2442.

[10] M. Chignoli and S. Kim, "Online trajectory optimization for dynamic aerial motions of a quadruped robot," in *IEEE Int. Conf. on Robotics and Automation*, 2021, pp. 7693–7699.

[11] Z. Song, L. Yue, G. Sun, Y. Ling, H. Wei, L. Gui, and Y.-H. Liu, "An optimal motion planning framework for quadruped jumping," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 366–11 373.

[12] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, and G. Martius, "Learning agile skills via adversarial imitation of rough partial demonstrations," in *Conference on Robot Learning*. PMLR, 2023, pp. 342–352.

[13] C. Mastalli, S. P. Chhatoi, T. Corbéres, S. Tonneau, and S. Vijayakumar, "Inverse-dynamics mpc via nullspace resolution," *IEEE Transactions on Robotics*, 2023.

[14] E. Dantec, M. Naveau, P. Fernbach, N. Villa, G. Saurel, O. Stasse, M. Taïx, and N. Mansard, "Whole-body model predictive control for biped locomotion on a torque-controlled humanoid robot," in *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2022, pp. 638–644.

[15] W. Jallet, A. Bambade, E. Arlaud, S. El-Kazdadi, N. Mansard, and J. Carpentier, "Proxddp: Proximal constrained trajectory optimization," 2023.

[16] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 1–9.

[17] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, "Biconmp: A nonlinear model predictive control framework for whole body motion planning," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 905–922, 2023.

[18] Y.-M. Chen, J. Hu, and M. Posa, "Beyond inverted pendulums: Task-optimal simple models of legged locomotion," *arXiv preprint arXiv:2301.02075*, 2023.

[19] J. Li, J. Ma, O. Kolt, M. Shah, and Q. Nguyen, "Dynamic loco-manipulation on hector: Humanoid for enhanced control and open-source research," *arXiv preprint arXiv:2312.11868*, 2023.

[20] Z. Gu, R. Guo, W. Yates, Y. Chen, and Y. Zhao, "Walking-by-logic: Signal temporal logic-guided model predictive control for bipedal locomotion resilient to external perturbations," *arXiv preprint arXiv:2309.13172*, 2023.

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

[22] H. Li, R. J. Frei, and P. M. Wensing, "Model hierarchy predictive control of robotic systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3373–3380, 2021.

[23] H. Li, W. Yu, T. Zhang, and P. M. Wensing, "A unified perspective on multiple shooting in differential dynamic programming," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9978–9985.

[24] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 137–142.

[25] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 4906–4913.

[26] 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*, vol. 40, no. 3, pp. 429–455, 2016.

[27] G. Bledt and S. Kim, "Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 6316–6323.

[28] T. Apgar, P. Clary, K. Green, A. Fern, and J. W. Hurst, "Fast online trajectory optimization for the bipedal robot cassie." in *Robotics: Science and Systems*, 2018.

[29] M. Hutter, H. Sommer, C. Gehring, M. Hoepflinger, M. Bloesch, and R. Siegwart, "Quadrupedal locomotion using hierarchical operational space control," *The Int. Journal of Robotics Research*, vol. 33, pp. 1047–1062, 2014.

[30] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3d-slip model," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 5134–5140.

[31] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT cheetah," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 4102–4109.

[32] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[33] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, pp. 161–176, 2013.

[34] 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*, 2014, pp. 295–302.

[35] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *Int. Journal of Humanoid Robotics*, vol. 13, 2016.

[36] G. Romualdi, S. Dafarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online non-linear centroidal mpc for humanoid robot locomotion with step adjustment," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 412–10 419.

[37] 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 Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[38] C. Mastalli, W. Merkt, J. Marti-Saumell, H. Ferrolho, J. Solà, N. Mansard, and S. Vijayakumar, "A feasibility-driven approach to control-limited ddp," *Autonomous Robots*, pp. 1–21, 2022.

[39] S. Katayama and T. Ohtsuka, "Structure-exploiting newton-type method for optimal control of switched systems," *International Journal of Control*, no. just-accepted, p. 1, 2023.

[40] A. Jordana, S. Kleff, A. Meduri, J. Carpentier, N. Mansard, and L. Righetti, "Stagewise implementations of sequential quadratic programming for model-predictive control," 2023.

[41] J. Carpentier and N. Mansard, "Analytical derivatives of rigid body dynamics algorithms," in *Robotics: Science and systems*, 2018.

[42] S. Singh, R. P. Russell, and P. M. Wensing, "Efficient analytical derivatives of rigid-body dynamics using spatial vector algebra," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1776–1783, 2022.

[43] J. Wang, S. Kim, S. Vijayakumar, and S. Tonneau, "Multi-fidelity receding horizon planning for multi-contact locomotion," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 53–60.

[44] Y. Liu, J. Shen, J. Zhang, X. Zhang, T. Zhu, and D. Hong, "Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8547–8553.

[45] C. Khazoom and S. Kim, "Humanoid arm motion planning for improved disturbance recovery using model hierarchy predictive control," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6607–6613.

[46] C. Khazoom, S. Heim, D. Gonzalez-Diaz, and S. Kim, "Optimal scheduling of models and horizons for model hierarchy predictive control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9952–9958.

[47] J. Norby, A. Tajbakhsh, Y. Yang, and A. M. Johnson, "Adaptive complexity model predictive control," *arXiv preprint arXiv:2209.02849*, 2022.

[48] S. Shin and V. M. Zavala, "Diffusing-horizon model predictive control," *IEEE Transactions on Automatic Control*, 2021.

[49] T. Brüdigam, D. Prader, D. Wollherr, and M. Leibold, "Model predictive control with models of different granularity and a non-uniformly spaced prediction horizon," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3876–3881.

[50] V. A. Laurense and J. C. Gerdes, "Long-horizon vehicle motion planning and control through serially cascaded model complexity," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 166–179, 2022.

[51] G. Kim, D. Kang, J.-H. Kim, S. Hong, and H.-W. Park, "Contact-implicit mpc: Controlling diverse quadruped motions without pre-planned contact modes or trajectories," *arXiv preprint arXiv:2312.08961*, 2023.

[52] V. Kurtz, A. Castro, A. Ö. Önol, and H. Lin, "Inverse dynamics trajectory optimization for contact-implicit model predictive control," *arXiv preprint arXiv:2309.01813*, 2023.

[53] S. Le Cleac'h, T. A. Howell, S. Yang, C.-Y. Lee, J. Zhang, A. Bishop, M. Schwager, and Z. Manchester, "Fast contact-implicit model predictive control," *IEEE Transactions on Robotics*, 2024.

[54] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*, 2006, pp. 65–93.

[55] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.

[56] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

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

[58] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on control and optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[59] 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.

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

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

[62] 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 Intelligent Robots and Systems*, pp. 3346–3351.

[63] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1502–1509, 2017.

[64] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 1168–1175.

[65] 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 Robotics and Automation*, 2017, pp. 93–100.

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

[67] T. A. Howell, B. E. Jackson, and Z. Manchester, "ALTRO: a fast solver for constrained trajectory optimization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 7674–7679.

[68] G. Lantoine and R. P. Russell, "A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 1: theory," *Journal of Optimization Theory and Applications*, vol. 154, no. 2, pp. 382–417, 2012.

[69] B. Plancher, Z. Manchester, and S. Kuindersma, "Constrained unscented dynamic programming," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 5674–5680.

[70] 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 Intelligent Robots and Systems*, 2018, pp. 1–9.

[71] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 2589–2594.

[72] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 505–518, 2005.

[73] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[74] H. Li, T. Zhang, W. Yu, and P. M. Wensing, "Versatile real-time motion synthesis via kino-dynamic mpc with hybrid-systems ddp," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9988–9994.

[75] 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.

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

[77] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," 2020.

[78] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4630–4637, 2022.

[79] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," 2023.

[80] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," *arXiv preprint arXiv:2309.14341*, 2023.

[81] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *arXiv preprint arXiv:2205.02824*, 2022.

[82] Y. Yang, G. Shi, X. Meng, W. Yu, T. Zhang, J. Tan, and B. Boots, "Cajun: Continuous adaptive jumping using a learned centroidal controller," *arXiv e-prints*, pp. arXiv–2306, 2023.

[83] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "Dtc: Deep tracking control." *Science Robotics*, vol. 9, no. 86, pp. eadh5401–eadh5401, 2024.

[84] H. Li and P. M. Wensing, "Hybrid systems differential dynamic programming for whole-body motion planning of legged robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5448 – 5455, 2020.

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

[86] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[87] J. Hauser and A. Saccon, "A barrier function method for the optimization of trajectory functionals with constraints," in *IEEE Conference on Decision and Control (CDC)*, 2006, pp. 864–869.

[88] S. Singh, R. P. Russell, and P. M. Wensing, "On second-order derivatives of rigid-body dynamics: Theory & implementation," *IEEE Transactions on Robotics (to appear)*, 2024.

[89] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer methods in applied mechanics and engineering*, vol. 1, no. 1, pp. 1–16, 1972.

[90] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansarc
dynamic programming for multi-phase rigid contact dynar
RAS 18th Int. Conf. on Humanoid Robots (Humanoids),

[91] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel,
O. Stasse, and N. Mansard, “The pinocchio c++ librar
flexible implementation of rigid body dynamics algorit
analytical derivatives,” in *IEEE International Symposi*
Integrations (SII), 2019.

[92] K. J. Waldron and J. Schmiedeler, “Kinematics,” *Spring*
robotics, pp. 11–36, 2016.

[93] X. Xiong and A. Ames, “3-d underactuated bipedal wa
based gait synthesis and stepping stabilization,” *IEEE T*
Robotics, vol. 38, no. 4, pp. 2405–2425, 2022.

[94] M. Focchi, G. A. Medrano-Cerda, T. Boaventura, M. Frige
J. Buchli, and D. G. Caldwell, “Robot impedance contro
analysis with inner torque and velocity feedback loops,”
and Technology, vol. 14, pp. 97–112, 2016.

[95] M. Posa, C. Cantu, and R. Tedrake, “A direct method
optimization of rigid bodies through contact,” *The I*
Robotics Research, vol. 33, pp. 69–81, 2014.

[96] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

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

[98] B. E. Jackson, K. Tracy, and Z. Manchester, “Planning with attitude,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5658–5664, 2021.

[99] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, “Contact model fusion for event-based locomotion in unstructured terrains,” in *IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 4399–4406.

Appendix: Offline TO Design of In-Place Barrel Roll

The contact sequence for the in-place barrel roll is given by {FS, FR-HR, FT, FS}, meaning full-stance → right-stance → flight → full-stance. Further, we append a recovery step ({FT, FS}) to the in-place barrel roll, in case the stance phase after landing is not sufficient to balance the robot. This choice is motivated by the additional steps that are often taken by human gymnasts after landing to assist in balancing. The contact timings for the barrel roll are heuristically determined by mimicking that of a backflip in previous work [3].

The in-place barrel roll TO employs the whole-body dynamics. It shares almost the same problem structure as the whole-body planning problem of Cafe-Mpc (Section IV-A), enabling tailoring available code for this specific task, and significantly reducing the amount of engineering efforts. There are two differences. First, we add to the in-place barrel roll TO a minimum body-height constraint at the full-stance phase after landing to avoid ground collision. Second, rather than tracking a reference trajectory, we manually design a few keyframes, one for each phase. The running cost and terminal cost of each phase are simply to track the allocated keyframes (Fig. 18). The swing trajectory cost and the foot placement cost are not used in the barrel roll TO.

The design of the keyframes starts with the floating base and then associates it with appropriate joint angles. The robot experiences a 2π rotation during the barrel roll. The roll rate is assumed constant and calculated by averaging 2π over the barrel roll duration. The lateral velocity is assumed constant and is given by the user. With the minimum landing height and flight duration, the vertical speed and position are determined assuming parabolic motion. Yaw and pitch angles are assumed to be always zero. An illustration of the keyframes is shown in Fig. 18, drawn from the front view of the robot. For stance legs in keyframes 1 and 2, the joint angles are solved via inverse kinematics. For swing legs in keyframes 2 and 3, the joint angles are set to a reasonable landing configuration. Joint angles in keyframe 4 are set via a default standing pose.

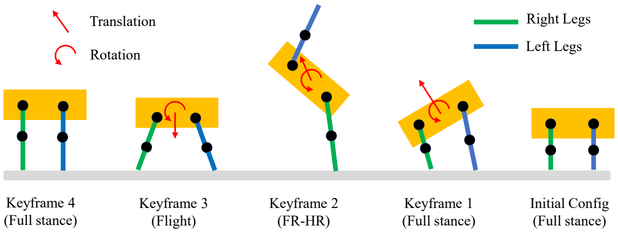
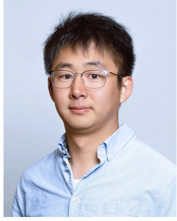


Fig. 18: Illustration of keyframes used for barrel roll TO, depicted from the front view of the robot. Each keyframe is the desired terminal state for the associated phase (as labeled in parenthesis).

The resulting trajectory obtained from this offline TO is then used as a reference and an initial guess for the online Cafe-Mpc. Though the offline TO is designed for the in-place barrel roll, the same trajectory is used to seed the running barrel roll online for multiple contact configurations. More specifically, the locomotion references Section VI-A are placed ahead and/or appended to the in-place barrel roll trajectory with simple manipulations, for instance, the roll angles of the appending trajectory are shifted by 2π. Though there are no advanced techniques used here to glue the trajectories together, Cafe-Mpc plays a role of online synthesizing a smooth transition.



He Li received the M.S. degree (2019) from Missouri University of Science and Technology, Rolla, USA, and the Ph.D. degree (2024) from the University of Notre Dame, USA, supervised by Prof. Patrick Wensing. He is currently a robotics research scientist at the Boston Dynamics AI Institute.

His research interest is in numerical optimal control, with application to highly dynamic motion planning and control of legged robots.

Patrick M. Wensing (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from The Ohio State University, Columbus, OH, USA, in 2009, 2013, and 2014, respectively.

He is currently the Wanzek Family Foundation Associate Professor in the Department of Aerospace and Mechanical Engineering, University of Notre Dame, where he directs the Robotics, Optimization, and Assistive Mobility (ROAM) Laboratory. Before joining Notre Dame, he was a Postdoctoral Associate with MIT, working on control system design for the MIT cheetah robots. His current research interests include aspects of dynamics, optimization, and control toward advancing the mobility of legged robots and assistive devices. He received the NSF CAREER Award in 2020. He currently serves as a Co-Chair for the IEEE RAS Technical Committee on Model-Based Optimization for Robotics and as a Senior Editor for the IEEE Transactions on Robotics.