

Momentum-Aware Trajectory Optimization and Control for Agile Quadrupedal Locomotion

Ziyi Zhou¹, Student Member, IEEE, Bruce Wingo¹, Student Member, IEEE,
Nathan Boyd¹, Student Member, IEEE, Seth Hutchinson¹, Fellow, IEEE, and Ye Zhao¹, Member, IEEE

Abstract—In this letter, we present a versatile hierarchical offline planning algorithm, along with an online control pipeline for agile quadrupedal locomotion. Our offline planner alternates between optimizing centroidal dynamics for a reduced-order model and whole-body trajectory optimization, with the aim of achieving dynamics consensus. Our novel momentum-inertia-aware centroidal optimization, which uses an equimomental ellipsoid parameterization, is able to generate highly acrobatic motions via “inertia shaping”. Our whole-body optimization approach significantly improves upon the quality of standard DDP-based approaches by iteratively exploiting feedback from the centroidal level. For online control, we have developed a novel convex model predictive control scheme through a linear transformation of the full centroidal dynamics. Our controller can efficiently optimize for both contact forces and joint accelerations in single optimization, enabling more straightforward tracking for momentum-rich motions compared to existing quadrupedal MPC controllers. We demonstrate the capability and generality of our trajectory planner on four different dynamic maneuvers. We then present one hardware experiment on the MIT Mini Cheetah platform to demonstrate the performance of the entire planning and control pipeline on a twisting jump maneuver.

Index Terms—Whole-body motion planning and control, legged robots, optimization and optimal control.

I. INTRODUCTION

WITH the recent rapid increase in capabilities of legged robots, the demand for more sophisticated approaches to trajectory optimization and online control has motivated a plethora of optimization-based motion planning strategies. Broadly speaking, these can be divided into methods that solve a single trajectory optimization (TO) problem, and those that decompose the overall problem into a set of decoupled subproblems. Single-optimization methods can be further subdivided into those that use reduced order models and those that optimize over the full-order dynamics.

Manuscript received 24 February 2022; accepted 7 June 2022. Date of publication 23 June 2022; date of current version 1 July 2022. This letter was recommended for publication by Associate Editor P. M. Wensing and Editor A. Kheddar upon evaluation of the reviewers' comments. This work was supported in part by NSF under Grant IIS-1924978, and in part by the Georgia Tech Research Institute IRAD Grant. (Ziyi Zhou and Bruce Wingo contributed equally to this work.) (Corresponding author: Ye Zhao.)

The authors are with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: zhouziyi@gatech.edu; bwingo@gatech.edu; nboyd2226@gmail.com; seth@gatech.edu; yezhao@gatech.edu).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2022.3185374>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3185374

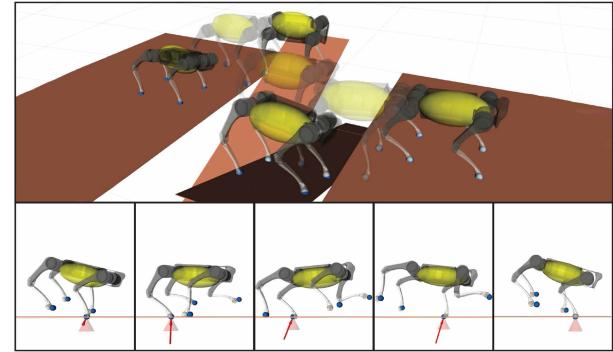


Fig. 1. Dynamic maneuvers that require high-fidelity control of angular momentum, such as “parkour” (top) and cantering (bottom).

Reduced-order models, such as the spring loaded inverted pendulum (SLIP) model, single rigid body model (SRBM), centroidal dynamics model, etc., have been widely adopted. High speed and robust running are demonstrated in [1]–[3], using SRBM and user defined heuristics; however, their framework can only generate cyclic and short-horizon gaits. To generate more complex motions, Winkler *et al.* proposed TOWR [4], which is capable of directly optimizing over contact sequences and timings, through its phase-based parameterization. Incorporating joint information to reason about the centroidal inertia is non-trivial with these models, which limits their ability to generate angular-momentum-rich motions using limb movements such as the maneuvers shown in Fig. 1. TOWR+ [5] and the lumped-leg SRBM [6] attempted to augment the standard SRBM by approximating the time-varying inertia tensor; however, neither use the full capabilities of the joint motions. Earlier works, such as [7] and [8] derived analytical models, known as equimomental ellipsoids, to parameterize centroidal inertia; however, discovering physically meaningful inertia trajectories solely through the ellipsoid parameterization has proven difficult.

It is also possible to use full-order dynamic models in single-optimization approaches. One option is to use full dynamics with either a hybrid [9]–[11] or contact-implicit [12]–[15] formulation. Alternatively, centroidal dynamics are utilized in [16]–[18] with full kinematics through the centroidal momentum matrix (CMM) [19]. Notably, the differential dynamic programming (DDP)-based indirect approaches, such as FDDP [10] and constrained SLQ [18], have shown impressive results in terms of computational efficiency and constraint handling. Additional efforts have been made in [20]–[22] to enforce general equality and inequality constraints; however, a both efficient and general-purpose solver, capable of handling nonlinear full-order models, does not yet exist, especially for highly agile acrobatic motions.

By decomposing trajectory optimization into simpler subproblems, it is possible to leverage the benefits of both reduced and full-order models. One strategy is through a hierarchical optimization. In [23], [24], a centroidal optimization solves for momentum and contact force trajectories, then inverse kinematics is applied to generate joint motions. Similarly, Nguyen *et al.* [25] proposed a pipeline, first solving an SRBM-based centroidal optimization, then using its solution as the tracking objective for a full dynamics-based whole-body optimization. However, as pointed out by [26], additional constraints projecting whole-body information to the centroidal level are required to ensure feasible solutions, which are often computationally expensive and difficult to define, especially when involving angular momentum (AM).

An alternative strategy is to iteratively solve the two optimizations in an alternating fashion until the convergence of a common set of dynamics variables is achieved (dynamics consensus) [26]–[29]. To date, existing frameworks compromise in either of the two optimizations: only full kinematics models (instead of dynamics ones) are considered in [27], [28] during its whole-body optimization, while the centroidal optimizations proposed in [26], [29] did not explicitly optimize over foot placements. To our knowledge, no alternating optimization frameworks currently exist that both optimize over contact and use inertia-aware reduced order models at the centroidal level, while formulating constrained full dynamics DDP at the whole-body level.

In this letter, we introduce a new alternating centroidal and whole-body optimization framework. Compared to existing alternating frameworks, we improve the centroidal optimization by incorporating the equimomental ellipsoid parameterization of centroidal inertia [8] with SRBM, and by leveraging the inertia feedback from whole-body optimization to enable a wide range of motions previously unachievable with standard centroidal optimization (e.g., “inertia shaping” [8] that aims to reach a goal pose through precise control of centroidal inertia). In addition, our whole-body optimization is designed to only incorporate minimal constraints to track nominal trajectories from the centroidal level in its objective; therefore, efficient methods such as DDP can be used. We ensure solution feasibility, in terms of constraint satisfaction, by handling most task-space constraints at the centroidal level via direct collocation, and improve solution accuracy at the whole-body level through dynamics consensus.

In order to implement our method on real robots, we have developed a novel discrete-time finite-horizon model predictive tracking controller (MPC). Many predictive tracking controllers exist in the legged locomotion control literature; however, to achieve reasonable control frequency, it is common to use reduced order models within the problem formulation [1], [3]. Although efficient and capable, these controllers are primarily designed for stabilizing fast periodic gaits, and often require expertly designed cost heuristics to generate desired motion [3]. These controllers also do not explicitly solve for joint variables, which makes precisely tracking angular-momentum-rich joint motions impossible without introducing additional layers of whole-body kinematics/dynamics optimization [2], [6]. Our implementation of MPC aims to address these short comings. Specifically, the SRBM is augmented with centroidal momentum terms, and the joint variables are introduced through the CMM. With some mild assumptions, the proposed model can be linearized, which not only allows for an efficient QP formulation of the MPC similar to [1], but also grants

us direct control over the joint variables for momentum tracking.

Our core contributions are highlighted as follows:

- We introduce an alternating centroidal, whole-body optimization scheme targeting a wide range of motions. Inertia shaping tasks can be performed due to our inertia-aware SRBM based centroidal optimization. We also demonstrate improvement in solution feasibility compared to the standard whole-body optimization, via dynamics consensus.
- We formulate a novel convex model predictive controller by modifying the original [1] with a linear transformation of the standard full centroidal dynamics. Our MPC is capable of tracking momentum-rich motions by jointly optimizing over contact forces and joint accelerations, while still retaining model simplicity for fast online computation.
- We show that the proposed trajectory optimization pipeline generates high-fidelity contact and momentum-rich agile maneuvers, and demonstrate the capability of our controller through numerical simulations and one experiment performing a 180-degree twisting jump.

II. SYSTEM OVERVIEW

We propose a hierarchical planning and control framework in this letter, as shown in Fig. 2. The inputs to our framework consist of the desired motions, specified by the user (e.g. trot, canter, twisting jump, etc.), along with the associated contact sequence, total motion duration, initial and final poses, and the terrain map. The offline trajectory planner solves centroidal and whole-body optimizations in an alternating fashion, until dynamics consensus is achieved. The resulting centroidal momentum, base motion, and joint trajectories are modified and tracked online using the proposed full centroidal MPC, which re-computes foot contact forces and joint accelerations based on feedback from an state estimator. The contact forces and joint accelerations are then sent to an inverse dynamics block for computing the feedforward joint torques, which are combined with the feedback term and passed to the PD.

A. System Modeling

Consider the standard floating base model of a legged robot, with an unactuated 6-DoF base and a set of n -DoF fully-actuated limbs. The equations of motion is given by:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} + \mathbf{C}(\mathbf{q}, \mathbf{v}) = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{B}} \boldsymbol{\tau} + \mathbf{J}_c^\top \mathbf{F}_c \quad (1)$$

where $\mathbf{q} = [\mathbf{q}_b^\top \ \mathbf{q}_j^\top]^\top \in \mathbb{R}^{n_q}$, $\mathbf{v} = [\boldsymbol{\nu}_b^\top \ \dot{\mathbf{q}}_j^\top]^\top \in \mathbb{R}^{n_v}$ are the generalized coordinates and velocities partitioned in base and joint variables. The subscripts b and j are base and joint related quantities, respectively. Base coordinates $\mathbf{q}_b = [\mathbf{p}_b^\top \ \boldsymbol{\theta}_b^\top]^\top \in \mathbb{R}^6$ are partitioned as base position and orientation. Base velocities $\boldsymbol{\nu}_b = [\dot{\mathbf{p}}_b^\top \ \boldsymbol{\omega}^\top]^\top \in \mathbb{R}^6$ are partitioned as base linear and angular velocities in the world frame. $\mathbf{M}(\mathbf{q})$ is the joint space mass matrix. $\mathbf{C}(\mathbf{q}, \mathbf{v})$ captures the nonlinear effects. $\boldsymbol{\tau} \in \mathbb{R}^{n_j}$ denotes joint torques. $\mathbf{J}_c(\mathbf{q})$ is the stacked contact Jacobian and \mathbf{F}_c is the stacked contact reaction force vector. Without loss of generality, an implicit assumption $\mathbf{v} = \dot{\mathbf{q}}$ is made for the subsequent development of centroidal and whole-body optimization. This model will be referred to as the **full dynamics** of the robot.

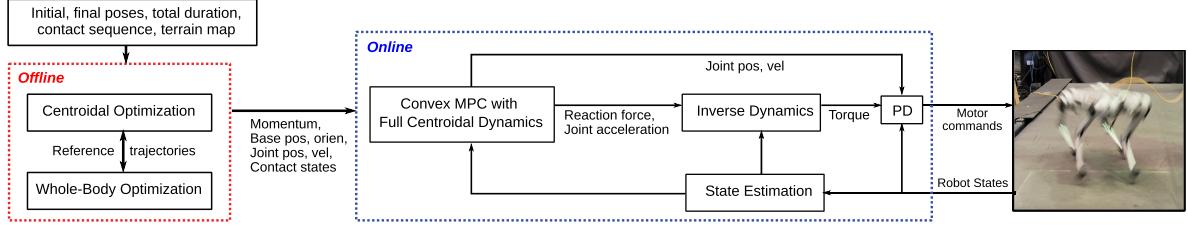


Fig. 2. Overall planning and control framework that consists of offline trajectory generation (red) and online execution (blue). The input to offline block is specifically decided by the desired motion. Full-body, centroidal, and contact information are passed from offline generation to online control.

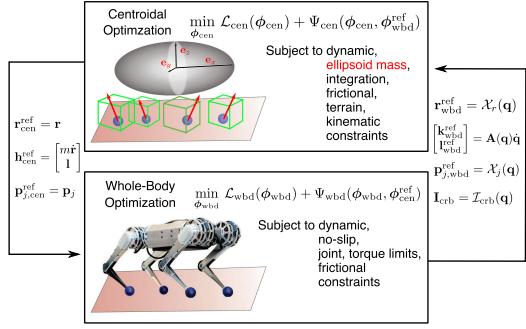


Fig. 3. The block diagram that demonstrates the proposed alternating centroidal, whole-body optimization.

Assuming full control authority over the joint variables q_j , (1) can be converted into Newton-Euler equations of motion about the robot's center-of-mass (CoM) [30]:

$$\dot{h} = \begin{bmatrix} \dot{k} \\ \dot{l} \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^{n_f-1} f_j + mg \\ \sum_{j=0}^{n_f-1} (p_j - r) \times f_j \end{bmatrix} \quad (2)$$

where it is assumed that the robot has n_f feet with point contacts. $h = [k^T \ 1^T]^T$ are the linear and angular centroidal momentum. p_j and r are the individual contact and CoM position in the world frame. f_j are the individual foot contact forces. This reduced-order model will be referred to as the **centroidal dynamics** throughout this letter.

III. OFFLINE TRAJECTORY GENERATION

We present an alternating centroidal, whole-body trajectory optimization scheme in this section. Detailed formulations for individual components are provided in Sections III-A and III-B. The alternating update scheme and the importance of dynamics consensus are discussed in Section III-C.

A. Centroidal Trajectory Optimization

Formulation 1 details the TO problem to be solved at the centroidal level, which utilizes the **centroidal dynamics** in (2). We transcribe the optimization problem into a nonlinear program through direct collocation. Total duration T of the desired motion is equally divided into N knot points.

The centroidal decision set ϕ_{cen} consists of CoM position r , CoM linear velocity \dot{r} , angular excursion with Euler angle parameterization θ and its rate $\dot{\theta}$, centroidal angular momentum l and its rate \dot{l} , j^{th} foot's position p_j and contact force f_j . e and γ are inertia variables characterizing the principle semi-axes and orientation of the equimomental ellipsoid [8] (illustrated in Fig. 3). Ellipsoid orientation is again parameterized using Euler

angles. All components of ϕ_{cen} belongs to \mathbb{R}^3 . The optimization objective contains a user-defined cost $\mathcal{L}_{cen}(\cdot)$ which specifies various heuristics, such as penalizing the velocity of foot position to discourage violent movements. In addition, a tracking cost $\Psi_{cen}(\phi_{cen}, \phi_{wbd})$ that minimizes the tracking error to the whole-body reference trajectories (solution to Formulation 2 and the variable ϕ_{wbd}^{ref} is defined in Formulation 1) are included. Benefits of this tracking cost will be discussed in Section III-C.

$$\begin{aligned} \Psi_{cen} = & \|r - r_{wbd}^{ref}\|_{Q_r^{cen}}^2 + \|m\dot{r} - k_{wbd}^{ref}\|_{Q_k^{cen}}^2 \\ & + \|l - l_{wbd}^{ref}\|_{Q_l^{cen}}^2 + \|I_{ellip} - I_{crb}\|_{Q_I^{cen}}^2 \\ & + \sum_{j=0}^{n_f-1} \left(\|p_j - p_{j,wbd}^{ref}\|_{Q_p^{cen}}^2 \right) \end{aligned}$$

Symmetric positive definite weighting matrices for tracking the reference CoM position r_{wbd}^{ref} , linear momentum (LM) k_{wbd}^{ref} , angular momentum (AM) l_{wbd}^{ref} , and foot positions $p_{j,wbd}^{ref}$ are denoted as Q_r^{cen} , Q_k^{cen} , Q_l^{cen} , and Q_p^{cen} , respectively. Inertia tracking is achieved through minimizing the Q_I^{cen} -weighted Frobenius norm of the difference between the inertia tensors of the equimomental ellipsoid I_{ellip} and that of the reference composite rigid body (CRB) I_{crb} , both expressed in the world frame. Constraints in centroidal optimization include:

Dynamic constraint: The discrete-time non-linear **centroidal dynamics** are used. Eq. (3a) explicitly includes the relationship between centroidal angular momentum, ellipsoid inertia tensor, and CoM angular velocity. The inertia tensor I_{ellip} in the world frame is constructed from the rotational transformation of the inertia tensor in the base frame, $I_{ellip} = R(\gamma)B I_{ellip}(e)R(\gamma)^\top$ with $R(\gamma)$ being the rotation transformation between the ellipsoid base frame and world frame. Inertia tensor in the base frame is defined as $B I_{ellip} = \text{diag}([I^{xx}, I^{yy}, I^{zz}]^T)$ with $I^{xx} = \frac{1}{5}m(e^{y^2} + e^{z^2})$, $I^{yy} = \frac{1}{5}m(e^{x^2} + e^{z^2})$, and $I^{zz} = \frac{1}{5}m(e^{x^2} + e^{y^2})$.

Ellipsoid mass constraint: To ensure the equimomental ellipsoid has the same mass as the robot, ellipsoid semi-axes, e , must be constrained. Ellipsoid mass $m = \frac{4}{3}\pi e^x e^y e^z \rho$ with ρ being the average mass density. ρ can be obtained from the reference CRB inertia tensor I_{crb} for each time step [19]. When the inertia reference is unavailable, ρ is kept constant using the initial robot configuration.

Integration constraint: By adopting trapezoidal integration scheme, the quantities r, \dot{r}, θ and l are approximated as piecewise linear functions as shown in (3c)–(3f). The final knot points are set to be free but subject to final conditions if needed. Δt denotes one time step.

Frictional constraint: Friction cone constraint and unilateral constraint on the contact forces are both modeled in (3g)–(3i).

Formulation 1: Centroidal trajectory optimization.

	$\min_{\phi_{\text{cen}}} \mathcal{L}_{\text{cen}}(\phi_{\text{cen}}) + \Psi_{\text{cen}}(\phi_{\text{cen}}, \phi_{\text{wbd}}^{\text{ref}})$
(Variables)	$\phi_{\text{cen}}[i] = [\mathbf{r}[i]^\top, \dot{\mathbf{r}}[i]^\top, \boldsymbol{\theta}[i]^\top, \dot{\boldsymbol{\theta}}[i]^\top, \mathbf{l}[i]^\top,$ $\mathbf{i}[i]^\top, \mathbf{p}_j[i]^\top, \mathbf{f}_j[i]^\top, \underbrace{\mathbf{e}[i]^\top, \boldsymbol{\gamma}[i]^\top}_{\text{inertia vectors}}$ $\phi_{\text{wbd}}^{\text{ref}}[i] = [\mathbf{r}_{\text{wbd}}^{\text{ref}}[i]^\top, \dot{\mathbf{r}}_{\text{wbd}}^{\text{ref}}[i]^\top, \mathbf{l}_{\text{wbd}}^{\text{ref}}[i]^\top,$ $\mathbf{p}_{j,\text{wbd}}^{\text{ref}}[i]^\top, \mathbf{I}_{\text{crb}}[i]^\top]$ $\forall i = 0, \dots, N-1, j = 0, \dots, n_f-1$
(Dynamics)	s.t. $\begin{bmatrix} \dot{\mathbf{r}}[i] \\ \dot{\mathbf{l}}[i] \\ \mathbf{l}[i] \end{bmatrix} = \begin{bmatrix} \sum_j \mathbf{f}_j[i] + mg \\ \sum_j (\mathbf{p}_j[i] - \mathbf{r}[i]) \times \mathbf{f}_j[i] \\ \mathbf{I}_{\text{ellip}}(\mathbf{e}[i], \boldsymbol{\gamma}[i]) \boldsymbol{\omega}[i] \end{bmatrix}$ (3a)
(Ellipsoid)	$m = \frac{4}{3} \pi \mathbf{e}^x[i] \mathbf{e}^y[i] \mathbf{e}^z[i] \rho[i]$ (3b)
(Integration)	$\mathbf{r}[i+1] - \mathbf{r}[i] = \frac{\Delta t}{2} (\dot{\mathbf{r}}[i+1] + \dot{\mathbf{r}}[i])$ (3c)
	$\dot{\mathbf{r}}[i+1] - \dot{\mathbf{r}}[i] = \frac{\Delta t}{2} (\ddot{\mathbf{r}}[i+1] + \ddot{\mathbf{r}}[i])$ (3d)
	$\boldsymbol{\theta}[i+1] - \boldsymbol{\theta}[i] = \frac{\Delta t}{2} (\dot{\boldsymbol{\theta}}[i+1] + \dot{\boldsymbol{\theta}}[i])$ (3e)
	$\mathbf{l}[i+1] - \mathbf{l}[i] = \frac{\Delta t}{2} (\dot{\mathbf{l}}[i+1] + \dot{\mathbf{l}}[i])$ (3f)
(Frictional)	$\mathbf{f}_j[i] \cdot \mathbf{n}(\mathbf{p}_j^{xy}[i]) \geq 0, \forall i \in \mathcal{C}_j$ (3g)
	$\mathbf{f}_j[i] \in \mathcal{F}(\mu, \mathbf{n}, \mathbf{p}_j^{xy}[i]), \forall i \in \mathcal{C}_j$ (3h)
	$\mathbf{f}_j[i] = 0, \forall i \notin \mathcal{C}_j$ (3i)
(Terrain)	$\mathbf{p}_j^{xy}[i+1] - \mathbf{p}_j^{xy}[i] = 0, \forall i \in \mathcal{C}_j$ (3j)
	$\mathbf{p}_j^z[i] = h_{\text{terrain}}(\mathbf{p}_j^{xy}[i]), \forall i \in \mathcal{C}_j$ (3k)
(Kinematics)	$\mathbf{p}_j[i] \in \mathcal{R}_j(\mathbf{r}[i], \boldsymbol{\theta}[i])$ (3l)

Specifically, friction cones are approximated as friction pyramids \mathcal{F} to remove nonlinearity. The contact normal forces are non-negative, and always equal to zero during the swing phase.

Terrain constraint: For each foot during the stance phase, (3j) defines the no-slip condition. While (3k) guarantees that foot placement remains on the terrain surface, as specified by the height map h_{terrain} (see [4] for details).

Kinematic constraint: Foot position range-of-motion constraints, (3l), are defined, to heuristically bound each foot's movement within a fixed sized box \mathcal{R} .

B. Whole-Body Trajectory Optimization

For legged systems, leveraging full-body dynamics, such as the angular momentum of limbs, is critical for generating dynamic motions. The whole-body optimization is formulated in a hybrid dynamics fashion. The objective is to track quantities computed from centroidal optimization. Formulation 2 details this whole-body optimization.

The decision set ϕ_{wbd} consists of the generalized position \mathbf{q} and velocity $\dot{\mathbf{q}}$, joint torques $\boldsymbol{\tau}$ and contact forces \mathbf{F}_c . Similar to the centroidal optimization, the objective includes a user-defined heuristic cost \mathcal{L}_{wbd} , such as torque minimization, and a tracking cost $\Psi_{\text{wbd}}(\phi_{\text{wbd}}, \phi_{\text{cen}}^{\text{ref}})$ to minimize the deviation from the centroidal reference trajectories (solution to Formulation 1).

$$\Psi_{\text{wbd}} = \|\mathcal{X}_r(\mathbf{q}) - \mathbf{r}_{\text{cen}}^{\text{ref}}\|_{\mathbf{Q}_r^{\text{wbd}}}^2 + \|\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} - \mathbf{h}_{\text{cen}}^{\text{ref}}\|_{\mathbf{Q}_h^{\text{wbd}}}^2$$

Formulation 2: Whole-body trajectory optimization.

	$\min_{\phi_{\text{wbd}}} \mathcal{L}_{\text{wbd}}(\phi_{\text{wbd}}) + \Psi_{\text{wbd}}(\phi_{\text{wbd}}, \phi_{\text{cen}}^{\text{ref}})$
(Variables)	$\phi_{\text{wbd}}[i] = [\mathbf{q}[i]^\top, \dot{\mathbf{q}}[i]^\top, \boldsymbol{\tau}[i]^\top, \mathbf{F}_c[i]^\top]$ $\phi_{\text{cen}}^{\text{ref}}[i] = [\mathbf{r}_{\text{cen}}^{\text{ref}}[i]^\top, \mathbf{h}_{\text{cen}}^{\text{ref}}[i]^\top, \mathbf{p}_{j,\text{cen}}^{\text{ref}}[i]^\top]$ $\forall i = 0, 1, \dots, N-1, j = 0, \dots, n_f-1$
(Dynamics)	s.t. $\begin{cases} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\boldsymbol{\tau} + \mathbf{J}_c^\top \mathbf{F}_c \\ \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}^+ - \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}^- = \mathbf{J}_c^\top \boldsymbol{\Lambda} \end{cases}$ (4a)
(Contact)	$\begin{cases} \mathbf{J}_c\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c\dot{\mathbf{q}} = 0 \\ \mathbf{J}_c\dot{\mathbf{q}}^+ = 0 \end{cases}$ (4b)
(Limits)	$\mathbf{q} \in \mathcal{J}, \boldsymbol{\tau} \in \mathcal{T}$ (4c)
(Frictioncone)	$\boldsymbol{\lambda}_c \in \mathcal{K}$ (4d)

$$+ \sum_{j=0}^{n_f} \left(\|\mathcal{X}_j(\mathbf{q}) - \mathbf{p}_{j,\text{cen}}^{\text{ref}}\|_{\mathbf{Q}_p^{\text{wbd}}}^2 \right)$$

where the quadratic weighting matrices for tracking the reference CoM position $\mathbf{r}_{\text{cen}}^{\text{ref}}$, centroidal momentum $\mathbf{h}_{\text{cen}}^{\text{ref}}$, and foot positions $\mathbf{p}_{j,\text{cen}}^{\text{ref}}$ are denoted as $\mathbf{Q}_r^{\text{wbd}}$, $\mathbf{Q}_h^{\text{wbd}}$, and $\mathbf{Q}_p^{\text{wbd}}$, respectively. $\mathcal{X}_r(\cdot)$ and $\mathcal{X}_j(\cdot)$ map the generalized position to CoM position and the j th end-effector (EE)'s position. The centroidal momentum matrix, $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{6 \times (6+n_f)}$, referring to (6), provides a linear relationship between the centroidal momentum \mathbf{h} and generalized velocities $\dot{\mathbf{q}}$ [19]. In (4a), the hybrid dynamics constraint includes either the continuous **full dynamics** (first row), or the impact dynamics (second row), where $\boldsymbol{\Lambda}$ stands for the contact impulse, and $\dot{\mathbf{q}}^-$ and $\dot{\mathbf{q}}^+$ are instantaneous joint velocities before and after the impact. Following (4a), a contact constraint (4b) is added for the stance or impact foot assuming a rigid contact, while a zero force is applied to the non-contact foot. Joint limit, torque limit, and friction cone are also modeled as shown in (4c)–(4d).

Considering the computational efficiency, we then choose the DDP-based method implemented in [10] to solve the whole-body optimization. Briefly, depending on the continuous (impact) phase, the dynamics and contact constraints are coupled to derive the joint acceleration $\ddot{\mathbf{q}}$ (joint velocity after impact $\dot{\mathbf{q}}^+$) and contact forces \mathbf{F}_c (contact impulse $\boldsymbol{\Lambda}$) simultaneously, so that the forward pass for DDP can be performed. The joint limit, torque limit and friction cone constraints are encoded as quadratic barrier functions inside the objective function.

C. Alternating CEN-WBD Optimization

We perform an alternating update scheme based on the aforementioned centroidal optimization (CEN) and whole-body optimization (WBD) problems. Fig. 3 demonstrates the proposed updating sequence, where the CEN optimization is first solved and then followed by the WBD optimization. Note that no hand-crafted reference trajectory is required and a constant MoI is employed for the first CEN optimization. The WBD or CEN reference trajectories are updated accordingly after each CEN or WBD optimization solve. \mathcal{I}_{crb} maps the generalized coordinates to the CRB inertia tensor \mathbf{I}_{crb} . Similar to [26]–[29], this

process, referred to as one alternating iteration, is iterated until the dynamics consensus between CEN and WBD is achieved. Moreover, our proposed consensus quantities include not only CoM position, centroidal momentum, and EE locations, but also the equimomental inertia tensors ($\mathbf{I}_{\text{ellip}}$ in CEN and \mathbf{I}_{crb} in WBD). This is achieved by enabling the variables, constraints, and costs that involve ellipsoid inertia vectors \mathbf{e} and \mathbf{v} in CEN, which allows generating robot behaviors that cannot be captured by only using centroidal dynamics such as zero-gravity reorientation (see Remark 1). For the other motions unnecessary to parameterize centroidal inertia, we disable the ellipsoid related terms in CEN and directly use \mathbf{I}_{crb} from WBD. This hybrid formulation reduces the complexity of directly solving Formulation 1 and saves the solve time. More results will be shown in Section V.

Remark 1: We introduce two sets of orientation representations in CEN: angular excursion θ and equimomental ellipsoid orientation \mathbf{v} . The former one can be seen as the angular analog of CoM and only changes under external torques [7], while the latter one can be affected given any limb movements even without external forces [8].

IV. FULL CENTROIDAL CONVEX MPC

Starting with the derivation of the linearized full centroidal dynamics, this section presents a QP-based MPC problem. We then discuss the benefits of our controller and provides the comparison with the original convex-MPC [1] in Section IV-C.

A. Full Centroidal Dynamics

Standard full centroidal dynamics as shown in (2) has been widely used in the legged systems literature [16], [17]. We propose a linearized variant of this model, and demonstrate its benefits within the context of MPC. To make this linear model unambiguous, the transformation \mathbf{T}_b between the time derivative of generalized base coordinates and the generalized base velocities is explicitly stated here:

$$\dot{\mathbf{q}}_b = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{4 \times 3} & \mathbf{W}(\theta_b) \end{bmatrix}}_{\mathbf{T}_b(\mathbf{q})} \nu_b \quad (5)$$

where $\mathbf{W}(\theta_b)$ represents the velocity transform for different choices of orientation parameterization, see [31].

Using the CMM, $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{6 \times (6+n_j)}$, to relate centroidal momentum to generalized velocities:

$$\mathbf{h} = \underbrace{\begin{bmatrix} \mathbf{A}_b(\mathbf{q}) & \mathbf{A}_j(\mathbf{q}) \end{bmatrix}}_{\mathbf{A}(\mathbf{q})} \begin{bmatrix} \nu_b \\ \dot{\mathbf{q}}_j \end{bmatrix} \quad (6)$$

Rearranging (6) for ν_b and using (5), one has

$$\dot{\mathbf{q}}_b = \begin{bmatrix} \underbrace{\mathbf{T}_b(\mathbf{q})\mathbf{A}_b^{-1}(\mathbf{q})}_{\widetilde{\mathbf{A}}_h(\mathbf{q})} - \mathbf{T}_b(\mathbf{q})\mathbf{A}_b^{-1}(\mathbf{q})\mathbf{A}_j(\mathbf{q}) & \underbrace{\mathbf{h}}_{\widetilde{\mathbf{A}}_j(\mathbf{q})} \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \dot{\mathbf{q}}_j \end{bmatrix} \quad (7)$$

where $\widetilde{\mathbf{A}}_h(\mathbf{q})$ and $\widetilde{\mathbf{A}}_j(\mathbf{q})$ will be referred to as transformed CMMs. Defining an augmented state $\mathbf{x} = [\mathbf{h}^\top, \mathbf{q}_b^\top, \mathbf{q}_j^\top, \dot{\mathbf{q}}_j^\top, \mathbf{g}^\top]^\top$, and combining (2) and (7), the

augmented state dynamics can be written as:

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{h}} \\ \dot{\mathbf{q}}_b \\ \dot{\mathbf{q}}_j \\ \ddot{\mathbf{q}}_j \\ \dot{\mathbf{g}} \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & m \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \\ \widetilde{\mathbf{A}}_h(\mathbf{q}) & 0 & 0 & \widetilde{\mathbf{A}}_j(\mathbf{q}) & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{H}(\mathbf{q})} \begin{bmatrix} \mathbf{h} \\ \mathbf{q}_b \\ \mathbf{q}_j \\ \dot{\mathbf{q}}_j \\ \mathbf{g} \end{bmatrix} \\ &+ \underbrace{\begin{bmatrix} \mathbf{I} & \cdots & \mathbf{I} & 0 \\ \widehat{(\mathbf{p}_0 - \mathbf{r})} & \cdots & \widehat{(\mathbf{p}_{n_f-1} - \mathbf{r})} & 0 \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & \mathbf{I} \\ 0 & \cdots & 0 & 0 \end{bmatrix}}_{\mathbf{G}(\mathbf{p}, \mathbf{r})} \begin{bmatrix} \mathbf{f}_0 \\ \vdots \\ \mathbf{f}_{n_f-1} \\ \mathbf{a} \end{bmatrix} \end{aligned} \quad (8)$$

We refer to this system as the **full centroidal dynamics**. The $\widehat{\cdot}$ operator transforms the cross-product operation into a matrix multiplication. Note that, \mathbf{H} only depends on the joint configuration \mathbf{q} , and \mathbf{G} only depends on contact and CoM locations, \mathbf{p} and \mathbf{r} . Each control vector \mathbf{u} consists of the foot contact forces \mathbf{f}_j and the joint accelerations $\mathbf{a} \in \mathbb{R}^{n_j}$. Similar to [1], we assume that the robot follows the desired reference trajectory generated from our offline optimization, then foot placements, CoM location, and transformed CMMs from offline trajectory can be directly substituted into (8).

$$\dot{\mathbf{x}} = \mathbf{H}(\mathbf{q}^{\text{ref}})\mathbf{x} + \mathbf{G}(\mathbf{p}^{\text{ref}}, \mathbf{r}^{\text{ref}})\mathbf{u} \quad (9)$$

which simplifies to a linear time-varying (LTV) system in the augmented state \mathbf{x} .

B. Full Centroidal Convex MPC

The discretized version of continuous LTV dynamics (9) is given as:

$$\mathbf{x}[i+1] = \widetilde{\mathbf{H}}[i]\mathbf{x}[i] + \widetilde{\mathbf{G}}[i]\mathbf{u}[i] \quad (10)$$

where $\widetilde{\mathbf{H}}$ and $\widetilde{\mathbf{G}}$ are matrix exponential of \mathbf{H} and \mathbf{G} from (9). Similar to the original convex-MPC [1], to reduce the size of the proposed MPC problem, state variables can be eliminated from the decision set by adopting a condensed dynamics formulation [32]. At time step k , assuming a MPC prediction horizon of $n \in \mathbb{N}^+$ and aggregating (10) from $i = k$ to $i = k + n - 1$ produces:

$$\overrightarrow{\mathbf{x}} = \widetilde{\mathbf{H}}^{\mathcal{A}}\mathbf{x}[k] + \widetilde{\mathbf{G}}^{\mathcal{A}}\overrightarrow{\mathbf{u}} \quad (11)$$

with aggregated states, $\overrightarrow{\mathbf{x}} = [\mathbf{x}[k+1]^\top \mathbf{x}[k+2]^\top \cdots \mathbf{x}[k+n]^\top]^\top$. Weighting matrix $\widetilde{\mathbf{H}}^{\mathcal{A}}$ is given as

$$\widetilde{\mathbf{H}}^{\mathcal{A}} = \left[\prod_{i=k}^k \widetilde{\mathbf{H}}[i]^\top \quad \prod_{i=k}^{k+1} \widetilde{\mathbf{H}}[i]^\top \quad \cdots \quad \prod_{i=k}^{k+n-1} \widetilde{\mathbf{H}}[i]^\top \right]^\top$$

and actuation matrix $\widetilde{\mathbf{G}}^{\mathcal{A}}$ is given as Unnumbered equation shown at the bottom of next page. The decision variables for this dense QP formulation are aggregated control vectors $\overrightarrow{\mathbf{u}}$ over the n -step prediction horizon, $\overrightarrow{\mathbf{u}} = [\mathbf{u}[k]^\top \mathbf{u}[k+1]^\top \cdots \mathbf{u}[k+n-1]^\top]^\top$. It is straightforward to define the QP objective as a sum

of tracking cost and a regularization term:

$$\Psi_{qp} = \|\widetilde{\mathbf{H}}^A \mathbf{x}[k] + \widetilde{\mathbf{G}}^A \vec{\mathbf{u}} - \vec{\mathbf{x}}^{\text{ref}}\|_{\mathbf{Q}_{qp}} + \|\vec{\mathbf{u}}\|_{\mathbf{R}_{qp}}$$

where $\vec{\mathbf{x}}^{\text{ref}} = [\mathbf{h}^{\text{ref}^T}, \mathbf{q}_b^{\text{ref}^T}, \mathbf{q}_j^{\text{ref}^T}, \dot{\mathbf{q}}_j^{\text{ref}^T}]^T$ are given by the desired reference, generated through offline optimization, for the entire horizon $i = k + 1$ to $i = k + n$. Friction pyramid constraint, joint acceleration constraint, and normal force constraint are also added into the QP formulation to ensure physically feasible contact forces and joint accelerations. The computed contact forces and joint accelerations from the proposed MPC-QP problem are fed into the inverse dynamics (ID) controller to generate joint torques as shown in Fig. 2.

C. Comparison With the Original Convex-MPC

Compared to the original Convex-MPC [1], our formulation added the ability to directly track centroidal momentum quantities. This offers a tighter integration with our offline trajectory generation, and fully utilizes the generated reference linear and angular momentum trajectories to faithfully execute momentum-rich motions, such as twisting jump and zero-gravity body reorientation (see Section V). Unlike the original convex-MPC controller [1], where cost heuristics must be designed to generate such motions [3], our controller allows momentum to be designed as part of the desired reference trajectory, and directly tracked online. In addition, leveraging CMMs in the system dynamics naturally allows the inclusion of joint variables as decision variables, thus eliminating the need for any additional kinematics optimizations commonly found in such control hierarchies [2], [6]. Another benefit when compared to the original is that our formulation does not assume zero roll direction rotation of the robot. On the downside, the ability to track momentum and joint trajectories directly increases the QP problem size, and our formulation is indeed slower to compute compared to the original, for the same length prediction horizon. Direct comparison against the original MPC is provided in Section V-B for a 180° twisting jump maneuver.

The crucial assumption in our formulation is that the robot must follow the reference trajectory generated by offline planning for the approximation in (9) to hold true. This assumption is reasonable because when the robot deviates from the offline reference, our MPC controller will generate new contact forces and joint accelerations, at a fast rate (see Section V-B), to bring the robot state back to the reference.

V. RESULTS

We perform various experiments both in simulation and hardware using the alternating CEN-WBD and MPC framework. Both the Unitree A1 and MIT Mini Cheetah [33] were used to verify the proposed methods on quadrupedal platforms. In simulation, dynamic maneuvers such as cantering, “parkour,” and zero gravity reorientation are shown to have stable and converging solutions. In addition, hardware tests were conducted with the Mini Cheetah to verify both the offline optimization and online MPC in the real world.

TABLE I
SOLVE TIMES AND RESIDUALS FOR TRAJECTORY GENERATION

Desired motions	Solve times CEN,WBD (s)	Solve times CEN,WBD (Hierarchical)	Residuals for CoM,EE,AM ($m, kg \cdot m^2/s$)	Residuals for CoM,EE,AM (Hierarchical)
Cantering	36.8, 4.9	14.0, 3.4	1.3, 6.9, 9.9	1.4, 8.3, 12.7
“Parkour”	213.0, 10.3	53.8, 7.4	5.2, 11.3, 42.9	53.8, 49.7, 197.2
Inertia shaping	35.9, 48.1	4.1, 45.8	0, 12.9, 0	0, 290.0, 0
Twisting jump	45.3, 5.3	2.5, 2.4	0.7, 1.5, 6.2	4.15, 22.1, 47.1

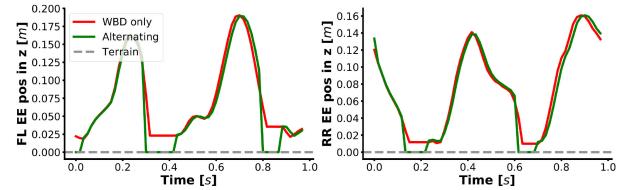


Fig. 4. EE in z-axis for FL and RR legs with a flat terrain (grey), generated by using only WBD optimization (red) and our alternating one (green).

A. Trajectory Generation and Simulation

Example dynamic maneuvers generated using our offline alternating CEN-WBD optimization are presented here. The direct transcription in CEN optimization is solved by IPOPT [34], and uses a universal 0.01 s timestep. For the inertia shaping example, the number of decision variables per knot point in CEN is 48, while the other examples use 42 since the inertia vector is turned off. The WBD optimization is solved by crocoddyl [10], and applies the same amount of knot points as the CEN. The number of decision variables in WBD remains 61 (49 for states and 12 for control inputs) for all examples. Aggregated solve times and residuals of both our alternating update scheme and a hierarchical one, i.e., running only one alternating iteration, are documented in Table I. We run 3 alternating iterations for all examples. The accumulated solve times for CEN and WBD optimizations are reported separately. We use the residuals to measure the consensus errors for CoM position (m), EE position (m), and AM ($kg \cdot m^2/s$), which are defined as the l_2 -norm of the error between CEN and WBD accumulated along the entire time horizon. Note that the residual for EE is also accumulated over all 4 legs. It is observed that although the solve times are longer, the residuals with more alternating iterations are significantly smaller and consensus performance is improved, which can be attributed to the benefit that one level of optimization iteratively receives feedback from the other.

Our offline TO pipeline is able to replicate the highly dynamic cantering gait. A one-second clip of motion capture (mocap) data from a real animal is used to first generate the initial seed trajectory for our robot’s morphology through re-targeting [35]. Then our TO pipeline uses the re-targeted inputs to produce base and joint trajectories which are not only faithful to the original mocap recording, but also physically-correct. We benchmark our solution accuracy against that of only running WBD optimization. Fig. 4 shows a comparison of the front left (FL) and front right (FR) z-direction foot position trajectories for the cantering example. It is clear that WBD optimization by itself

$$\widetilde{\mathbf{G}}^A = \begin{bmatrix} \widetilde{\mathbf{G}}[k] & & & & 0 \\ \prod_{i=k+1}^{k+1} \widetilde{\mathbf{H}}[i] \widetilde{\mathbf{G}}[k] & \widetilde{\mathbf{G}}[k+1] & & & \\ \vdots & \vdots & \ddots & & \\ \prod_{i=k+1}^{k+n-1} \widetilde{\mathbf{H}}[i] \widetilde{\mathbf{G}}[k] & \prod_{i=k+2}^{k+n-1} \widetilde{\mathbf{H}}[i] \widetilde{\mathbf{G}}[k+1] & \cdots & \widetilde{\mathbf{G}}[k+n-1] & \end{bmatrix}$$

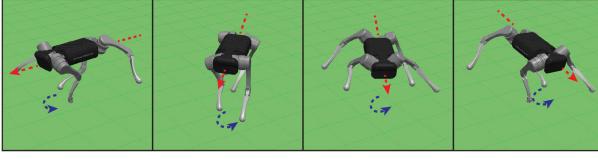


Fig. 5. Simulated motion of A1 turning 90° in a zero-gravity environment.

has difficulty constraining foot positions on the ground during stance phases, but our alternating pipeline can correctly enforce terrain constraints, which improves the overall solution accuracy. The residuals in Table I also reflect the improved solution accuracy, considering that most constraints to be satisfied in CEN optimization, such as the terrain constraints, are indirectly enforced in the WBD optimization through the tracking tasks.

A “parkour” scenario, where the robot traverses over a large gap via inclined surfaces, using hopping gaits, is also successfully computed through our TO pipeline. We set up the CEN optimization by manually providing the desired contact sequence, timings, and a terrain map with a 2.4 s time horizon. Intermediate states, for each landing configuration on the inclined surfaces, are also specified as hard constraints to improve solver convergence rate. The cantering and “parkour” examples are visualized in Fig. 1.

To demonstrate the benefit of momentum-inertia-aware CEN optimization in the offline TO, we study the zero-gravity body reorientation problem. Fig. 5 shows the simulation result using Drake [36]. Starting from stationary initial configuration, the robot is tasked to turn its trunk 90° within two seconds. Since no gravity nor external contacts are present in the problem setup, the robot is forced to rely on its limb motions to produce the turning behavior. In the absence of external forces, CEN optimization by itself and all other SRBM-based approaches cannot discover meaningful EE trajectories for this specific example. However, with inertia feedback from the WBD level, the two optimization levels indeed reach dynamics consensus, as shown in Table I with the EE residual decreasing from 290 to 12.9. This further validates that inclusion of the momentum inertia parameterization at the CEN level is physically meaningful. It is also worth noting that for this example, the WBD optimization takes significantly longer than the other examples, although the total motion horizon is short. This is because during the first alternating iteration, WBD optimization lacks good tracking reference from the CEN level, due to the absence of any contact related constraints. This further illustrates that a decent tracking reference generated from CEN level helps drastically reduce the solve time for the WBD optimization.

The 180° twisting jump maneuver is also studied, consisting of three contact phases: takeoff (0-0.3 s), aerial (0.3-0.7 s), and landing (0.7-1.0 s). The robot must gain enough AM during takeoff to achieve the goal orientation after landing. We provide a comparison study against a SRBM with inverse kinematics (SRBM+IK)-based trajectory optimization scheme. Different from [37], no hand-crafted reference trajectory is employed for SRBM+IK as a fair comparison. Centroidal AM trajectories are compared in the right sub-figure of Fig. 6. By achieving momentum consensus through CRB inertia feedback \mathbf{I}_{crb} from WBD, our CEN-WBD optimization leverages additional leg movements [a swift counter rotation observed in Fig. 7(1b)] during the take-off phase, which leads to a larger centroidal angular momentum. The slight but non-trivial change of

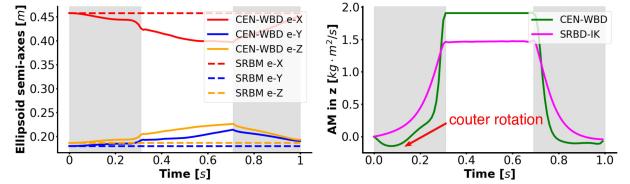


Fig. 6. Equimomental ellipsoid semi-axes changes (left) and centroidal angular momentum in z direction (right) generated by CEN-WBD and SRBM+IK for 180° twisting jump.

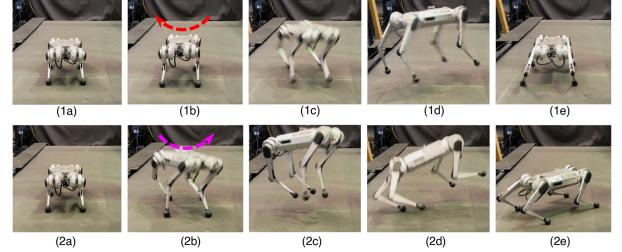


Fig. 7. Demonstration of the Mini Cheetah executing a 180° twisting jump. The first row (1a)–(1e) shows the performance on tracking the trajectory generated by our CEN-WBD, while the trajectory used in the second experiment (2a)–(2e) is from a SRBM combined with IK.

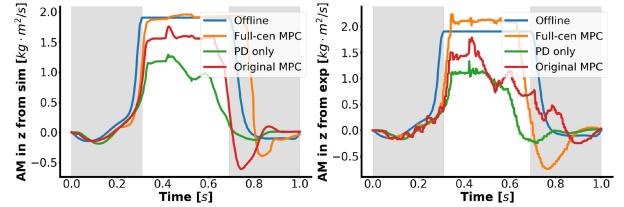


Fig. 8. Tracking performances from simulations (left) and experiments (right) for a twisting jump by comparing the offline reference trajectories and the measured data during online execution with different control strategies.

equimomental ellipsoid semi-axes in Fig. 6 guarantees a high-fidelity momentum trajectory to be tracked by our controller. When tested on hardware and using identical controller setups, trajectories generated using SRBM+IK fail to guide the robot to the full 180° rotation, as shown in Fig. 7. More hardware implementation details are described in the subsequent section.

B. Hardware Demonstration

We validate the tracking performance of the proposed MPC controller, by reproducing the twisting jump maneuver on the Mini-Cheetah platform. Since our MPC lacks the ability to directly optimize contact timings, a heuristics-based contact detection scheme is devised to deal with the early touchdown scenario. For each leg, touchdown is estimated by detecting an abrupt velocity change for the knee joints after the takeoff phase. For each leg, if a touchdown is early, tracking reference is immediately shifted to the point of touchdown on the offline trajectory. When all four legs land, base and momentum trajectories are also shifted to their respective points of touchdown.

Our controller is deployed directly on the Mini-Cheetah on-board computer, and updates at 100 Hz with a prediction horizon of 50 ms. Compare to the original convex MPC controller [1], which updates at around 30 Hz, our version updates at a higher frequency, but with a shorter prediction horizon (up to 0.5 s prediction horizon for the original). Fig. 8 shows the performance of both our proposed and the original convex MPC for the 180°

twisting jump maneuver. A baseline benchmark using a joint PD controller is also included in the figure. Our proposed controller gives the best tracking performance and is able to generate the largest angular momentum. In both simulation and on hardware, the original convex MPC and the PD controllers fail to achieve the full 180° rotation, while our proposed controller successfully achieves the desired rotation. We attribute the improvement in performance over the original convex MPC to the choice of centroidal momentum as decision variables, rather than the base linear and angular velocities employed by the original. The inclusion of joint states as decision variables also allows us to better reason about joint tracking, and not relying on Cartesian PD as is the case for original convex MPC.

VI. CONCLUSION AND FUTURE WORK

We presented a trajectory optimization and QP-based MPC framework for versatile and agile quadrupedal locomotion. The novelty of our proposed TO framework lied in the inclusion of equimomental ellipsoid, and an alternating scheme to achieve dynamics consensus between CEN and WBD optimizations. The MPC controller used the generated trajectory and re-computed contact forces and joint accelerations for online execution. Our proposed pipeline has been verified through a range of dynamic motions in both simulation and real world. Future works include optimizing over foot placements and contact timings in our MPC controller to realize cantering and “parkour” on hardware.

ACKNOWLEDGMENT

The authors would like to thank the MIT Biomimetic Robotics Lab and NAVER LABS for providing the Mini Cheetah simulation software and lending the Mini Cheetah for experiments.

REFERENCES

- 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, IROS, 2018, pp. 1–9.
- 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,” 2019, *arXiv:1909.06586*.
- G. Bledt, “Regularized predictive control framework for robust dynamic legged locomotion,” Ph.D. dissertation, Dept. Mech. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2020.
- 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 Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1560–1567, Jul. 2018.
- J. Ahn, S. J. Jorgensen, S. H. Bang, and L. Sentis, “Versatile locomotion planning and control for humanoid robots,” *Front. Robot. AI*, vol. 8, 2021, Art. no. 712239.
- K. Wang, G. Xin, S. Xin, M. Mistry, S. Vijayakumar, and P. Kormushev, “A unified model with inertia shaping for highly dynamic jumps of legged robots,” 2021, *arXiv:2109.04581*.
- V. Zordan, D. Brown, A. Macchietto, and K. Yin, “Control of rotational dynamics for ground and aerial behavior,” *IEEE Trans. Visual. Comput. Graph.*, vol. 20, no. 10, pp. 1356–1366, Oct. 2014.
- S.-H. Lee and A. Goswami, “Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 4667–4672.
- A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, “3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics,” in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2016, pp. 1447–1454.
- C. Mastalli *et al.*, “Crocoddyl: An efficient and versatile framework for multi-contact optimal control,” in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2020, pp. 2536–2542.
- H. Li and P. M. Wensing, “Hybrid systems differential dynamic programming for whole-body motion planning of legged robots,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5448–5455, Oct. 2020.
- M. Posa, C. Cantu, and R. Tedrake, “A direct method for trajectory optimization of rigid bodies through contact,” *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 69–81, 2014.
- M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, “Trajectory optimization through contacts and automatic gait discovery for quadrupeds,” *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1502–1509, Jul. 2017.
- Y. Tassa, T. Erez, and E. Todorov, “Synthesis and stabilization of complex behaviors through online trajectory optimization,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 4906–4913.
- I. Mordatch, E. Todorov, and Z. Popović, “Discovery of complex behaviors through contact-invariant optimization,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–8, 2012.
- H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 295–302.
- F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, “An efficient optimal planning and control framework for quadrupedal locomotion,” in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2017, pp. 93–100.
- J.-P. Sleiman, F. Farshidian, and M. Hutter, “Constraint handling in continuous-time DDP-based model predictive control,” in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2021, pp. 8209–8215.
- D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a humanoid robot,” *Auton. Robots*, vol. 35, no. 2, pp. 161–176, 2013.
- Y. Tassa, N. Mansard, and E. Todorov, “Control-limited differential dynamic programming,” in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2014, pp. 1168–1175.
- Z. Xie, C. K. Liu, and K. Hauser, “Differential dynamic programming with nonlinear constraints,” in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2017, pp. 695–702.
- T. A. Howell, B. E. Jackson, and Z. Manchester, “ALTRO: A fast solver for constrained trajectory optimization,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, IROS, 2019, pp. 7674–7679.
- J. Carpentier and N. Mansard, “Multicontact locomotion of legged robots,” *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1441–1460, Dec. 2018.
- M. Kudruss *et al.*, “Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations,” in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots*, 2015, pp. 684–689.
- C. Nguyen and Q. Nguyen, “Contact-timing and trajectory optimization for 3D jumping on quadruped robots,” 2021, *arXiv:2110.06764*.
- R. Budhiraja, J. Carpentier, and N. Mansard, “Dynamics consensus between centroidal and whole-body models for locomotion of legged robots,” in *Proc. Int. Conf. Robot. Automat.*, ICRA, 2019, pp. 6727–6733.
- A. Herzog, S. Schaal, and L. Righetti, “Structured contact force optimization for kino-dynamical motion generation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, IROS, 2016, pp. 2703–2710.
- B. Ponton, M. Khadiv, A. Meduri, and L. Righetti, “Efficient multicontact pattern generation with sequential convex approximations of the centroidal dynamics,” *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1661–1679, Oct. 2021.
- Z. Zhou and Y. Zhao, “Accelerated ADMM based trajectory optimization for legged locomotion with coupled rigid body dynamics,” in *Proc. IEEE Amer. Control Conf.*, ACC, 2020, pp. 5082–5089.
- M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, vol. 3, Hoboken, NJ, USA: Wiley, 2006.
- C. Gehring *et al.*, “Kindr library-kinematics and dynamics for robotics,” 2016. [Online]. Available: https://docs.leggedrobotics.com/kindr/cheatsheet_latest.pdf
- J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides, “A condensed and sparse QP formulation for predictive control,” in *Proc. IEEE 50th Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 5217–5222.
- B. Katz, J. Di Carlo, and S. Kim, “Mini cheetah: A platform for pushing the limits of dynamic quadruped control,” in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2019, pp. 6295–6301.
- A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.
- X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, “Learning agile robotic locomotion skills by imitating animals,” in *Proc. Robot.: Sci. Syst.*, 2020.
- R. Tedrake and the Drake Development Team, “Drake: Model-based design and verification for robotics,” 2019. [Online]. Available: <https://drake.mit.edu>
- M. Chignoli and S. Kim, “Online trajectory optimization for dynamic aerial motions of a quadruped robot,” in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2021, pp. 7693–7699.