Momentum-Aware Trajectory Optimization and Control for Agile Quadrupedal Locomotion

Ziyi Zhou*[1], Bruce Wingo*[1], Nathan Boyd[1], Seth Hutchinson[1], and Ye Zhao[1]

***Abstract*— In this paper, we present a versatile hierarchical offline planning algorithm, along with and 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 linearization of the full centroidal dynamics, and incorporated these into a convex model predictive control scheme. 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 hardware experiments on the MIT Mini Cheetah platform to demonstrate performance of the entire planning and control pipeline on a twisting jump maneuver.**

I. Introduction

A plethora of optimization-based motion planning strategies exist in the legged locomotion literature. Solving a single nonlinear trajectory optimization (TO) problem is a rather common approach. Single-optimization based methods can be categorized based on the types of dynamics representation. 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 within the community. 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.

For reduced-order models, incorporating joint information to reason about the centroidal inertia is non-trivial, which limits its ability to generate angular-momentum-rich motions using limb movements. TOWR+ [5] improved upon the original TOWR framework by augmenting the standard SRBM with a centroidal inertia regressor network. However, it has difficulty predicting inertia trajectories for gaits not present in the training set. Wang et al. [6] attempted to model the leg dynamics through a point mass approximation for each limb. But, the resulting lumped-leg SRBM still cannot utilize the full capabilities of the joint motions. Earlier works, such as [7] and [8] derived analytical models, know as equimomental ellipsoids, to parameterize centroidal inertia. However, they introduce highly nonlinear constraints into the optimization, and discovering physically meaningful inertia trajectories solely through the ellipsoid parameterization is difficult.

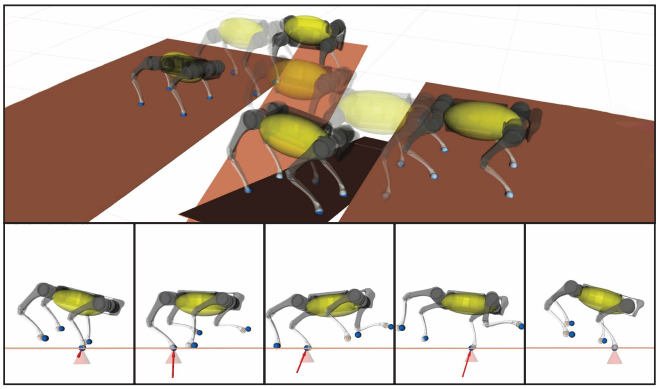


Fig. 1: Dynamic maneuvers that require high-fidelity control of angular momentum, such as "parkour" (top) and cantering (bottom).

Directly incorporating full-order models into a single-optimization is another common approach to complex whole-body motion generation. 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 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 also been made in [20]–[22] to enforce general state, input 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 decoupling single-optimization into simpler subproblems, it is still 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.

*The first two authors equally contributed to this work

[1]Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332, USA {zhouziyi, bwingo, nboyd31, yezhao, seth}@gatech.edu

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.

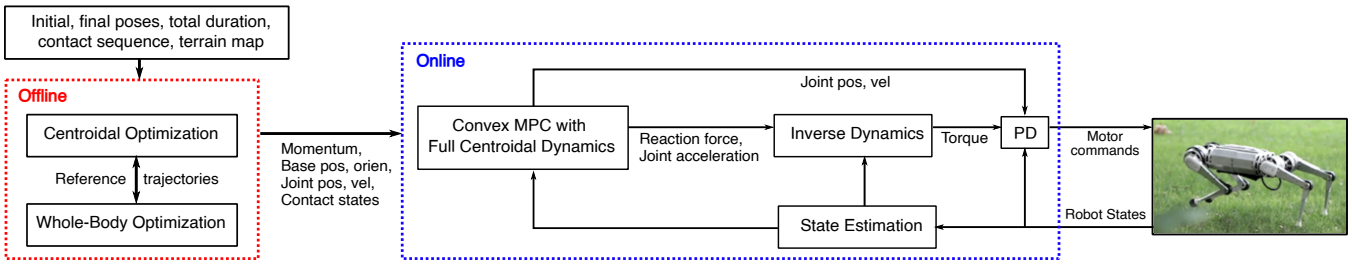


Fig. 2: Overall planning and control framework that consists of offline trajectory generation (red) and online execution (blue). The input to offline block is specifically decided by the desired motion. Full-body, centroidal, and contact information are passed from offline generation to online control.

Similarly, Nguyen et al. [25] proposed a pipeline, first solving an SRBM-based centroidal optimization, then using its solution as tracking objective for a full dynamics based whole-body optimization. However, additional constraints projecting whole-body information to the centroidal level are required to ensure feasible solutions. As pointed out by [26], these constraints are often computationally expensive and difficult to define, especially when involving angular momentum (AM). Another strategy would be to iteratively solve the two optimizations in an alternating fashion until the convergence of a common set of dynamics variables is achieved (dynamic consensus) [26]–[29]. However, 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 work, we propose an 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 leveraging the inertia feedback from whole-body optimization to enable a wide range of motions previously unachievable with standard centroidal optimization, such as "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 quality at the whole-body level through dynamics consensus.

To validate the solution trajectory generated by our alternating optimization framework 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 a fast and efficient QP formulation of the MPC problem similar to [1], but also grants us direct control over the joint variables for direct momentum tracking.

Our core contributions are highlighted as follows:

- We propose 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 convex model predictive controller through a novel 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 framework generates high-fidelity contact and momentum-rich agile maneuvers, and demonstrate the capability of our controller on the MIT Mini-Cheetah platform.

II. System Overview

We propose a hierarchical planning and control framework in this paper, 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 (not including contact locations), total motion duration, initial and final poses, and the terrain map. The offline trajectory planner takes these inputs, and 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 state feedback from the robot. First of all, let us establish the system models.

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:

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

where $\mathbf{q} = \begin{bmatrix}\mathbf{q}_{\mathrm{b}}^{\top} & \mathbf{q}_{\mathrm{j}}^{\top}\end{bmatrix}^{\top} \in \mathbb{R}^{n_q}$, $\mathbf{v} = \begin{bmatrix}\boldsymbol{\nu}_{\mathrm{b}}^{\top} & \dot{\mathbf{q}}_{\mathrm{j}}^{\top}\end{bmatrix}^{\top} \in \mathbb{R}^{n_v}$ 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 $\mathbf{q}_{\mathrm{b}} = \begin{bmatrix}\mathbf{p}_{\mathrm{b}}^{\top} & \boldsymbol{\theta}_{\mathbf{b}}^{\top}\end{bmatrix}^{\top} \in \mathbb{R}^6$ are partitioned as base position and orientation. Base velocities $\boldsymbol{\nu}_{\mathrm{b}} = \begin{bmatrix}\dot{\mathbf{p}}_{\mathrm{b}}^{\top} & \boldsymbol{\omega}^{\top}\end{bmatrix}^{\top} \in \mathbb{R}^6$ 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. $\mathbf{J}_{\mathrm{c}}(\mathbf{q})$ is the stacked contact Jacobian and $\mathbf{F}_{\mathrm{c}}$ 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.

Assuming full control authority over the joint variables $\mathbf{q}_{\mathrm{j}}$, and applying appropriate transformations, Eq. (1) can be converted into Newton-Euler equations of motion about the robot's center-of-mass (CoM) [30]:

$$\dot{\mathbf{h}} = \begin{bmatrix}\dot{\mathbf{k}}\\ \dot{\mathbf{l}}\end{bmatrix} = \begin{bmatrix}\sum_{\mathrm{j}=0}^{n_f-1}\mathbf{f}_{\mathrm{j}} + m\mathbf{g}\\ \sum_{\mathrm{j}=0}^{n_f-1}(\mathbf{p}_{\mathrm{j}} - \mathbf{r})\times\mathbf{f}_{\mathrm{j}}\end{bmatrix} \tag{2}$$

where it is assumed that the robot has n_f feet with point contacts. $\mathbf{h} = \begin{bmatrix}\mathbf{k}^{\top} & \mathbf{l}^{\top}\end{bmatrix}^{\top}$ are the linear and angular centroidal momentum. $\mathbf{p}_{\mathrm{j}}$ and $\mathbf{r}$ are the individual contact and CoM position in the world frame. $\mathbf{f}_{\mathrm{j}}$ are the individual foot contact forces. This reduced-order model will be referred to as the **centroidal dynamics** throughout this paper. Unless otherwise stated, all vector quantities in this paper are expressed in the world frame.

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 Sec. III-A and III-B. The alternating update scheme and the importance of dynamics consensus are discussed in Sec. III-C.

A. Centroidal Trajectory Optimization

The TO problem to be solved at the centroidal level utilizes the **centroidal dynamics** in Eq. (2). We transcribe the optimization problem into a nonlinear program through direct collocation, due to its natural capability of constraint handling. Total duration T of the desired motion is equally divided into N knot points. Contact sequence and timing per foot are also predefined. Formulation 1 details the centroidal optimization.

The centroidal decision set $\boldsymbol{\phi}_{\mathrm{cen}}$ consists of CoM position $\mathbf{r}$, CoM linear velocity $\dot{\mathbf{r}}$, angular excursion [7] with Euler angle parameterization $\boldsymbol{\theta}$ and its rate $\dot{\boldsymbol{\theta}}$, centroidal angular momentum $\mathbf{l}$ and its rate $\dot{\mathbf{l}}$, j^{th} foot's position $\mathbf{p}_{\mathrm{j}}$ and contact force $\mathbf{f}_{\mathrm{j}}$. $\mathbf{e}$ and $\boldsymbol{\gamma}$ are inertia variables characterizing the principle semi-axes and orientation of the equimomental ellipsoid [8] (see the ellipsoid illustration in Fig. 5). Ellipsoid orientation is again parameterized using Euler angles. All components of $\boldsymbol{\phi}_{\mathrm{cen}}$ belongs to $\mathbb{R}^3$.

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

$$\begin{aligned}\Psi_{\mathrm{cen}} =& \|\mathbf{r} - \mathbf{r}^{\mathrm{ref}}_{\mathrm{wbd}}\|^2_{\mathbf{Q}^{\mathrm{cen}}_{\mathrm{r}}} + \|m\dot{\mathbf{r}} - \mathbf{k}^{\mathrm{ref}}_{\mathrm{wbd}}\|^2_{\mathbf{Q}^{\mathrm{cen}}_{\mathrm{k}}} +\\ & \|\mathbf{l} - \mathbf{l}^{\mathrm{ref}}_{\mathrm{wbd}}\|^2_{\mathbf{Q}^{\mathrm{cen}}_{\mathrm{l}}} + \|\mathbf{I}_{\mathrm{ellip}} - \mathbf{I}_{\mathrm{crb}}\|^2_{\mathbf{Q}^{\mathrm{cen}}_{\mathrm{I}}} +\\ & \sum_{j=0}^{n_f-1}(\|\mathbf{p}_j - \mathbf{p}^{\mathrm{ref}}_{j,\mathrm{wbd}}\|^2_{\mathbf{Q}^{\mathrm{cen}}_{\mathrm{p}}})\end{aligned}$$

Symmetric positive definite weighting matrices for tracking CoM position, linear momentum (LM), angular momentum (AM), and foot positions are denoted as $\mathbf{Q}^{\mathrm{cen}}_{\mathrm{r}}$, $\mathbf{Q}^{\mathrm{cen}}_{\mathrm{k}}$, $\mathbf{Q}^{\mathrm{cen}}_{\mathrm{l}}$, and $\mathbf{Q}^{\mathrm{cen}}_{\mathrm{p}}$, respectively. Inertia tracking is achieved through minimizing the $\mathbf{Q}^{\mathrm{cen}}_{\mathrm{I}}$-weighted Frobenius norm of the difference between the inertia tensors of the equimomental ellipsoid $\mathbf{I}_{\mathrm{ellip}}$ and that of the reference composite rigid body (CRB) $\mathbf{I}_{\mathrm{crb}}$, both expressed in the world frame. Centroidal optimization includes the following constraints:

Dynamic constraint: Without any simplification, 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 $\mathbf{I}_{\mathrm{ellip}}$ in the world frame is constructed from the rotational transformation of the inertia tensor in the base frame, $\mathbf{I}_{\mathrm{ellip}} = \mathbf{R}(\boldsymbol{\gamma})_{\mathcal{B}}\mathbf{I}_{\mathrm{ellip}}(\mathbf{e})\mathbf{R}(\boldsymbol{\gamma})^{\top}$ with $\mathbf{R}(\boldsymbol{\gamma})$ being the rotation transformation between the ellipsoid base frame and world frame. Inertia tensor in the base frame is defined as ${}_{\mathcal{B}}\mathbf{I}_{\mathrm{ellip}} = diag([\mathbf{I}^{\mathrm{xx}}, \mathbf{I}^{\mathrm{yy}}, \mathbf{I}^{\mathrm{zz}}]^T)$ with $\mathbf{I}^{xx} = \frac{1}{5}m(\mathbf{e}^{\mathrm{y}2} + \mathbf{e}^{\mathrm{z}2})$, $\mathbf{I}^{yy} = \frac{1}{5}m(\mathbf{e}^{\mathrm{x}2} + \mathbf{e}^{\mathrm{z}2})$, and $\mathbf{I}^{zz} = \frac{1}{5}m(\mathbf{e}^{\mathrm{x}2} + \mathbf{e}^{\mathrm{y}2})$.

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

Formulation 1 Centroidal trajectory optimization

$$\min_{\boldsymbol{\phi}_{\text{cen}}} \ \mathcal{L}_{\text{cen}}(\boldsymbol{\phi}_{\text{cen}}) + \Psi_{\text{cen}}(\boldsymbol{\phi}_{\text{cen}}, \boldsymbol{\phi}^{\text{ref}}_{\text{wbd}})$$

(Variables)
$$\boldsymbol{\phi}_{\text{cen}}[i] = [\mathbf{r}[i]^\top, \dot{\mathbf{r}}[i]^\top, \boldsymbol{\theta}[i]^\top, \dot{\boldsymbol{\theta}}[i]^\top, \mathbf{l}[i]^\top, \dot{\mathbf{l}}[i]^\top, \mathbf{p}_j[i]^\top, \mathbf{f}_j[i]^\top, \underbrace{\mathbf{e}[i]^\top, \boldsymbol{\gamma}[i]^\top}_{\text{inertia vectors}}]^\top$$
$$\boldsymbol{\phi}^{\text{ref}}_{\text{wbd}}[i] = [\mathbf{r}^{\text{ref}}_{\text{wbd}}[i]^\top, \dot{\mathbf{r}}^{\text{ref}}_{\text{wbd}}[i]^\top, \mathbf{l}^{\text{ref}}_{\text{wbd}}[i]^\top, \mathbf{p}^{\text{ref}}_{j,\text{wbd}}[i]^\top, \mathbf{I}_{\text{crb}}[i]^\top]^\top$$
$$\forall i = 0, \dots, N-1, j = 0, \dots, n_f - 1$$

(Dynamics) s.t.
$$\begin{bmatrix} m\ddot{\mathbf{r}}[i] \\ \dot{\mathbf{l}}[i] \\ \mathbf{l}[i] \end{bmatrix} = \begin{bmatrix} \sum_j \mathbf{f}_j[i] + m\mathbf{g} \\ \sum_j (\mathbf{p}_j[i] - \mathbf{r}[i]) \times \mathbf{f}_j[i] \\ \mathbf{I}_{\text{ellip}}(\mathbf{e}[i], \boldsymbol{\gamma}[i])\boldsymbol{\omega}[i] \end{bmatrix} \tag{3a}$$

(Ellipsoid)
$$m = \frac{4}{3}\pi \mathbf{e}^{\text{x}}[i]\mathbf{e}^{\text{y}}[i]\mathbf{e}^{\text{z}}[i]\rho[i] \tag{3b}$$

(Integration)
$$\mathbf{r}[i+1] - \mathbf{r}[i] = \frac{\Delta t}{2}(\dot{\mathbf{r}}[i+1] + \dot{\mathbf{r}}[i]) \tag{3c}$$
$$\dot{\mathbf{r}}[i+1] - \dot{\mathbf{r}}[i] = \frac{\Delta t}{2}(\ddot{\mathbf{r}}[i+1] + \ddot{\mathbf{r}}[i]) \tag{3d}$$
$$\boldsymbol{\theta}[i+1] - \boldsymbol{\theta}[i] = \frac{\Delta t}{2}(\dot{\boldsymbol{\theta}}[i+1] + \dot{\boldsymbol{\theta}}[i]) \tag{3e}$$
$$\mathbf{l}[i+1] - \mathbf{l}[i] = \frac{\Delta t}{2}(\dot{\mathbf{l}}[i+1] + \dot{\mathbf{l}}[i]) \tag{3f}$$

(Frictional)
$$\mathbf{f}_j[i] \cdot \mathbf{n}(\mathbf{p}_j^{xy}[i]) \geq 0, \ \forall i \in \mathcal{C}_j \tag{3g}$$
$$\mathbf{f}_j[i] \in \mathcal{F}(\mu, \mathbf{n}, \mathbf{p}_j^{xy}[i]), \ \forall i \in \mathcal{C}_j \tag{3h}$$
$$\mathbf{f}_j[i] = 0, \ \forall i \notin \mathcal{C}_j \tag{3i}$$

(Terrain)
$$\mathbf{p}_j^{xy}[i+1] - \mathbf{p}_j^{xy}[i] = 0, \ \forall i \in \mathcal{C}_j \tag{3j}$$
$$\mathbf{p}_j^{z}[i] = h_{\text{terrain}}(\mathbf{p}_j^{xy}[i]), \ \forall i \in \mathcal{C}_j \tag{3k}$$

(Kinematics)
$$\mathbf{p}_j[i] \in \mathcal{R}_j(\mathbf{r}[i], \boldsymbol{\theta}[i]) \tag{3l}$$

Integration constraint: By adopting trapezoidal integration scheme, the quantities $\mathbf{r}$, $\dot{\mathbf{r}}$, $\boldsymbol{\theta}$ and $\mathbf{l}$ are approximated as piecewise linear functions as shown in Eqs. (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 Eqs. (3g) - (3i). 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, Eq. (3j) defines the no-slip condition. While Eq. (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, Eq. (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. Since the contact sequence is provided beforehand, 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.

Formulation 2 Whole-body trajectory optimization

$$\min_{\boldsymbol{\phi}_{\text{wbd}}} \ \mathcal{L}_{\text{wbd}}(\boldsymbol{\phi}_{\text{wbd}}) + \Psi_{\text{wbd}}(\boldsymbol{\phi}_{\text{wbd}}, \boldsymbol{\phi}^{\text{ref}}_{\text{cen}})$$

(Variables)
$$\boldsymbol{\phi}_{\text{wbd}}[i] = [\mathbf{q}[i]^\top, \dot{\mathbf{q}}[i]^\top, \boldsymbol{\tau}[i]^\top, \mathbf{F}_{\text{c}}[i]^\top]^\top$$
$$\boldsymbol{\phi}^{\text{ref}}_{\text{cen}}[i] = [\mathbf{r}^{\text{ref}}_{\text{cen}}[i]^\top, \mathbf{h}^{\text{ref}}_{\text{cen}}[i]^\top, \mathbf{p}^{\text{ref}}_{j,\text{cen}}[i]^\top]^\top$$
$$\forall i = 0, 1, \dots, N-1, j = 0, \dots, n_f - 1$$

(Dynamics) s.t.
$$\begin{cases} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\boldsymbol{\tau} + \mathbf{J}_{\text{c}}^\top \mathbf{F}_{\text{c}} \\ \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}^+ - \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}^- = \mathbf{J}_{\text{c}}^\top \boldsymbol{\Lambda} \end{cases} \tag{4a}$$

(No-slip)
$$\mathbf{J}_c\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c\dot{\mathbf{q}} = 0 \tag{4b}$$

(Limits)
$$\mathbf{q} \in \mathcal{J}, \boldsymbol{\tau} \in \mathcal{T} \tag{4c}$$

(Friction cone)
$$\boldsymbol{\lambda}_c \in \mathcal{K} \tag{4d}$$

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

$$\Psi_{\text{wbd}} = \|\mathcal{X}_r(\mathbf{q}) - \mathbf{r}^{\text{ref}}_{\text{cen}}\|^2_{\mathbf{Q}^{\text{wbd}}_{\text{r}}} + \|\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} - \mathbf{h}^{\text{ref}}_{\text{cen}}\|^2_{\mathbf{Q}^{\text{wbd}}_{\text{h}}} + \sum_{j=0}^{n_f} (\|\mathcal{X}_j(\mathbf{q}) - \mathbf{p}^{\text{ref}}_{j,\text{cen}}\|^2_{\mathbf{Q}^{\text{wbd}}_{\text{p}}})$$

where the quadratic weighting matrices for tracking CoM position/orientation, centroidal momentum, and foot positions are denoted as $\mathbf{Q}^{\text{wbd}}_{\text{r}}$, $\mathbf{Q}^{\text{wbd}}_{\text{h}}$, and $\mathbf{Q}^{\text{wbd}}_{\text{p}}$, respectively. $\mathcal{X}_r(\cdot)$ and $\mathcal{X}_j(\cdot)$ map the generalized position to CoM position and the jth end-effector (EE)'s position. The centroidal momentum matrix, $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{6\times(6+n_j)}$, referring to Eq. (6), provides a linear relationship between the centoridal momentum $\mathbf{h}$ and generalized velocities $\dot{\mathbf{q}}$ [19]. The hybrid dynamic constraint includes both the non-impact **full dynamics** Eq. (1), and the impact phases described in Eq. (4a), 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. Additionally, a no-slip contact constraint (4b) is added for the stance foot. Joint limit, torque limit, and friction cone are also modeled as shown in Eqs. (4c) - (4d).

Considering the computational efficiency, we then choose the DDP-based method implemented in [10] to solve the whole-body optimization. Briefly, the dynamic and no-slip constraints are coupled to derive the joint acceleration and contact forces 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. Readers are referred

to [10] for more details. Based on the above whole-body trajectory optimization, we are able to generate versatile whole-body motions through tracking reference trajectories provided by the centroidal optimization.

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. 5 demonstrates the proposed updating sequence, where the CEN optimization is first solved and then followed by the WBD optimization. The reference trajectories are updated accordingly after each CEN or WBD optimization solve. $\mathcal{I}_{\text{crb}}$ maps the generalized coordinates to the CRB inertia tensor $\mathbf{I}_{\text{crb}}$. 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 $\mathbf{I}_{\text{crb}}$ 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 $\mathbf{I}_{\text{crb}}$ from WBD. This hybrid formulation reduces the complexity of directly solving Formulation 1 and saves the solve time. More results will be shown in Sec. V.

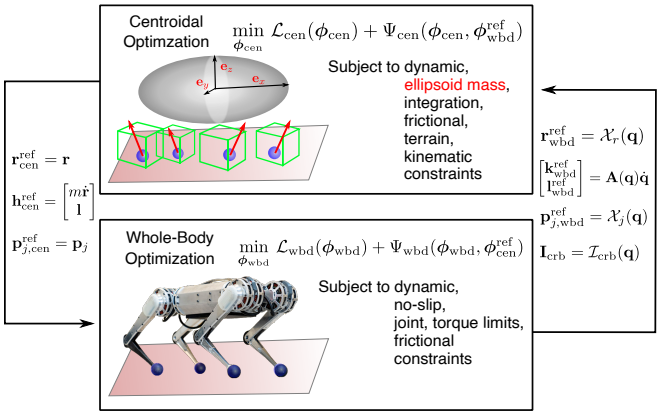


Fig. 5: The block diagram that demonstrates the proposed alternating centroidal, whole-body optimization.

Remark 1. *We introduce two sets of orientation representations in CEN: angular excursion $\boldsymbol{\theta}$ and equimomental ellipsoid orientation $\mathbf{v}$. The former one can be seen as the angular analog of CoM and only changed 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

This section describes the mathematical formulation of our proposed tracking controller. Starting with the derivation of the linearized full centroidal dynamics, we present a QP-based convex-MPC problem. We then discuss the benefits of our controller and provides the comparison with the original convex-MPC [1] in Sec. IV-C.

A. Full Centroidal Dynamics

Standard full centroidal dynamics as shown in Eq. (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 between the time derivative of generalized base coordinates and the generalized base velocities is explicitly stated here:

$$\dot{\mathbf{q}}_{\text{b}} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{4\times 3} & \mathbf{W}(\boldsymbol{\theta}_{\text{b}}) \end{bmatrix}}_{\mathbf{T}_{\text{b}}(\mathbf{q})} \boldsymbol{\nu}_{\text{b}} \tag{5}$$

where $\mathbf{W}(\boldsymbol{\theta}_{\text{b}})$ 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:

$$\mathbf{h} = \underbrace{\begin{bmatrix} \mathbf{A}_{\text{b}}(\mathbf{q}) & \mathbf{A}_{\text{j}}(\mathbf{q}) \end{bmatrix}}_{\mathbf{A}(\mathbf{q})} \begin{bmatrix} \boldsymbol{\nu}_{\text{b}} \\ \dot{\mathbf{q}}_{\text{j}} \end{bmatrix} \tag{6}$$

Rearranging Eq. (6) for $\boldsymbol{\nu}_{\mathbf{b}}$ and using Eq. (5), one has

$$\dot{\mathbf{q}}_{\text{b}} = \begin{bmatrix} \underbrace{\mathbf{T}_{\text{b}}(\mathbf{q})\mathbf{A}_{\text{b}}^{-1}(\mathbf{q})}_{\widetilde{\mathbf{A}_{\text{h}}}(\mathbf{q})} & \underbrace{-\mathbf{T}_{\text{b}}(\mathbf{q})\mathbf{A}_{\text{b}}^{-1}(\mathbf{q})\mathbf{A}_{\text{j}}(\mathbf{q})}_{\widetilde{\mathbf{A}_{\text{j}}}(\mathbf{q})} \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \dot{\mathbf{q}}_{\text{j}} \end{bmatrix} \tag{7}$$

where $\widetilde{\mathbf{A}_{\text{h}}}(\mathbf{q})$ and $\widetilde{\mathbf{A}_{\text{j}}}(\mathbf{q})$ will be referred to as transformed CMMs. Defining an augmented state $\mathbf{x} = \left[\mathbf{h}^\top, \mathbf{q}_{\text{b}}{}^\top, \mathbf{q}_{\text{j}}{}^\top, \dot{\mathbf{q}}_{\text{j}}^\top, \mathbf{g}^\top\right]^\top$, and combining Eqs. (2) and (7), the augmented state dynamics can be written as:

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{h}} \\ \dot{\mathbf{q}}_{\text{b}} \\ \dot{\mathbf{q}}_{\text{j}} \\ \ddot{\mathbf{q}}_{\text{j}} \\ \dot{\mathbf{g}} \end{bmatrix} &= \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & m\begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \\ \widetilde{\mathbf{A}_{\text{h}}}(\mathbf{q}) & \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{A}_{\text{j}}}(\mathbf{q}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{H}(\mathbf{q})} \begin{bmatrix} \mathbf{h} \\ \mathbf{q}_{\text{b}} \\ \mathbf{q}_{\text{j}} \\ \dot{\mathbf{q}}_{\text{j}} \\ \mathbf{g} \end{bmatrix} \\ &+ \underbrace{\begin{bmatrix} \mathbf{I} & \cdots & \mathbf{I} & \mathbf{0} \\ \widehat{(\mathbf{p}_0 - \mathbf{r})} & \cdots & \widehat{(\mathbf{p}_{\text{n}_\text{f}-1} - \mathbf{r})} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{G}(\mathbf{p},\mathbf{r})} \underbrace{\begin{bmatrix} \mathbf{f}_0 \\ \vdots \\ \mathbf{f}_{\text{n}_\text{f}-1} \\ \mathbf{a} \end{bmatrix}}_{\mathbf{u}} \end{aligned} \tag{8}$$

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 $\mathbf{r}$. If 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 Eq. (8).

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

which simplifies to a linear time-varying (LTV) system in the augmented state $\mathbf{x}$, with contact forces and joint accelerations as control inputs. This linearized dynamics will be used in the subsequent QP-based MPC formulation.

B. Full Centroidal Convex MPC

Continuous LTV dynamics Eq. (9) can be discretized. The resulting LTV discrete dynamics are given as:

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

where $\widetilde{\mathbf{H}}$ and $\widetilde{\mathbf{G}}$ are matrix exponential of $\mathbf{H}$ and $\mathbf{G}$ from Eq. (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 Eq. (10) from $i = k$ to $i = k + n - 1$ produces:

$$\vec{\mathbf{x}} = \widetilde{\mathbf{H}^{\mathcal{A}}}\mathbf{x}[k] + \widetilde{\mathbf{G}^{\mathcal{A}}}\vec{\mathbf{u}} \tag{11}$$

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

$$\widetilde{\mathbf{H}^{\mathcal{A}}} = \begin{bmatrix} \prod_{i=k}^{k} \widetilde{\mathbf{H}}[i]^\top & \prod_{i=k}^{k+1} \widetilde{\mathbf{H}}[i]^\top & \cdots & \prod_{i=k}^{k+n-1} \widetilde{\mathbf{H}}[i]^\top \end{bmatrix}^\top$$

and actuation matrix $\widetilde{\mathbf{G}^{\mathcal{A}}}$ is given as

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

The decision variables for this dense QP formulation are aggregated control vectors $\vec{\mathbf{u}}$ over the n-step prediction horizon, $\vec{\mathbf{u}} = \begin{bmatrix}\mathbf{u}[k]^\top & \mathbf{u}[k+1]^\top & \cdots & \mathbf{u}[k+n-1]^\top\end{bmatrix}^\top$, where each control vector $\mathbf{u}$ consists of the foot contact forces $\mathbf{f}_j$ and the joint accelerations $\mathbf{a}$ for each actuated joint. It is straightforward to define the QP objective as a sum of tracking cost and a regularization term:

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

where $\vec{\mathbf{x}}^{\text{ref}} = [\mathbf{h}^{\text{ref}\top}, \mathbf{q}_{\text{b}}^{\text{ref}\top}, \mathbf{q}_{\text{j}}^{\text{ref}\top}, \dot{\mathbf{q}}_{\text{j}}^{\text{ref}\top}]^\top$ 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 Sec. 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. However, by trading off between update frequency and prediction horizon, we still managed to achieve satisfactory tracking performance on real hardware (see Sec. V-B).

The main assumption in our formulation is that the robot must follow the reference trajectory generated by offline planning for the approximation in Eq. (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 Sec. 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.

A. Trajectory Generation and Simulation

Example dynamic maneuvers generated using our offline alternating CEN-WBD optimization are presented here. We demonstrate the solution quality of these maneuvers either in visualization or physics simulation, and discuss the importance of dynamics consensus. Aggregated solve times and residuals for all the desired motions are documented in Table I. We run 3 alternating iterations for all examples. The accumulated solve times over 3 alternating iterations for CEN and WBD optimizations are reported separately. The residuals for CoM position (m), EE position (m), and AM ($kg \cdot m^2/s$) are defined as the $l2$-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. We show both the residuals from our alternating update scheme and a hierarchical one, i.e., running only one alternating iteration. It is observed that the residuals with more alternating iterations are significantly smaller and consensus performance is improved, which can be attributed to

TABLE I: Solve times and residuals for trajectory generation

Desired motions	Solve times CEN,WBD (s)	Residuals for CoM,EE,AM $(m,\ kg \cdot m^2/s)$	Residuals for CoM,EE,AM (Hierarchical)
Cantering	36.8, 4.9	1.3, 6.9, 9.9	1.4, 8.3, 12.7
"Parkour"	213.0, 10.3	5.2, 11.3, 42.9	53.8, 49.7, 197.2
Inertia shaping	35.9, 48.1	0, 12.9, 0	0, 290.0, 0
Twisting jump	23.2, 5.8	0.6, 1.8, 3.2	1.5, 2.1, 23.7

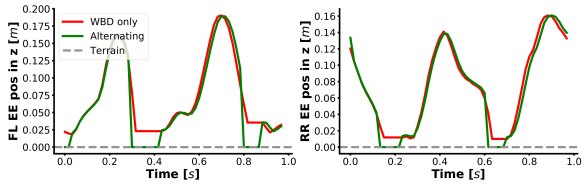


Fig. 6: EE in z-axis for FL and RR legs with a flat terrain (grey), generated by using only WBD optimization (red) and our alternating one (green).

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. 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 [34]. 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 quality against that of only running WBD optimization. Fig. 6 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 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 quality.

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.

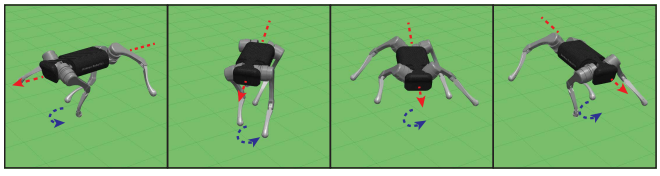


Fig. 7: Simulated motion of A1 turning 90° in a zero-gravity environment

To demonstrate the benefit of momentum-inertia-aware CEN optimization in the offline TO, we study the zero-gravity body reorientation problem. Fig. 7 shows the simulation result using Drake [35]. Starting from stationary initial configuration, the robot is tasked to turn its trunk 90° around the world z-axis. 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.

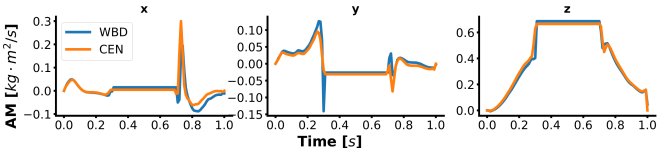


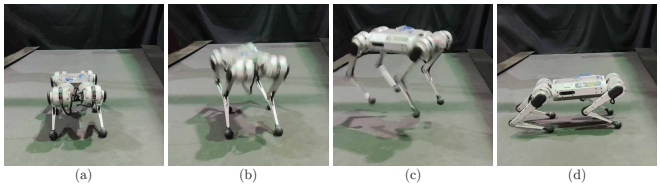
Fig. 8: Dynamics consensus of angular momentum for 90° twisting jump.

The 90° twisting jump maneuver is also studied, shown in Fig. 9. This maneuver consists of three contact phases: takeoff, aerial, and landing. The robot must gain enough AM during takeoff to achieve the goal orientation after landing. In addition to the solve times and residuals provided in Table I, Fig. 8 illustrates desirable consensus performance, for centroidal angular momentum, between the two optimization layers. In the subsequent section, we will discuss the hardware implementation of this maneuver using the convex MPC controller proposed in Sec. IV-B.

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 onboard 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. 10 demonstrates the tracking performance for the twisting jump maneuver. During the takeoff and



(a) (b) (c) (d)

Fig. 9: Demonstration of the Mini Cheetah executing a 90° twisting jump.

aerial phases, both joint and momentum are tracked properly. When early touchdown is detected, tracking references are shifted heuristically as described in the previous paragraph. However, the momentum tracking during the landing phase still leaves much to be desired. As part of our future work, we expect that by properly optimizing contact timing online, instead of employing simple touchdown detection heuristics, the landing phase momentum tracking performance can be improved significantly. Nonetheless, our proposed controller successfully replicates the 90° twisting jump on hardware, as shown in Fig. 9.

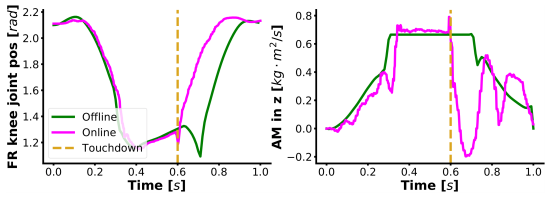


Fig. 10: Tracking performance for twisting jump by comparing the offline reference trajectories (green) and the recorded data during online execution (pink). The brown vertical dashed line denotes the early touchdown.

VI. Conclusions and Future Work

We presented a trajectory optimization and QP-based MPC framework for versatile and agile quadrupedal locomotion. The novelty of our proposed framework lied in the incorporation of both CEN and WBD during the trajectory optimization process. The offline trajectory planner adopted an alternating scheme to achieve dynamics consensus between centroidal and whole-body 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.

References

[1] 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 International Conference on Intelligent Robots and Systems*, 2018, pp. 1–9.

[2] 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.

[3] G. Bledt, "Regularized predictive control framework for robust dynamic legged locomotion," Ph.D. dissertation, Massachusetts Institute of Technology, 2020.

[4] 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.

[5] J. Ahn, S. J. Jorgensen, S. H. Bang, and L. Sentis, "Versatile locomotion planning and control for humanoid robots," *Frontiers in Robotics and AI*, p. 257, 2021.

[6] 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," *arXiv preprint arXiv:2109.04581*, 2021.

[7] V. Zordan, D. Brown, A. Macchietto, and K. Yin, "Control of rotational dynamics for ground and aerial behavior," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 10, pp. 1356–1366, 2014.

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

[9] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1447–1454.

[10] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2536–2542.

[11] 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.

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

[13] 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.

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

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

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

[17] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 93–100.

[18] J.-P. Sleiman, F. Farshidian, and M. Hutter, "Constraint handling in continuous-time ddp-based model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8209–8215.

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

[20] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1168–1175.

[21] Z. Xie, C. K. Liu, and K. Hauser, "Differential dynamic programming with nonlinear constraints," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 695–702.

[22] T. A. Howell, B. E. Jackson, and Z. Manchester, "Altro: A fast solver for constrained trajectory optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7674–7679.

[23] J. Carpentier and N. Mansard, "Multicontact locomotion of legged robots," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, 2018.

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

[25] C. Nguyen and Q. Nguyen, "Contact-timing and trajectory optimization for 3d jumping on quadruped robots," *arXiv preprint arXiv:2110.06764*, 2021.

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

[27] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2703–2710.

[28] B. Ponton, M. Khadiv, A. Meduri, and L. Righetti, "Efficient multicontact pattern generation with sequential convex approximations of the centroidal dynamics," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1661–1679, 2021.

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

[30] M. W. Spong, S. Hutchinson, M. Vidyasagar, *et al.*, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[31] C. Gehring, D. Bellicoso, M. Bloesch, H. Sommer, P. Fankhauser, M. Hutter, and R. Siegwart, "Kindr library-kinematics and dynamics for robotics," 2016.

[32] J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides, "A condensed and sparse qp formulation for predictive control," in *IEEE Conference on Decision and Control*. IEEE, 2011.

[33] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6295–6301.

[34] 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 *Robotics: Science and Systems*, 07 2020.

[35] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019.