From centroidal to whole-body models for legged locomotion: a comparative analysis

Ewen Louis Dantec, Wilson Jallet, Justin Carpentier

From centroidal to whole-body models for legged locomotion: a comparative analysis

Ewen Dantec [a,*], Wilson Jallet [a,b] and Justin Carpentier [a]

***Abstract*— Model predictive control is one of the most common methods for stabilizing the dynamics of a legged robot. Yet, it remains unclear which level of complexity should be considered for modeling the system dynamics. On the one hand, most embedded pipelines for legged locomotion rely on reduced models with low computational load in order to ensure real-time capabilities at the price of not exploiting the full potential of the whole-body dynamics. On the other hand, recent numerical solvers can now generate whole-body trajectories on the fly while still respecting tight time constraints. This paper compares the performances of common dynamic models of increasing complexity (centroidal, kino-dynamics, and whole-body models) in simulation over locomotion problems involving challenging gaits, stairs climbing and balance recovery. We also present a 3-D kino-dynamics model that reformulates centroidal dynamics in the coordinates of the base frame by efficiently leveraging the centroidal momentum equation at the acceleration level. This comparative study uses the humanoid robot Talos and the augmented Lagrangian-based solver ALIGATOR to enforce hard constraints on the optimization problem.**

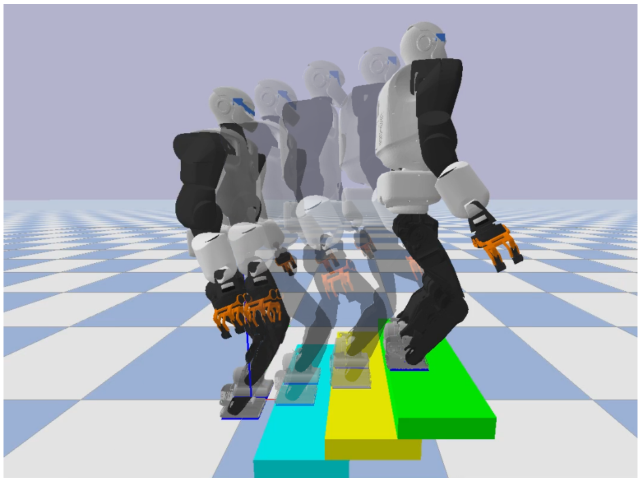

Fig. 1. Snapshot of the Talos humanoid robot climbing a 10-cm stairstep in Bullet simulation using a full dynamics MPC scheme.

I. Introduction

Achieving reliable and efficient biped locomotion in dynamic environments remains challenging. The unstable component of biped dynamics, the intrinsic limitations of ground reaction forces, and the complexity of multi-body systems greatly hinder the capability of a humanoid robot to adapt its motion plan in real time [1], [2].

In the robotics community, model predictive control (MPC) methods [3] are a powerful tool used to perform online re-planning while considering system dynamics and path constraints over a finite horizon. Because they repeatedly solve a receding trajectory optimization problem in close-loop, MPC methods can deal with model discrepancies, state drift as well as unpredicted external disturbances. However, the performances of MPC methods are dependent on the choice of the dynamic model and the predicted horizon length, which both affect the computational complexity of the locomotion problem.

In the context of online optimal control, reduced models such as the linear inverted pendulum (LIP) [4]–[6] or the single rigid body (SRB) [7]–[9] isolate the most important features of locomotion by predicting the evolution of the centroidal part of the dynamics, thus keeping the problem tractable. Interestingly, the work of [10] demonstrates that the sampling frequency of centroidal schemes can be chosen reasonably low without impacting the overall stability of the motion. Consequently, since computation time is not a limiting factor, such schemes can easily be combined with step adaptation heuristics to increase performance and recovery capabilities [11]–[13]. As for drawbacks, these approaches generally struggle to achieve kinematic and dynamical feasibility [14], relying on finely-tuned instantaneous controllers to execute the desired motion plan.

To overcome the kinematics feasibility issue, researchers have proposed to leverage kino-dynamics models in which joint positions are combined with the centroidal dynamics [15]–[19]. However, such models fail to describe the effect of contact interactions on the multi-body system. If the robot features heavy limbs with non-negligible inertia, it may still experience failure. Furthermore, when applied to quadrupedal systems [16], [17], they tend not to properly integrate the effects of whole-body dynamics on the angular momentum of the base frame.

Recent progress in optimization and computational power [1], [2], [20], [21] now allows to leverage the full dynamics of the robot along the predictive horizon without neglecting the inertial effects of the limbs. Contrary to reduced model schemes, full dynamics controllers directly output the feasible torque command needed to execute the desired behavior on torque-controlled systems. This control strategy has been successfully implemented on quadrupeds [22], [23] and humanoids with heavy limbs [24], [25], illustrating the versatility of the method. Nevertheless, full dynamics MPC features a heavy computational load that may hinder their re-planning frequency when working

[a] INRIA - École Normale Supérieure - PSL Research University, Paris, France
[b] LAAS-CNRS, 7 av. du Colonel Roche, 31400 Toulouse
[*]corresponding author: ewen.dantec@inria.fr

with longer horizons, resulting in a loss of reactivity and robustness. In order to achieve a compromise between complexity and tractability, some works have proposed to consider the complete dynamics along the first steps of the trajectory. At the same time, subsequent phases are modeled with reduced dynamics [26], [27]. To this day, it remains unclear whether full dynamics approaches perform better than centroidal ones when solving complex bipedal locomotion tasks.

This paper aims to evaluate the performances of three different modeling approaches of increasing complexity, frequently used in MPC literature:

- the **centroidal** or SRB scheme, minimalist and computationally light;
- the **kino-dynamics** scheme, which stands as a middle ground in terms of complexity and computation load;
- the **full dynamics** scheme, exhaustive but costly to compute.

All three locomotion problems are iteratively solved through the proximal differential dynamic programming (DDP) method introduced in [28], which is able to enforce equality and inequality constraints along the motion path. Among other trajectory optimization techniques, DDP is particularly suited to perform MPC because of its linear complexity in horizon length, local optimal feedback policy, and efficient exploitation of the sparse temporal structure of the problem [20].

As an additional contribution, this paper introduces a 3-D second-order kino-dynamics formulation which does not make any simplifying assumption about the angular momentum of the system (contrary to [16], [17]) and which naturally enforces the consensus between the centroidal model and its kinematic counterpart without relying on additional equality constraints as in [15], [29]. A 2-D version of this formulation has been introduced in [19].

II. Full dynamics MPC scheme

This section presents the details of our multi-contact whole-body optimal control formulation used to produce robust locomotion trajectories.

A. Generic formulation of the optimal control problem

All along this paper, we are interested in solving discrete optimization problems of the following form:

$$\begin{aligned} \min_{\underline{\boldsymbol{x}}, \underline{\boldsymbol{u}}} \sum_{t=0}^{T-1} & \ell_t(\boldsymbol{x}_t, \boldsymbol{u}_t) + \ell_T(\boldsymbol{x}_T) \\ s.t. \qquad & \boldsymbol{x}_0 = \hat{\boldsymbol{x}}_0 \\ & 0 = h_T(\boldsymbol{x}_T) \\ \forall t = 0..T-1, \quad & \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) \\ & 0 = h_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ & 0 \leq g_i(\boldsymbol{x}_i, \boldsymbol{u}_i), \end{aligned} \tag{1}$$

with $\hat{\boldsymbol{x}}_0$ initial measured state of the system, $\underline{\boldsymbol{x}} = (\boldsymbol{x}_t)_{t=0..T}$ and $\underline{\boldsymbol{u}} = (\boldsymbol{u}_t)_{t=0..T-1}$ the decision variables concatenated into state and control trajectories, ℓ_t and ℓ_T the discretized running and terminal cost functions, f the discretized dynamics and $\boldsymbol{h}_i, \boldsymbol{g}_i$ equality and inequality constraints at each knot of the horizon. We also consider the option of adding an equality constraint h_T acting on the terminal state of the problem. Finally, we note δt the problem time step.

B. Whole-body modeling

Let us first consider a humanoid robot system with n_j actuated joints, one free-floating joint, and K end-effector frames in contact with the environment. For this model, we define the whole-body configuration vector of size n_q to be $\boldsymbol{q} \in \mathbf{Q}$ where $\mathbf{Q} = \mathbf{SE}(3) \times \mathbb{R}^{n_j}$ is the configuration manifold, and the corresponding velocity vector of size n_v to be $\dot{\boldsymbol{q}} \in T_e\mathbf{Q} = \mathbf{se}(3) \times \mathbb{R}^{n_j}$. Likewise, the joint acceleration of size n_v is noted $\ddot{\boldsymbol{q}}$.

Following the works of [30], [31] and assuming non-slippage condition (from which it results that contact acceleration is null), the rigid contact dynamics of the multi-body system is described by the Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{J}_c(\boldsymbol{q})^\top \\ \boldsymbol{J}_c(\boldsymbol{q}) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}^\top \boldsymbol{\tau} - \boldsymbol{b}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ -\dot{\boldsymbol{J}}_c(\boldsymbol{q})\dot{\boldsymbol{q}} \end{bmatrix}, \tag{2}$$

where $\boldsymbol{M} \in \mathbb{R}^{n_v \times n_v}$ is the joint-space inertia matrix, $\boldsymbol{b} \in \mathbb{R}^{n_v}$ is the generalized nonlinear forces accounting for centrifugal, Coriolis, and gravitational terms, $\boldsymbol{\lambda} \in \mathbb{R}^{6K}$ and $\boldsymbol{J}_c \in \mathbb{R}^{6K \times n_v}$ stand for the concatenated contact wrenches and contact Jacobians, $\boldsymbol{\tau} \in \mathbb{R}^{n_j}$ is the vector of commanded joint torques and $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_{n_j} \end{bmatrix} \in \mathbb{R}^{n_j \times n_v}$ is the actuation matrix. The whole-body dynamics is under-actuated because the command cannot act directly on the free-floating joint acceleration due to the six first lines of $\boldsymbol{S}^\top$ being null.

In this section only, the state and control of the system are defined to be $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}}) \in T\mathbf{Q}$ and $\boldsymbol{u} = \boldsymbol{\tau}$. As shown in [32], the KKT system of Eq. (2) can be reformulated as a force-free equation and solved using a special Cholesky decomposition [30], [31] during the integration of the dynamics, resulting in the discrete-time equation $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$. For the sake of simplicity and contrary to [20], we do not consider impulse dynamics in this study.

C. Costs formulation

Based on [33], we design a tailored cost function for locomotion composed of 4 weighted running costs: state regularization, control regularization, end-effector tracking, and wrench tracking. The terminal cost is composed of only one state regularization term. The final cost formulation writes:

$$\begin{aligned} \ell_t(\boldsymbol{x}, \boldsymbol{u}) &= \|\boldsymbol{x} - \boldsymbol{x}^*\|^2_{D_x} + \|\boldsymbol{u} - \boldsymbol{u}^*(t)\|^2_{D_u} \\ &\quad + \|\boldsymbol{p}(\boldsymbol{x}) - \boldsymbol{p}^*(t)\|^2_{D_p} + \|\boldsymbol{\lambda}(\boldsymbol{x}, \boldsymbol{u}) - \boldsymbol{\lambda}^*(t)\|^2_{D_\lambda} \\ \ell_T(\boldsymbol{x}) &= \|\boldsymbol{x} - \boldsymbol{x}^*\|^2_{D_x}, \end{aligned} \tag{3}$$

where the symbols $\boldsymbol{D}_*$ represent diagonal positive weight matrices, $\boldsymbol{x}^*$ is the initial state of the robot (defined as the initial half-sitting configuration with zero velocity), $\boldsymbol{u}^*$ is the gravity-compensating torque at a nominal configuration, $(\boldsymbol{p}(\boldsymbol{x}), \boldsymbol{p}^*(t))$ are the current and desired foot placement in $\mathbf{SE(3)}$ and $(\boldsymbol{\lambda}(\boldsymbol{x}, \boldsymbol{u}), \boldsymbol{\lambda}^*(t))$ are the current and desired contact wrenches. The user-defined foot trajectory $\boldsymbol{p}^*(t)$ connects the current and next contacts in the sequence with a parameterized spline whose vertical acceleration is set to be orthogonal to the ground at take-off and landing times. Additionally, vertical velocity is close to zero at impact times in order to neglect impulse dynamics. Note that the placement cost remains active as long as the corresponding foot is in a flying phase. Likewise, we interpolate the vertical part of the force reference $\boldsymbol{\lambda}^*(t)$ to ensure a linear transition of the Center of Pressure (CoP) from one supporting foot to another.

D. Constraints formulation

A classical whole-body locomotion problem features kinematics and dynamics constraints on the state, torque, and contact forces. In [33], such constraints were approximated by quadratic penalization with large weights, resulting in a less numerically robust optimization. This paper, on the other hand, leverages a recently introduced augmented Lagrangian DDP algorithm [28] to handle constraints through a dedicated primal-dual augmented Lagrangian optimization scheme.

Defining the kinematics' lower and upper joint limits as $\boldsymbol{q}_l, \boldsymbol{q}_u \in \mathbb{R}^{n_j}$ and the torque limits as $\boldsymbol{\tau}_u \in \mathbb{R}_+^{n_j}$, the corresponding inequality constraints read

$$g_i^l(\boldsymbol{x}_i, \boldsymbol{u}_i) = \begin{bmatrix} \boldsymbol{X}^j \boldsymbol{x}_i - \boldsymbol{q}_l \\ \boldsymbol{q}_u - \boldsymbol{X}^j \boldsymbol{x}_i \\ \boldsymbol{\tau}_u - \boldsymbol{u}_i \\ \boldsymbol{\tau}_u + \boldsymbol{u}_i \end{bmatrix}, \tag{4}$$

with $\boldsymbol{X}^j \in \mathbb{R}^{n_j \times n_x}$ selecting the positions of the n_j actuated joints.

When dealing with rectangular contact frames, contact wrenches can be approximated by considering that each vertex of the foot applies on the ground a 3D force laying inside a friction cone. The resulting sum of these forces produces a 6D contact wrench $\boldsymbol{\lambda} = (\boldsymbol{f}, \boldsymbol{\tau})$ which, in case of non-sliding contacts, belongs to the classical nonlinear "ice-cream" cone defined in [34]. This cone constraint can be rearranged into a linear formulation by considering a 4-facet friction cone (see [33] for details), leading to the following inequality:

$$g_{i,k}^\lambda(\boldsymbol{x}_i, \boldsymbol{u}_i) = \boldsymbol{A}\boldsymbol{\lambda}_k(\boldsymbol{x}_i, \boldsymbol{u}_i), \tag{5}$$

with $\boldsymbol{A} \in \mathbb{R}^{17\times 6}$ a matrix depending only on friction coefficient and foot parameters, and $\boldsymbol{\lambda}_k(\boldsymbol{x}_i, \boldsymbol{u}_i)$ the constrained wrench at contact k computed from state and torque values at knot i. More details on this formulation are available in [20].

Additionally, a terminal 3D capturability constraint over the Divergent Component of Motion (DCM) [35] position is added to the formulation to prevent fall at the end of the horizon:

$$h_T(\boldsymbol{x}_T) = \boldsymbol{c}_T + \alpha \dot{\boldsymbol{c}}_T - \boldsymbol{d}^*, \tag{6}$$

with $\boldsymbol{c}_T, \dot{\boldsymbol{c}}_T$ position and velocity of the Center of Mass (CoM) at the end of the horizon, α time-constant of the DCM dynamics and $\boldsymbol{d}^*$ a target DCM position corresponding to the center of the support polygon at the end of the horizon.

E. Low-level feedback control

Even with current computation capabilities, full dynamics schemes may not reach the low-level frequency of modern torque-control platforms, which generally run at 1 kHz or more. Handtuned proportional-derivative feedback policies are generally used to mitigate delay and track desired tasks between each command update; however, when using DDP-based solvers to generate full dynamics motions, a Riccati state-feedback policy can be extracted from the backward pass at no additional cost. As demonstrated in other works [16], [23], [36], this particular policy can achieve adaptive impedance based on the costs and constraints of the high-level control problem.

We leverage such a Riccati state-feedback policy in our scheme implementation and compute the commanded torque sent to the simulation as:

$$\boldsymbol{u} = \boldsymbol{\tau} = \boldsymbol{u}_0 + \boldsymbol{K}_0(\boldsymbol{x}_0 \ominus \hat{\boldsymbol{x}}), \tag{7}$$

with $\boldsymbol{u}_0, \boldsymbol{K}_0$ and $\boldsymbol{x}_0$ respectively the feedforward torque, Riccati gain and initial state provided by the MPC block at 100 Hz, and $\hat{\boldsymbol{x}}$ the current state measured at 1 kHz. Operator $\ominus$ refers to the difference operator between two points on the state manifold.

III. Centroidal MPC scheme

In this section, we introduce a nonlinear centroidal formulation coupled with an external whole-body control layer.

A. Centroidal dynamics modelling

Let us consider a centroidal model of mass m subjected only to gravity and ground reaction forces. Abusing the notations, the corresponding state can be written as $\boldsymbol{x} = (\boldsymbol{c}, \boldsymbol{h}_l, \boldsymbol{h}_a)$ with $\boldsymbol{c} \in \mathbb{R}^3$ its CoM, $\boldsymbol{h}_l = m\dot{\boldsymbol{c}} \in \mathbb{R}^3$ its linear momentum and $\boldsymbol{h}_a \in \mathbb{R}^3$ its angular momentum. We write $\boldsymbol{h}_g = (\boldsymbol{h}_l, \boldsymbol{h}_a) \in \mathbb{R}^6$ the total centroidal momentum of the system. The nonlinear dynamics of this reduced model is given by the following equations [37]:

$$\dot{\boldsymbol{h}}_{\boldsymbol{l}} = \sum_{k=1}^{K} \boldsymbol{f}_k + m\boldsymbol{g}, \ \dot{\boldsymbol{h}}_{\boldsymbol{a}} = \sum_{k=1}^{K} (\boldsymbol{r}_k^* - \boldsymbol{c}) \times \boldsymbol{f}_k + \boldsymbol{\tau}_k, \tag{8}$$

with K number of active contacts, $\boldsymbol{r}_k^* \in \mathbb{R}^3$ translation of contact k in world frame, $\boldsymbol{f}_k \in \mathbb{R}^3$ and $\boldsymbol{\tau}_k \in \mathbb{R}^3$ linear and angular part of the corresponding ground reaction forces and $\boldsymbol{g} \in \mathbb{R}^3$ gravitational component.

Denoting $\boldsymbol{\lambda}_k = (\boldsymbol{f}_k, \boldsymbol{\tau}_k)$ as the wrench produced by contact k, the control vector is set to be $\boldsymbol{u} = (\boldsymbol{\lambda}_1, .., \boldsymbol{\lambda}_K) \in \mathbb{R}^{6K}$. Some works integrate the desired contact positions into the control in order to perform step adaptation [13]. As we wish to compare our dynamics models based on the same locomotion task, we do not consider this option here.

B. Costs formulation

Given a sequence of user-defined contact points, the cost formulation of the centroidal scheme aims to regularize the CoM trajectory of the system and prevent unstable behaviours. By design, it cannot provide any way to act on the multi-body configuration of the robot, making it impossible to track end-effector trajectories and maintain a feasible posture. We formulate our cost function as:

$$\begin{aligned} \ell_t(\boldsymbol{x}, \boldsymbol{u}) &= \|\boldsymbol{x} - \boldsymbol{x}^*\|^2_{D_x} + \|\boldsymbol{u} - \boldsymbol{u}^*(t)\|^2_{D_u} + \|\dot{\boldsymbol{h}}_g\|^2_{D_h} \\ \ell_T(\boldsymbol{x}) &= \|\boldsymbol{x} - \boldsymbol{x}^*\|^2_{D_x}, \end{aligned} \tag{9}$$

where the regularization functions now operate over the centroidal state and contact forces. State reference $\boldsymbol{x}^*$ is again constant while control reference $\boldsymbol{u}^*$ is null except for the components corresponding to the vertical part of contact forces, which follow the same heuristics as in Sec. II-C.

C. Constraints formulation

The simplicity of the centroidal model does not allow to guarantee kinematics and dynamics feasibility at the joint level along the predictive horizon. Thus, only two constraints are implemented inside our centroidal problem:

- A contact wrench inequality similar to Eq. (5), except that it operates on the control variable $\boldsymbol{u}_{i,k}$ representing the wrench applied at contact k for node i;
- A DCM capturability condition on the terminal node similar to the one defined in Eq. (6).

D. Low-level torque computation

Our reduced MPC scheme generates an optimal centroidal trajectory as well as the necessary contact forces to execute it. Next, we compute the joint torque vector associated to this motion in two steps. First, the Riccati gains are again leveraged to correct the command and mitigate the effect of control delay; the resulting forces are obtained through a Riccati feedback law similar to the one defined in Eq. (7):

$$\boldsymbol{u} = \boldsymbol{\lambda} = \boldsymbol{u}_0 + \boldsymbol{K}_0(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}(\boldsymbol{q}, \dot{\boldsymbol{q}})), \tag{10}$$

with $\hat{\boldsymbol{x}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ the centroidal state computed from whole-body measures $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ available at 1 kHz.

Second, an inverse dynamics Quadratic Program (QP) solved with the library ProxQP [38] is used to output the final joint torques at a frequency of 1 kHz. Aside from following the motion plan devised by the centroidal MPC, the QP architecture should aim to ensure the feasibility of the whole-body dynamics, keep the robot base straight, track user-defined foot references and stay close to the initial whole-body configuration. The formulation boils down to the inverse dynamics stack-of-task approach used for example in [39] to make a humanoid walk on non-planar surfaces:

$$\begin{aligned} \min_{\boldsymbol{\tau}, \boldsymbol{\delta}\ddot{\boldsymbol{q}}, \boldsymbol{\delta\lambda}} \quad & \|\boldsymbol{\delta}\ddot{\boldsymbol{q}}\|^2_{W_q} + \|\boldsymbol{\delta\lambda}\|^2_{W_\lambda} + \|\dot{\boldsymbol{h}}_g - \dot{\boldsymbol{h}}^*_g\|^2_{W_h} \\ & + \|\ddot{\boldsymbol{w}}_b - \ddot{\boldsymbol{w}}^*_b\|^2_{W_r} + \sum_{k=1}^{K} \|\ddot{\boldsymbol{p}}_k - \ddot{\boldsymbol{p}}^*_k\|^2_{W_p} \\ s.t. \quad & \boldsymbol{M}(\ddot{\boldsymbol{q}}^* + \boldsymbol{\delta}\ddot{\boldsymbol{q}}) + \boldsymbol{b} = \boldsymbol{S}^\top \boldsymbol{\tau} + \boldsymbol{J}_c^\top(\boldsymbol{\lambda} + \boldsymbol{\delta\lambda}) \\ & \boldsymbol{J}_c(\ddot{\boldsymbol{q}}^* + \boldsymbol{\delta}\ddot{\boldsymbol{q}}) + \dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}} = 0 \\ & \boldsymbol{\lambda} + \boldsymbol{\delta\lambda} \in \mathcal{K} \\ & -\boldsymbol{\tau}_u \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_u, \end{aligned} \tag{11}$$

with $\boldsymbol{p}_k \in \mathbf{SE}(3)$ placement of foot k, $\dot{\boldsymbol{p}}_k, \ddot{\boldsymbol{p}}_k \in \mathbb{R}^6$ the corresponding spatial velocity and acceleration, $\boldsymbol{w}_b \in \mathbf{SO(3)}$ rotation of the base frame and $\dot{\boldsymbol{w}}_b, \ddot{\boldsymbol{w}}_b \in \mathbf{so}(3)$ corresponding angular velocity and acceleration, and $\mathcal{K}$ the wrench cone defined by Eq. (5). Weight matrices W_* are chosen to be diagonal. The reference for rate of momentum change $\dot{\boldsymbol{h}}^*_g$ is obtained from the dynamics equation (8) computed at the first knot of the horizon, while the references for joint acceleration, base rotation and end-effector placements are given by:

$$\begin{aligned} \ddot{\boldsymbol{q}}^* &= \boldsymbol{K}^q_p(\boldsymbol{q_0} \ominus \boldsymbol{q}) + \boldsymbol{K}^q_d\dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{w}}^*_b &= \boldsymbol{K}^w_p(\log_3(\boldsymbol{w}_b)) + \boldsymbol{K}^w_d\dot{\boldsymbol{w}}_b \\ \ddot{\boldsymbol{p}}^*_k &= \boldsymbol{K}^p_p(\boldsymbol{p}^*_k \ominus \boldsymbol{p}_k) + \boldsymbol{K}^p_d(\dot{\boldsymbol{p}}^*_k - \dot{\boldsymbol{p}}_k), \end{aligned} \tag{12}$$

with $\boldsymbol{K}_p, \boldsymbol{K}_d$ classical proportional-derivative gains corresponding to each whole-body task. The accelerations associated to base rotation and feet placement, as well as the time derivative of momentum, are computed through:

$$\begin{aligned} \ddot{\boldsymbol{w}}_b &= \boldsymbol{J}_b\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_b\dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{p}}_k &= \boldsymbol{J}_k\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{h}}_g &= \boldsymbol{A}_g\ddot{\boldsymbol{q}} + \dot{\boldsymbol{A}}_g\dot{\boldsymbol{q}}, \end{aligned} \tag{13}$$

with $\boldsymbol{J}_b \in \mathbb{R}^{3\times n_v}$ rotational part of the base Jacobian, $\boldsymbol{J}_k \in \mathbb{R}^{6\times n_v}$ Jacobian of the foot k and $\boldsymbol{A}_g$ the centroidal momentum matrix [37].

IV. Kino-dynamics MPC scheme

This section introduces a kino-dynamics formulation that computes the centroidal dynamics and full kinematics of the system over the predictive horizon.

A. Dynamics formulation

In this section, we now consider that the state contains the full kinematics of the robot, $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n_q+n_v}$, while the control becomes the concatenation of the commanded joint acceleration and contact forces, $\boldsymbol{u} = (\ddot{\boldsymbol{q}}_a, \boldsymbol{\lambda}_1, .., \boldsymbol{\lambda}_K) \in \mathbb{R}^{n_j+6K}$. This formulation offers a trade-off between the extensive but complex full dynamics model and the simple but under-constrained centroidal view. In particular, it allows to consider kinematic tasks and constraints along the predictive horizon without inverting the costly KKT system defined by Eq. (2).

Computing the next state of such a kino-dynamics system amounts to casting the dynamics of the CoM to the robot base frame, or in other words to express the base acceleration with respect to the state, commanded joint acceleration and external forces. To do so, we decompose the centroidal momentum matrix into actuated and unactuated blocks and isolate the unactuated part of the acceleration $\ddot{\boldsymbol{q}}_u \in \mathbf{se}(3)$:

$$\begin{aligned}\dot{\boldsymbol{h}}_g &= \begin{bmatrix}\boldsymbol{A}_{g,u} & \boldsymbol{A}_{g,a}\end{bmatrix}\begin{bmatrix}\ddot{\boldsymbol{q}}_u\\ \ddot{\boldsymbol{q}}_a\end{bmatrix} + \dot{\boldsymbol{A}}_g\dot{\boldsymbol{q}} \\ \Rightarrow \ddot{\boldsymbol{q}}_u &= \boldsymbol{A}_{g,u}^{-1}(\dot{\boldsymbol{h}}_g - \dot{\boldsymbol{A}}_g\dot{\boldsymbol{q}} - \boldsymbol{A}_{g,a}\ddot{\boldsymbol{q}}_a)\end{aligned} \tag{14}$$

The 6×6 matrix $\frac{1}{m}\boldsymbol{A}_{g,u}$ links the base velocity with the CoM spatial velocity and is always invertible [37]. Based on Eq. (8), $\dot{\boldsymbol{h}}_g$ can be computed from current state and control vectors, taking into account that contact translations $\boldsymbol{r}_k$ are now drawn from the kino-dynamics state of the robot rather than defined by a high-level planner:

$$\dot{\boldsymbol{h}}_g = \begin{bmatrix}\sum_{k=1}^{K} \boldsymbol{f}_k + m\boldsymbol{g} \\ \sum_{k=1}^{K}(\boldsymbol{r}_k(\boldsymbol{q}) - \boldsymbol{c}(\boldsymbol{q})) \times \boldsymbol{f}_k + \boldsymbol{\tau}_k\end{bmatrix}. \tag{15}$$

Hence, the optimization scheme can naturally adapt the pose of its contacts without adding any other state variables to the problem as in the centroidal formulation.

B. Costs formulation

In the kino-dynamics setting, the optimization process is again given a contact sequence and a set of user-defined foot reference trajectories that can directly be tracked at the level of the solver. The baseline formulation integrates the state and control regularization and feet tracking costs of Sec. II-C, the only difference being that the default control reference is now defined as the concatenation of a null joint acceleration vector and interpolated wrench distribution. We then complete the cost formulation by adding the total acceleration momentum cost of Sec. III-B:

$$\begin{aligned}\ell_t(\boldsymbol{x},\boldsymbol{u}) &= \|\boldsymbol{x} - \boldsymbol{x}^*\|^2_{D_x} + \|\boldsymbol{u} - \boldsymbol{u}^*(t)\|^2_{D_u} \\ &\quad + \|\dot{\boldsymbol{h}}_g\|^2_{D_h} + \|\boldsymbol{p}(\boldsymbol{x}) - \boldsymbol{p}^*(t)\|^2_{D_p} \\ \ell_T(\boldsymbol{x}) &= \|\boldsymbol{x} - \boldsymbol{x}^*\|^2_{D_x}.\end{aligned} \tag{16}$$

C. Constraints formulation

Contrary to Sec. II, contact acceleration constraints are not naturally enforced at the level of the dynamics and must be actively taken into account. For all foot frames in contact, we add the following equality constraint to the problem formulation:

$$h_i^v(\boldsymbol{x}_i, \boldsymbol{u}_i) = \boldsymbol{J}_c\dot{\boldsymbol{q}} \tag{17}$$

Additionally, the wrench cone constraint defined in Eq. (5) and CoM final constraint defined in Eq. (6) are re-implemented in the kino-dynamics cost formulation to ensure the dynamic stability of the centroidal motion. Similarly, the kinematics' constraint introduced in Eq. (4) is injected into the formulation to account for joint limits.

D. Inverse dynamics controller

As in Sec. III-D, the commanded torque is computed through an inverse dynamics QP which minimizes the acceleration and wrench corrections needed to ensure dynamics feasibility. Contrary to the centroidal case, kino-dynamics equations provide a reference joint acceleration $\ddot{\boldsymbol{q}}$ in addition to the reference wrench $\boldsymbol{\lambda}$. Note that these quantities are again adjusted through the usual Riccati feedback law to account for state drift during MPC computation. Lastly, as the feet tracking and base stabilization tasks are solved at the level of the optimal control problem, the QP formulation takes the following form, inspired from the work of [40]:

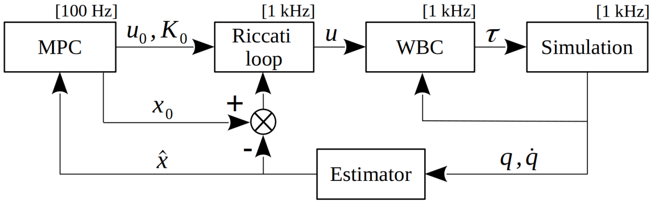

Fig. 2. Generic architecture of the three MPC controllers. In the full dynamics case, no WBC block is needed as the feedback control $\boldsymbol{u}$ is directly the commanded joint torque $\boldsymbol{\tau}$. In the centroidal case, the estimator block is non-trivial and maps whole-body measures to CoM motions.

$$\begin{aligned}\min_{\boldsymbol{\tau},\boldsymbol{\delta\ddot{q}},\boldsymbol{\delta\lambda}} \quad & \|\boldsymbol{\delta\ddot{q}}\|^2_{W_q} + \|\boldsymbol{\delta\lambda}\|^2_{W_\lambda} \\ s.t. \quad & \boldsymbol{M}(\ddot{\boldsymbol{q}} + \boldsymbol{\delta\ddot{q}}) + \boldsymbol{b} = \boldsymbol{S}^\top\boldsymbol{\tau} + \boldsymbol{J}_c^\top(\boldsymbol{\lambda} + \boldsymbol{\delta\lambda}) \\ & \boldsymbol{J}_c(\ddot{\boldsymbol{q}} + \boldsymbol{\delta\ddot{q}}) + \dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}} = 0 \\ & \boldsymbol{\lambda} + \boldsymbol{\delta\lambda} \in \mathcal{K} \\ & -\boldsymbol{\tau}_u \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_u.\end{aligned} \tag{18}$$

V. Comparative Analysis

In this section, we present the results of an extensive benchmark of Bullet simulations [41] which feature the Talos humanoid robot [42] performing dynamic walking motions on flat floor. Our objective is to emphasize the advantages and limits of each model through a series of various experiments involving large stepping motion and push recovery. All MPC schemes are solved with the proximal DDP algorithm with a multithreaded Riccati solver [21] and are based on the real-time iteration heuristics introduced in [43]. Only one DDP iteration is performed to solve the problem at each control cycle. The control architecture for all three schemes is presented in Fig. 2. A video of the experiments is available at https://youtu.be/5_PhNAdqGA, and the source code can be found at https://github.com/edantec/MPC_benchmark.

A. Experimental setting

In the following set of experiments, the robot is required to track a user-defined contact sequence with fixed timings and footstep placements. Contrary to [33], we control the entire body of the robot, including the arms, in order to better highlight the angular momentum dissipation due to upper body motions. The number of actuated joint is set to be $n_j = 22$ (12 leg joints, 2 torso joints, 8 arm joints), so that $n_q = 29, n_v = 28$. Desired foot trajectories are defined as spline curves with variable step lengths. Step duration is fixed at $t_{step} = 1$ s, with $t_{ds} = 0.2$ s and $t_{ss} = 0.8$ s respectively double and simple support duration. The predictive horizon size is chosen to be $T = 100$ with a $\delta t = 10$ ms timestep. The Bullet simulation runs at a frequency of 1 kHz, meaning

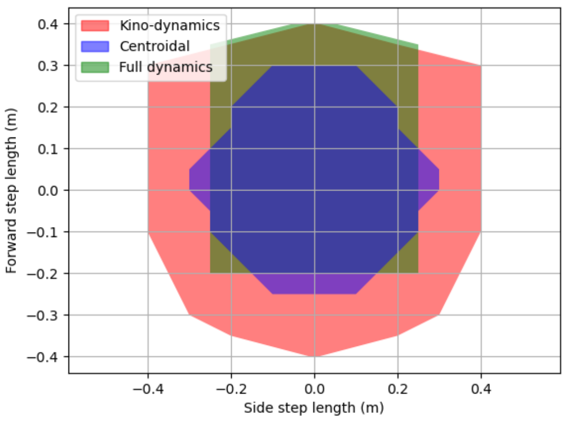

Fig. 3. Maximum footstep size for all models when considering a stepping period of 1 second.

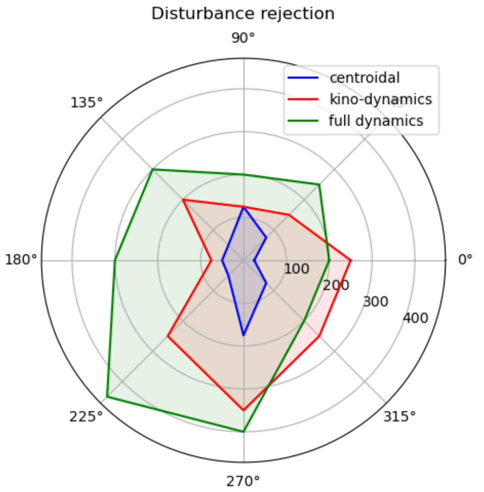

Fig. 4. Maximum force push the MPC schemes can withstand. The force intensity was made to vary from 0 to 450 N and was applied at the center of the base frame during 0.1 s.

that a new MPC policy is recomputed and applied every 10 simulation steps. Control delay is simulated by solving the optimization problem over the state measured 10 ms before instead of the most recent one. For each formulation, the same set of gains is used across all experiments.

The experimental campaign was conducted on a Dell XPS laptop with an Intel i9-13900k CPU composed of 8 performance (P) cores and 16 efficiency (E) cores. Parallelizing the computation over more than 8 threads does not bring any additional gain since the optimization will start to be distributed over less powerful cores.

Because of the increasing complexity of our models, the three optimization schemes feature different computational loads and are not solved at the same frequency, even when accounting for parallelization. As for now, only the centroidal model solution can be recomputed every 10 ms, while kino-dynamics and full dynamics trajectories take twice as long to be obtained (see Table I). Lowering the computation load by increasing the timestep length and decreasing the number of discretization nodes inside the horizon has not allowed to produce stable behaviours for the whole-body MPC. In order to draw a relevant comparison between our models, it thus has been decided to use the same time parametrization for each of them while making the assumption that they all run at a frequency of 100 Hz. Given the constant improvement of optimization techniques and computer power, we consider that this assumption will be verified in a foreseeable future.

TABLE I
AVERAGE COMPUTATION LOAD FOR 1 MPC ITERATION

Computation Model	Mean Time (ms)	Standard Deviation (ms)
Centroidal	2.67	0.280
Kino-dynamics	20.3	1.62
Full dynamics	24.1	2.69

B. Step length evaluation

For this experiment, we measured the maximum step length that the proposed models were able to achieve while sticking to a standard gait period of 1 second. The simulation is considered successful if the robot manages to perform 4 steps before standing still. The results of this experiment are presented in Fig. 3. The kino-dynamics scheme appears to be surprisingly robust as compared to the full dynamics and centroidal ones, which are quite comparable.

C. Push recovery

Next, we evaluated the recovery capabilities of the proposed schemes by applying a virtual force on the center of the base frame during an interval of 0.1 s. The push is scheduled to happen when the right foot is at its highest point while the robot performs a 20-cm forward walk. Changing the orientation of the push allows to draw a capturability region for each model; the simulation is validated only when the robot is able to stop within the next 2 steps. The stability region obtained through this approach is presented in Fig. 4.

Both kino-dynamics and full dynamics models feature better recovery capabilities than the centroidal one. Contrary to what could have been expected, the full dynamics region does not include the kino-dynamics one, despite relying on a more complex dynamics model. The kino-dynamics scheme has been found to be more robust to backward pushes, while the full dynamics recovers more easily when pushed from the front. Disturbances oriented toward the right side are better handled because they take place when the CoM lays above the left foot, meaning that the robot can rely on the soon-to-be right contact to dissipate the exceeding energy of the push.

D. Stairstep trajectory comparison

In this subsection, we benchmark the MPC strategies over a 10-cm stairstep climbing experiment and discuss the resulting motions. The CoP trajectories for all models are displayed in Fig. 5 along with the corresponding support

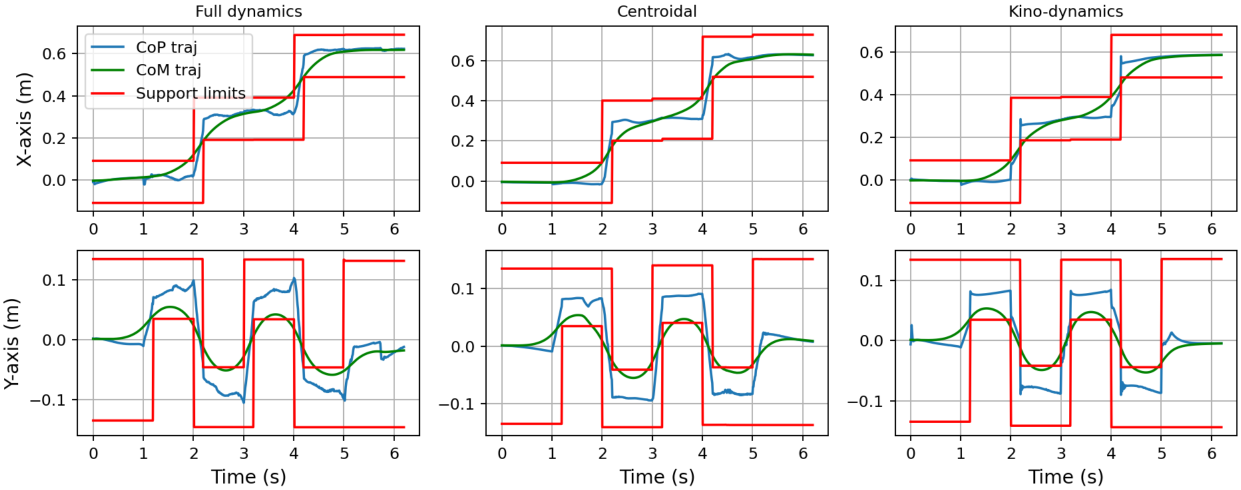

Fig. 5. From left to right: full dynamics, centroidal and kino-dynamics CoP trajectory for a 10-cm stairstep climbing motion.

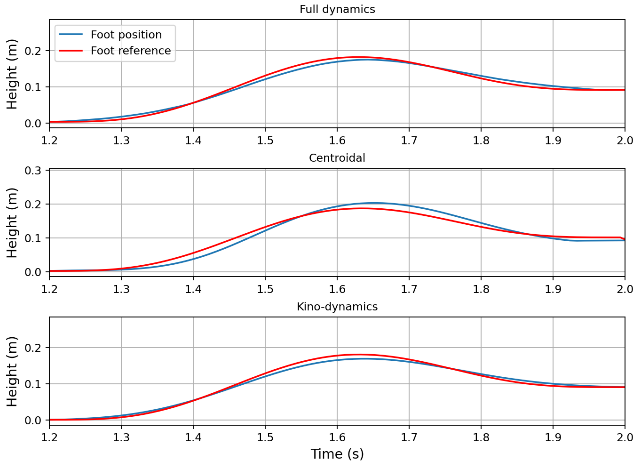

Fig. 6. Comparison of the vertical component of the left foot position during stairstep climbing.

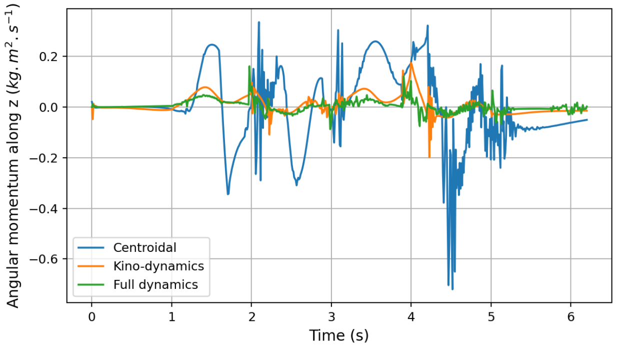

Fig. 7. Comparison of the vertical component of the angular momentum during stairstep climbing.

limits. It can be observed that the CoP stability margins tend to be smaller in the kino-dynamics case.

Likewise, the resulting trajectories of the left foot along the vertical axis are presented in Fig. 6. As can be noted, the centroidal scheme features a slight delay in foot tracking due to not possessing information about the future reference motion. On the contrary, kino-dynamics and full dynamics schemes anticipate the rise of the foot and better follow the reference spline.

We display in Fig. 7 the angular momentum trajectories along the Z-axis of the world frame. While kino-dynamics and full dynamics formulations achieve small momentum, the centroidal scheme struggles to do the same. This is a direct consequence of the fact that the influence of whole-body motions on angular momentum is considered only at the level of the inverse dynamics QP in the centroidal case. Kino-dynamics and full dynamics, on the other hand, take into account whole-body motions when predicting the angular momentum along the horizon.

Finally, the mechanical joint power and total dissipated energy for the stairstep experiment are presented in Table II. The centroidal model appears to perform poorly in terms of energy consumption, as compared to the others schemes.

TABLE II

ENERGY CONSUMPTION FOR STAIRSTEP EXPERIMENT

Computation Model	**Mean power (W)**	**Dissipated energy (J)**
Centroidal	94.8	588.0
Kino-dynamics	63.5	393.8
Full dynamics	55.9	346.7

VI. Conclusion

This work draws an empirical comparison between three control schemes for locomotion based on a set of increasingly complex models of the robot system. Experiments performed in simulation illustrate the capabilities and limitations of each architecture in terms of push recovery, stepping region size and CoP stability margins. Results reveal that given a fixed contact schedule, full dynamics schemes do not always outperform reduced model approaches, in particular kino-dynamics ones. Computing the full kinematics of the robot, on the other hand, greatly improves the walking performance and allows to perform step adaptation, at the price of a computation load one order of magnitude larger than the centroidal scheme. In order to better highlight the benefits of full dynamics approaches, it may be necessary to consider more complex motions with unstructured environments and challenging multi-body tasks.

In this study, control delays have not been considered in order to be able to compare the control schemes on an equal basis; yet the additional load induced by whole-body models computation cannot be ignored in real-world applications. We plan to reduce this load in future works by improving the optimization process and exploring the possibility to use longer timesteps combined with more precise integrators.

References

[1] J. Carpentier and P.-B. Wieber, "Recent progress in legged robots locomotion control," *Current Robotics Reports*, vol. 2, no. 3, pp. 231–238, 2021.

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

[3] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation and Design*. Nob Hill Publishing, 2017.

[4] P.-B. Wieber and C. Chevallereau, "Online adaptation of reference trajectories for the control of walking systems," *Robotics and Autonomous Systems*, vol. 54, no. 7, pp. 559–566, Jul. 2006.

[5] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, 2017.

[6] F. M. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "From Walking to Running: 3D Humanoid Gait Generation via MPC," *Frontiers in Robotics and AI*, vol. 9, 2022.

[7] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, "A versatile and efficient pattern generator for generalized legged locomotion," in *IEEE ICRA*, May 2016.

[8] 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, Jul. 2018.

[9] N. Rathod, A. Bratta, M. Focchi, M. Zanon, O. Villarreal, C. Semini, and A. Bemporad, "Model Predictive Control with Environment Adaptation for Legged Locomotion," *IEEE Access*, vol. 9, pp. 145 710–145 727, 2021.

[10] N. A. Villa, J. Englsberger, and P.-B. Wieber, "Sensitivity of Legged Balance Control to Uncertainties and Sampling Period," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3665–3670, Oct. 2019.

[11] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Foot Step Placement," *Advanced Robotics*, vol. 24, pp. 719–737, Apr. 2010.

[12] T. Corbères, T. Flayols, P.-A. Léziart, R. Budhiraja, P. Souères, G. Saurel, and N. Mansard, "Comparison of predictive controllers for locomotion and balance recovery of quadruped robots," in *IEEE ICRA*, May 2021.

[13] G. Romualdi, S. Dafarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online Non-linear Centroidal MPC for Humanoid Robot Locomotion with Step Adjustment," in *IEEE ICRA*, May 2022.

[14] J. Carpentier and N. Mansard, "Multicontact Locomotion of Legged Robots," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, Dec. 2018.

[15] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE-RAS Humanoids*, 2015.

[16] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," in *IEEE/RSJ IROS*, 2019.

[17] H. Li, T. Zhang, W. Yu, and P. M. Wensing, "Versatile Real-Time Motion Synthesis via Kino-Dynamic MPC with Hybrid-Systems DDP," in *IEEE ICRA*, Sep. 2022.

[18] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *IEEE-RAS Humanoids*, 2017.

[19] M. Y. Galliker, N. Csomay-Shanklin, R. Grandia, A. J. Taylor, F. Farshidian, M. Hutter, and A. D. Ames, "Bipedal Locomotion with Nonlinear Model Predictive Control: Online Gait Generation using Whole-Body Dynamics," in *IEEE-RAS Humanoids*, 2022.

[20] C. Mastalli, R. Budhiraja *et al.*, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *IEEE ICRA*, 2020.

[21] W. Jallet, E. Dantec, E. Arlaud, N. Mansard, and J. Carpentier, "Parallel and proximal constrained linear-quadratic methods for real-time nonlinear mpc," in *Robotics: Science and Systems*, 2024.

[22] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, 2018.

[23] C. Mastalli, W. Merkt, G. Xin, J. Shim, M. Mistry, I. Havoutis, and S. Vijayakumar, "Agile Maneuvers in Legged Robots: a Predictive Control Approach," Jul. 2022, arXiv.

[24] S. Katayama and T. Ohtsuka, "Whole-body model predictive control with rigid contacts via online switching time optimization," in *IEEE/RSJ IROS*, Oct. 2022.

[25] E. Dantec, M. Naveau, P. Fernbach, N. Villa, G. Saurel, O. Stasse, M. Taix, and N. Mansard, "Whole-Body Model Predictive Control for Biped Locomotion on a Torque-Controlled Humanoid Robot," in *IEEE-RAS Humanoids*, Nov. 2022.

[26] H. Li, R. J. Frei, and P. M. Wensing, "Model hierarchy predictive control of robotic systems," *IEEE Robotics and Automation Letters*, vol. 6, 2021.

[27] M. H. Yeganegi, M. Khadiv, A. D. Prete, S. A. A. Moosavian, and L. Righetti, "Robust Walking Based on MPC With Viability Guarantees," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2389–2404, Aug. 2022.

[28] W. Jallet, A. Bambade, N. Mansard, and J. Carpentier, "Constrained Differential Dynamic Programming: A primal-dual augmented Lagrangian approach," in *IEEE/RSJ IROS*, Oct. 2022.

[29] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, Mar. 2016.

[30] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2014.

[31] J. Carpentier, R. Budhiraja, and N. Mansard, "Proximal and sparse resolution of constrained dynamic equations," in *R:SS*, 2021.

[32] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "DDP for multi-phase rigid contact dynamics," in *IEEE-RAS Humanoids*, 2018.

[33] E. Dantec, M. Naveau, P. Fernbach, N. Villa, G. Saurel, O. Stasse, M. Taix, and N. Mansard, "Whole-Body Model Predictive Control for Biped Locomotion on a Torque-Controlled Humanoid Robot," in *IEEE-RAS Humanoids*, Nov. 2022.

[34] S. Caron, Q.-C. Pham, and Y. Nakamura, "Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas," in *IEEE ICRA*, 2015.

[35] A. Pajon and P.-B. Wieber, "Safe 3d bipedal walking through linear mpc with 3d capturability," in *IEEE ICRA*, 2019.

[36] E. Dantec, M. Taix, and N. Mansard, "First order approximation of model predictive control solutions for high frequency feedback," *IEEE Robotics and Automation Letters*, vol. 7, 2022.

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

[38] A. Bambade, F. Shramm, A. Taylor, and J. Carpentier, "QPLayer: efficient differentiation of convex quadratic optimization," 2023.

[39] O. E. Ramos, N. Mansard, O. Stasse, and P. Souères, "Walking on non-planar surfaces using an inverse dynamic stack of tasks," in *IEEE-RAS Humanoids*, Nov. 2012.

[40] D. Kim, J. Lee, O. Campbell, H. Hwang, and L. Sentis, "Computationally-robust and efficient prioritized whole-body controller with contact constraints," in *IEEE/RSJ IROS*, 2018.

[41] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[42] O. Stasse, T. Flayols *et al.*, "Talos: A new humanoid research platform targeted for industrial applications," *IEEE-RAS Humanoids*, 2017.

[43] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on control and optimization*, vol. 43, p. 1714–1736, 2005.