Reference-Free Model Predictive Control for Quadrupedal Locomotion

Gianni Lunardi, Thomas Corbères, Carlos Mastalli, Nicolas Mansard, Thomas Flayols, Steve Tonneau, Andrea Del Prete

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 October 2023, accepted 8 December 2023, date of publication 19 December 2023,
date of current version 3 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3345157

RESEARCH ARTICLE

Reference-Free Model Predictive Control for Quadrupedal Locomotion

GIANNI LUNARDI[1], THOMAS CORBÈRES[2], (Graduate Student Member, IEEE), CARLOS MASTALLI[3,4], (Member, IEEE), NICOLAS MANSARD[5], THOMAS FLAYOLS[5], STEVE TONNEAU[2], AND ANDREA DEL PRETE[1], (Member, IEEE)

[1]Department of Industrial Engineering, University of Trento, 38122 Trento, Italy
[2]School of Informatics, The University of Edinburgh, EH8 9AB Edinburgh, U.K.
[3]School of Engineering and Physical Sciences, Heriot-Watt University, EH14 4AS Edinburgh, U.K.
[4]Florida Institute for Human and Machine Cognition, Penscola, FL 32502, USA
[5]LAAS/CNRS, 31400 Toulouse, France

Corresponding author: Gianni Lunardi (gianni.lunardi@unitn.it)

This work was supported in part by the EU H2020 Project Memory of Motion (MEMMO) under Grant 780684.

ABSTRACT Full-dynamics model predictive control (MPC) has recently been applied to quadrupedal locomotion in semi-unstructured environments. These advances have been fueled by the availability of efficient trajectory optimization (TO) algorithms and inexpensive computational power. The main advantages of full-dynamics MPC are (i) enabling complex locomotion manoeuvres, (ii) considering actuation limits, and (iii) improving robot stability. However, to make the TO problem sufficiently simple to be solved at run time, reference swing foot trajectories are usually tracked in the MPC formulation. These trajectories are often computed independently of the motion of the joints, limiting the approach generality and capability. To address this limitation, we present a full-dynamics MPC formulation that does not require reference swing-foot trajectories, featuring a novel cost function targeting swing foot motion and considering environmental information. Removing the need for reference swing foot trajectories, our approach can also automatically adjust footstep locations, as long as the contact surfaces are predefined. We have validated our MPC formulation through simulations and experiments on the ANYmal B robot. Our approach has similar computational efficiency to state-of-the-art formulations, while displaying superior push-recovery capabilities on various terrains.

INDEX TERMS Full-body MPC, collision avoidance, DDP.

I. INTRODUCTION

Fast growth in legged robot hardware quality has led to more impressive locomotion behaviors. To achieve so, a significant part of the research community has focused on efficient optimal control (OC) techniques [1], [2], [3], [4], [5]. Full-dynamics MPC [6], [7] can improve robot stability and enable complex locomotion manoeuvres while considering the actuation limits.

Early work on MPC for legged locomotion mostly relied on simplified models, such as linear inverted pendulum (LIP) [8], spring-loaded inverted pendulum [9], or centroidal dynamics [10]. However, these models rely on assumptions that limit their applicability, e.g., walking or running on flat ground. In particular, simplified models omit several whole-body features such as swing foot trajectories, which are crucial for obstacle avoidance. Obstacles can still be accounted for in a swing-foot planner, for instance via cubic spline [11] or Bezier curve [12] optimization. However, these methods cannot guarantee the robot's balance and limb momenta compensation.

For these reasons, incorporating full-body dynamics into MPC is crucial for legged locomotion over *obstacles*, and a number of works have exploited it for this purpose. For instance, an obstacle potential field, as a function of the joint configuration, can be added to the cost to obtain

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo.

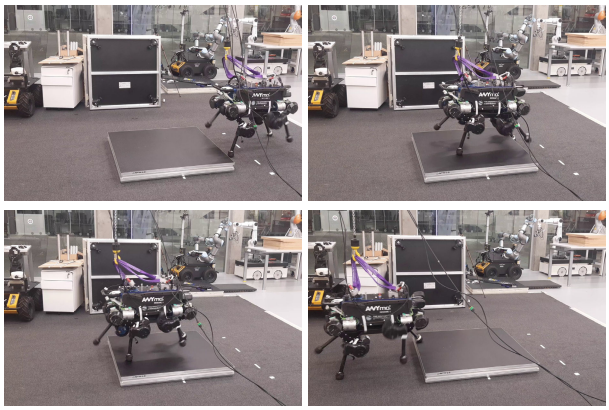


FIGURE 1. ANYmal trotting on a stair (10 height and 1 m length) using the novel cost formulation. Experiments can be seen at https://youtu.be/3lK3TxqPrjw.

a smooth state trajectory [13]. Alternatively, we adopt a continuous-time collision avoidance harmony potential for modelling fixed or mobile objects as in [14]. Wermelinger et. al. [15] use exteroceptive sensors to build traversability map based on different properties of the terrain, such as roughness, slope, obstacles, etc. In general, obstacles can be accounted for through penalty functions depending on the distance between objects, which can be computed efficiently using state-of-the-art algorithms [16], [17].

Full-body dynamics lead to non-convex and high-dimensional Optimal Control Problems (OCPs), which are challenging to solve inside fast control loops (between 50 and 100 Hz). Neunert et. al. [18] develop an explicit spring-damper contact model to directly optimize contact locations, sequence and timings. However, this could results in local minima and nonphysical modeling, since the contact forces never vanish during the swing motion. To increase full-body MPC convergence rates, priors can be injected into the OCP (e.g. reference CoM and swing-foot trajectories) [6], [19]. Swing-foot references can be computed as polynomials [20] or Bezier curves [12], [21]. In [22] nominal footholds can be obtained from base pose reference and elevation terrain, then are projected into convex footstep location constraints; for the swing trajectories, heuristic splines are still used. Although these tailored costs lead to faster convergence, locomotion robustness can be compromised if external disturbances make reference trajectories unfeasible.

A. CONTRIBUTION

In this paper, we propose a novel reference-free MPC formulation to achieve robust locomotion behaviors by optimizing swing foot motions. The key challenge to enable this is to avoid collisions between the swinging foot and the environment. This typically requires moving slowly when close to obstacles to prevent accidentally bumping into them. To achieve this, we introduce a differentiable nonlinear cost function that leverages geometric information to increase foot-velocity penalization as the foot gets closer to an obstacle. With respect to reference-based MPC [6], that we identify as *Ref* throughout the paper, our approach allows a larger range of motions and shows improved push-recovery capabilities. It recovers from pushes up to twice as strong, while avoiding obstacles with the swing feet. We validated our approach through simulations and real experiments (see the snapshots in Fig. 1) with a full-body MPC controller.

The paper is organized as follows: Section II briefly discusses full-body dynamics subject to contact constraints. Section III presents the main contributions: our novel cost function and the MPC formulation. Simulation results and experimental trials are reported in Sections IV and V, respectively. Section VI draws the conclusions.

II. BACKGROUND

In this section, we provide a brief description of the equation of motion for rigid body systems subjected to contact holonomic constraints. This is also known as contact and impulse dynamics.

A. CONTACT DYNAMICS

Given the state $\mathbf{x} = \begin{bmatrix}\mathbf{q}^\top & \mathbf{v}^\top\end{bmatrix}^\top$, the rigid body dynamics of a legged robot can be described as follows [23]:

$$\begin{bmatrix}\mathbf{M} & \mathbf{J}_c^\top \\ \mathbf{J}_c & \mathbf{0}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{v}} \\ -\boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{S}^\top\boldsymbol{\tau} - \mathbf{b} \\ -\mathbf{a}_0\end{bmatrix}, \tag{1}$$

where $\begin{bmatrix}\mathbf{v}^\top & \dot{\mathbf{v}}^\top\end{bmatrix}^\top$ lies in the tangent space of the state manifold; $\mathbf{q} = \begin{bmatrix}{}^b\mathbf{q}^\top & {}^j\mathbf{q}^\top\end{bmatrix}^\top \in \mathbb{R}^{n_q}$ is the generalized position, containing both the base's position (modeled as a $\mathbb{SE}(3)$ group) and the joint configuration, while $\mathbf{v} = \begin{bmatrix}{}^b\mathbf{v}^\top & {}^j\mathbf{v}^\top\end{bmatrix}^\top$, $\dot{\mathbf{v}} \in \mathbb{R}^{n_v}$ are the respective generalized velocity and acceleration; $\mathbf{M} \in \mathbb{R}^{n_v \times n_v}$ is the mass matrix, $\mathbf{J}_c \in \mathbb{R}^{n_c \times n_v}$ is the contact Jacobian, $\mathbf{S} \in \mathbb{R}^{n_j \times n_v}$ is a selection matrix for the actuated joints, $\boldsymbol{\tau} \in \mathbb{R}^{n_j}$ is the joint efforts vector, and $\mathbf{b} \in \mathbb{R}^{n_v}$ accounts for the nonlinear effects; $\boldsymbol{\lambda} \in \mathbb{R}^{n_c}$ are the contact forces and $\mathbf{a}_0 \in \mathbb{R}^{n_c}$ is the desired acceleration in the constraint space. To improve numerical integration stability, $\mathbf{a}_0$ is defined in a Baumgarte stabilization fashion [24] with the following PD law:

$$\mathbf{a}_0 = \mathbf{a}_\lambda - k_p\, {}^oM_\lambda^{\text{ref}}\, {}^oM_\lambda{}^{-1} - k_d\mathbf{v}_\lambda, \tag{2}$$

where $\mathbf{v}_\lambda$, $\mathbf{a}_\lambda$ are the spatial velocity and acceleration, using the notation in [25], at the parent body of the contact, ${}^oM_\lambda^{\text{ref}}\, {}^oM_\lambda{}^{-1}$ is the difference between the reference and current contact pose, represented as transformation matrix [26], while k_p, k_d are positive stabilization gains.

B. IMPACT DYNAMICS

When the robot foot collides with the environment, it is subject to an impulse, and the robot dynamics are [1]:

$$\begin{bmatrix}\mathbf{M} & \mathbf{J}_c^\top \\ \mathbf{J}_c & \mathbf{0}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}^+ \\ -\boldsymbol{\Lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{M}\dot{\mathbf{q}}^- \\ -e\,\dot{\mathbf{J}}_c\dot{\mathbf{q}}^-\end{bmatrix}, \tag{3}$$

where $\dot{\mathbf{q}}^+$ and $\dot{\mathbf{q}}^-$ are the generalized velocities after and before the impact, respectively, $\mathbf{\Lambda}$ is the impulse vector, while $e \in [0, 1]$ is the restitution coefficient. Note that for inelastic contacts, the restitution coefficient is 0.

III. MPC FORMULATION

This section presents the MPC formulation used for generating locomotion behaviors, with particular focus on the novel *fly-high* cost. The MPC problem can be expressed as:

$$\begin{aligned} \min_{\mathbf{X}_j, \mathbf{U}_j} \quad & \sum_{k=j}^{j+N-1} \ell_k(\mathbf{x}_k, \mathbf{u}_k) + \ell_{j+N}(\mathbf{x}_{j+N}) \\ \text{s.t.} \quad & \mathbf{x}_j = \hat{\mathbf{x}}_j, \\ & \mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k), \\ & \underline{\mathbf{u}} \le \mathbf{u}_k \le \overline{\mathbf{u}}, \quad \forall\, k \in [j, j+N-1], \end{aligned} \tag{4}$$

where $\mathbf{X}_j = (\mathbf{x}_j, \dots, \mathbf{x}_{j+N})$ and $\mathbf{U}_j = (\mathbf{u}_j, \dots, \mathbf{u}_{j+N-1})$ are respectively the concatenations of the states and the controls, j is the current time step, N is the horizon length, $\hat{\mathbf{x}}_j$ is the estimated initial state; $\ell(\cdot)$ describes the running or terminal costs, $\mathbf{f}(\cdot)$ is the discretized nonlinear robot dynamics, which can be either (1) or (3), and $\underline{\mathbf{u}}$, $\overline{\mathbf{u}}$ are the control bounds.

In this work, we have used the box feasibility-driven differential dynamic programming (Box-FDDP) algorithm [27] to solve the OCPs. This solver is a direct multiple-shooting version of the DDP algorithm, which provides better numerical stability and more initialization options. Additionally, Box-FDDP can handle control bounds (i.e., joint torque limits).

A. FLY-HIGH COSTS

In previous works (e.g., [6], [21], [28]), $\ell(\cdot)$ contains a cost to stiffly track some user-defined reference swing foot trajectory, limiting the range of motions that the solver can discover. We introduce a novel cost, called "fly-high", to replace the classic trajectory tracking cost. Our key idea is to increasingly penalize the foot $f \in \{\text{lf}, \text{lh}, \text{rf}, \text{rh}\}$ proximity to obstacles, as its velocity ${}^f\dot{\mathbf{p}}$ grows. In this way, we can expect the foot to move away from obstacles when swinging fast. In contrast, it should slow down when it needs to approach an obstacle in order to make contact. This heuristic behavior resembles biological systems' movement, as shown in [29] for humans walking and running. This locomotion strategy leads to robust results. Below, we suggest two versions of this strategy defined in terms of cost functions.

1) SQUARE FLY-HIGH COST

Our first definition has a square in its formulation, i.e.,

$$\ell^{f,o}_{\text{sqr}}(\mathbf{x}) = \begin{cases} \left\| \dfrac{\beta}{(\phi^{f,o})^2 + \beta} {}^f\dot{\mathbf{p}}^2_{\text{xy}} \right\|^2, & \text{if } \phi^{f,o} \ge 0 \\ \left\| {}^f\dot{\mathbf{p}}^2_{\text{xy}} \right\|^2, & \text{if } \phi^{f,o} < 0, \end{cases} \tag{5}$$

where ${}^f\dot{\mathbf{p}}^2_{\text{xy}} = \left[{}^f\dot{p}^2_x \;\; {}^f\dot{p}^2_y \right]^\top \in \mathbb{R}^2$ contains the longitudinal and lateral linear velocities of the swinging foot f, $\phi^{f,o} \in \mathbb{R}$ is the signed distance between the foot and the o-th obstacle, while $\beta \in \mathbb{R}^+$ is a small positive constant used to smooth the cost when $\phi^{f,o} \to 0$. Signed distances can be efficiently computed using the GJK algorithm [16], for example, available in [17] and [30]. Despite its piecewise definition, this cost is continuous and differentiable.

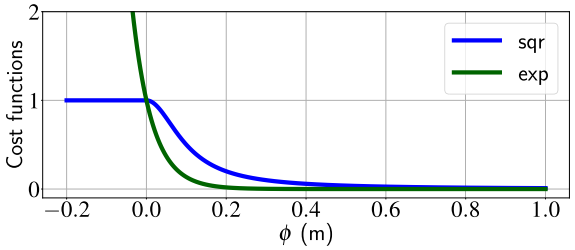


FIGURE 2. Fly-high cost functions (5) and (6) for ${}^f\dot{\mathbf{p}}_{xy} = (1, 1)$ and $\beta = 10^{-2}$, $\gamma = 40$.

2) EXPONENTIAL FLY-HIGH COST

Our second definition relies on the exponential function, i.e.,

$$\ell^{f,o}_{\text{exp}}(\mathbf{x}) = \left\| {}^f\dot{\mathbf{p}}_{\text{xy}}\, e^{-\gamma\phi^{f,o}} \right\|^2, \quad {}^f\dot{\mathbf{p}}_{\text{xy}} = \left[{}^f\dot{p}_x \;\; {}^f\dot{p}_y \right]^\top \in \mathbb{R}^2, \tag{6}$$

where γ is a hyperparameter to ponder the effect of $\phi^{f,o}$ on the cost. Thanks to the monotonic properties of the exponential function,[1] this cost also deals with colliding objects, so with $\phi^{f,o} < 0$.

Fig. 2 shows the fly-high costs argument as a function of the distance $\phi^{f,o}$, considering a fixed value of velocity ${}^f\dot{\mathbf{p}}_{\text{xy}} = (1, 1)$ (chosen for visualization purposes). The two costs are both monotonically non-increasing along $\phi^{f,o}$. Thus, the solver will penalize the foot proximity to the obstacle. Function (5) is constant in $\phi^{f,o}$ for negative distances to ensure differentiability $\forall\phi^{f,o}$. With a velocity close to zero, both costs reach their minimum value, so foot contact with the environment ($\phi^{f,o} = 0$) is allowed. During the MPC loop resolution, $\phi^{f,o} < 0$ happens rarely, thus the constant behaviour of function (5) does not degrade performances.

3) ANALYTICAL DERIVATIVES

Since most of the costs, including the fly-high ones, depend on the squared norm function applied to an argument function $\mathbf{r}$: $\mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \longrightarrow \mathbb{R}^{n_r}$, we can consider the following notation:

$$\ell(\mathbf{x}, \mathbf{u}) = a(\mathbf{r}(\mathbf{x}, \mathbf{u})), \quad a(\mathbf{r}) = \|\mathbf{r}\|^2 \in \mathbb{R}_{\ge 0}, \tag{7}$$

where cost derivatives can be computed using the chain rule. The cost Hessian $\boldsymbol{\ell}_{\mathbf{xx}} = \partial^2\ell/\partial\mathbf{x}^2$ is computed through the Gauss-Newton approximation, to avoid the tensor $\mathbf{R}_{\mathbf{xx}}$ and ensure positive definiteness:

$$\boldsymbol{\ell}_{\mathbf{xx}} = \mathbf{R}_{\mathbf{x}}^\top \mathbf{A}_{\mathbf{rr}} \mathbf{R}_{\mathbf{x}} + \mathbf{R}_{\mathbf{xx}}^\top \mathbf{A}_{\mathbf{r}} \approx \mathbf{R}_{\mathbf{x}}^\top \mathbf{A}_{\mathbf{rr}} \mathbf{R}_{\mathbf{x}}, \tag{8}$$

[1] Although penetration distances can be more difficult to define [31].

where $\mathbf{A_r} = \partial a/\partial \mathbf{r}$, $\mathbf{A_{rr}} = \partial^2 a/\partial \mathbf{r}^2$. The Jacobian $\mathbf{R_x}$ is obtained through the chain rule:

$$\mathbf{R_x} = \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \frac{\partial \mathbf{r}}{\partial {}^f\dot{\mathbf{p}}_{xy}} \frac{\partial {}^f\dot{\mathbf{p}}_{xy}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}}{\partial \phi^{f,o}} \frac{\partial \phi^{f,o}}{\partial \mathbf{x}}. \tag{9}$$

Further details on the computations of object distance derivatives can be found in Appendix A.

B. OTHER COST FUNCTIONS

To achieve quadrupedal locomotion, other cost functions are summed to the fly-high cost. Let us consider the following notation for cost weights $\mathbf{w}$:

$$\|\boldsymbol{\eta}\|^2_{\mathbf{w}} = \sum_{i=1}^{n} w_i \eta_i^2, \quad \mathbf{w} = (w_1, \ldots, w_n), \tag{10}$$

where some weights will be reduced to scalars.

1) REGULARIZATION

This cost penalizes large deviations of the state from a reference configuration:

$$\ell_x(\mathbf{x}) = \left\|\mathbf{x} - \mathbf{x}^*\right\|^2_{\mathbf{w_x}}, \tag{11}$$

where $\mathbf{x}^*$ refers to the backward and forward configurations for the front and rear legs respectively.

2) BOUNDS

Since the Box-FDDP solver cannot handle state constraints, we penalize violations of the state bounds $\underline{\mathbf{x}}$ and $\overline{\mathbf{x}}$:

$$\ell_{\text{bx}}(\mathbf{x}) = \left\|\max(\underline{\mathbf{x}} - \mathbf{x}, 0)\right\|^2_{\mathbf{w}_{\text{bx}}} + \|\max(0, \mathbf{x} - \overline{\mathbf{x}})\|^2_{\mathbf{w}_{\text{bx}}}, \tag{12}$$

where the max operator acts element-wise.

3) FRICTION CONES

Considering the usual linear approximation of friction cones [32], the Coulomb inequalities are:

$$\left|\lambda^f_x\right| \leq \mu\, \lambda^f_z \qquad \left|\lambda^f_y\right| \leq \mu\, \lambda^f_z \qquad \lambda^f_z \geq \lambda^{\min}_z. \tag{13}$$

These can be written in matrix form as $\mathbf{A}\boldsymbol{\lambda} \geq \mathbf{c}$. Then, the bounds on the contact forces are accounted for by the following penalty term:

$$\ell_{\text{cone}}(\mathbf{x}) = \|\min(\mathbf{A}\boldsymbol{\lambda} - \mathbf{c}, 0)\|^2_{\mathbf{w}_{\text{cone}}}. \tag{14}$$

4) CONTACT SEQUENCE

Contrary to the standard formulation *Ref*, in this work we have removed the costs related to tracking the reference trajectories of the Center of Mass (CoM) and the feet. However, the MPC needs guidance to move inside an environment. We have considered two approaches: in the first one, fixed footstep positions ${}^f\mathbf{p}^*$ are imposed through the following cost at the moment contacts are made:

$$\ell_c(\mathbf{x}) = \left\|{}^f\mathbf{p} - {}^f\mathbf{p}^*\right\|^2_{\mathbf{w}_c}. \tag{15}$$

This method will be used in all the environments with stairs. In the second approach, the contact sequence is left free, except for the vertical direction, by setting $\mathbf{w}_c = (0, 0, w_c)$. This procedure can be useful for flat terrain scenario, where footstep locations can be freely optimized. To ensure motion in a particular direction, a reference linear velocity ${}^b\mathbf{v}^* \in \mathbb{R}^3$ is tracked by the robot's base:

$$\ell_t(\mathbf{x}) = \left\|{}^b\mathbf{v} - {}^b\mathbf{v}^*\right\|^2_{\mathbf{w}_t}. \tag{16}$$

C. CONTACT SCHEDULE

In this work, we use the same contact schedule as in [6]. The contact schedule can be defined as a sequence of phases, which can be active or inactive: during the active phase, the foot position is constrained to be fixed, while in the inactive phase the motion of the foot is left free. The transitions between contact phases occur at each:

$$\sum_{j=1}^{n_f} T^a_j + T^i_j, \tag{17}$$

where n_f is the number of feet and T^a, T^i are the duration of the active and inactive phases, respectively. Depending on the type of locomotion, the phases can have different durations, including zero. Each timing is defined a priori before the MPC pipeline starts. We have considered quadruped trotting motions, in which the legs move in diagonal pairs (e.g. the right-front leg moves with the left-hind leg).

D. LOW-LEVEL WHOLE-BODY-CONTROLLER

The robot is controlled by a whole-body controller running at 1 kHz: it is composed of a feed-forward and a proportional-derivative (PD) term. The MPC provides joint torque commands with a time step of 10 ms. Whole-body linear feedback is realized through Riccati gains [33]:

$$\bar{\boldsymbol{\tau}} = \boldsymbol{\tau}_0 + \mathbf{K}_0(\mathbf{x} \ominus \mathbf{x}_0), \tag{18}$$

where $\boldsymbol{\tau}_0$, and $\mathbf{x}_0$ are respectively the optimal initial torque and state computed by the MPC step, and $\mathbf{x}$ is the current estimated state at 1 kHz. The feedback gain $\mathbf{K}_0$ is computed in the backward pass of the Box-FDDP solver [27].

We sum the action of a PD controller to the previous term, to get the torques sent to the robot:

$$\boldsymbol{\tau} = \bar{\boldsymbol{\tau}} + \mathbf{K}_p({}^j\mathbf{q}^d - {}^j\mathbf{q}) + \mathbf{K}_d({}^j\mathbf{v}^d - {}^j\mathbf{v}), \tag{19}$$

with $\mathbf{K}_p, \mathbf{K}_d$ being the proportional and derivative gain diagonal matrices, and ${}^j\mathbf{q}, {}^j\mathbf{v}$ being the current joint positions and velocities. The targets ${}^j\mathbf{q}^d$ and ${}^j\mathbf{v}^d$, given by the MPC at 100 Hz, are updated in the low-level controller at 1 kHz: the interpolated values are obtained through cubic smoothing univariate splines.

IV. SIMULATION RESULTS

We have evaluated our novel problem formulation considering the fly-high cost through various simulations, answering the following questions:

TABLE 1. Cost weights for the nominal formulation with reference foot trajectories.

	w_q^{base}		w_v^{base}		w_x^{joints}		w_{bx}	w_{cone}		w_c	
	Pos	Rot	Lin	Ang	Pos	Vel	weight	min. force	weight	Pos	Vel
Continuous	0	500	10	10	0.01	1	10^3	0	10	10^6	10^4
Impact/Contact	1	1	10	10	10	10				10^7	10^2

TABLE 2. Cost weights for our formulation w/o reference foot trajectories.

	w_q^{base}		w_v^{base}		w_x^{joints}		w_{bx}	w_{cone}		w_c		Obstacle		
	Pos	Rot	Lin	Ang	Pos	Vel	weight	min. force	weight	Pos	Vel	Sqrt	Exp	w_t
Continuous	0	260	8	8	0.06	1.4	10^3	0	10	10^6	10^4	$2 \cdot 10^4$	10^3	$2 \cdot 10^5$
Impact/Contact	0	0	5	10	30	10				10^7	10^2			

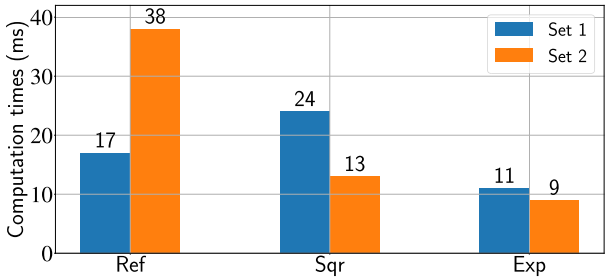


FIGURE 3. Number of iterations to solve the full-horizon OCP (ANYmal on flat ground). For the different problem formulations we have compared the cost weights of Set 1 (Table 1) and Set 2 (Table 2).

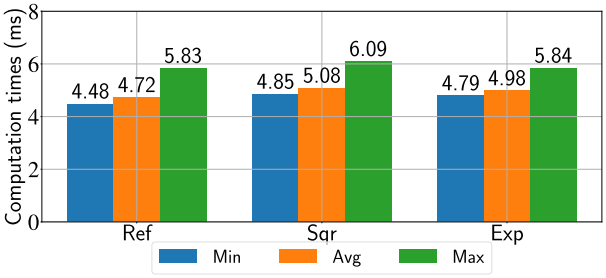


FIGURE 4. Minimum, average and maximum computation time, in ms, for each MPC cycle (given only one iteration of the OCP solver, ANYmal on flat ground).

1) How many iterations does the Box-FDDP solver needs to solve an OC problem? Are these numbers comparable to the nominal formulation (w/ reference trajectories)? (see Section IV-A)
2) Is our approach sufficiently fast for real-time applications? (see Section IV-B)
3) Is our approach more robust to external disturbances than other methods? (see Section IV-C)
4) What is the shape of the obtained feet trajectories when using the *Exponential* cost or the *Squared* cost? (see Section IV-D1)
5) How are foot velocity and obstacle distance related? (see Section IV-D2)

We have used Pinocchio for robot dynamics and derivatives [34], Crocoddyl [1] for solving OCPs and HPP-FCL [30] for distance computations. For the purpose of this section, we identify our modified formulations with the associated fly-high costs with the terms *Sqr* and *Exp*. For all simulations and experiments on flat ground, we consider free footstep locations.

A. NUMBER OF ITERATIONS & COST WEIGHTS

To compare our trajectory-free formulation to the state-of-the-art formulation that considers reference trajectories for the feet, we had to tune the weights of the different terms in the cost function. Table 1 shows the weights used in *Ref* [6]: the parameters are ordered as in Section III-B.

For the trajectory-free formulation, we have changed some weights, as reported in Table 2 to improve computational performance. To motivate this choice, we have solved a full-horizon OC problem, considering 5 steps of trotting of the quadruped robot ANYmal (18 Degrees of Freedom), moving on flat ground. Fig. 3 shows the number of Box-FDDP iterations required for convergence. With the first set of weights, the novel fly-high square cost requires more iterations (24) than the nominal formulation (17), while the exponential formulation needs fewer iterations (11). In contrast, with the second set of weights, specifically tuned for the fly-high costs, the iterations are less with both the fly-high costs.

B. COMPUTATION TIMES

For real-time applications, the time spent on each MPC cycle is fundamental. We have run the simulations on a computer with AMD Ryzen 9 5950X CPU @ 3.4 GHz. Fig. 4 reports the minimum, average and maximum periods for each cycle, considering only one Box-FDDP iteration. The MPC horizon is composed of 100 nodes, with a total duration of 1 s. Before the MPC pipeline, a first warm-start OC problem is solved: starting from an initial standing configuration $\mathbf{x}_0^{\text{stand}}$ of the quadruped, with all the feet in contact with the ground, we solve problem (4) for $j = 0$ until the solver reaches its convergence criteria. With the proposed costs, we have a larger deviation between the minimum and maximum values. The average computation time of both the *Sqr* and

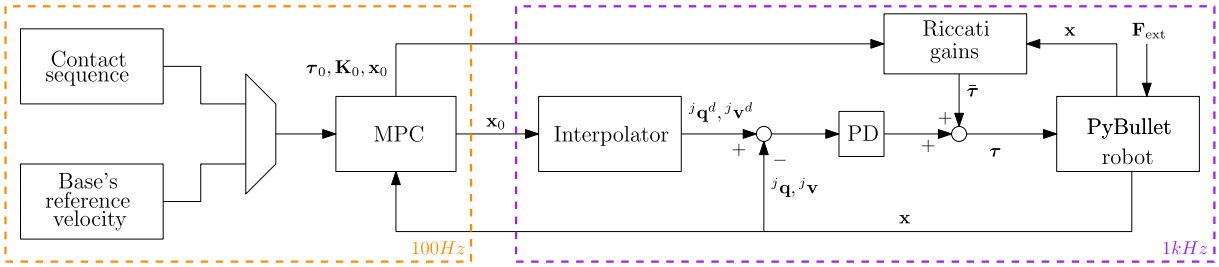


FIGURE 5. Simulation architecture including the Whole-Body MPC, low-level controller and PyBullet simulator.

Exp methods are considerably close to the *Ref* approach. In general, we can state that the periods are small enough to lead to MPC frequencies of about 100 Hz. This is a typical reference for full-body MPC on legged robots.

C. FULL-BODY DYNAMICS MPC

We have validated the novel cost models through a simulator developed in Python, whose architecture is reported in Fig. 5. On the left, the multiplexer box identifies the two possible guidelines for the MPC cycles, which can be a predefined contact sequence or a base reference velocity (as stated in Section III-B). The orange and purple blocks, containing respectively the MPC and the low-level controller, are executed sequentially: every 10 iterations of the low-level controller, we run an MPC cycle using the last state value. During the computation of each MPC step, which lasts 10 ms, we use the previous optimal state $\mathbf{x}_0$, control $\boldsymbol{\tau}_0$, and Riccati gains $\mathbf{K}_0$ in the low-level controller.

1) PYBULLET

PyBullet physics engine [35] simulates robot dynamics. With this library, we have built an environment composed of the floor and some stairs. External forces $\mathbf{F}_{\text{ext}}$ can be applied to the robot directly from the simulator: in our approach, we have modeled external disturbances like momentum, integrating a force with a fourth-order polynomial profile.

2) PUSH RECOVERY

We have studied the quadruped's ability to balance without falling down after external disturbances. In particular, we have analyzed the effects of a pushing force on the robot trunk, in the x-y plane and in different directions. The results are summarized by the polar plots in Fig. 6: the radius indicates the intensity of the pushing force, ranging from 50 to 470 N with a discretization step of 30 N, while the plane has been divided into 10 sectors, each representing a direction of the force. The 0° angle corresponds to the positive x-axis, while 90° identifies the positive y-axis. Each polygon represents the pair's force/direction for which the robot rejected the perturbation without colliding with the ground (except for the feet) for the given models. In our formulations *Sqr* and *Exp*, we consider free footstep locations, given a longitudinal reference velocity ${}^b v_x^* = 0.26$ m/s in the cost (16). The push disturbance is applied as an external force, with a duration of 0.2 s, in the PyBullet environment. This is done following the procedure described in subsection IV-C1. The external force is applied starting from different instants, namely the start ($t = 0.8$ s), mid ($t = 0.9$ s) and end ($t = 1$ s) of the foot's swing motion: the coverage areas are different in the distinct moments, as the fall avoidance capability depends on the joint configuration. Both our formulations showed higher reliability than the nominal case, as shown by the fact that the red patch is always contained inside the other polygons. This is mainly due to our formulation's ability to optimize footstep positions, given the absence of reference foot trajectories. We can also notice that the blue patches (*Sqr*) are larger than the green ones (*Exp*), except for a few peaks.

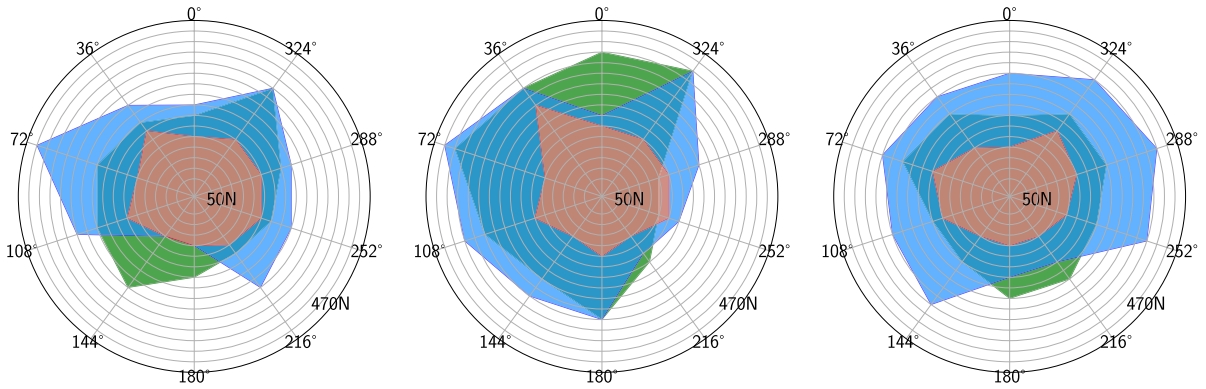
3) CLIMBING UPSTAIRS

Given the dependency on the distance between feet and obstacles, our novel formulation can also be used for other scenarios, such as climbing stairs, for which we can make the same analysis as in subsection IV-B (see Fig. 7). In this case we consider "fixed footsteps" ($w_{c,i} \neq 0$ for all the 3D components in (15)) because the robot cannot freely decide where to step. The periods related to the novel costs are a bit larger than the *Ref* case, but they are still less than 10 ms. To achieve these results, the line search of the forward pass inside the Box-FDDP has been limited to 2 iterations. However, performance still remains high thanks to the warm-start.

D. SWING TRAJECTORIES

1) SQR VS EXP

Using the two fly-high costs, we quantitatively compare the foot trajectories. Fig. 8 shows ANYmal's foot trajectories along the x and z directions for both *Sqr* and *Exp* costs, giving also the comparison with the *Ref* method. In these simulations, ANYmal climbed two stairs given fixed footstep positions. Only the front foot trajectories are shown since the hind feet remain on the ground. In both cases, the feet avoided collision with the stairs. The path generated with the fly-high costs have also greater altitude with respect to *Ref* trajectories, which might be beneficial in case of terrain uncertainty.



(a) $t = 0.8$ s (b) $t = 0.9$ s (c) $t = 1$ s

FIGURE 6. Push recovery capability of ANYmal moving on flat terrain, comparison between the *Ref* (red patch), *Sqr* (blue) and *Exp* (green) models in three different instants of the motion (the force F_{ext} is applied at time t).

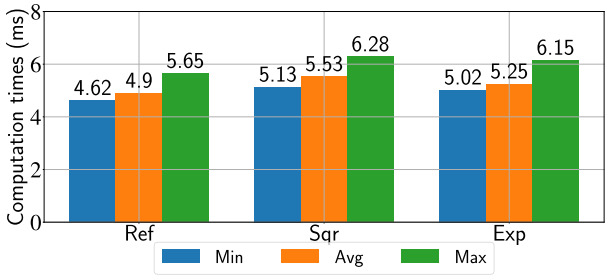


FIGURE 7. Minimum, average and maximum computation periods, in ms, for each MPC cycle (given only one iteration of the OCP solver, ANYmal climbing two steps).

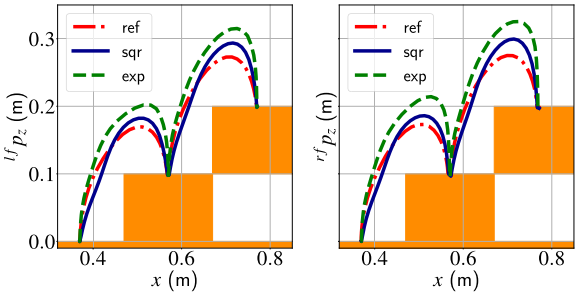


FIGURE 8. Feet trajectories, along the x and z directions, of ANYmal climbing two steps, obtained with the *Ref* method and the two fly-high costs.

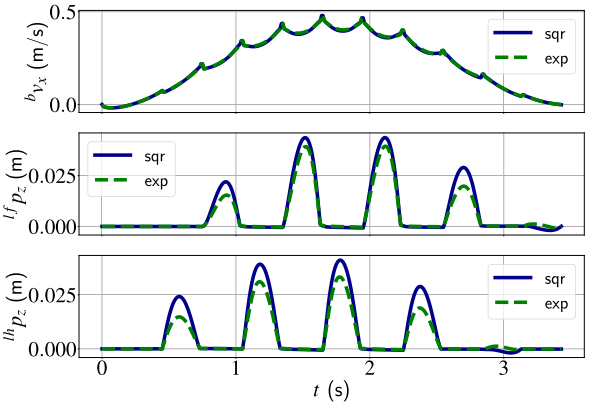


FIGURE 9. Comparison between the base's longitudinal velocity and the height of the left feet during locomotion.

2) HEIGHT-SPEED RELATIONSHIP

Fig. 9 shows one of the main benefits of the novel cost formulations: the quadruped follows a time-varying longitudinal base velocity reference, through cost (16). In the mid and bottom subplots are reported the height $^{f}p_z$ of the left front and hind feet respectively, which swing alternatively during the trotting motion. For the two costs, the base velocities $^{b}v_x$ (reported in the top subplot) are very similar and, as expected, larger velocities correspond to larger foot heights, which is typical behaviour of human walking as shown in [29].

V. EXPERIMENTAL RESULTS

In this section, we evaluate the novel cost functions' performances by generating trotting maneuvers on the ANYmal B robot [36]. To reduce the number of experiments, we choose only one of the two costs, namely the *Exp* model since it reports lower computation times for flat terrain locomotion (see Fig. 4).

We use the state estimator already equipped on the ANYmal B robot [37], which relies on an EKF to fuse legs odometry with inertial sensing from the IMU. LiDAR scans adjust the drift due to inaccuracies in kinematic and contact information, using the iterative closest point algorithm [38].

The MPC runs at 50 Hz on an offboard PC (Intel(R) Core(TM) i9-9900KF CPU @ 3.60 GHz). The offboard computer is connected to the onboard PC via Ethernet and communication is achieved through ROS TCP/IP.

Fig. 10 shows the x-z trajectories, of the left-front and right-hind feet, during the trotting motion reported in Fig. 1. The green lines represent the paths measured directly from the feet by the point cloud. The robot crosses the stair with both feet successfully.

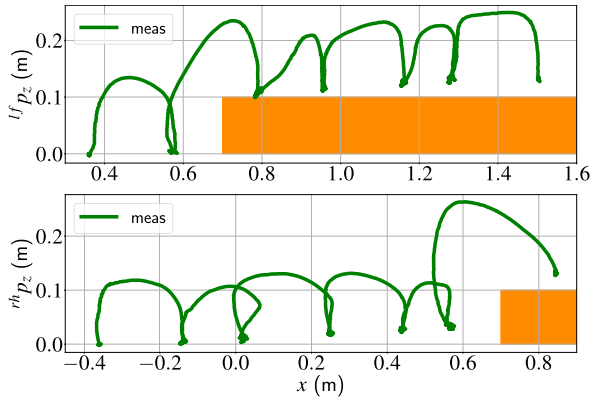


FIGURE 10. Feet trajectories, along the x and z directions, of ANYmal trotting on a big flat stair.

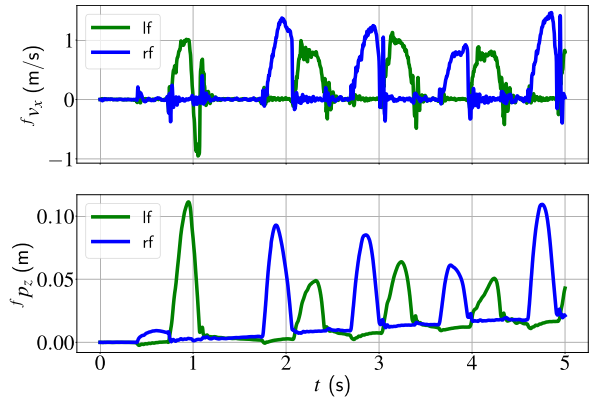


FIGURE 11. Longitudinal velocities versus height of the front feet during trotting, following a longitudinal reference velocity of 0.5 m/s with free footsteps positions.

TABLE 3. RMSEs (in mm) of the feet x and z positions.

Experiment	Measurand	lf	lh	rf	rh
free footsteps	$^{f}p_x$	5.14	3.54	4.97	5.99
	$^{f}p_z$	3.29	2.89	3.62	3.86
big stair	$^{f}p_x$	15.56	16.04	21.88	13.64
	$^{f}p_z$	11.35	8.68	16.53	10.49

We investigate the relationship, given by the novel cost function, between the foot's velocity and the distance from the terrain or the obstacle. Fig. 11 reports the longitudinal velocities against the heights of the front feet during locomotion with free footsteps locations: peaks at the altitudes correspond to large velocities as expected. This behaviour is clearly visible in the experiment with the joystick (see the link in Fig. 1).

We also evaluated the performances of the state-feedback and low-level controller [6]. Table 3 reports the Root Mean Square Errors (RMSEs) associated with the longitudinal and vertical positions of all the feet during the aforementioned experiments. The RMSE is computed on the difference between the measured and predicted foot trajectories. In all positions, RMSEs are always lower than 6 mm for the free footsteps scenario. However, they are a bit higher but still acceptable for the big stair. Thus the tracking performances are superior in the simple scenario as expected.

VI. CONCLUSION

In this work, we have proposed a full-body MPC approach with a novel cost formulation that does not require reference swing foot trajectories. This is made possible by introducing a cost function able to produce collision-free foot trajectories without reference paths to follow. Using the same solver and MPC pipeline, we have shown that our approach achieves better push-recovery capabilities than the state-of-the-art approach, while keeping almost identical computational times. To validate our approach, we have performed simulations with different agile trotting maneuvers, including flat ground and stairs. In addition, we have carried out hardware experiments to demonstrate the applicability of our approach to the MPC pipeline.

Future research directions include the incorporation of a contact planner for the stair scenarios, removing the constraint of pre-defined contact locations. Since most of the real environments are unstructured and cannot be modeled as combination of geometrical primitives, triangular meshes and point clouds can be adopted to model intricate shapes while preserving the continuity of the distance function. Another improvement can be the application of the MPC pipeline with fly-high costs for biped locomotion, which would require some adaptation of the novel costs to consider foot orientations.

APPENDIX A
DISTANCE DERIVATIVE

Let us consider each foot modeled as a sphere with zero radii, thus collapsed into a point ${}^{f}\mathbf{p}$. The environmental obstacles, like the ground and the stairs, are modeled with boxes. By referring to the obstacle point closest ${}^{o}\mathbf{p}$, the distance derivative can be written as:

$$\frac{\partial \phi^{f,o}}{\partial \mathbf{q}} = \mathbf{n}_d^{\top} \left(\frac{\partial {}^{f}\mathbf{p}}{\partial \mathbf{q}} - \frac{\partial {}^{o}\mathbf{p}}{\partial \mathbf{q}} \right), \tag{20}$$

with $\mathbf{n}_d$ being the unit vector aligned with ${}^{f}\mathbf{p} - {}^{0}\mathbf{p}$. The previous definition can be further simplified if we assume static obstacles, which is reasonable for ground and stairs. Fig. 12 shows the x-z cross-section of a stair, with the red and blue dots depicting possible positions of ${}^{f}\mathbf{p}$ and ${}^{o}\mathbf{p}$ for each highlighted area. During the foot motion, ${}^{o}\mathbf{p}$ can move on the box faces if ${}^{f}\mathbf{p} \in S_1$ or ${}^{f}\mathbf{p} \in S_2$, while it moves along the corner line for ${}^{f}\mathbf{p} \in S_3$. Given the following definitions:

$${}^{f}\mathbf{p} \in S_1 \implies \begin{aligned} \frac{\partial {}^{o}\mathbf{p}}{\partial \mathbf{q}} &= \begin{bmatrix} \frac{\partial {}^{o}p_x}{\partial \mathbf{q}}^{\top} & \frac{\partial {}^{o}p_y}{\partial \mathbf{q}}^{\top} & \mathbf{0}^{\top} \end{bmatrix}^{\top} \\ \mathbf{n}_d &= [0\ 0\ 1]^{\top} \end{aligned}$$

$${}^{f}\mathbf{p} \in S_2 \implies \begin{aligned} \frac{\partial {}^{o}\mathbf{p}}{\partial \mathbf{q}} &= \begin{bmatrix} \mathbf{0}^{\top} & \frac{\partial {}^{o}p_y}{\partial \mathbf{q}}^{\top} & \frac{\partial {}^{o}p_z}{\partial \mathbf{q}}^{\top} \end{bmatrix}^{\top} \\ \mathbf{n}_d &= [1\ 0\ 0]^{\top} \end{aligned}$$

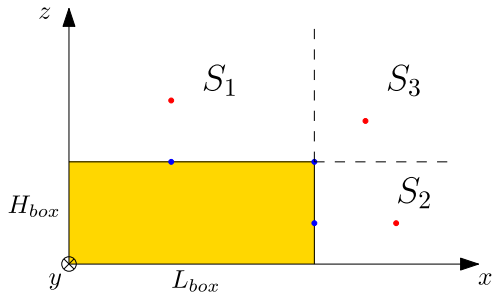


FIGURE 12. Cross-section of an axis-aligned stair.

$$ {}^f\mathbf{p} \in S_3 \implies \begin{matrix} \frac{\partial^o \mathbf{p}}{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{0}^\top & \frac{\partial^o p_y}{\partial \mathbf{q}}^\top & \mathbf{0}^\top \end{bmatrix}^\top \\ \mathbf{n}_d = [n_x \; 0 \; n_z]^\top \end{matrix} \tag{21} $$

we can infer that the following always holds:

$$ \mathbf{n}_d^\top \frac{\partial^o \mathbf{p}}{\partial \mathbf{q}} = 0, \quad \forall \mathbf{q}. \tag{22} $$

This statement holds also for the other faces of the box due to symmetry. Therefore, (20) can be simplified into:

$$ \frac{\partial \phi^{f,o}}{\partial \mathbf{q}} = \mathbf{n}_d^\top \frac{\partial^f \mathbf{p}}{\partial \mathbf{q}} = \mathbf{n}_d^\top {}^f\mathbf{J}_p, \tag{23} $$

where ${}^f\mathbf{J}_p$ is the Jacobian of the foot position with respect to $\mathbf{q}$ [39]. In conclusion, we observe that the distance function is continuous and differentiable everywhere, even during the transitions among sectors.

REFERENCES

[1] 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 *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2536–2542.

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

[3] T. A. Howell, B. E. Jackson, and Z. Manchester, "ALTRO: A fast solver for constrained trajectory optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 7674–7679.

[4] N. Rathod, A. Bratta, M. Focchi, M. Zanon, O. Villarreal, C. Semini, and A. Bemporad, "Model predictive control with environment adaptation for legged locomotion," 2021, *arXiv:2105.05998*.

[5] W. Jallet, A. Bambade, N. Mansard, and J. Carpentier, "Constrained differential dynamic programming: A primal-dual augmented Lagrangian approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 13371–13378.

[6] C. Mastalli, W. Merkt, G. Xin, J. Shim, M. Mistry, I. Havoutis, and S. Vijayakumar, "Agile maneuvers in legged robots: A predictive control approach," 2022, *arXiv:2203.07554*.

[7] C. Mastalli, S. P. Chhatoi, T. Corbéres, S. Tonneau, and S. Vijayakumar, "Inverse-dynamics MPC via nullspace resolution," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 3222–3241, Jan. 2023.

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. IEEE Int. Conf. Robot. Autom.*, Mar. 2003, pp. 1620–1626.

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

[10] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Auto. Robots*, vol. 35, nos. 2–3, pp. 161–176, Oct. 2013.

[11] J. Z. Kolter and A. Y. Ng, "Task-space trajectories via cubic spline optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 1675–1682.

[12] F. Risbourg, T. Corbères, P.-A. Léziart, T. Flayols, N. Mansard, and S. Tonneau, "Real-time footstep planning and control of the solo quadruped robot in 3D environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 12950–12956.

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

[14] W. Merkt, V. Ivan, and S. Vijayakumar, "Continuous-time collision avoidance for trajectory optimization in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 7248–7255.

[15] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1184–1189.

[16] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 193–203, Apr. 1988.

[17] J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for collision and proximity queries," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3859–3866.

[18] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1458–1465, Jul. 2018.

[19] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–9.

[20] P.-A. Léziart, T. Flayols, F. Grimminger, N. Mansard, and P. Souères, "Implementation of a reactive walking controller for the new open-hardware quadruped solo-12," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5007–5013.

[21] T. Corbères, C. Mastalli, W. Merkt, I. Havoutis, M. Fallon, N. Mansard, T. Flayols, S. Vijayakumar, and S. Tonneau, "Perceptive locomotion through whole-body MPC and optimal region selection," 2023, *arXiv:2305.08926*.

[22] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3402–3421, Oct. 2023.

[23] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential dynamic programming for multi-phase rigid contact dynamics," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2018, pp. 1–9.

[24] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Comput. Methods Appl. Mech. Eng.*, vol. 1, no. 1, pp. 1–16, Jun. 1972.

[25] R. Featherstone, *Rigid Body Dynamics Algorithms*. Cham, Switzerland: Springer, 2014.

[26] J. L. Blanco-Claraco, "A tutorial on SE(3) transformation parameterizations and on-manifold optimization," 2021, *arXiv:2103.15980*.

[27] C. Mastalli, W. Merkt, J. Marti-Saumell, H. Ferrolho, J. Solà, N. Mansard, and S. Vijayakumar, "A feasibility-driven approach to control-limited DDP," *Auto. Robots*, vol. 46, no. 8, pp. 985–1005, Dec. 2022.

[28] 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 *Proc. IEEE-RAS 21st Int. Conf. Humanoid Robots*, Nov. 2022, pp. 638–644.

[29] R. Desai and H. Geyer, "Robust swing leg placement under large disturbances," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2012, pp. 265–270.

[30] L. Montaut, Q. Le Lidec, V. Petrik, J. Sivic, and J. Carpentier, "Collision detection accelerated: An optimization perspective," in *Proc. Robot., Sci. Syst.*, Jun. 2022.

[31] L. Montaut, Q. L. Lidec, A. Bambade, V. Petrik, J. Sivic, and J. Carpentier, "Differentiable collision detection: A randomized smoothing approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 3240–3246.

[32] 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 *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 5107–5112.

[33] E. Dantec, M. Taïx, and N. Mansard, "First order approximation of model predictive control solutions for high frequency feedback," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4448–4455, Apr. 2022.
[34] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *Proc. IEEE Int. Symp. Syst. Integr. (SII)*, Aug. 2019, pp. 614–619.
[35] E. Coumans and Y. Bai. (2021). *Pybullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning*. [Online]. Available: http://pybullet.org
[36] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Bloesch, H. Kolvenbach, M. Bjelonic, L. Isler, and K. Meyer, "ANYmal—Toward legged robots for harsh environments," *Adv. Robot.*, vol. 31, no. 17, pp. 918–931, Sep. 2017.
[37] M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C. Remy, and R. Siegwart, "State estimation for legged robots—Consistent fusion of leg kinematics and IMU," in *Proc. Robot., Sci. Syst.*, Jul. 2012.
[38] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auto. Robots*, vol. 34, no. 3, pp. 133–148, Apr. 2013.
[39] J. Carpentier and N. Mansard, "Analytical derivatives of rigid body dynamics algorithms," in *Proc. Robot., Sci. Syst.*, Jun. 2018.

GIANNI LUNARDI received the B.Sc. degree in industrial engineering and the M.Sc. degree in mechatronics engineering from the University of Trento, Italy, in 2018 and 2021, respectively, where he is currently pursuing the Ph.D. degree under the supervision of Andrea Del Prete. His research interests include model predictive control, hybrid systems with impacts, and legged locomotion.

THOMAS CORBÈRES (Graduate Student Member, IEEE) received the M.Sc. degree in informatics and computing engineering from the University of Toulouse, Toulouse, France, in 2020. He is currently pursuing the Ph.D. degree in robotics and autonomous systems with The University of Edinburgh, Edinburgh, U.K., under the supervision of S. Tonneau. His research interests include legged locomotion, model predictive control, and contact planning.

CARLOS MASTALLI (Member, IEEE) received the M.Sc. degree in mechatronics engineering from Simón Bolívar University, Caracas, Venezuela, in 2013, and the Ph.D. degree in bio-engineering and robotics from Istituto Italiano di Tecnologia, Genoa, Italy, in 2017.

He is currently an Assistant Professor with Heriot-Watt University, Edinburgh, U.K. He is also the Head of the Robot Motor Intelligence (RoMI) Laboratory, National Robotarium and Edinburgh Centre for Robotics. He is also appointed as a Research Scientist with IHMC, USA. Previously, he conducted cutting-edge research in several world-leading laboratories: Istituto Italiano di Tecnologia, Italy, LAAS-CNRS, France, ETH Zürich, Switzerland, and The University of Edinburgh, U.K. His research interest includes building athletic intelligence for robots with legs and arms. His research work is at the intersection of model predictive control, numerical optimization, machine learning, and robot co-design.

NICOLAS MANSARD received the M.Sc. degree in computer science from University Grenoble Alpes, Grenoble, France, in 2003, and the Ph.D. degree in robotics from the University of Rennes, Rennes, France, in 2006. He was a Postdoctoral Researcher with Stanford University, Stanford, CA, USA, in 2007, and with the Joint Robotics Laboratory, Tsukuba, Japan, in 2008. He was an invited Researcher with the University of Washington, Seattle, WA, USA, in 2014. He is currently a CNRS Researcher ("chargé de recherche CNRS") with CNRS, Paris, France, since 2009. He published more than 70 papers in international journals and conferences and supervised ten Ph.D. thesis. His main research interests include the motion generation and planning and control of complex robots, with a special regard in humanoid robotics. His expertise covers sensor-based (vision and force) control, numerical mathematics for control, bipedal locomotion, and locomotion planning. He was the recipient of the CNRS Bronze Medal in 2015 (one medal is awarded in France in automatic/robotic/signal-processing every year). He was previously an Associate Editor of IEEE TRANSACTIONS ON ROBOTICS.

THOMAS FLAYOLS received the master's degree in embedded systems and industrial computing from ENS and the Ph.D. degree, in 2018. He is currently a Contract Researcher with LAAS-CNRS, Gepetto Group. He is also a Mechatronics Engineer, he values open hardware design in robotic perception and actuation. He also contributes to the open dynamic robot initiative by MPI, Tübingen, particularly in creating open hardware legged robots like Solo12. His research interest includes force control in legged robots, along with validation of estimation and control techniques.

STEVE TONNEAU received the Ph.D. degree in humanoid robotics from INRIA/IRISA, Rennes, France, in 2015. He is currently a Lecturer with The University of Edinburgh, Edinburgh, U.K. He was a Postdoctoral Researcher with LAAS-CNRS, Toulouse, France. His research interests include motion planning based on the biomechanical analysis of motion invariants, with applications, including computer graphics animation and robotics.

ANDREA DEL PRETE (Member, IEEE) has been an Associate Professor with the Industrial Engineering Department, University of Trento, Italy, since 2022. From 2019 to 2021, he was a tenure-track Assistant Professor (RTD-B) with the Industrial Engineering Department. In 2018, he was a Research Scientist with the Max-Planck Institute for Intelligent Systems, Tübingen, Germany. From 2014 to 2017, he was an Associate Researcher with LAAS/CNRS, Toulouse, France, where he has been working with the humanoid robot HRP-2. Before going to LAAS, he had spent four years (three of Ph.D. + one of postdoctoral) with the Italian Institute of Technology (IIT), Genoa, Italy, working on the iCub humanoid robot. His research interests include the intersection between control, optimization, and machine learning.