Multicontact Locomotion of Legged Robots

Justin Carpentier and Nicolas Mansard, *Member, IEEE*

***Abstract*—Locomotion of legged robots on arbitrary terrain using multiple contacts is yet an open problem. To tackle it, a common approach is to rely on reduced template models (e.g., the linear inverted pendulum). However, most of existing template models are based on some restrictive hypotheses that limit their range of applications. Moreover, reduced models are generally not able to cope with the constraints of the robot complete model, such as the kinematic limits. In this paper, we propose a complete solution relying on a generic template model, based on the centroidal dynamics, able to quickly compute multicontact locomotion trajectories for any legged robot on arbitrary terrains. The template model relies on exact dynamics and is thus not limited by arbitrary assumption. We also propose a generic procedure to handle feasibility constraints due to the robot's whole body as occupancy measures, and a systematic way to approximate them using offline learning in simulation. An efficient solver is finally obtained by introducing an original second-order approximation of the centroidal wrench cone. The effectiveness and the versatility of the approach are demonstrated in several multicontact scenarios with two humanoid robots both in reality and in simulation.**

***Index Terms*—Humanoid robots, legged robots, machine learning, multicontact locomotion, optimal control.**

I. Introduction

Legged robots are underactuated systems, which create and break contacts with their environment in order to move. The motion of a robot is the consequence of the interaction forces created at each contact point. These contact forces are constrained to remain inside the so-called friction cones to prevent the slippage and falls. Maintaining these forces deep inside the cones is one of the main tasks of the locomotion pattern generator (LPG).

In its generic form, an LPG deals with a high-dimensional and complex optimal control problem (OCP), seeking both the sequence of contacts and the whole-body trajectory while ensuring the feasibility of the contact constraints. This generic formulation of the locomotion problem is currently intractable by modern computers at sufficient control rate (e.g., 10 Hz or more). To tackle the computational complexity, many strategies have been proposed in the literature. Most of them are based on reduced models: instead of working with the full dynamics, only a subpart is considered, covering the essential properties of the whole-body dynamics.

A. Reduced Models

In the context of bipedal locomotion, the most famous reduced model is the linear inverted pendulum model (LIPM) [1]. The locomotion is then reduced to the problem of finding a trajectory for the reduced model which will in turn drive the whole-body system. Since the seminal work of Kajita *et al.* [2], various optimal control formulations have been proposed by the community, to either tackle the robustness problem [3], include viability conditions [4], allow altitude variations of the center of mass (CoM) [5], or also include foot placements as parameters of the problem [6].

However, LIPM-based methods are restricted to basic environments and cannot deal with more complex scenarios such as noncoplanar contact cases, climbing stairs using handrail, etc. Considering noncoplanar contacts invalidates the linearization leading to the LIPM model. The first approach to handle the nonlinear dynamics was proposed in [7]; however, it requires technical and dedicated developments based on limiting assumptions (e.g., prior knowledge of the force distribution). In another vein, it has been proposed to simplify the whole-body optimization problem by, e.g., assuming unconstrained torque capabilities [8]. Both approaches indeed boil down to optimizing the so-called centroidal dynamics [9] as a reduced model. Direct resolution of the underlying OCP is then possible [10], [11] (preliminary version of this paper), resulting in real-time performances. Other contributions have also been suggested that exhibit approximate dynamics (with possibly bounded approximations) leading to convex optimization problems, thus, ensuring global optimality at convergence [5], [12], [13]. In most cases, the footstep sequence is a given, although some solvers are also able to discover it while optimizing the centroidal dynamics [13], [14], at the cost of larger computation times.

B. Feasibility Constraints

The reduced model (either LIPM or centroidal) is subject to feasibility constraints implied by the whole body (e.g., kinematic or torque limits, footstep length, etc.). For instance, the CoM trajectory must be achievable (e.g., stay in the robot workspace) by the whole-body kinematics. Such constraints are difficult to express as solely function of the reduced model. These constraints can be tackled explicitly, by adding the corresponding whole-body variable in the optimization scheme [8],

Manuscript received April 13, 2018; accepted June 27, 2018. Date of current version December 4, 2018. This paper was recommended for publication by Associate Editor P.-C. Lin and Editor A. Billard upon evaluation of the reviewers' comments. This work was supported in part by the European Research Council through the Actanthrope Project (ERC grant agreement 340050) and in part by the European Commission through the MEMMO and RoboCom++ Projects. *(Corresponding author: Justin Carpentier.)*

The authors are with the Department of Robotics, LAAS-CNRS, Université de Toulouse, CNRS, Toulouse 31400, France (e-mail: jcarpent@laas.fr; nmansard@laas.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2018.2862902

[14]. However, this direct representation is also the most expensive in terms of computation.

Such constraints can also be represented at the level of the reduced model by using the so-called proxy constraints [15]. In most previous works, proxy constraints are defined using rough approximations [12], [13], [16] (box constraints, elliptic bounds, etc.) that lead to conservatism, or are simply ignored in many formulations [17], [18]. Footstep limits have been defined from data, for example, using hyperplanes based on a data set of robot success and failure inside a dynamic simulator [19]. Similar constraints can be obtained by training a neural network [20]. In [15], bounds of the capturability regions are obtained by extensive computations of the viability set of reduced models.

In [21], no proxy is set at the centroidal level. As the centroidal trajectory may not be feasible by the whole body, it is proposed to iteratively solve the centroidal then the whole-body motion problems. Such a loop empirically tends to quickly converge on a motion satisfying the constraints. However, no guarantee can be formulated as the loop may converge to invalid movement if no proxy is defined in the centroidal problem.

In general, two important constraints impact the centroidal dynamics: the motion of the CoM is limited by the kinematics bounds of the whole-body; the contact forces are constrained to lie inside the friction cones while motor torques must exist to actuate these forces. The common issue lies in the fact that it is hard (possibly hopeless) to find analytic formulas to exactly represent and express these constraints in the centroidal models. Approximations are needed, which can be incorporated in problems reasoning about the centroidal dynamics, while offering a good accuracy; such representations must also offer a simple and smooth geometry, so that being adequately handled by any numerical algorithm used to solve the motion problem.

C. Organization

In this paper, we introduce a complete formulation of LPG able to cope with multiple nonflat contacts, footstep timings, and whole-body "proxy" constraints, with a generic and versatile approach, tractable at robot control rate (from 20 to 100 Hz). Our solution is based on a generic optimal control formulation presented in Section II, which computes the centroidal dynamics trajectory according to a given sequence of contacts while enforcing two sets of constraints. On the one hand, the feasibility with respect to the whole-body constraints is tackled using a systematic approach introducing occupancy measure inside the optimal control formulation. We then propose a complete solution to learn the occupancy measure offline, by sampling the robot motion capabilities in simulation (see Section III). On the other hand, the feasibility of the contact model (friction cone constraints) is handled either by directly working with the contact forces or with the centroidal wrench. For that aim, we leverage on the double cone description [22], [23] and provide an efficient and original quadratic approximation of the centroidal wrench cone (CWC) (see Section IV). Both contact and proxy constraints are solved in near real time inside the proposed optimal control formulation presented in Section V. A complete experimental analysis is finally proposed, exhibiting the versatility and the efficiency of the approach, based on various locomotion scenarios with the new robot TALOS [24] in simulation and in reality with the humanoid robot HRP2.

D. Contribution

The main contribution of this paper is to propose the first complete formulation of a LPG able to generate realistic trajectories for multicontact locomotion in near real time. It relies on four technical contributions:

1) the exact and generic formulation of the OCP;
2) an efficient approach to handle proxyconstraints as an occupancy measure;
3) an original and efficient quadratic approximation of the CWC;
4) the proposition to rely on multiple shooting for computing the OCP solution.

The paper is based on two previous papers [11] (which introduced the centroidal OCP) and [25] (which focused on the proxy). In this paper, we additionally propose an efficient approximation of the gravito-inertial wrench cone, unify the formulation so that both forces or acceleration trajectories can be handled and experimentally compare these formulations. All the movements on the robot have been generated with the latest formulation (i.e., they are new compared to [11], even if based on similar scenarios).

II. Generic Optimal Control Formulation

In this section, we briefly recall the fundamental equations which drive the dynamics of a polyarticulated system in contact. We then introduce a generic OCP formulation for multicontact locomotion of legged systems. For that purpose, we first recall how the whole-body dynamics can be reduced to the centroidal dynamics under a simple assumption. We also demonstrate how the centroidal dynamics can be driven with two different controls leading to two OCP formulations with complementary properties. We conclude this section by highlighting how most LPGs in the literature are subcases of this generic OCP. Although this section contains known materials, we believe that both the exact formulation of the dynamics decoupling and the development of a generic formulation for the multicontact problem are a contribution. They are indeed prerequisites to the introduction of proxy constraints and centroidal cone approximations in the next sections.

A. Contact Model

The interaction between a robot and its environment is defined through a set of contact points $\{\boldsymbol{p_k} \in \mathbb{R}^3, k = 1, \dots, K\}$. For instance, for a humanoid robot equipped with rectangular feet, the contact points correspond to the four vertices of the rectangular shape. At each contact point, $\boldsymbol{p}_k$ is defined a contact force $\boldsymbol{f}_k$. In the case of unilateral contacts, $\boldsymbol{f}_k$ must lie inside a 3-D friction cone $\mathcal{K}_k^3$ (also denoted quadratic Lorentz "ice-cream" cone) characterized by a positive friction coefficient μ_k.

In this paper, we only consider rigid contact interaction which is a reasonable assumption for most modern multiped robots that are mostly equipped with rigid soles.

A *contact phase* is defined by a constant set of contact points. In the context of bipedal walking, two examples of contact phases are the single and double support phases. As soon as a creation or a rupture of contact point occurs, the contact set is modified, defining a new contact phase. The concatenation of contact phases describes what we name a *contact sequence*, inside which all the contact phases have their own duration.

The computation of such contact sequences in arbitrary environment is computationally challenging. Since Bretl [26] efficient algorithms have been proposed by the motion planning community either to plan only for robot with feet [27], [28] or more generically for any kind of multiped robots [23], [29]. In our current approach, we use the open source and efficient implementation of [29] proposed in [30] to compute in real time a feasible contact sequence inside complex environments.

B. Whole-Body Dynamics and Centroidal Dynamics

A legged robot is a free-floating-base system composed of $6 + n$ degrees of freedom (DoF). Its dynamics is governed by $6 + n$ equations of motion, which link the joint configuration $\boldsymbol{q}$ and its time derivatives $\dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ to the torque actuation $\boldsymbol{\tau}_a$ and the contact forces $\boldsymbol{f}_k$

$$\begin{bmatrix} H_u \\ H_a \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_a \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_u \\ \boldsymbol{g}_a \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_6 \\ \boldsymbol{\tau}_a \end{bmatrix} + \sum_{k=1}^{K} \begin{bmatrix} J_{k,u}^{\top} \\ J_{k,a}^{\top} \end{bmatrix} \boldsymbol{f}_k \quad (1)$$

where subscripts u and a stands for the underactuated and the actuated parts, respectively, H is the generalized mass matrix, $\boldsymbol{b}$ covers the centrifugal and Coriolis effects, $\boldsymbol{g}$ is generalized gravity vector and J_k is the Jacobian of the kth contact.

The 6 first rows of (1) corresponds to the underactuated dynamics of the robot, also called the centroidal dynamics [9]. This centroidal dynamics coincides with the Newton–Euler equations of motion, which links the variation of the linear momentum and the angular momentum of the whole system expressed around its CoM to the contact forces. Denoting by $\boldsymbol{h} \overset{\text{def}}{=} m\dot{\boldsymbol{c}}$, the linear momentum ($m$ being the total mass of the robot and $\boldsymbol{c}$ the CoM position) and $\boldsymbol{L_c}$ the angular momentum, the 6 first rows of (1) can be written as

$$\dot{\boldsymbol{h}} = \sum_{k=1}^{K} \boldsymbol{f}_k + m\boldsymbol{g} \qquad \dot{\boldsymbol{L}}_{\boldsymbol{c}} = \sum_{k=1}^{K} (\boldsymbol{p}_k - \boldsymbol{c}) \times \boldsymbol{f}_k \quad (2)$$

where $\boldsymbol{g} \overset{\text{def}}{=} (0, 0, -9.81)$ is the gravity vector and $\times$ denotes the cross product operator.

The n last rows of (1) are the Lagrange dynamics of a robot manipulator making contacts with its environment. At this stage, two important hypotheses are made. First, the manipulator dynamics is fully actuated through $\boldsymbol{\tau}_a$. In the context of robots with passive joints, this formulation no longer holds: a precise account of flexibilities and additional underactuation must be taken into account in order to correctly control them. Second, we suppose that the underactuated robot must be equipped with sufficient actuated joints in order to allow the underactuation to be fully controllable through the joint coordination (gesticulation) [31]. Both hypotheses are very classical when considering humanoid and quadruped robots, but might be limiting if trying to generalize the work to passive walkers.

C. Hierarchical Decoupling Between Centroidal and Manipulator Dynamics

Equation (1) can be interpreted as: when supplying a certain amount of joint torque $\boldsymbol{\tau}_a$, the environment reacts by producing the contact forces $\boldsymbol{f}_k$. These forces act on the centroidal dynamics to enable the robot to move along the environment.

Under the assumption that the system can produce sufficient torque (which current high-performance legged robots usually have), the centroidal and manipulator dynamics can be decoupled one from the other. The locomotion problem can then be split into two consecutive stages. In a first stage, it is sufficient to find the force trajectories which drive the centroidal dynamics. In a second stage, the required joint torque trajectory can be retrieved through the manipulator trajectory, knowing the centroidal trajectory and under the hypothesis of nonsliding contacts. In other words, the torque may be seen as a slack variable.[1] This decoupling has been highly exploited in humanoid robotics, for instance, in the work of Dai *et al.* [8] and later in [13], [32], and [21].

To ensure the effective decoupling, two additional restrictions must be respected by the first stage:

1) in case of unilateral contacts, the corresponding forces must belong to the friction cone;
2) the centroidal dynamics may be feasible by the system in terms of both kinematics and dynamics;

The first constraint stems directly from the contact model introduced in Section II-A. The second constraint comes from the fact that the centroidal dynamics is linked to the joint configuration and its derivatives through the centroidal mapping

$$\begin{bmatrix} \boldsymbol{h} \\ \boldsymbol{L_c} \end{bmatrix} = A_g\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}}, \quad (3)$$

with A_g the so-called centroidal momentum matrix [9].

D. State and Control of the Centroidal Dynamics

Substituting $\boldsymbol{h}$ by its value $m\dot{\boldsymbol{c}}$, (2) can be rewritten as

$$m(\ddot{\boldsymbol{c}} - \boldsymbol{g}) = \sum_{k=1}^{K} \boldsymbol{f}_k \quad (4a)$$

$$\dot{\boldsymbol{L}}_{\boldsymbol{c}} + m\,\boldsymbol{c} \times (\ddot{\boldsymbol{c}} - \boldsymbol{g}) = \sum_{k=1}^{K} \boldsymbol{p}_k \times \boldsymbol{f}_k. \quad (4b)$$

Equation (4) defines an affine dynamical system with the state vector $\boldsymbol{x} \overset{\text{def}}{=} (\boldsymbol{c}, \dot{\boldsymbol{c}}, \boldsymbol{L_c})$ and the control vector $\boldsymbol{u}_f \overset{\text{def}}{=} (\boldsymbol{f}_k, k = 1, \ldots, K)$ with $\boldsymbol{f}_k \in \mathcal{K}_k^3$. One drawback of this formulation is that the control input grows linearly with the number of contacts. To overcome that, one can write (4) by condensing all the forces and torques with a single control input

[1] Torque bounds can later be treated as a proxy constraint following the approach that we introduced in Section III.

$\boldsymbol{u}_c \overset{\text{def}}{=} (\boldsymbol{f}_c, \boldsymbol{\tau}_c)$ such that the centroidal dynamics reads

$$m(\ddot{\boldsymbol{c}} - \boldsymbol{g}) = \boldsymbol{f}_c \tag{5a}$$

$$\dot{\boldsymbol{L}}_{\boldsymbol{c}} = \boldsymbol{\tau}_c - \boldsymbol{c} \times \boldsymbol{f}_c \tag{5b}$$

with $\boldsymbol{f}_c \overset{\text{def}}{=} \sum_{k=1}^{K} \boldsymbol{f}_k$ and $\boldsymbol{\tau}_c \overset{\text{def}}{=} \sum_{k=1}^{K} \boldsymbol{p}_k \times \boldsymbol{f}_k$, $\boldsymbol{u}_c$ being the gravitoinertial wrench exerted by the environment on the robot and expressed in the world frame. The constraints on the individual contact cone is then reduced to the 6-D constraint

$$(\boldsymbol{f}_c, \boldsymbol{\tau}_c) \in \mathcal{K}_c^6 \tag{6}$$

where

$$\mathcal{K}_c^6 \overset{\text{def}}{=} \oplus_{k=1}^{K} \mathcal{K}_k^3 = \left\{ \sum_{k=1}^{K} (\boldsymbol{f}_k, \boldsymbol{p}_k \times \boldsymbol{f}_k), \boldsymbol{f}_k \in \mathcal{K}_k^3 \right\} \tag{7}$$

being the Minkowski sum of the contact cones translated by the contact point positions. This cone was first introduced in [7]. Following the wordings "centroidal dynamics" introduced in [9], we name it CWC.[2]

At this stage, the following observations come:

1) $\mathcal{K}_k^3$ contact cones have analytic description as Lorentz ("ice-cream") cone [35] while there is no explicit formula for the Minkowski sum $\mathcal{K}_c^6$ of Lorentz cones.
2) $\mathcal{K}_c^6$ explicitly depends on the contact point positions $\boldsymbol{p}_k$, but it is independent from the CoM position $\boldsymbol{c}$.
3) Equation (4) is a dynamical system whose control input grows linearly with the number of contacts while (5) has a fixed size control vector.
4) There is a forward map to pass from (4) to (5). The reverse is not true: in case of multiple contacts, we cannot uniquely retrieve the contact forces resulting in a given contact wrench vector. But it is even harder as we cannot ensure the existence of a contact wrench which lies in the Minkowski sum and which gives rise to a given centroidal dynamics value.

E. Generic Optimal Control Formulation

From Section II-D, it appears that the centroidal dynamics can be driven either by the force input $\boldsymbol{u}_f$ or by the centroidal input $\boldsymbol{u}_c$. In both cases, the dynamics can be written as an affine dynamical system equation

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = F_x \boldsymbol{x} + F_u(\boldsymbol{x})\boldsymbol{u} \tag{8}$$

where F_x and $F_u(\boldsymbol{x})$ are two matrices directly deduced from (4) or (5) and $\boldsymbol{u}$ indifferently represents $\boldsymbol{u}_f$ or $\boldsymbol{u}_c$ and must belong to the corresponding set denoted by $\mathcal{K}$. Equation (8) is a bilinear dynamics through the term $F_u(\boldsymbol{x})\boldsymbol{u}$ (the CoM position $\boldsymbol{x}$ multiplies for forces $\boldsymbol{u}$, as more clearly expressed in (5b) with $\boldsymbol{c} \times \boldsymbol{f}_c$).

We are now able to describe the generic problem of locomotion as follows.

> *From a given contact sequence and an initial centroidal state, find a feasible centroidal trajectory, satisfying the Newton–Euler equations, the contact constraints, and leading to a viable state.*

This problem can be directly translated as an OCP with path and terminal constraints:

$$\min_{\underline{\boldsymbol{x}}, \underline{\boldsymbol{u}}, (\Delta t_s)} \sum_{s=1}^{S} \int_{t_s}^{t_s + \Delta t_s} \ell_s(\boldsymbol{x}, \boldsymbol{u})\, dt \tag{9a}$$

$$\text{s.t.} \quad \forall t \quad \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{9b}$$

$$\forall t \quad \boldsymbol{u} \in \mathcal{K} \tag{9c}$$

$$\forall t \quad \exists\, (\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \text{ s.t.}, \boldsymbol{x}, \dot{\boldsymbol{x}} \text{ is feasible} \tag{9d}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{9e}$$

$$\boldsymbol{x}(T) \in \mathcal{X}_* \tag{9f}$$

where s is the index of the contact phase, $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{u}}$ are the state and control trajectories,[3] and t_s is the start time of the contact phase s with $t_1 = 0$ and $t_{s+1} = t_s + \Delta t_s$. Constraints (9b) and (9c) enforce the consistent dynamics with respect to the contact model. As (9b) is bilinear in the decision variables, this constraint makes the problem nonconvex. While other works have focused on the problems induced by this property [13], we did not observe in practice any trouble to handle the nonconvexity in our solver. Equation (9d) is the constraint enforcing the feasibility of the centroidal dynamics with respect to the whole-body problem: it handles kinematics limits, bounds on the angular momentum quantity, etc. We show in Section III how it can be handled with proxy constraints in an automatic way. Constraint (9e) constrains the trajectory to start from a given state (typically estimated by the sensor of the real robot) while (9f) enforces a viable terminal state [36]. Finally, ℓ_s is the cost function typically decoupled in $\ell_x(\boldsymbol{x}) + \ell_u(\boldsymbol{u})$ whose parameters may vary according to the phase. ℓ_x is generally used to smooth the state trajectory while ℓ_u tends regularize the control.

F. Previous Formulations

In the following, we present a state of the art in terms of the main LPG which are present in the literature. In particular, we detail how those LPGs correspond to a specific choice of the exact formulation (9).

1) Walking Patterns in 2-D: One major difficulty of (9) comes from the bilinear form of the dynamics (8). When the contacts are all taken on a same plane, a clever reformulation of the dynamics makes it linear [2], by neglecting the dynamics of both the CoM altitude and the angular momentum. In that case, $\mathcal{K}$ boils down to the constraint of the zero momentum point to lie in the support polygon.

Kajita *et al.* [2] did not explicitly check the constraint (9c); in exchange, ℓ_u is used to keep the control trajectory close to a reference trajectory provided *a priori*. Similarly, (9f) is not checked; in exchange, ℓ_x tends to stop the robot at a static equilibrium at the end of the trajectory by minimizing the jerk of the CoM. These three simplifications turn (9) into an unconstrained

[2] The same cone is also named admissibility polytope [33], contact wrench cone [8], gravitoinertial wrench cone [34]. We believe that the wording CWC is the most suitable for standardization; hence, we have chosen to use it in the paper.

[3] In this paper, trajectories are denoted as underline variables.

problem of linear-quadratic regulation (LQR) that is implicitly solved by integrating the corresponding Riccati equation.

Kajita's LQR was reformulated into an explicit OCP [37], directly solved as a quadratic program. The OCP formulation makes it possible to explicitly handle the inequality constraints: (9c) is then explicitly checked under its ZMP reformulation. A modification of this OCP is proposed in [4] where (9f) is approximated by the capturability constraint, which constrains the CoM position and velocity in the context of coplanar contacts.

2) Walking Patterns in 3-D: An iterative scheme is proposed in [38] that can be written as an implicit optimization scheme whose cost function is the distance to a given CoM trajectory and a given force distribution. The resulting forces satisfy (9c) by construction of the solution. There is no condition on the angular momentum (9d) neither on the viability of the final state (9f); however, the reference trajectory enforced by the cost function is likely to play the same role.

In [18], $\dot{\boldsymbol{L}}_{\boldsymbol{c}}$ is null by construction of the solution. Moreover, (9c) is supposed to always hold by hypothesis and is not checked, while (9f) is not considered but tends to be enforced by minimizing the norm of the jerk of the CoM, as in [2]. These assumptions result in a (bilinear)-constrained quadratic program that is solved with a dedicated numerical method.

In [33], (9c) is explicitly handled (using the classic linear approximation of the quadratic cones). As in [2], (9f) is indirectly handled by minimizing the jerk. No condition in (9d) on the angular momentum is considered. Additionally, the proposed cost function maximizes the robustness of the computed forces and minimizes the execution time. Finally, constraints are added to represent the limitation of the robot kinematics.

In [32], (9c) is handled under a simple closed form solution, while (9f) is not considered. To stabilize the resolution, the cost function tends to stay close to an initial trajectory of both the CoM and the angular momentum, computed beforehand from a kinematic path. Consequently, (9d) is not considered either (as it will simply stay close to the initial guess). The method is later extended by reoptimizing the centroidal dynamics once the whole-body trajectory has been computed [21]. This tends to drive the centroidal dynamics away from the whole-body constraints. It also generates a movement where the angular momentum (due to the limbs quickly moving from contact to contact) is better coordinated with other centroidal effects. The same team later focused on also deciding the contact placement in the centroidal optimizer and the optimization of the timings of the contact sequence [39]. A specific approximation of the dynamics was also introduced to formulate the centroidal problem as a convex optimization problem [13].

In [40], the centroidal problem is written as a collocation problem with both phase timings and contact placements on a height-map. Kinematic feasibility is checked using ad hoc boxes representing the effector reachability limits, while the angular momentum is set to zero. The authors empirically show that their (off-the-shelf) solver can often cope with discontinuities of the height-map, hence, also solving the contact sequence problem.

In [10], the conic constraint is directly handled. The angular momentum is treated through the orientation of the system ($\boldsymbol{L_c} \approx \tilde{I}\boldsymbol{\omega} + \boldsymbol{\tau}_{\boldsymbol{L_c}}$, with $\tilde{I}$ the compound (rigid) inertia of the robot and $\boldsymbol{\tau}_{\boldsymbol{L_c}}$ the angular momentum due to the internal gesticulation). $\tilde{I}\boldsymbol{\omega}$ is kept low by penalizing the large rotation $\boldsymbol{\omega}$ but $\boldsymbol{\tau}_{\boldsymbol{L_c}}$ is unlimited, resulting in (9d) not being checked. The viability (9f) is not checked neither, but as seen previously, it is approximately handled by minimizing the derivatives of the state in the cost function (however, the first derivatives instead of the third), while a reference trajectory of the CoM is provided to keep the nice behavior of the numerical scheme. Additionally, hard constraints on the CoM position are added to represent the kinematic limits of the whole body.

In [41], Caron and Kheddar work only with the CoM acceleration and neglect the contribution of the angular momentum quantity setting it to 0 as in [2]. They approximate the Minkowski sum of contact cones $\mathcal{K}_c^6$ with a conservative linear approximation following the method proposed in [33]. The proposed cost function regularizes the control vector and tries to minimize the distance between the final and the desired states. No proxy constraint is provided to ensure the feasibility of the CoM trajectory w.r.t. the whole body.

In [12], Dai and Tedrake do not directly consider the angular momentum quantity but instead, they chose to minimize an upper bound of its L_1 norm. Similar to [41], they consider a linear approximation of $\mathcal{K}_c^6$ and try to maximize the margin on the CWC. In addition to those previous criteria, the cost function is augmented with a regularization term on the CoM acceleration.

In [42], Serra *et al.* propose an efficient OCP formulation to compute a CoM trajectory for horizontal contacts. The cost function is composed of regularization terms on the contacts forces and moments as well as on the CoM jerk. In addition, they try to follow at best a reference trajectory both for the ZMP and the CoM. In their work, the authors do not consider friction cones, they just restrict the ZMP to lie in the convex hull of the contact points.

G. *From Generic Formulation to Its Implementation*

OCP (9) corresponds to a generic formulation of the problem, also exploited by several other works within the last five years [8], [10], [13], [32]. It contains several terms that are difficult, complex, or even impossible to make explicit: whole body constraints, angular momentum set, viability set. The stake is now twofold: we need to decide how to represent these functions, and how to solve the OCP.

1) Representing the constraint functions implies a tradeoff between accuracy of the model and efficiency of the resolution. In the following sections, we propose an original contribution to formulate approximate proxy constraints representing the whole-body limits with a generic offline learning approach (see Section III). We also propose an efficient approximation of the contact constraints then allowing the formulation of the OCP with the reduced variable $\boldsymbol{u}_c$ (see Section IV). Both constraints could be used in any OCP, for example, directly applying to [10], [13], [17], [21], [43].

2) Solving the OCP first requires its transcription into a numerical problem (using a finite-dimension representation of both control and state trajectories), then to solve it with an adequate nonlinear optimizer. When formulating the transcription, naive approaches (e.g., for representing the dynamics, the trajectories or the constraints) very often give very bad numerical behaviors (e.g., numerical instability, long active-set iterations, etc.). We propose a complete transcription of the OCP into an efficient numerical problem, based on multiple shooting, smooth trajectory parametrization, smooth constraint representation, and continuous proxy temporal integration (see Section V). The proposed transcription is then solved using the off-the-shelf solver MUSCOD-II.

Thanks to the original constraints (proxy and force) and to the clean transcription, we are able to demonstrate an efficient and reliable implementation of our approach, leading to real-time performances (i.e., tractable at robot control rate—from 20 to 100 Hz) and able to produce efficient movement on the real robot (see Section VI).

III. Learning Feasibility Constraints for the Centroidal Problem

In this section, we first present a mathematical encoding of the feasibility constraints as probability measures. We discuss the interest of this representation with respect to more-classical set-membership and show how it can be used to efficiently implement (9d) in the OCP. We then present a complete solution to efficiently approximate the CoM feasibility, i.e., we want to guarantee that the CoM produced by the reduced formulation is achievable by the whole-body kinematics. Handling this sole constraint first is a proper way of validating our proxy formulation. It is also interesting in practice, as the feasibility of the CoM is one of the most limiting constraint. Generalizing this proxy of the CoM position to its velocity and its acceleration with respect to joint velocity and acceleration limits would be straightforward. The extension to the construction of the proxy on the torque limits is left as a perspective. Finally, we conclude this section by validating our learning process on the HRP-2 robot.

A. Handling Feasibility Constraints

1) Mathematical Representation of Feasibility Constraints: Our objective is to efficiently implement the feasibility constraint (9d) in our OCP. This constraint explicitly depends on the robot configuration, which is not a variable of the centroidal OCP. A straightforward implementation is to add the robot configuration in the variables of the OCP [8]. However, this would surely lead the OCP to optimize the whole-body trajectory in order to handle all the robot constraints, which is yet not tractable especially if targeting real-time performances. We rather believe that it is possible to represent this constraint with an equivalent "proxy" constraint not dependent on the robot configuration.

Various ways to encode proxy constraints have been proposed in the literature. Most of them rely on set-membership. Denoting by γ the centroidal projection function

$$\gamma : (\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \rightarrow (\boldsymbol{x}, \dot{\boldsymbol{x}}) = \gamma(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$$

the proxy can be written as the constraint to have the state variables in the range space of γ. Set-membership proxies are used for instance in [6] and [28] to encode the maximal step size in biped walking, or in [12] to bound the CoM position by simple geometric shape. In all these cases, the set boundaries are represented by very simple mathematical structures (typically linear inequalities) in order not to burden the OCP solver. Remarkably, there are few papers about the automatic synthesis of the set boundaries [15], [19], [20].

Despite its popularity, the set-membership representation has important drawbacks. First, it is often difficult to handle with the OCP solver, in particular when the feasible set is not convex. The boundary, which is a singular mathematical object, is also complex to describe or numerically approximate. Finally, the OCP solver often tends to saturate the set boundary, where the inverse kinematics γ^{-1} is likely to fail. Consequently, the set is often arbitrarily reduced to improve the robustness of the whole-body solution.

2) Proxy as Occupancy Measure: In this paper, we rather state that the proxy is best represented by the notion of occupancy measure over $\boldsymbol{x}, \dot{\boldsymbol{x}}$ [44], [45]. In its generic form, given a set $A \subset \mathbb{R}^n$, a time interval $I \subset \mathbb{R}$ and a trajectory $\underline{s} : I \rightarrow \mathbb{R}^n$, the occupancy measure μ of the trajectory $\underline{s}$ on A is defined as

$$\mu(A) \overset{\text{def}}{=} \int_I \mathbb{1}_A(s(t))dt \tag{10}$$

with $\mathbb{1}_A(.)$ the indicator function of the set A. It gives the duration spent in the set A on the interval I by the trajectory $\underline{s}$.

Now, consider a state trajectory $\underline{\boldsymbol{x}}$. With (9d), we want to maximize the likelihood that the inverse-kinematics solver converges on a trajectory $\underline{\boldsymbol{q}}$ such that $\underline{\boldsymbol{x}}$ is the image of $\underline{\boldsymbol{q}}$ by γ. For that purpose, it is desirable that to any state $\boldsymbol{x}$ corresponds as many robot configurations as possible, so that the inverse kinematics is likely to converge to a solution $\boldsymbol{q}$ meeting continuity constraints.

We define the centroidal occupancy measure as the image of the uniform distribution in configuration space through the centroidal projection γ

$$\mu_o(\tilde{\boldsymbol{x}}) \overset{\text{def}}{=} \int_{\tilde{\boldsymbol{q}}\,\text{s.t.}\,\gamma(\tilde{\boldsymbol{q}})=\tilde{\boldsymbol{x}}} d\tilde{\boldsymbol{q}} = \int_{\mathcal{Q}} \mathbb{1}_{\gamma(\tilde{\boldsymbol{q}})=\tilde{\boldsymbol{x}}} d\mu_{\mathcal{Q}}$$

where $\tilde{\boldsymbol{x}} \overset{\text{def}}{=} (\boldsymbol{x}, \dot{\boldsymbol{x}})$, $\tilde{\boldsymbol{q}} \overset{\text{def}}{=} (\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, $\mathcal{Q}$ is the whole-body motion range, and $\mu_{\mathcal{Q}}$ is the uniform distribution on $\mathcal{Q}$.

Measured μ_o has several properties of the set-membership representation. First, the support of μ_o is equal to the feasibility set, which means that μ_o contains at least as much information as the set boundaries. It indeed contains more information, as for example, the level sets of μ_o can be used as boundaries of the interior of the feasibility set, used to improve the robustness.

In practice, it is desirable that OCP (9) promotes centroidal states $\tilde{\boldsymbol{x}}$ where μ_o is the highest. First, it makes it easier to then compute a corresponding configuration $\tilde{\boldsymbol{q}}$. Second, the configuration is well inside the kinematic feasibility set, where redundancy will help the robot to handle disturbances.

Finally, the measure also eases the life of the OCP solver, compared to handling directly the feasibility set membership, as explained next.

3) Maximizing the Occupancy Measure: Before deriving an effective solution to represent μ_o for the specific case of the kinematic feasibility, we quickly show how μ_o can be integrated in the OCP (9).

In practice, the measure can be normalized and represented by the corresponding probability density function (PDF), denoted by $p(\boldsymbol{x}, \dot{\boldsymbol{x}})$. It is then possible to directly exploit the measure to represent the set-membership constraint (by imposing the integral of the measure to be positive on any small neighborhood around the trajectory). In addition, we could use the PDF to directly optimize the robustness, either by optimizing over a level set of the PDF, or by maximizing the neighborhood around the trajectory where the measure is nonzero.

However, adding a PDF as a constraint of an OCP is not straightforward. Therefore, we propose to remove the hard constraint (9d) and penalize the OCP cost with the log-PDF. The new cost formulation $\tilde{\ell}_s$ is the composition of two terms: the previous cost function ℓ_s which regularize the dynamics, plus the log-PDF of the feasibility constraints, leading to

$$\tilde{\ell}_s(\boldsymbol{x}, \boldsymbol{u}) = \overbrace{\ell_s(\boldsymbol{x}, \boldsymbol{u})}^{\text{regularization term}} - \overbrace{\log(p(\boldsymbol{x}, \dot{\boldsymbol{x}}))}^{\text{feasibility constraint}} . \tag{11}$$

In practice, the logarithm prevents the solver from selecting nonfeasible states $\boldsymbol{x}$ and controls $\boldsymbol{u}$ through the dynamics equation $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$. Constraint (9d) is always satisfied. It also penalizes nonrobust behavior where no redundancy $\boldsymbol{q}$ is available, and avoids saturation of the hard constraint. Finally, the OCP solver is gently pushed away from the constraint, instead of searching for a solution living on the boundaries, which greatly improves its efficiency. Furthermore, it is unlikely that the OCP solver is trapped in local minima of μ_o, as it manipulates a full trajectory $\underline{\boldsymbol{x}}$ and not a single state $\boldsymbol{x}$. Experimentally, we observed that our OCP solver robustly computes a good local minimum when optimizing over a cost penalizing the log-PDF, while it is unlikely to converge to a solution when optimizing over set-membership.

B. Learning the CoM Reachability Proxy

We now present a complete solution to efficiently approximate the CoM feasibility, i.e., for any time t, there exists a joint configuration $\boldsymbol{q}(t)$ such that the contact placements are respected and the CoM of the polyarticulated system matches $\boldsymbol{c}(t)$.

1) Probabilistic Model: The geometric condition can be stated as the *conditional probability* of the CoM to be at the position $\boldsymbol{c}$ given the current set of K contact points $\{\boldsymbol{p_k} \in \mathbb{R}^3, k = 1 \ldots K\}$. This conditional probability is denoted by $p(\boldsymbol{c}|\boldsymbol{p}_k, k = 1 \ldots K)$, which means that for a given CoM position $\boldsymbol{c}$, it returns a positive real number which returns the probability of the CoM to be reachable from the contact points $\{\boldsymbol{p}_k, k = 1 \ldots K)\}$. Its domain is the whole $\mathbb{R}^{3(K+1)}$ on which it sums to one. It is a quantity hard to compute in general due to its inherent complexity.

The probability domain can be exactly reduced by gathering together the contact points belonging to the same rigid end-effector (e.g., the four vertices of the humanoid foot belongs to the same end-effector). We denote by $M_i = (R_i, \boldsymbol{p}_i) \in SE(3)$ the placement (position and orientation) of the contact body i. The conditional probability is then reduced to $p(\boldsymbol{c}|M_i, i = 1 \ldots K_c)$, where K_c is the number of end-effectors in contact.

2) Kernel Density Estimation by CoM Sampling: There is in general no closed form to encode $p(\boldsymbol{c}|M_i)$. Nevertheless, this conditional probability can be approximated by extensive sampling of the CoM position. The points must be sampled for a given contact configuration $M_i, i = 1 \ldots K_c$, by projecting random configurations to satisfy the given contacts. We discuss below this particular aspect of the problem.

The probability distribution can be approximated from the cloud of CoM points by the kernel density estimators (KDE) [46]. KDE are in some sense the analogs of histograms but for continuous domains. For each point of the data set, it associates one kernel centered on the point and all kernels share the same parameters. In this paper, we use isotropic Gaussian kernel.

3) Reduction of Dimension: One drawback of the KDE representation is its computational complexity: evaluating the exponential function contained in the Gaussian kernel takes around 10 ns on modern CPU. Evaluating the PDF of the KDE takes approximately $10.\text{N}_{\text{samples}}$ ns which becomes rapidly a bottleneck when the number of points is huge ($\text{N}_{\text{samples}}$ greater than 100 points).

We propose to then approximate the KDE with a Gaussian mixture model (GMM) [47]. GMMs are particularly suited to approximate a PDF with only few Gaussians in the mixture. The GMMs are learned for each end-effector from the corresponding cloud of samples by means of the expectation-maximization (EM) algorithm [48].

The quality of the GMM approximation can be estimated using the Kullback–Leibler (KL) divergence between the KDE (ground-truth) and the learned GMM (approximation) using the Monte Carlo estimator proposed in [49]. Depending on the number of Gaussians in the mixture, the divergence can reveal under or overfitting effects. The optimal number of Gaussians is easily selected for each end effector by dichotomy, as exemplified in next section.

4) Independence Hypothesis: The distribution $p(\boldsymbol{c}|M_i, i = 1..K_c)$ varies when the contact changes. It must be evaluated each time a new contact set $M_i, i = 1 \ldots K_c$ is chosen (which is typically at run-time, when a new contact phase is added to the sequence). Sampling 200 configurations in contact takes approximately 0.5 s (1 to 5 ms per configuration) and can be parallelized; fitting the GMM using EM takes approximately 0.1 s for ten Gaussians. The entire evaluation of $p(\boldsymbol{c}|M_i, i = 1 \ldots K_c)$ is then feasible at run time. We propose here an approximation to offline evaluate the distribution and get rid of this cost at run time.

To that end, we assume that variables M_i are all independent. This assumption is abusive, however is a reasonable approximation under knowledge of $\boldsymbol{c}$, later discussed. Under this assumption, the conditional probability reads

$$p(\boldsymbol{c}|M_i, i = 1 \ldots K_c) \propto \prod_{i=1}^{K_c} p_i(\boldsymbol{c}) \tag{12}$$

where $p_i(\boldsymbol{c})$ stands for $p(\boldsymbol{c}|M_i)$ and $\propto$ stands for "is proportional to." $p_i(\boldsymbol{c})$ is the probability distribution of the CoM to be at position $\boldsymbol{c}$ w.r.t the frame defined by M_i.

The assumption of independence of M_i is commonly used inside the machine-learning community as a trick to make the problem numerically tractable. In this particular case, it simplifies a lot the learning process: instead of working in a high-dimensional space, the problem is restricted to a subset of $\mathbb{R}^3$. In addition, the independence of end-effector placements plays the role of an upper-bound for the real probability: if a CoM is not feasible for at least one of the end-effectors (i.e., one of the $p_i(\boldsymbol{c})$ is equal to 0), then the joint probability is also zero. The converse is not true. We empirically show in next section that this approximation, although intuitively rough, is reasonable in practice and leads to good experimental results.

5) Summary of the Learning Procedure: In summary, for each end effector, N_{samples} configurations are sampled offline and the corresponding CoM is computed in the end-effector frame. The resulting KDE is approximated by fitting a GMM using EM. Finally, the probability of CoM occupancy is approximated as the product of $p_i(\boldsymbol{c})$, for i the end effectors in contact with the environment. The OCP cost function is then given by

$$\tilde{\ell}_s(\boldsymbol{x},\boldsymbol{u}) = \ell_s(\boldsymbol{x},\boldsymbol{u}) - \sum_{i=1}^{K_c} \log(p_i(\boldsymbol{c})). \qquad (13)$$

C. Empirical Validation of the CoM Proxy

We first validate the proposed approximation of the CoM proxy using the model of the HRP-2 robot. This unit testing is completed with an integration test in the complete LPG in the result section of this paper. For that purpose, we illustrate the learning procedure and then validate the independence assumption.

1) Illustration of the Learning Procedure: We only expose (for space reasons) the learning of the accessibility space of the CoM w.r.t. the RF. A similar study can be conducted on the three other end-effectors.

The learning process is made from a set of 20 000 points sampled uniformly in the configuration space. The KDE of this set is represented on the first row of Fig. 1. The first observation is that the PDF of the RF is not convex and follows a kind of banana distribution on the X-Z sagittal plane. In other words, this means that the distribution cannot be approximated by a single normal distribution but must be composed of several ones. The second row of Fig. 1 represents the color map of the GMM used inside the OCP. At this stage, it is important to notice that the approximation with GMMs does not fit perfectly the maximal values of the real distribution. However, this approximation is conservative with respect to the support and the level sets of the original distribution.

Fig. 2 highlights the experimental procedure suggested in Section III-B3 and shows the evolution of the KL-divergence with respect to the size of the GMMs. For the right and left feet, the KL-divergence stagnates from seven kernels in the mixture. In other words, it is sufficient to choose a GMM of size 7 to represent the CoM distribution in the foot frame. For the right and left grippers, it is not the case. The KL-divergence first decreases and then increases from 14 kernels. This behavior can be explained by the fact that the EM algorithm does not optimize the KL divergence, but the likelihood of observation (expectation). We thus represent the CoM distribution w.r.t. the grippers with a GMM of size 14.

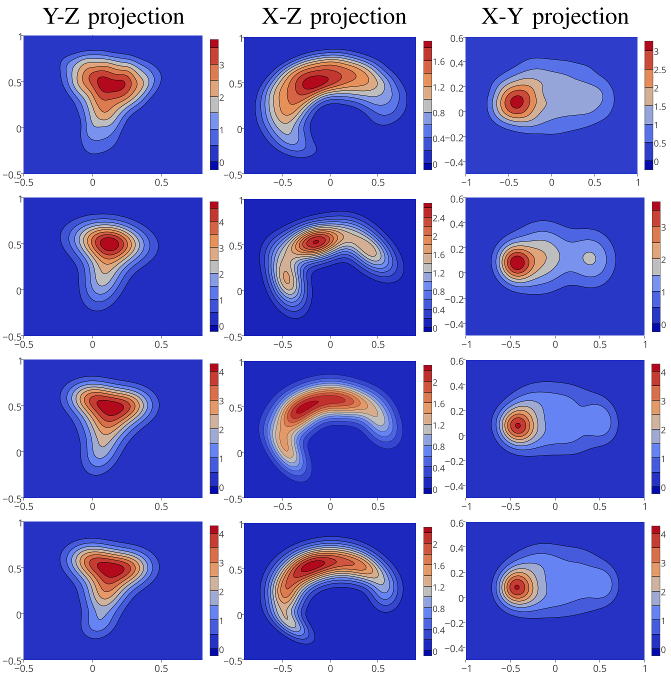

Fig. 1. Illustration of the probability density distribution of the CoM w.r.t. the right foot (RF) frame of HRP-2, projected along the three axis X, Y, Z. The first row corresponds to the ground truth distribution estimated through KDE (20 000 points). Next rows depict the learned GMM with respectively 3, 5, and 7 kernels in the mixture. All the units on these axes are expressed in meter.

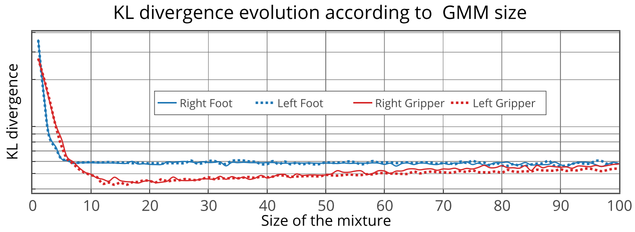

Fig. 2. Evolution of the KL divergence between the KDE distribution and GMMs of different sizes for the four end-effectors of the HRP-2 robot.

2) Validation of the Independence Assumption: In Section III-B1, we make the hypothesis of independence between the end effectors in order to simplify the learning process. We test this assumption empirically for 2, 3, and 4 contacts.

For that purpose, we use an analytical inverse-kinematics solver to uniformly sample configurations with respect to end-effector placements. These samples give a ground-truth estimation of the constrained CoM distribution, which is then compared to the estimate (12).

Fig. 3 shows the results of this validation protocol for two, three, and four contacts. We computed the cumulative distribution function for both the ground-truth distribution and the distribution obtained from the independence assumption. We

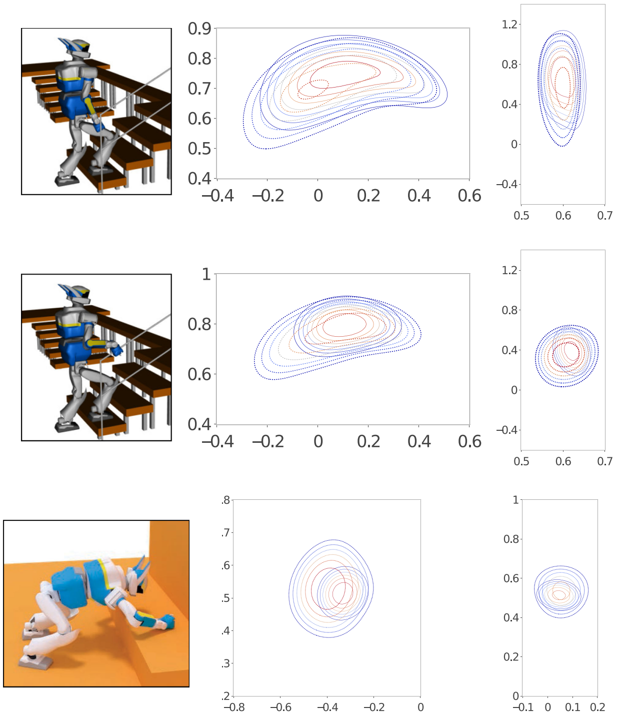

Fig. 3. Validation of the CoM independence hypothesis for various scenarios. For each scenario shown on the left column, the proxy constraint are plotted by the level sets of the cumulative distribution of the CoM occupancy measures (ground truth in solid lines and approximations $\Pi_i p_i(\boldsymbol{c})$ in dash lines), along the lateral (sagittal) and horizontal (frontal) planes. The approximations are satisfactory for two and three contacts (properly centered, with the high probability level set inside the feasibility support). For four contacts, the approximation is pushed to its limits. All the units are expressed in meters.

plot the isolines of the cumulative distribution, indexed by the volume inside the curve.

The CoM reachability volume decreases with the number of contacts for both real and approximated distributions, which is expected: with more contacts, less DoF are available to freely move the CoM.

For scenarios with two and three contacts, the independence hypothesis leads to a fair approximation of the ground truth. The approximations are centered. This is positive, as the OCP will then drive the CoM to the region of high probability, hence makes the computation of the whole-body movement easy. The important part of the workspaces are properly captured. Considering only the support of the distribution (i.e., the area where the probability is not negligible), we may consider it optimistic. However, the strength of considering the measure and not only its support is that the OCP will push the trajectory in the zone of high probability. This effect is strengthen by the simultaneous optimization of several steps. We show in the results section (Section VI) that such an approximation leads to a feasible whole-body trajectory in 100% of the observed cases.

With four contacts, the approximation is pushed to its limit. The distribution is less adequately centered, and we are missing half of the workspace (here, the back of the robot workspace). In that case, it might be more interesting to sample $p(\boldsymbol{c}|M_i, i = 1\ldots 4)$ without relying to the independence hypothesis. Yet, we have been able to compute a complete movement even in this situation. Here, we have chosen to rely on a hypothesis of independence. Other models could be taken to benefit from offline computations. For example, we may prefer to approximate the probability by a minimum over single-contact distribution (i.e., always take the most limiting constraint):

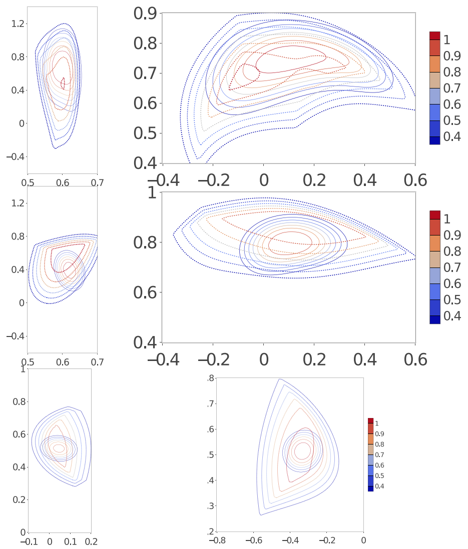

Fig. 4. Alternative approximation, using minimum of probability instead of product.

$$p(\boldsymbol{c}|M_i, i = 1\ldots K_c) \approx p_{\min}(\boldsymbol{c}) = \min_{i=1,\ldots,K_c} p_i(\boldsymbol{c}).$$

The resulting distributions for 2, 3, and 4 contacts are plotted in Fig. 4. Such an approximation leads to a weaker model than considering the hypothesis of probabilistic independence.

D. Discussion

We have proposed to represent the whole-body proxy constraints with an occupancy measure rather than a feasibility set. This presents the advantage of better capturing the information. It is more suitable when used inside an OCP, leading to smoother, more robust trajectories and smaller computation costs.

We proposed a complete implementation of this idea for representing the CoM feasibility. The occupancy measure can be either estimated on the fly from the placement of the contacts (by sampling online some contact configurations), or be approximated as a product of independent single-contact distributions (evaluated offline).

In the results section (see Section VI), we will show that the approximation is a good solution for classical locomotion scenarios. The occupancy measure properly captures the constraints, even with the independence hypothesis, and empirically always lead to a feasible whole-body movement.

For more complex scenarios, like the robot standing up using four limbs, the approximation resulting from the independence hypothesis is weaker but is also often usable. In these cases, it might be more suitable to rather use the exact distribution estimated on the fly. We did not investigate the case where the end-effector has less DoF than the dimension of the contact constraint (i.e., six for humanoid in planar contact, three for the quadruped with point contact). While theoretically valid, the chosen hypotheses might make the method less suitable.

In the future, the proposed proxy will be extended to encompass velocity, acceleration, and forces constraints. We will also work to build an offline model of the conditional probability $p(\boldsymbol{c}|M_i, i = 1 \ldots K_c)$. Finally, we also believe that constraints imposed by the environment (e.g., collision) might be included in the OCP by a similar approach. We will also consider how these proxy interacts with the method used to compute the whole-body trajectory, for example, by using some results of the whole-body motion generator to optimize a variation of the LPG, as done in [21].

IV. CWC Approximation

As mentioned in Section II-D, the linear and angular momentum variations must lie in the CWC, which is defined by (7) as the Minkowski sum of Lorentz cones. In general, there is no closed-form formulation of such Minkowski sum. Moreover, either an exact representation or the classic linear forms that are typically used to approximate it are composed of many facets, hence many inequalities to be added in the optimization problem. This is not suitable as it would be hardly tractable when solving the OCP.

We propose here an efficient approximation (both easy to compute and very suitable to use in the numerical solver) under the form of a single quadratic inequality. We empirically show in the result section (Section VI) that, despite being simple, the approximation is suitable for classical instances of the locomotion problem. We also give future directions to extend the proposed approximation while keeping its good properties, in case more complex scenarios out of the scope of this paper should be investigated.

A. State of the Art

The CWC formally corresponds to the projection in the centroidal wrench space of the Cartesian sum of the friction cones at each contact bodies. Two main approaches can then be separated: either the CWC is handled by manipulating directly the contact forces; or a representation of the projection must be written. Like other previous works in centroidal optimization [13], [32], [40], we first worked with the first solution by optimizing the trajectory of the contact forces [11]. The side effect is that the solver must also decide the contact forces, which increases the dimension of the numerical problem and makes it slower to solve. Here, we rather try to follow the second approach, and formulate an adequate representation of the CWC, so that the contact forces can be removed from the final problem to speed up its solution.

A first attempt to compute the exact supporting area has been limited to the context of static equilibrium [50]. However, this method is limited to very specific cases called "tame stances." No attempt has yet been published to express the full CWC. A linear version of the CWC is classically obtained by replacing contact cones with their linear approximations [33]. The set-membership constraint (6) is then reduced to a set of linear inequalities thanks to the double-description property of linear cones [22].

Several recent approaches now rely on the double-description of the CWC [29], [41]. Yet, the computation of the double-description is numerically unstable for 2, 3 contacts and more [41] with no polynomial-time guarantee [22]. More important, the implicit description leads to high number of inequalities (about 150 inequalities with two contacts, more than 300 with three contacts) which depend on the contact placements, thus increasing the dimensionality of the global problem. This is a bad property when aiming at using the CWC representation in a numerical problem.

In the perspective of using the CWC in a numerical problem, it is important to find a representation of the set that is smooth and simple (i.e., not composed of multiple facets, linear [22], or nonlinear [50]). We must remember that the CWC is simply the projection of the Cartesian product of quadratic cones, so it is the image of a simple and smooth geometric object. We must keep a simple representation, or the resulting numerical problem would be artificially complex.

In the following, we propose a CWC approximation composed of a single quadratic inequality, no matter the number of contacts. This approximation is obtained by first computing an outer approximation of the CWC (the so-called Löwner–John ellipsoid minimally encompassing a set of rays of the nominal CWC—see Section IV-B), which then enables us to obtain a conservative inner approximation (see Section IV-C). In practice, this inner approximation can be relaxed to obtain a less conservative approximation, which we empirically show to be suitable for various instances of the locomotion problem (see Section IV-D) and we discuss how the approximation could be extended, should a more accurate representation of the CWC be needed in more complex scenarios (see Section IV-E).

B. Outer Approximation

To keep the description of the method simple, we directly work in the 6-D space $\mathbb{R}^6$ (all the developments apply in any dimension larger than 3). In its generic form, the friction cones are Lorentz (or "ice-cream") cones, classically defined as

$$\mathcal{K}^6 \overset{\text{def}}{=} \left\{ \boldsymbol{y} = (\boldsymbol{\tau}, \eta) \in \mathbb{R}^6, \|\boldsymbol{\tau}\|_2 \leq \eta \right\} \tag{14}$$

For example, for the 3-D Coulomb cone described in Section II-A, the component $\boldsymbol{\tau}$ corresponds to the tangential forces and the η variable is the normal force scaled by the friction coefficient. With a more geometric view, $\mathcal{K}^6$ can rather be represented with a hyperplane Π intersecting the cone (the so-called conic section) and a 5-D ellipsoid $\mathcal{E}^5$ in this hyperplane

$$\mathcal{K}^6 = \left\{ \boldsymbol{y} \in \mathbb{R}^6, \boldsymbol{P}_\Pi(\boldsymbol{y}) \in \mathcal{E}^5 \right\} \tag{15}$$

with $\boldsymbol{P}_\Pi(\boldsymbol{y})$ the normal projection of $\boldsymbol{y}$ in Π. The conic section Π is easily represented by its normal direction $\boldsymbol{d}$. The projection is then $P_\Pi(\boldsymbol{y}) = \boldsymbol{y} - (\boldsymbol{y}^\top \boldsymbol{d})\boldsymbol{d}$. The ellipsoid $\mathcal{E}^5$ can be represented by its center $\boldsymbol{b} \in \Pi$ and a symmetric definite positive (SDP) matrix $\boldsymbol{Q}$ ($\mathcal{E}^5$ is the spectral ellipsoid of $\boldsymbol{Q}$)

$$\mathcal{K}^6 = \left\{ \boldsymbol{y} \in \mathbb{R}^6, \|\boldsymbol{y} - (\boldsymbol{y}^\top \boldsymbol{d})\boldsymbol{d} - \boldsymbol{b}\|_{\boldsymbol{Q}} \leq \boldsymbol{y}^\top \boldsymbol{d} \right\}. \tag{16}$$

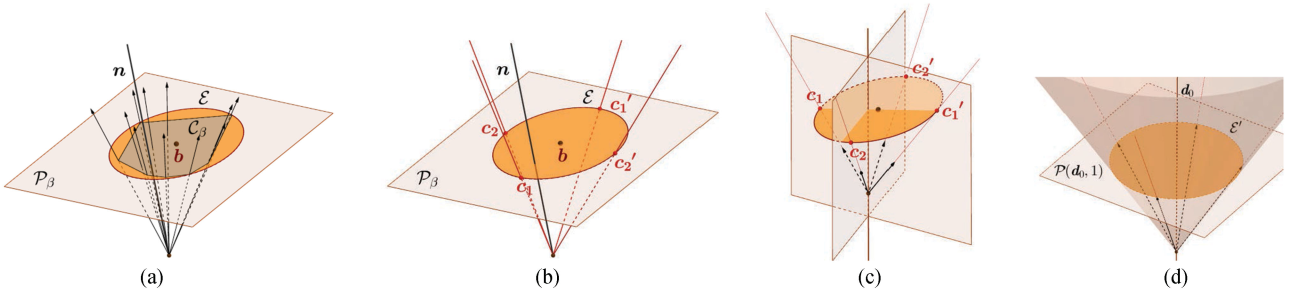

Fig. 5. Illustration of the procedure to build the outer approximation of the CWC from the collection of rays coming from the linearization of the contact cones. (a) Estimation of the plane direction $\boldsymbol{n}$ from the collection of rays contained in K_c^6. Outer ellipsoid $\mathcal{E}$ estimated from the convex-hull $\mathcal{C}(\boldsymbol{n}, \beta)$ obtained from the ray projection onto the plane $P(\boldsymbol{n}, \beta)$. (b) Computation of the antipodal points from the ellipsoid $\mathcal{E}^5$. (c) Estimation of the outer cone direction from the collection of antipodal points. (d) Computation of the minimal conic from the projection of rays onto the plane $P(\boldsymbol{d}_0, 1)$.

Several triplets $(\boldsymbol{d}, \boldsymbol{b}, \boldsymbol{Q})$ can be chosen to represent the same cone $\mathcal{K}^6$. Among all triplets, the specific case where $\boldsymbol{b}$ is null (i.e., $\mathcal{E}^5$ is centered on the normal direction $\boldsymbol{d}$) also corresponds to the spectral radius of $\boldsymbol{Q}$ being minimal. Finally, we can equivalently work with $\boldsymbol{Q}$ being a 5-matrix, or a 6-matrix with arbitrary-given norm.

Our goal is to find the best outer Lorentz approximation $\mathcal{K}_o^6$ of the CWC $\mathcal{K}_c^6$ using the generic form (16), i.e., to find the direction $\boldsymbol{d}$ and SDP matrix $\boldsymbol{Q}$ such that $\mathcal{K}_c^6 \subset \mathcal{K}_o^6$ and $\mathcal{K}_o^6$ is minimal (the center $\boldsymbol{b}$ being null at the optimum). This is equivalent to minimize the spectral radius of $\boldsymbol{Q}$ so that a sufficiently large family of rays of $\mathcal{K}_c^6$ are inside the resulting outer approximation. This statement can be translated into the following optimization problem:

$$\min_{Q\succeq 0,\, \boldsymbol{d}\in\mathbb{R}^6} \quad \det(Q) \tag{17a}$$

$$\text{s.t.} \quad \boldsymbol{\lambda}_i \in \mathcal{K}_o^6(Q, \boldsymbol{d}), \qquad i = 1, \dots, \tilde{N} \tag{17b}$$

$$\|\boldsymbol{d}\| = 1 \tag{17c}$$

$$\boldsymbol{d}^\top Q \boldsymbol{d} = 1 \tag{17d}$$

where $(\boldsymbol{\lambda}_i)_{i=1\dots N}$ is a family of rays of $\mathcal{K}_c^6$ (typically obtained by concatenation of regular rays of the 3-D contact cones $\mathcal{K}_k^3$).[4] The cost (17a) induces the minimization of the area of the section, with (17d) required to avoid trivial solutions. Constraint (17c) enforces the unitary norm of the direction vector. Constraint (17b) means that all the rays must belong to the Lorentz cone $\mathcal{K}_o^6$ parametrized by Q_o and $\boldsymbol{d}_o$. In practice, we take the same number of rays than what is typically used to compute a linear approximation of the CWC by double description [33] (i.e., four rays per contact cone). Here, we have the advantage that the complexity of problem (17) typically scales linearly with the number of rays while it induces a combinatorial when using the double description.

Nevertheless, (17) is hard to solve. To simplify its resolution, we propose to better use its geometric structure and rely on a dedicated alternate descent strategy which iteratively and independently optimize the plan and the ellipsoid. The procedure is summarized in Fig. 5:

1) we first find a suboptimal direction $\boldsymbol{d}$;
2) then a suboptimal (noncentered) ellipsoid $\mathcal{E}^5 = (\boldsymbol{Q}, \boldsymbol{b})$;
3) this ellipsoid is used to compute the optimal direction $\boldsymbol{d}$ where the ellipsoid would be centered;
4) the optimal ellipsoid is then obtained by optimizing the sole matrix $\boldsymbol{Q}$.

1) Choosing a Initial Direction d: We can choose the normal direction $\boldsymbol{n}$ by only considering the family of rays $(\boldsymbol{\lambda}_i)$ (if the family is large enough, which is the same hypothesis—implicitly—done with the double-description approach, and is always true in practice). Each ray $\boldsymbol{\lambda}_i$ defines a half-space (the linearized cone is the intersection of all the half-spaces). Clearly, if the normal direction is not in this half-space, the normal hyperplane Π will not properly intersect the cone (i.e., the intersection of $\mathcal{K}_c^6$ and Π is not an ellipsoid) [51]. We then search $\boldsymbol{d}$ as close as possible to the mean of the family of rays, while respecting this constraint. It can be computed with the following quadratic program (QP):

$$\min_{\boldsymbol{d}\in\mathbb{R}^6} \quad \frac{1}{2}\sum_i \|\boldsymbol{d} - \boldsymbol{\lambda}_i\|_2^2 \tag{18a}$$

$$\text{s.t.} \quad \Lambda^\top \boldsymbol{d} > \boldsymbol{0} \tag{18b}$$

where Λ is the matrix where columns are the rays $\boldsymbol{\lambda}_k$ [see Fig. 5(a).

2) Computing an Outer Ellipsoid on the Plane: Any hyperplane $\Pi \overset{\text{def}}{=} \{\boldsymbol{x} \in \mathbb{R}^6, \boldsymbol{n}^\top \boldsymbol{x} = \beta\}$ with $\beta > 0$ can be considered (we typically take $\beta = 1$). The intersection of the rays $(\boldsymbol{\lambda}_i)_i$ with Π defines a family of points $(\boldsymbol{p}_i) \overset{\text{def}}{=} \left(\beta\boldsymbol{\lambda}_i / \boldsymbol{n}^\top \boldsymbol{\lambda}\right)_i$ in Π. The convex hull of (p_i) is the intersection of the linear inner approximation with Π. We search $\mathcal{E}^5$ as the minimum-volume ellipsoid that encloses the set of points $(p_i)_i$ (also called the Löwner–John ellipsoid [52]) represented by its center $\boldsymbol{b}$ and spectral matrix $\boldsymbol{Q}$. The pair of parameters is obtained by the following second-order conic program (SOCP):

$$\min_{\boldsymbol{b}\in\mathbb{R}^5,\, \boldsymbol{Q}\in\mathbb{R}^{5\times 5}} \quad \det Q \tag{19a}$$

$$\text{s.t.} \quad Q \succeq 0 \tag{19b}$$

$$\forall i = 1 \dots N, \ \|Q\boldsymbol{p}_i - \mathbf{b}\| \leq 1. \tag{19c}$$

[4] This family of rays span a linear approximation of $\mathcal{K}_c^6$ which is typically handled by the double-description approach [33].

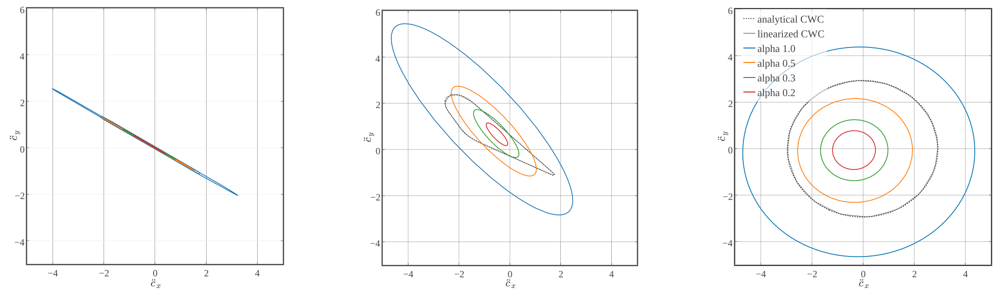

Fig. 6. Illustration of the contact wrench approximations for scenarios of Fig. 3. The exact CWC is obtained by sampling and taking the convex hull of the samples. The exact CWC and its linear approximation closely match. The outer approximation is obtained with $\alpha = 1$, and the inner approximation with $\alpha = 0.2$. The approximation $\alpha = 0.3$ appears to be an efficient tradeoff. For two contacts, the exact CWC is a quite flat objects (as torques along the contact are strongly limited by the size of the body in contact—here the feet): it is properly approximated by a flat ellipsoid. For three contacts, the CWC is clearly composed of several nonlinear facets. In all cases, the inner approximation is conservative, while a more accurate (but nonconservative) approximation can be found with a unique value of α.

This SOCP reads as: find the minimal (by minimizing the determinant) ellipsoid (by making Q positive) containing each of the N rays (by satisfying the last inequalities for each *i* below N). Details can be found in [35, p. 410].

3) Choosing the Optimal Direction: As previously explained, the minimal outer approximation is found when the ellipsoid $\mathcal{E}^5$ is centered on direction $\boldsymbol{d}$. We can directly obtain the optimal direction by considering the antipodal points of the initial ellipsoid $\mathcal{E}^5$ (the opposite points on the ellipsoid corresponding to Eigen directions of $\boldsymbol{Q}$). Consider the bisecting planes B_i defined from antipodal points [$i = 1 \ldots 5$, see Fig. 5(c)], i.e., B_i is the hyperplane containing the center of the cone and for which the pair of antipodal points are reflections. Then, the optimal direction $\boldsymbol{d}^*$ is defined by the intersections of the five hyperplanes B_i.

4) Computing the Optimal Ellipsoid: The minimal section is finally computed by solving again SOCP (19), but this time in the plane defined by the optimal direction $\boldsymbol{d}^*$. For that, we first define the intersecting plane $\diamond$ with normal $\boldsymbol{d}^*$ and level value $\beta = 1$. Then, we project the rays onto this plane and compute the minimal ellipsoid $\mathcal{E}^*$ defined by $\boldsymbol{Q}^*$ and $\boldsymbol{b} = 0$ enclosing those projected points by imposing its center to be zero in the plane frame [see Fig. 5(d)].

C. Inner Approximation

From the previous section, we have an outer approximation, i.e., any feasible wrench is guaranteed to be in the outer approximation. We want the opposite, i.e., an inner (conservation) approximation. A property of the Löwner–John *outer* ellipsoid is that an inner ellipsoid can be obtained by simply scaling down the outer approximation by the dimension of the space (here, dimension 5 of the plane $\diamond$) [52]. Moreover, if the convex-hull is symmetric with respect to the center of the ellipsoid, it can be simply reduced by a factor $\sqrt{5}$ as detailed in [35, p. 410].

Using this property, we obtain an inner approximation of $\mathcal{K}_c^6$, denoted by $\mathcal{K}_i^6$ and having the same direction as $\mathcal{K}_o^6$.

The proposed approximation is guaranteed to strictly lie inside the CWC by construction. While it is conservative—less centroidal wrench variations are allowed—the proposed approximation can be used in the context of robust control where the contact forces must be sufficiently inside the contact cones to avoid contact slippage.

In practice, the reduction factor α can be chosen in the interval $[1/5; 1/\sqrt{5}]$ depending on the invariance properties of the set of wrenches (symmetries, alignment, etc.). Using the lower bound leads to theoretical guarantee, while choosing a greater value allows us to exploit geometric properties of the contact set, like symmetries. Theoretically, the factor α can be adjusted on the fly by a quick dichotomy in the range $[\frac{1}{5}; \frac{1}{\sqrt{5}}]$. In practice, we will see in the following that choosing a fixed α larger than these nominal values also leads to an effective inner approximation of the CWC.

D. Validation of the Centroidal Cone Approximation

We illustrate here this inner approximation with respect to both the real CWC and the linearized version of the CWC on the scenarios already used in Section III-C2. Given a contact configuration of the robot (i.e., contact placements and COM position), we uniformly sample values of the centroidal wrench and check whether they are in the true CWC cone, in its linear approximation and in either α-approximation of the CWC. The resulting sampling live in a 6-D manifold. For visualization purposes, we only plot a 2-D cross section of the cones. Fig. 6 shows the cross section corresponding to $\dot{\boldsymbol{L}}_{\boldsymbol{c}} = \boldsymbol{0}$ and $\ddot{\mathbf{c}}^z = 0$ (i.e., corresponding to the LIPM dynamics), in the case of 2, 3, and 4 contacts. The true CWC is obtained by a rejection sampling approach: for a random value of the centroidal wrench, we check whether there exists a contact force distribution which gives rise to this centroidal wrench by solving a SOCP problem. If such distribution exists, we keep track of this wrench, otherwise it is invalid. From this set of valid contact wrenches, we are able to compute its projection on the 2-D surface and then get its convex-hull, corresponding to the black thin curves in Fig. 6. In other words, the more wrench samples, the better the ground-truth approximation of the CWC.

As theoretically expected, the outer ($\alpha = 1$) approximation contains the true CWC. The linear approximation is inside the true CWC and closely matches it (to the cost of high computation costs when solving the resulting OCP). A fair but nonconservative approximation is obtained for $\alpha = 0.5$, while $\alpha = 0.3$

is in the CWC but is conservative. In practice during the experiments with the robot, we used $\alpha = 0.3$. The resulting cone indeed corresponds to the inside of CWC where it is the most desirable to select the forces achieved in the context of legged locomotion. For the four contacts scenario, we can observe that the approximation $\alpha = 0.5$ is also contained in the real CWC because of the symmetries in the contact placements. However, we did not find useful in practice to adjust α in order to take advantage of the larger volume. Note that the outer approximation in general does not touch the true cone when plot in an arbitrary 2-D section (while it does in the 6-D space).

A side result is obtained from comparing the cone resulting two contacts to the cone resulting three contacts. The CWC remarkably grows with the addition of a new contact. This goes in favor of multicontact locomotion: adding contacts enable the robot to increase its dynamics capabilities while constraining more its kinematics.

E. Discussion

We proposed a quadratic approximation that is suitable to represent the CWC in a numerical problem. It can be made conservative or adjusted by tuning α to better fit to the real CWC. In practice, we show in the result section (see Section VI) that slightly relaxing the inner approximation without fine tuning α leads to excellent results.

It may happen that the approximation is not satisfactory (either too conservative—α too small—leading to unfeasible forces—α too large). This is easy to detect, and thanks to the reported computation speed, the parameter can easily be adjusted on the fly. In practice, we have not observed this to be necessary.

For scenarios where the proposed approximation is not suitable, two approaches can be considered. First, our OCP solver can work directly with the contact cones by optimizing the contact forces [11]: the CWC approximation is optional. This comes with an extra computation cost, quantified in the result section (see Section VI). Thanks to sparsity and low number of forces, the performances when also deciding the contact forces are acceptable. Alternatively, the approximation might be extended. In the cases where the CWC does not span the entire 6-D space (for example, in case of one or two point contact), or if the forces are not bounded (for example, when climbing a chimney), it is easy to detect the trivial direction and only compute the ellipsoid in the orthogonal space. If more accuracy is needed, the ellipsoid can be replaced by another smooth object, for example, a superquadric. Optimizing the shape of the superquadric to get an outer approximation would be a nonlinear yet tractable optimization problem, that we expect to be similarly easy to solve.

The main message of this section is that the CWC must be represented by a smooth and simple object. In particular, representing the CWC by a linear polytope using the double description is not the best choice and might lead to bad numerical performances. The proposed approximation is a fair tradeoff between performances and accuracy, able to cope with most of the relevant locomotion scenarios. If needed, a better approximation can be used for extra computation cost. In the context of locomotion, we advocate for our approximation which we observed to be good enough, and knowing that reduced computation time is a true advantage when preparing the implementation of model-predictive control (MPC).

V. Final Formulation of the OCP

In this section, we formulate the tailored optimal control used in the experimental section. It is based on the generic OCP (9) and uses the results of Section III for the CoM proxy and of Section IV for the constraints on the control vector. In addition to that, we proposed an effective way to solve it in order to reach real-time computations.

A. Tailored OCP

Based on previous sections, the OCP (9) is finally implemented under the following from:

$$\min_{\underline{\boldsymbol{x}},\underline{\boldsymbol{u}},(\Delta t_s)} \quad \sum_{s=1}^{S} \int_{t_s}^{t_s+\Delta t_s} \ell_s(\boldsymbol{x},\boldsymbol{u}) - \sum_i \log(p_i(\boldsymbol{x}))\, dt \tag{20a}$$

$$\text{s.t.} \quad \forall t \quad \dot{\boldsymbol{x}} = f(\boldsymbol{x},\boldsymbol{u}) \tag{20b}$$

$$\forall t \quad \boldsymbol{u} \in \mathcal{K} \tag{20c}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{20d}$$

$$\boldsymbol{x}(T) = (\boldsymbol{c}_f, \boldsymbol{0}, \boldsymbol{0})\,, \dot{\boldsymbol{x}}(T) = \boldsymbol{0} \tag{20e}$$

where the feasibility constraints (9d) is replaced by the additional log likelihood sum in the cost function, as explained in Section III. The control variable $\boldsymbol{u}$ can be either $\boldsymbol{u}_f$ [the contact forces with the Lorentz cone constraints for (20c)] or as $\boldsymbol{u}_c$ (centroidal wrench with approximate quadratic CWC for (20c), as explained in Section IV). We discuss this choice in the result section. We reduce the terminal viability constraints (9f) to the constraint of the robot to be at rest at the end of the motion (20e). Here, the ith mixture of Gaussians $p_i(\boldsymbol{c})$ has been replaced by $p_i(\boldsymbol{x})$ to be generic and the cost function is given by

$$\ell_s(\boldsymbol{x},\boldsymbol{u}) = w_{\ddot{\boldsymbol{c}}}\|\ddot{\boldsymbol{c}}\|_2^2 + w_{\dot{\boldsymbol{L}}_{\boldsymbol{c}}}\|\dot{\boldsymbol{L}}_{\boldsymbol{c}}\|_2^2.$$

For all the experiments and robots presented in the next section, we use the same weighting in the cost function: $w_{\ddot{\boldsymbol{c}}} = w_{\dot{\boldsymbol{L}}_{\boldsymbol{c}}} = 10$. This weighting allows us to balance between the contribution of the log-PDF terms and the regularizations of the dynamic variables ensuring a smooth state trajectory.

Concerning the cost tuning, it is important to mention that there are none. In other words, we use a fixed ℓ_s for all the experiments, as a regularization criterion in order to select among all the feasible solutions, the ones which are smooth with respect to the state variables of the centroidal dynamics.

B. Transcription Into a Numerical Problem

The decision variables of problem (20) are the trajectory in state $\underline{\boldsymbol{x}}$ and control $\underline{\boldsymbol{u}}$ spaces. The study of the geometrical properties of its solution, through the optimality principles of Hamilton–Jacobi–Belman or Pontryagin Maximum, seems yet out of reach [because of the bilinear constraint (20b)]. Like previous works about LPG, we rather follow a "direct" approach, by doing the transcription of this OCP into a static optimization problem where $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{u}}$ are expressed in a truncated (finite dimension) basis of function.

Several choices arise during the transcription and they have a direct impact on the performances. They are summarized here, with their justification. We mostly follow the recommendations coming from the Control community (MPC), adapted to the specificities of our robotic problem.

1) Representing the Trajectories: They are typically represented by piecewise polynomials, with degrees varying from 0 (piecewise constant) to 3 (piecewise cubic). Other discretization, or higher degrees, might be chosen, but polynomials are known to be universal approximations with good properties. A tradeoff must be chosen, between the polynomial degree and the number of intervals. Our experience is that better results are obtained with higher degree and large intervals (opposite to low degree and shorter intervals). We used degree three polynomials and set two to three intervals per contact phases. It is also important to get smooth forces, as the robot would not be able to track the discontinuities in the reference forces.

2) Representing the Dynamics: An important choice is how the dynamics constraint (20b) is implemented to constrain $\underline{\boldsymbol{x}}$ with respect to $\underline{\boldsymbol{x}}$. Collocation is often used in robotics [21], [33], [40]. $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{u}}$ are independently represented and the dynamics (20b) is only checked on a fixed temporal grid. For accuracy, a fine grid must be chosen. Consequently, a finer discretization of $\underline{\boldsymbol{x}}$ must be taken so that it can fit to the real dynamics at every collocation points. This leads to artificially large numerical problems.

On the other hand, single shooting consists in only representing $\underline{\boldsymbol{u}}$ by some polynomials, and obtaining $\underline{\boldsymbol{x}}$ by temporal integration of the dynamics [18]. It follows that the dynamics can be tracked as accurately as needed. On the other hand, single-shooting transcription leads to an unstable numerical problem. Because of the unstable bilinear equation (20b), single shooting is poorly applicable to our problem.

Multiple shooting can be seen as a tradeoff between both [53]. The dynamics is integrated over the shooting intervals, while $\underline{\boldsymbol{x}}$ is only represented by its static value at the beginning of each interval (the values on the rest of the interval being obtained by integration). An additional constraint must then be added to enforce the continuity at the shooting nodes. It follows that the dynamics can be accurately tracked with a low-dimension representation of $\underline{\boldsymbol{x}}$ and only few additional constraints. Additionally, the resulting optimization problem is numerically stable.

3) Continuous Integration: On each shooting interval, the dynamics is integrated, typically using a Runge–Kutta integrator. The integrator must be set with adaptive time samplings; otherwise, it leads to numerical inaccuracies that tend to jam the numerical solver. A side effect is that the derivatives of any function of $\boldsymbol{x}$ can no longer be written in a closed form. However, they can be easily integrated using the same integration scheme (using the so-called sensitivities of the integrator), resulting in an accurate computation of the true derivatives.

4) Smooth Force Constraints: As already explained when describing the CWC, force constraints must be simply and smoothly represented. Using a large number of inequalities would make the problem artificially complex. In the best case, it would slow the solver converge, by increasing its size and preventing proper use if the derivatives of the set border. In the worst case, the active-set algorithm internal to the solver would not converge and fail.

5) Continuous Validation of the Kinematic Proxy: OCP constraints should typically hold at any time, but are indeed only checked at a finite number of points. However, the kinematic proxy is formulated as an additional cost term with a log barrier. It is then continuously integrated and checked at any time.

C. Numerical Algorithm

Once the OCP is transcribed into a numerical problem, a numerical algorithm must be chosen and implemented.

1) Numerical Solver: Two main classes might be considered. As interior-point solver are difficult to initialize, we rely on a SQP. The solver must be sparse to take advantage of the OCP temporal structure: we used OOQP [54] to sparsely solve the QPs inside the SQP.

2) Warm Start: The SQP can work from scratch (e.g., 0 control and constant shooting states). However, we found that it is easy to build a good warm start thanks to the multiple-shooting transcription. We initialize the OCP with a constant CoM position at each shooting interval. The velocity of each node is set to 0, with forces compensating the gravity. The state is then discontinuous, but the solver is able to address this issue and converge from this guess to a feasible trajectory.

3) Optimal Control Toolbox: We relied on the software MUSCOD-II to easily transcript our problem [55]. MUSCOD offers efficient integrators with sensitivities, transcription from the continuous OCP into a static optimization problem, and a SQP solver tailored for multiple shooting. However, the proposed transcription would apply to any off-the-shelf solver, as no specific point would be difficult to implement outside MUSCOD (except maybe its tailored SQP). In particular, previous attempts to solve the centroidal OCP with MUSCOD but without adopting the same transcription led to minutes of computation [10].

VI. Experimental Results

We first quickly present the complete pipeline used to compute the robot movements, from generating the sequence of contacts, then optimizing the locomotion patterns and finally computing the whole-body trajectory. We summarize the computation performances of our method. We report several movements with the real HRP-2 humanoid robot in industrial scenarios, along with the same last movement in simulation on the new TALOS robot. Finally, we validate the proposed approximations on two sets of 1000 simulations in random debris environments.

A. Description of the Complete Pipeline

Our locomotion framework is composed of three stages.

1) Contact Sequence Planning: Depending on the experiments, the contact sequences are either manually designed or automatically generated using the contact planner [29]. We also

TABLE I
TIME SCORES, WITH DISTRIBUTION, FOR SOLVING THE OCP

	Exp.1 (a)	Exp.1 (b)	Exp.2	Exp.3
control type	$\boldsymbol{u}_c$	$\boldsymbol{u}_f$	$\boldsymbol{u}_c$	$\boldsymbol{u}_c$
motion duration	8 s	8 s	24 s	36 s
computation time	1.23 s	3.89 s	8 s	10 s
iterations	22	40	42	15
time / iteration	56 ms	97 ms	0.19 s	0.66 s
QP	42 %	77 %	70 %	70 %
sensitivity computation	53 %	14 %	20 %	16 %

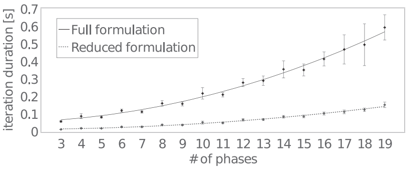

Fig. 7. Evolution of the mean duration to compute a single iteration of the multiple-shooting algorithm according to the number of contact phases.

manually design the end-effector trajectories by using splines with zero acceleration and velocity at take off and landing instants.

2) Centroidal Resolution: From the contact sequence and the learned CoM feasibility constraints, we solve the optimal control formulation (20). We initialize the OCP with a linear interpolation of the CoM positions between the initial and final postures. In addition, the OCP initial guess considers the system to be at rest on each multiple-shooting interval. The state is then discontinuous at each multiple-shooting node, which is not a problem for the multiple-shooting solver. The control inputs are encoded as cubic splines, allowing the control variable to be continuous and differentiable along all the motions.

3) Whole-Body Resolution: From the OCP, we obtain a reference trajectory for the centroidal dynamics that we follow using a second-order inverse kinematics (IK) solver similar to [56]. In addition, the IK must track the end-effector trajectories. Optimal forces are also extracted from the OCP (if $\boldsymbol{u}_f$ is the control variable) and can be used as references to control the robot with an inverse dynamics low-level controller.

B. Time Scores

Table I summarizes the performances of our approach on the different scenarios, either using the centroidal wrench u_c or the contact forces u_f as control input. The two last rows of this table show the percentage of the time spent either in solving the QP inside the SQP or in computing the numerical sensitivities of the multiple-shooting problem using finite differences. All the computations were performed on a single thread of a i7 CPU running at 2.2 GHz (similar to the one we have on the real robot).

The solver takes between 7 and 15 ms to make one step of optimization for one second of motion. If using our method as a model-predictive controller, it would be necessary to take 2 to 3 s of horizon length, allowing the solver to run at 20 Hz. This matches the application needs [57]. Fig. 7 shows the evolution of the mean duration to compute a single iteration of the OCP solver according to the number of contact phases (hence duration of the horizon) in the sequence. The reduced OCP is ten times faster for ten steps or less. For more than ten steps, the cost grows quadratically (likely due to internal finite-differencing implementation; theoretically, the cost is linear in the number of shooting nodes). The quadratic term is not predominant until 20 steps for the reduced formulation.

There exists no open-source software that would have allowed us to benchmark our method with respect to existing works Time scores are given in some previous works. In [10], 30 min are needed for some few steps. In [12], 8 min are needed per iteration for long movements. In [13], 100 ms are needed per iterations for five contact phases, but using a relaxation of the dynamics (results are not demonstrated on a real robot). From our own experience on preliminary implementations, optimizing whole-body movements with the real robot constraints implies several 10 min of computation. Whole-body optimization using MUSCOD-II [58], [59] requires hours of computation to generate biped gaits. In [43], the solver needs 3 h to generate multicontact movements. Model-predictive control is targeted in [60] and [61], while one step of optimization (with horizon length of 0.5 s) implies 100 ms of computation; however, the results are yet not realistic enough to generate locomotion on a real robot.

In summary, our approach is the first one that is able to generate effective movements that the robot can execute, with a versatile and exact formulation, while matching the computation performances imposed by the application.

C. Experiment 1—Long Steps Walking

In this first experiment, we aim at comparing the influence of both types of controls $\boldsymbol{u}_c$ and $\boldsymbol{u}_f$ on the solution. For that purpose, we use a simple benchmark which consists in long step walking with a stride of 0.9 m with the HRP-2 robot. This stride is close to the limit of what a humanoid robot of 1.6 m height can achieve. Then, starting from a resting position and ending at another resting position, the solver has to find a crouching gait in order to satisfy the kinematics feasibility constraints.

The results of such motion are depicted in Fig. 8. The state trajectories have similar shapes, with smooth trajectories at the position and velocity levels on the x-axis and y-axis directions. On the z-axis, we can observe some weak oscillations of the CoM position mainly when optimizing the forces $\boldsymbol{u}_f$. This might appear as the conflict between the least-square cost on the CoM acceleration and the feasibility constraint. For the contact forces control, the angular momentum trajectory is more jerky. This is because the angular momentum is not a direct control of the systems, but a consequence of the contact wrenches action. Then, the least-square minimization of such a quantity is affected by the sensitivities and the conic constraints on the contact forces. The noise is mostly below the threshold of numerical noise. While direct OCP resolution (e.g., multiple shooting) is sensitive to local minima, it is likely that the two obtained trajectories are numerical approximation of a same minimum, with the formulation $\boldsymbol{u}_c$ better able to approximate it thanks to the more direct correlation between the centroidal variables and the resulting motion.

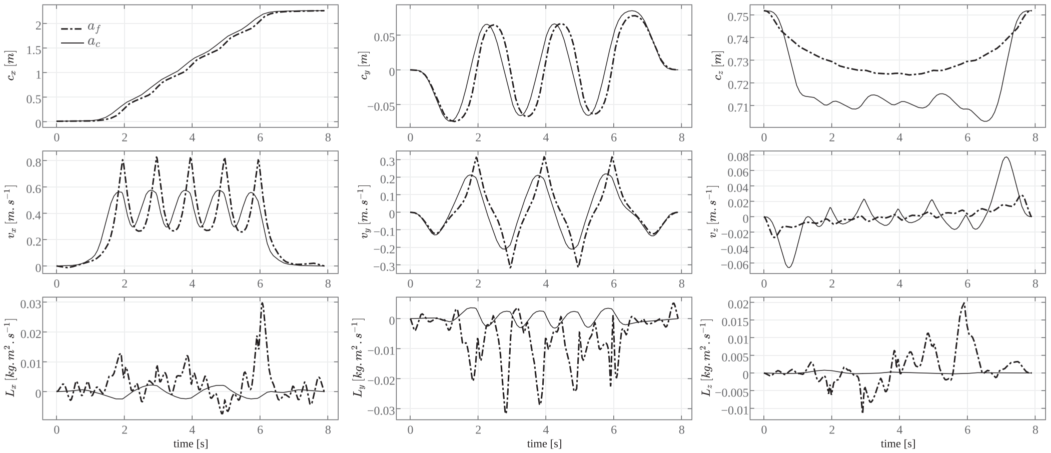

Fig. 8. Experiment 1: Comparison of the state trajectories obtained with either the force-based OCP (simple and exact 3-D cones, nonminimal parameters – a_f) or the motion-based OCP (approximate 6-D cone, minimal 6-D parameters – a_c). In theory, the optimum of both problems should be the same, however the numerical properties of each OCP leads to minor variations. The CoM trajectories have similar shape, but the dynamic marginally varies. The motion-based OCP leads to marginally smoother trajectories. Much more oscillations appear at the angular momentum level when optimizing the forces, but they mostly correspond to numerical noise.

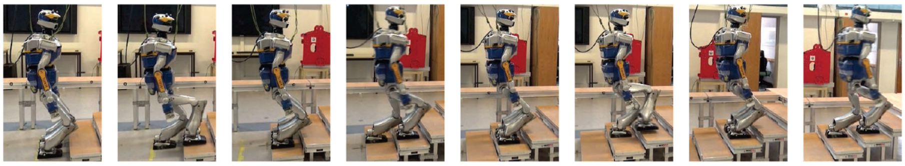

Fig. 9. Experiment 2: Snapshots of the climbing up 10-cm high steps motion with the HRP-2 robot.

D. Experiment 2—Climbing Up 10-cm High Steps

The experimental setup is an industrial stairs made of six 10-cm high steps. The steps have a length of 30 cm. The durations of the single and double support phases are 1.4 s and 0.2 s, respectively. The resulting motion is depicted in Fig. 9. During execution, the reference posture is tracked as well as the reference foot forces using the robot low-level control system (named HRP "stabilizer").

Fig. 10 shows two trajectories of the CoM projected in the RF frame: the black curve takes into account the log-pdf term in the cost function, while the green one does not. The figure also includes the level sets of the GMM of RF (depicted in Fig. 1). It appears that the OCP tends to maximize the inclination of the CoM to stay in the most feasible region, i.e., closed to the maxima of the PDF. On the contrary, if we do not add the log-pdf term, the CoM tends to be infeasible.

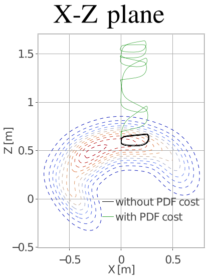

Fig. 10. Experiment 2: Projection of the CoM trajectory inside the RF frame with and without taking into account the log-pdf term in the cost function. The level set corresponds to the GMM distribution used in our OCP.

E. Experiment 3—Climbing Up 15-cm High Steps With Handrail Support

The experimental setup is another industrial staircase made of four 15-cm high steps and equipped with a handrail. The steps have a length of 30 cm too. The durations of the double and triple support phases are 1.8 s and 0.4 s, respectively. Here, the double support phases correspond either to the case of two feet on the steps or one feet plus the right gripper on the handrail. Snapshots of the entire motion are shown in Fig. 11.

We reproduce the climbing stairs with handrail scenario, but this time with the TALOS robot in simulation. Compared to HRP-2, TALOS is a 1.78-m high-humanoid robot weighing around 100 kg. For this experiment, only the end-effector trajectories and the GMMs are different: the cost function remains the same. The complete motion is depicted in Fig. 11.

F. Experiment 4—Empirical Validation of the Kinematic Proxy and the Conic Inner Approximation

We empirically check the proposed formulation: the robustness of the OCP numerical solver, and the validity of the two proposed approximations (the kinematic proxy and the conic inner approximation). We randomly sampled environments similar to the debris of the Darpa Robotics Challenge (DRC), composed of 12 tilted stepstones. Altitude, stride, and angles of the stones were randomly chosen. Each stone is placed relatively to the

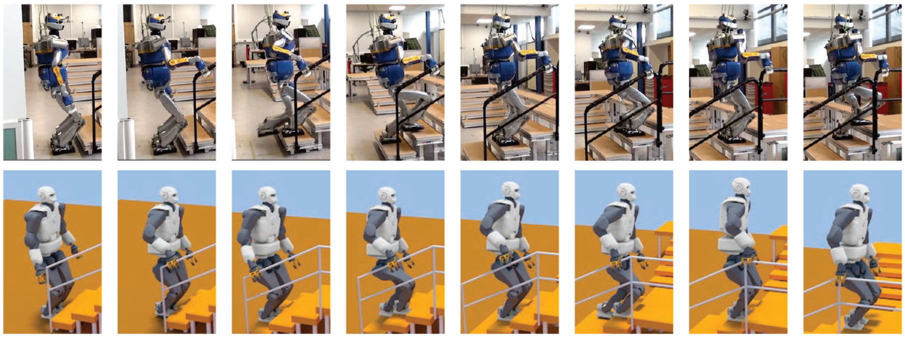

Fig. 11. Experiment 3: Snapshots of the climbing up 15-cm high steps motion with the HRP-2 or TALOS robot using the handrail.

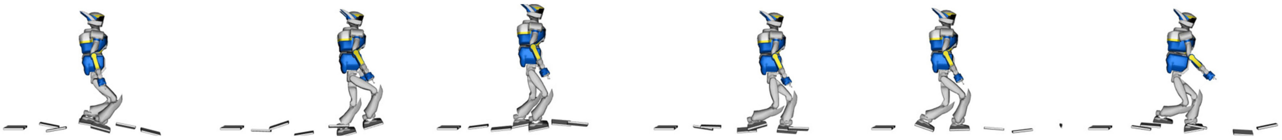

Fig. 12. Experiment 4: Snapshots of the trajectory on six different debris scenarios (simple random distribution on the top row, difficult distribution on the bottom row).

previous stone, with a delta in the forward (x), lateral (y), and vertical (z) directions uniformly sampled in a given interval, and roll (α_r) and pitch (α_p) angles relative to horizontal also taken uniformly in an interval. A first set of 1000 scenarios were sampled with uniform distribution where $(x, y, z, \alpha_r, \alpha_p)$ is taken between (0.2, −0.05, −0.05, −0.35, −0.35) and (0.4, 0.05, 0.05, 0.35, 0.35). A second set of more difficult 1000 scenarios were sampled with intervals between (0.3, −0.1, −0.1, −0.35, −0.35) and (0.9, 0.1, 0.1, 0.35, 0.35). An overview of the scenarios is shown in Fig. 12.

The protocol is made of two consecutive stages: we check first the validity and interest of the CWC approximation by computing a centroidal pattern with three different formulations; we then measure the validity of the proxy constraint when generating the whole-body movement from the centroidal pattern.

1) In the First Stage (LPG): we solve the centroidal optimal control formulation with three variants:

a) the full control formulation with one ice-cream cone at each contact surface (full force wrench cone—FWC);

b) the reduced control formulation using the proposed conic approximation of the CWC (conic centroidal wrench cone—CWC);

c) the reduced control formulation using a high-dimensional linear constraint inequality to represent the CWC obtained by using the double description (linear centroidal wrench cone—LWC).

Each LPG formulation computes the momentum trajectory and the contact timings, while optimizing the proxy integral. We took (FWC) as ground truth to assess the feasibility of the two conic approximations (CWC) and (LCW). We then measure the validity of (CWC) and (LWC) and the capabilities of the OCP solver to handle both.

TABLE II
SUCCESS RATES IN TWO SETS OF 1000 RANDOMIZED DEBRIS

		FWC	CWC	LWC
		$\boldsymbol{u}_f \in \mathsf{X}_{k=1}^{K} \mathcal{K}_k^3$	$\boldsymbol{u}_c \in \mathcal{K}_0^6$	$\boldsymbol{u}_c \in \mathcal{K}_c^6$
Simple	success rate OCP resolution	–	89.2 %	63.8 %
	success rate IK resolution	100 %	100 %	100 %
Difficult	success rate OCP resolution	–	67.8 %	42.9 %
	success rate IK resolution	100 %	100 %	100 %

The results are summarized in the white rows of Table II. First, (CWC) leads to more success than (LWC) for the two data sets (simple debris and difficult debris). LWC indeed is a better approximation than CWC (i.e., the volume of the LWC is larger than CWC), but LWC is not suitable for LPG. The failures of LWC corresponds to either failure of the double description algorithm [22] (5% of the cases) and failure of the active set of the QP solver[5] after 1 000 000 iterations (95% of the cases). On the opposite, failures of CWC correspond to situations where the needed actuation capabilities to cross a large gap between two stones are not captured by the quadratic approximation. As both CWC and LWC are conservative approximation, any solution found by the two LPG is dynamically consistent. The conclusion is that LWC is not a sane approximation when used inside an OCP. CWC is better used by the OCP, although its conservatism might be limiting in difficult scenarios. However, when a scenario is in the difficulty range of CWC, it can be repeatedly handled as the OCP will always safely converge.

2) In the Second Stage (IK): we feed the IK solver with the computed centroidal trajectories and some suitable foot trajectories. As expected, the success rate does not depend on FWC, CWC, or LWC formulations. The results are summarized in the

[5] for (LPG), we had to rely on the QP solver QPOPT [62], whose active set is more robust, as QP solver OOQP (used for FWC and CWC) was not able to deal with the linear cone constraint.

gray rows of Table II. The IK is always able to find a whole-body movement. This validates the fact that the kinematic proxy accurately approximates the kinematic limits, at least for debris environments.

VII. Discussion

The proposed method is able to generate various centroidal locomotion patterns for the most classical scenarios of locomotion: walking on any quasi-flat floors, biped, or quadruped, climbing stairs, with or without contacts of the hand on, e.g., a handrail or a wall. Thanks to the computational efficiency, it is aiming at controlling the robot and could be used to recover after disturbances or to adapt to an unexpected contact. The kinematic proxy, automatically generated from data, makes the centroidal trajectory easy to track when generating the whole-body movement. As the trajectory is smooth by design, the resulting robot movement is also smooth and directly apply to the physical platform.

Yet, some problems remain open, to extend it to more complex scenarios or to apply it to control the robot using a receding-horizon scheme.

First, the kinematics and dynamics constraint are handled through approximations. We claim that these approximations are a fair tradeoff between accuracy and efficiency, and empirically demonstrated that they are valid for, e.g., most of the motion problems of the Darpa Robotics Challenge. *The proposed kinematic proxy* can be either accurately estimated on the fly, or, as we rather propose, approximated as a product of offline distributions. For more complex scenarios, the proxy should be extended to take into account other kinematic constraints (e.g., collisions with the environment). Actuation constraints (e.g., motor limits [63]) should also be added. In the end, the proxy might measure the occupancies of CoM position, velocity, and acceleration, hence also encompass a representation of the CWC. While the general idea to learn such a proxy is clear (generate many data in simulation and learn the occupancy measure by any suitable representation), we believe that some developments in machine learning are required to scale to the problem dimension and geometry (most of the variables describing a robot and its dynamics lie on manifolds whose geometry must be well captured by the approximation).

Second, *the proposed CWC approximation* is simple, which makes it easy to compute and to use in the OCP solver. The side effect is that it might lack accuracy to properly capture the shape of the exact CWC. A parameter α can be used to more accurately scale the approximation, although its importance is limited to keep a conservative approximation. This approximation is easy to extend, either by eliminating trivial wrench dimensions, or by fitting some more expressive curve on the cone section (e.g., a superquadric). *Both approximations* carry a same message: in order to be useful inside an OCP solver, the constraint representation must be kept simple and smooth, as they are indeed the projection in the 6-D centroidal wrench space of some simple and smooth objects of larger dimension. Failing to keep the simplicity and smoothness of the representation means that we make the OCP problem artificially complex (and that, likely, it would be more efficient to solve it in some higher dimension).

Some effects are more difficult to capture at the centroidal level. The approach would not be very suitable for passive robots, or to finely capture the actuator dynamics. The angular momentum, mostly due to the movement of the off-contact limbs, is not properly decided at the centroidal level. A LPG is rather able to find a feasible trajectory, that must then be optimized at the whole-body level to get the full robot efficiency. On the other hand, the classic solvers used to solve the whole-body movement are not properly able to capture the subtle CoM effects when they rely on instantaneous linearization (e.g., Inverse Kinematics). Investigating how the LPG and the whole-body solver can interact, for example, by iterating several times on the same movement [21] is an interesting direction that we are currently implementing. We are also investigating how to solve the whole-body problem using a nonlinear trajectory optimizer (e.g., based on differential dynamic programming [61]), and how the LPG and whole-body OCP solver would interact. In general, the proposed LPG is a part of a complete framework [64], where a contact planner first decides the sequence of contacts [29], followed by the centroidal optimizer. Another planner, the “limb-RRT” then decides the trajectory of the feet in the free space [11]. The whole-body trajectory is finally obtained by task-space inverse dynamics [65]. All the elements of this cascade have been designed to allow fast motion generation, and could be run online during the robot motion.

The next step for us is to bring this methodology for controlling the robot. Concerning the LPG, the main issues are to connect it: in input with a centroidal state estimator [66]; and in output with a whole-body controller. The main stake is to show that the LPG endowed with a receding horizon scheme is able to control the unstable part of the centroidal dynamics. Results have already been obtained, but only for the LIPM version of the MPC [6]. We also believe that the MPC must be coupled with some torque-control capabilities, as it seems to be the only possible solution to handle the impact when creating a new contact, especially under uncertainties.

Finally, some technological issues prevent us to release our method as an open-source component. Our LPG relies on MUSCOD-II for solving the OCP, which is a proprietary software whose academic license is not easy to obtain. Behind this technological lock, the current numerical solvers used by the robotics community are not quite suitable for MPC of complex serial-chain robot. They mostly rely on a sequential quadratic programming strategy (like MUSCOD or SNOPT) whose underlying active-set QP solver is not satisfactory for MPC, or on Interior Point methods (like IPOPT, MOSEK, or GUROBI) whose log-barrier strategy is difficult to warm-start, hence, not suitable for MPC neither. We are currently working on implementing a numerical solver dedicated to robotics, upon which our LPG software will be released. It will also be the opportunity to explore in depth how the bilinear dynamics, resulting into a nonconvex numerical problem, is handled by convex optimization, and more particularly how our method would be complementary (or not) to the convex approximation proposed in [13] and [39].

VIII. Conclusion and Future Work

In this paper, we have proposed a complete solution to generate a centroidal trajectory for the locomotion of legged robots with multiple contacts. The method is able to produce efficient and smooth trajectories, suitable for their execution by the physical robot. Several elements make the method computationally efficient and able to cope with the complex constraints of the real robot. First, we proposed a clean transcription of the generic centroidal OCP into an efficient numerical problem, by using multiple shooting and by preserving important geometric properties (e.g., smoothness, simplicity) of the nominal problem. Second, we proposed an original approach to represent the whole-body constraints at the centroidal level, by using occupancy measures. This approach is suitable both for properly capturing the whole-body constraints despite their projection into the centroidal dynamics, and for its application in the numerical solver. Finally, we proposed to approximate the force constraint by a quadratic CWC, hence, simplifying the numerical problem while keeping the main part of the robot actuation capabilities.

We have demonstrated the validity of the method by showing that the generated movements can be executed by the real robot, and by statistically analyzing the behavior of the method on randomly generated debris environments. As our method consider by design the smoothness of the dynamics, the generated movements are suitable for their executions on the robot. The analysis on random environments showed that the proposed approximations are relevant for the classical locomotion scenarios. For more complex scenarios, we have proposed additional solutions or research directions to extend the proposed approximations while keeping their computational properties.

While we are still working on extending the proposed method (by adding the computation of the footstep position near the initial guess, improving the computation efficiency and the integration in the whole-body generator), our next step is to apply the OCP to control the real robot, using a receding horizon scheme. This also implies an important experimental work in centroidal perception and to control the robot by reference torques. We are also working on the numerical solver, in order to release an open source version of our method.

Acknowledgment

The authors would like to thank the anonymous reviewers for their fruitful comments and also the Simulation and Optimization Group of Heidelberg University for providing MUSCOD-II.

References

[1] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, “The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, pp. 239–246.

[2] S. Kajita *et al.*, “Biped walking pattern generation by using preview control of zero-moment point,” in *Proc. IEEE-RAS Int. Conf. Robot. Autom.*, 2003, pp. 1620–1626.

[3] P.-B. Wieber, “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2006, pp. 137–142.

[4] A. Sherikov, D. Dimitrov, and P.-B. Wieber, “Whole body motion controller with long-term balance constraints,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2014, pp. 444–450.

[5] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P.-B. Wieber, “A robust linear MPC approach to online generation of 3D biped walking motion,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2015, pp. 595–601.

[6] A. Herdt, N. Perrin, and P.-B. Wieber, “Walking without thinking about it,” in *Proc. IEEE/RSJ Int. Conf. Int. Robots Syst.*, 2010, pp. 190–195.

[7] H. Hirukawa *et al.*, “A universal stability criterion of the foot contact of legged robots-adios zmp,” in *Proc. IEEE-RAS Int. Conf. Robot. Autom.*, 2006, pp. 1976–1983.

[8] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2014, pp. 295–302.

[9] D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a humanoid robot,” *Auton. Robots*, vol. 35, pp. 161–176, 2013.

[10] M. Kudruss *et al.*, “Optimal control for whole-body motion generation using COM dynamics for predefined multi-contact configurations,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2015, pp. 684–689.

[11] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, “A versatile and efficient pattern generator for generalized legged locomotion,” in *Proc. IEEE-RAS Int. Conf. Robot. Autom.*, 2016, pp. 3555–3561.

[12] H. Dai and R. Tedrake, “Planning robust walking motion on uneven terrain via convex optimization,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2016, pp. 579–586.

[13] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, “A convex model of humanoid momentum dynamics for multi-contact motion generation,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2016, pp. 842–849.

[14] I. Mordatch, E. Todorov, and Z. Popović, “Discovery of complex behaviors through contact-invariant optimization,” *ACM Trans. Graph.*, vol. 31, no. 4, p. 43, 2012.

[15] P. Zaytsev, “Using controllability of simple models to generate maximally robust walking-robot controllers,” Ph.D. dissertation, Cornell Univ., Ithaca, NY, USA, 2015.

[16] S. Caron, Q.-C. Pham, and Y. Nakamura, “ZMP support areas for multicontact mobility under frictional constraints,” *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 67–80, Feb. 2017.

[17] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida, “Model preview control in multi-contact motion-application to a humanoid robot,” in *Proc. IEEE/RSJ Int. Conf. Int. Robots Syst.*, 2014, pp. 4030–4035.

[18] N. Perrin, D. Lau, and V. Padois, “Effective generation of dynamically balanced locomotion with multiple non-coplanar contacts,” in *Proc. Int. Symp. Robot. Res.*, 2015, pp 201–216.

[19] N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux, and E. Yoshida, “Fast humanoid robot collision-free footstep planning using swept volume approximations,” *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 427–439, Apr. 2012.

[20] A. Orthey and O. Stasse, “Towards reactive whole-body motion planning in cluttered environments by precomputing feasible motion spaces,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2013, pp. 274–279.

[21] A. Herzog, S. Schaal, and L. Righetti, “Structured contact force optimization for kino-dynamic motion generation,” in *Proc. IEEE/RSJ Int. Conf. Int. Robots Syst.*, pp. 2703–2710, 2016.

[22] K. Fukuda and A. Prodon, “Double description method revisited,” in *Proc. Conf. Comb. Comput. Sci.*, 1996, pp. 91–111.

[23] A. Escande, A. Kheddar, and S. Miossec, “Planning support contact-points for humanoid robots and experiments on HRP-2,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 2974–2979.

[24] O. Stasse *et al.*, “TALOS: A new humanoid research platform targeted for industrial applications,” in *Proc. IEEE-RAS 17th Int. Conf. Humanoid Robot.*, 2017, pp. 689–695.

[25] J. Carpentier, R. Budhiraja, and N. Mansard, “Learning feasibility constraints for multi-contact locomotion of legged robots,” in *Proc. Robot., Sci. Syst.*, 2017.

[26] T. Bretl, “Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem,” *Int. J. Robot. Res.*, vol. 25, pp. 317–342, 2006.

[27] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, “Planning biped navigation strategies in complex environments,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2003.

[28] R. Deits and R. Tedrake, “Footstep planning on uneven terrain with mixed-integer convex optimization,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2014, pp. 279–286.

[29] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 586–601, Jun. 2018.

[30] J. Mirabel *et al.*, "HPP: A new software for constrained motion planning," in *Proc. IEEE/RSJ Int. Conf. Int. Robots Syst.*, 2016, pp. 383–389.

[31] P.-B. Wieber, "Holonomy and nonholonomy in the dynamics of articulated motion, " in, *Fast Motions in Biomechanics and Robotics*. New York, NY, USA: Springer, 2006.

[32] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2015, pp. 874–880.

[33] Z. Qiu, A. Escande, A. Micaelli, and T. Robert, "Human motions analysis and simulation based on a general criterion of stability," in *Proc. Int. Symp. Digit. Human Model.*, pp. 1–8, 2011.

[34] S. Caron, Q.-C. Pham, and Y. Nakamura, "Leveraging cone double description for multi-contact stability of humanoids with applications to statics and dynamics," *Robot., Sci. Syst.*, 2015.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[36] P.-B. Wieber, "Viability and predictive control for safe locomotion," in *Proc. IEEE/RSJ Int. Conf. Int. Robots Syst.*, 2008, pp. 1103–1108.

[37] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Adv. Robot.*, vol. 24, pp. 719–737, 2010.

[38] H. Hirukawa *et al.*, "A pattern generator of humanoid robots walking on a rough terrain," in *Proc. IEEE-RAS Int. Conf. Robot. Autom.*, 2007, pp. 2181–2187.

[39] C. Bishop, "Pattern recognition and machine learning," Jan. 2006. [Online]. Available: https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/

[40] A. Winkler, D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1560–1567, Jul. 2018.

[41] S. Caron and A. Kheddar, "Multi-contact walking pattern generation based on model preview control of 3D COM accelerations," in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2016, pp. 550–557.

[42] D. Serra, C. Brasseur, A. Sherikov, D. Dimitrov, and P.-B. Wieber, "A newton method with always feasible iterates for nonlinear model predictive control of walking in a multi-contact situation," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robot.*, 2016, pp. 932–937.

[43] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *Int. J. Robot. Res.*, vol. 32, pp. 1104–1119, 2013.

[44] J. Pitman, "Occupation measures for Markov chains," *Adv. Appl. Probab.*, vol. 9, pp. 69–86, 1977.

[45] J. B. Lasserre, D. Henrion, C. Prieur, and E. Trélat, "Nonlinear optimal control via occupation measures and LMI-relaxations," *SIAM J. Control Optim.*, vol. 47, no. 4, pp. 1643–1666, 2008.

[46] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.

[47] C. M. Bishop, "Pattern recognition and machine learning," 2006.

[48] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Statist. Soc., Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[49] J. Hershey and P. Olsen, "Approximating the Kullback Leibler divergence between gaussian mixture models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2007, pp. IV-317–IV-320.

[50] Y. Or and E. Rimon, "Characterization of frictional multi-legged equilibrium postures on uneven terrains," *Int. J. Robot. Res.*, vol. 36, pp. 105–128, 2017.

[51] Y. Tian, Q. Jin, and Z. Deng, "Quadratic optimization over a polyhedral cone," *J. Ind. Manage. Optim.*, vol. 12, pp. 269–283, 2016.

[52] F. John, "Extremum problems with inequalities as subsidiary conditions," in *Traces Emergence Nonlinear Programming*. Springer, 2014, pp. 197–215.

[53] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions Biomech. Robot.*New York, NY, USA: Springer, 2006.

[54] E. M. Gertz and S. J. Wright, "Object-oriented software for quadratic programming," *ACM Trans. Math. Softw.*, vol. 29, pp. 58–81, 2003.

[55] D. Leineweber, I. Bauer, H. G. Bock, and J. P. Schlöder, "An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: Theoretical aspects," *Comput. Chem. Eng.*, vol. 27, pp. 167–174, 2003.

[56] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Souères, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Trans. Robot.*, vol. 29, no. 2, pp. 346–362, Apr. 2013.

[57] K. Nishiwaki and S. Kagami, "High frequency walking pattern generation based on preview control of ZMP," in *Proc. IEEE-RAS Int. Conf. Robot. Autom.*, 2006, pp. 2667–2672.

[58] K. H. Koch, K. Mombaur, and P. Souères, "Optimization-based walking generation for humanoid robot," *IFAC Proc. Vols.*, vol. 45, pp. 498–504, 2012.

[59] D. Clever, M. Harant, K. Mombaur, M. Naveau, O. Stasse, and D. Endres, "Cocomopl: A novel approach for humanoid walking generation combining optimal control, movement primitives and learning and its transfer to the real robot HRP-2," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 977–984, Apr. 2017.

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

[61] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *Proc. IEEE-RAS Int. Conf. Robot. Autom.*, 2014.

[62] P. Gill, W. Murray, and M. Saunders, "User's guide for qpopt 1.0: A fortran package for quadratic programming," *Tech. Rep.*, 1995.

[63] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. Caldwell, and C. Semini, "Application of wrench based geasibility analysis to the online trajectory optimization of legged robots," *IEEE Robt. Autom. Lett.*, 2018.

[64] J. Carpentier *et al.*, "Multi-contact locomotion of legged robots in complex environments — the loco3d project," in *Proc. RSS Workshop Challenges Dyn. Legged Locomotion*, 2017.

[65] A. Del Prete, F. Nori, G. Metta, and L. Natale, "Prioritized motion-force control of constrained fully-actuated robots: Task space inverse dynamics," *Robot. Auton. Syst.*, vol. 63, pp. 150–157, 2015.

[66] J. Carpentier, M. Benallegue, N. Mansard, and J.-P. Laumond, "Center-of-mass estimation for a polyarticulated system in contact—A spectral approach," *Trans. Robot.*, vol. 32, no. 4, pp. 810–822, 2016.

Justin Carpentier received the M.Sc. degree in applied mathematics and computer science from École Normale Supérieure Paris-Sacaly, France, in 2013. In 2014, he was a visiting student with the Optimization, Robotics and Biomechanics Group, University of Heidelberg, Germany. He received the Ph.D. degree in robotics from the University of Toulouse, Toulouse, France, in 2017.

Since 2017, he has been a Postdoctorate Researcher with the Gepetto team at the Laboratory for Analysis and Architecture of Systems, Toulouse, France. His research interests include the motion generation for humanoid robots, bipedal locomotion and numerical optimal control.

Nicolas Mansard received the M.Sc. degree in computer science from the University of Grenoble, Grenoble, France, in 2003 and the Ph.D. degree in robotics from the University of Rennes, Rennes, France, in 2006.

He has been a CNRS Researcher ("*chargé de recherche CNRS*") since 2009. He was then Postdoctoral Researcher at Stanford University, Stanford, CA, USA with O. Khatib in 2007 and in JRL-Japan with A. Kheddar in 2008. He was Invited Researcher at the University of Washington with E. Todorov in 2014. He received the CNRS Bronze Medal in 2015 (one medal is award in France in automatic/robotic/signal-processing every year). His main research interests include the motion generation, 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 published more than 70 papers in international journals and conferences and supervised 10 Ph.D. thesis.

Dr. Mansard is currently an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS.