Multi-contact Locomotion of Legged Robots
Justin Carpentier, Nicolas Mansard

To cite this version:

HAL Id: hal-01520248
https://hal.laas.fr/hal-01520248
Submitted on 10 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - ShareAlike International License
Multi-contact Locomotion of Legged Robots

Justin Carpentier, Student Member, IEEE, 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, like 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 multi-contact 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 whole body as occupation measures, and a systematic way to approximate them using off-line 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 is demonstrated in several multi-contact scenarios with two humanoid robots both in reality and in simulation.

Index Terms—Humanoid Robots, Legged Robots, Multi-contact Locomotion, Optimal Control, Machine Learning

I. INTRODUCTION

Legged robots are under-actuated 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 which then avoids slippage and falls (see Fig. 1). Maintaining these forces deep inside the cones is one of the main tasks of the locomotion pattern generator (LPG).

In its generic form, a LPG deals with a high-dimensional and complex optimal control problem (OCP), seeking both for 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. 10Hz or more). To tackle the computational complexity, many strategies have been proposed in the literature. Most of them are based upon reduced models: instead of working with the full dynamics, only a subpart is considered, covering the essential properties of the whole dynamics.

1) Reduced models: In the context of bipedal locomotion, the most famous reduced model is the linear inverted pendulum model (LIPM) [3]. 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. Starting with [4], various optimal control formulations have been proposed by the community, to either tackle the robustness problem [5], include viability conditions [6], allow altitude variations of the center of mass (CoM) [7], or also include foot placements as parameters of the problem [8].

However, LIPM-based methods are restricted to basic environments and cannot deal with more complex scenarios as non-coplanar contact cases, climbing stairs using handrail, etc. Considering non-coplanar contacts invalidates the nice linearization leading to the LIPM model. A first approach to handle the non-linear dynamics was proposed in [9], however it requires technical and dedicated developments based on limiting assumptions (e.g. prior knowledge of the force distribution). In quite another vein, it has been proposed to simplify the whole-body optimization problem by e.g. assuming unconstrained torque capabilities [10]. Both approaches indeed boil down to optimizing the so-called centroidal dynamics [11] as a reduced model. Direct resolution of the underlying optimal control problem is then possible [1], [12](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 [7], [13], [14]. In most cases, the footstep sequence is assumed given, although some solvers are also able to discover it while optimizing the centroidal dynamics [15], [16], at the cost of heavier computational times.

2) 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). 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 a solely function of the reduced model. These constraints can be tackled explicitly, by adding the corresponding whole-body variable in the optimization scheme \cite{10, 15}. 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 so-called proxy constraints \cite{17}. In most previous works, proxy constraints are defined by some rough approximations \cite{14, 18, 19} (box constraints, elliptic bounds, etc) leading to a certain conservatism when they are not simply ignored in many formulations \cite{20, 21}. For example, footstep limits have been encoded using hyper-planes based on a dataset of robot success and failure inside a dynamic simulator \cite{22}. Similar constraints can be obtained by training a neural network \cite{23}. In \cite{17}, bounds of the captureability regions are obtained by extensive computations of the viability set of reduced models.

An important constraint limits the motion of the CoM reflecting the kinematics bounds of the whole-body. It is also necessary to consider the constraints related to the contact forces \cite{24} which must lie inside the friction cones, the capacity of robots to generate sufficient variations of angular momentum, etc. The common issues lies in the fact that it is hard to find analytic formulas to represent and express these constraints.

3) Organization: In this paper, we introduce a complete formulation of LPG able to cope with multiple non-flat contact, footstep timings and whole-body "proxy" constraints, with a generic and versatile approach, tractable at robot control rate (from 20 to 100Hz). Our solution is based on a generic optimal control formulation presented in Sec. 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 occupation 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 Sec. II). 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 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 sub-cases of this generic OCP. Although this section contains known materials, we believe that both the clean formulation of the dynamics decoupling and the development of a generic formulation for the multi-contact 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 \( \{p_k \in \mathbb{R}^3, k = 1, \ldots, K\} \). 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 \( p_k \) is defined a contact force \( f_k \). In the case of unilateral contacts, \( f_k \) must lie inside a 3-dimensional friction cone \( K_k \) (also denoted quadratic “ice-cream” cone) characterized by a positive friction coefficient \( \mu_k \). Fig. 4 depicts a humanoid robot making contact with its environment.

In this work, we only consider rigid contact interaction which is a reasonable assumption for most modern multi-legged robots which 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 [28], efficient algorithms have been proposed by the motion planning community either to plan only for footed robot [16], [29] or more generically for any kind of multiped robots [26], [30]. In our current approach, we use the open source and efficient implementation of [30] proposed in [31] to compute in real time a feasible contact sequence inside complex environments.

B. Whole-body dynamics and centroidal dynamics

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

$$
\begin{bmatrix}
H_u \\
H_a
\end{bmatrix} \ddot{q} + \begin{bmatrix}
0_b \\
0_a
\end{bmatrix} + \begin{bmatrix}
g_u \\
g_a
\end{bmatrix} + \begin{bmatrix}
0_b \\
0_a
\end{bmatrix} \tau_a + \sum_{k=1}^{K} J_{k,u}^T f_k
$$

(1)

where subscripts $u$ and $a$ stands for the under-actuated and the actuated parts respectively, $H$ is the generalized mass matrix, $b$ covers the centrifugal and Coriolis effects, $g$ covers the gravito-inertial forces. Denoting by $h$ the cross product operator.

On one side, the 6 first rows of (1) corresponds to the under-actuated dynamics of the robot, also called the centroidal dynamics [11]. 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 $h \overset{\text{def}}{=} m\dot{c}$ the linear momentum ($m$ being the total mass of the robot and $c$ the CoM position), and $L_c$ the angular momentum, the 6 first rows of (1) can be simply written as:

$$
\begin{aligned}
\dot{h} &= \sum_{k=1}^{K} f_k + mg \\
\dot{L}_{c} &= \sum_{k=1}^{K} (p_k - c) \times f_k,
\end{aligned}
$$

(2a, 2b)

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

On the other side, the $n$ last rows of (1) are the classic Lagrange dynamics of a robot manipulator in contact.

C. Hierarchical decoupling between centroidal and manipulator dynamics

From a phenomenological point of view, (1) reads as follows: when supplying a certain amount of joint torque $\tau_a$, the environment reacts by producing the contact forces $f_k$. Those very same forces act on the centroidal dynamics to enable the robot to move inside the environment.

Under the mild 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 throw the manipulator trajectory, knowing the centroidal trajectory and under the hypothesis of non sliding contacts. In other words, the torque may be seen as a slack variable.

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 kinematics;

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

$$
\begin{bmatrix}
h \\
L_c
\end{bmatrix} = A_q (q) \dot{q}
$$

(3)

with $A_q$ the so-called centroidal momentum matrix (CMM) [11].

In the rest of the paper, we reduce the whole-body dynamics to its centroidal dynamics.

D. State and control of the centroidal dynamics

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

$$
\begin{aligned}
m(\ddot{c} - g) &= \sum_{k=1}^{K} f_k \\
\dot{L}_{c} + mc \times (\ddot{c} - g) &= \sum_{k=1}^{K} p_k \times f_k,
\end{aligned}
$$

(4a, 4b)

where $f_k \overset{\text{def}}{=} (f_{k, k = 1, \ldots, K})$ with $f_k \in K_6^3$. One drawback of this formulation is that the control input grows linearly with the number of contacts. To overcome that, one can write (4a) by condensing all the forces and torques with a single control input $u_c \overset{\text{def}}{=} (f_c, \tau_c)$ such that the centroidal dynamics reads:

$$
\begin{aligned}
m(\ddot{c} - g) &= f_c \\
\dot{L}_{c} &= \tau_c - c \times f_c,
\end{aligned}
$$

(5a, 5b)

with $f_c \overset{\text{def}}{=} \sum_{k=1}^{K} f_k$ and $\tau_c \overset{\text{def}}{=} \sum_{k=1}^{K} p_k \times f_k$, $u_c$ being the gravito-inertial 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 dimensional constraint:

$$
(f_c, \tau_c) \in K_c^6
$$

(6)

where

$$
K_c^6 \overset{\text{def}}{=} \bigoplus_{k=1}^{K} K_c^3 = \left\{ \sum_{k=1}^{K} (f_k, p_k \times f_k) \mid f_k \in K_c^3 \right\}
$$

(7)

being the Minkowski sum of the contact cones translated by the contact point positions. This cone is named the centroidal wrench cone (CWC) [9].

At this stage, several observations come:

1° Torque bounds can later be treated as a proxy constraint following the approach that we introduce in Sec. III.
are the estimated by the sensor of the real robot) while (9f) enforces with proxy constraints in an automatic way. Constraint (9e) quantity, etc. We will show in Section III how it can be handled on some heuristics to strip away the ambiguity on the force driving either by the force input \( u \) or (5) and \( u \) indifferently represents \( u_f \) or \( u_c \), and must belong to the corresponding set denoted by \( K \).

We are now able to describe the generic problem of locomotion which can merely be stated 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 transcribed as an optimal control problem with path and terminal constraints:

\[
\begin{align*}
\min_{\hat{x}, u} \quad & \sum_{s=1}^{S} \int_{t_s}^{t_{s+1}} \ell_s(x, u) \, dt \\
\text{s.t.} \quad & \forall t \quad \dot{x} = f(x, u) \quad (9b) \\
& \forall t \quad u \in K \\
& \exists (q, \dot{q}, \ddot{q}) \text{ s.t. } x, \dot{x} \text{ is feasible} \quad (9c) \\
& x(0) = x_0 \quad (9d) \\
& x(T) \in X_s \quad (9f)
\end{align*}
\]

where \( s \) is the index of the contact phase, \( x \) and \( u \) are the state and control trajectories, \( t_s \) 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 consistent dynamics with respect to the contact model. Eq. (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 will show in Section III how it can be handled with proxy constraints in an automatic way. Constraint (9c) constrains the trajectory to start with a given state (typically estimated by the sensor of the real robot) while (9f) enforces a viable terminal state \( x(T) \). Finally, \( \ell_s \) is the cost function typically decoupled in \( \ell_s(x) + \ell_u(u) \) whose parameters may vary according to the phase. \( \ell_s \) is generally used to smooth the state trajectory while \( \ell_u \) tends regularize the control. The resulting control is stable as soon as \( \ell_s \) involves the \( L_2 \) norm of one of the time derivatives of \( c \) \( [34] \).

### F. Previous formulations

In the following, we present a state-of-the-art the main LPG formulations present in the literature. In particular, we detail how those LPGs correspond to specific choices of the generic formulation \( [9] \).

#### 1) Walking patterns in 2D: 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 \( [4] \), by neglecting the dynamics of both the CoM altitude and the angular momentum. In that case, \( K \) boils down to the constraint of the zero-momentum point to lie in the support polygon.

Kajita et al. \( [4] \) did not explicitly check the constraint \( (9c) \); in exchange, \( \ell_u \) is used to keep the control trajectory close to a reference trajectory provided a priori. Similarly, \( [37] \) is not checked; in exchange, \( \ell_s \) tends to stabilize the robot at the end of the trajectory by minimizing the jerk of the CoM. These three simplifications turns \( (9) \) into a simple unconstrained problem of linear-quadratic regulation (LQR) that is implicitly solved by integrating the corresponding Riccati equation.

The Kajita’s LQR was reformulated into an explicit OCP \( [35] \), directly solved as quadratic program. The OCP formulation makes it possible to explicitly handle inequality constraints: \( (9c) \) is then explicitly checked under its ZMP reformulation. A modification of this OCP is proposed in \( [6] \) where \( (9c) \) is nicely approximated by the capturability constraint, which constrains the CoM position and velocity in the context of coplanar contacts.

In \( [21] \), \( L_c \) 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, like in \( [4] \). These assumptions result in a (bilinear)-constrained quadratic program that is solved by a dedicated numerical method.

In \( [37] \), \( (9c) \) is explicitly handled (using the classic linear approximation of the quadratic cones). As in \( [4] \), \( (9d) \) is indirectly handled by minimizing the jerk. No condition \( (9d) \) 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 \cite{12}, the conic constraint is directly handled. The angular momentum is approximated through the system \( L_c \approx \dot{I} \omega + t_{Le} \), with \( \dot{I} \) the compound (rigid) inertia of the robot and \( t_{Le} \), the angular momentum due to the internal gesticulation. \( \dot{I} \omega \) is kept low by penalizing the large rotation \( \omega \) but \( t_{Le} \) is unlimited, resulting in \( \gamma \) not being checked. The viability \( \gamma \) is not checked neither, but like 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 a 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 \cite{38}, the authors work only with the CoM acceleration and neglect the contribution of the angular momentum quantity setting it to 0 as in \cite{4}. They approximate the Minkowski sum of contact cones \( K_c \) with a conservative linear approximation following the method proposed in \cite{37}. 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 \cite{14}, the authors do not directly consider the angular momentum quantity but instead, they chose to minimize an upper bound of its \( L_1 \) norm. Similar to \cite{38}, they consider a linear approximation of \( K_c \) 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 \cite{39}, the authors propose an efficient OCP formulation to compute 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 \cite{4} corresponds to a generic formulation of the problem, but contains several terms that are difficult, complex or impossible to make explicit: whole body constraints, angular momentum set, viability set. The stake is now twofold: we need to decide (i) how to represent these functions and (ii) how to solve the OCP.

(i) Representing the constraint functions implies a trade off between accuracy of the model and efficiency of the resolution. In the following sections, we propose original contributions to formulate approximate proxy constraints representing the whole-body limits with a generic offline learning approach (Sec. III). We also propose an efficient approximation of the contact constraints then allowing the formulation of the OCP with the reduce variable \( u_c \) (Sec. IV). Both constraints could be used in any OCP, for example directly applying to \cite{1}, \cite{12}, \cite{20}, \cite{40}, \cite{41}.

(ii) We then propose to solve the resulting OCP using a multiple shooting solver, then enabling efficient and reliable implementation on the robot (Sec. V).

III. LEARNING FEASIBILITY CONSTRAINTS OF THE CENTROIDAL PROBLEM

In this section, we first present a mathematical coding 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 \( \gamma \) in the OCP. We then present a complete solution to efficiently approximate the CoM feasibility. 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 the most limiting constraint. Generalization to velocity and acceleration of the CoM with respect to joint velocity and acceleration limits would be straight-forward. 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 \( \gamma \) in our OCP. This constraint explicitly depends on the robot configuration, which is not a variable of the centroidal OCP. A straight-forward implementation is to add the robot configuration in the variables of the OCP \cite{10}. 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 by 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 \( \gamma \) the centroidal projection function:

\[
\gamma : (q, \dot{q}, \ddot{q}) \rightarrow (x, \dot{x}) = \gamma(q, \dot{q}, \ddot{q})
\]

the proxy can be written as the constraint to have the state variables in the range space of \( \gamma \). Set-membership proxies are used for instance in \cite{8}, \cite{16} to encode maximal step size in biped walking, or in \cite{14} 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 geometric shapes) in order not to burden the OCP solver. Remarkably, there are few papers about the automatic synthesis of the set boundaries \cite{17}, \cite{22}, \cite{23}.

Despite its popularity, the set-membership representation has important drawbacks. First, it is often difficult to handle by 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 \( \gamma^{-1} \) 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 \( x, \dot{x} \) \([42, 43]\). In its generic form, given a set \( A \subset \mathbb{R}^n \), a time interval \( I \subset \mathbb{R} \) and a trajectory \( \tilde{\gamma} : I \rightarrow \mathbb{R}^n \), the occupation measure \( \mu \) of the trajectory \( \tilde{\gamma} \) on \( A \) is defined as:

\[
\mu(A) \equiv \int_I 1_A(s(t))dt
\]  

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

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

We defined the centroidal occupation measure as the image of the uniform distribution in configuration space through the centroidal projection \( \gamma \):

\[
\mu_o(\tilde{x}) \equiv \int_{\tilde{q} \in \tilde{Q} \ s.t. \ \gamma(\tilde{q}) = \tilde{x}} d\tilde{q} = \int_{\tilde{Q}} 1_{\gamma(\tilde{q}) = \tilde{x}} \tilde{q} d\mu_{\tilde{Q}}
\]

where \( \tilde{x} \equiv (x, \dot{x}) \), \( \tilde{q} \equiv (q, \dot{q}) \), \( \tilde{Q} \) is the whole-body motion range and \( \mu_{\tilde{Q}} \) is the uniform distribution on \( \tilde{Q} \).

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

In practice, it is desirable that OCP \([9]\) promotes centroidal states \( \tilde{x} \) where \( \mu_o \) is the highest. First, it makes it easier to then compute a corresponding configuration \( \tilde{q} \). 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 \( \mu_o \) for the specific case of the kinematic feasibility, we quickly show how \( \mu_o \) 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 \( p(x, \dot{x}) \). 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 neighbourhood 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 neighbourhood 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 \( \ell_o \) is the composition of two terms: the previous cost function \( \ell_s \) which regularize the dynamics, plus the log-PDF of the feasibility constraints, leading to:

\[
\ell_o(x, u) = \ell_s(x, u) - \log(p(x, \dot{x}))
\]

In practice, the logarithm prevents the solver from selecting non-feasible states \( x \) and controls \( u \) through the dynamics equation \( \dot{x} = f(x, u) \). Constraints \([9d]\) is always satisfied. It also penalizes non-robust behavior where no redundancy \( q \) 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 \( \mu_o \), as it manipulates a full trajectory \( \tilde{x} \) and not a single state \( x \). 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 \( q(t) \) such that (i) the contact placements are respected and (ii) the CoM of the poly-articulated system matches \( p \).

1) **Probabilistic model**: The geometric condition can be stated as the conditional probability of the CoM to be at the position \( c \) given the current set of \( K \) contact points \( \{p_k \in \mathbb{R}^3, k = 1...K\} \). This probability is denoted by \( p(c|p_k, k = 1...K) \). It lives in the high dimensionality domain \( \mathbb{R}^{3(K+1)} \) and it is hard to compute in general.

The probability domain can be exactly reduced by gathering together the contact points belonging to the same rigid end-effector (e.g., the 4 vertices of the humanoid foot belongs to the same end-effector). We denote by \( M_i = (R_i, p_i) \in SE(3) \) the placement (position and orientation) of the contact body \( i \). The conditional probability is then reduced to \( p(c|M_i, i = 1...K_c) \) where \( K_c \) is the number of end-effectors in contact.

We now assume that variables \( M_i \) are all independent. This assumption is clearly abusive, however is a reasonable approximation under knowledge of \( c \). It is later discussed. Under this assumption, the conditional probability reads:

\[
p(c|M_i, i = 1...K_c) \propto \prod_{i=1}^{K_c} p_i(c)
\]

where \( p_i(c) \) stands for \( p(c|M_i) \) and \( \propto \) stands for “is proportional to”. \( p(c|M) \) is nothing more than the probability
distribution of the CoM to be at position \( c \) w.r.t the frame defined by \( M_i \).

The assumption of independence of the \( M_i \) is commonly employed 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 \( p_i(c) \) 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 quite reasonable in practice and leads to good experimental results.

2) Kernel density estimation by CoM sampling: There is in general no closed form to encode \( p_i(c) \) for a particular legged robot. Nevertheless, this conditional probability can be easily approximated by extensive sampling of the CoM position expressed in the end-effector frames.

Sampling \( N_{\text{samples}} \) of the CoM position expressed in the frame \( M_i \) does not raise particular difficulties. For each sample, a configuration \( q_a \) of the actuated joints is randomly sampled and the CoM position is computed (expressed in placement frame) by forward kinematics. The sample is rejected if joint limits or self collision are violated.

The probability distribution can be approximation from the cloud of CoM points by the kernel density estimators (KDE) \([44]\). KDE are in some sense the analogues 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 the present work, 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. So, roughly speaking, evaluating the PDF of the KDE takes approximately 10, \( N_{\text{samples}} \) ns which becomes rapidly a bottleneck when the number of points is huge (\( N_{\text{samples}} \) greater than 100 points).

We propose to then approximate the KDE by a Gaussian mixture model (GMM) \([45]\). 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 \([46]\).

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 \([47]\). Depending on the number of Gaussians in the mixture, the divergence can reveal under or over fitting effects. The optimal number of Gaussians is easily selected for each end effector by dichotomy, as exemplified in next section.

4) Summary of the learning procedure: In summary, for each end effector, \( N_{\text{samples}} \) configurations are sampled 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 \( p_i(c) \), for \( i \) the end effector in contact with the environment. The OCP cost function is then given by:

\[
\ell_s(\mathbf{x}, \mathbf{u}) = \ell_s(\mathbf{x}, \mathbf{u}) - \sum_{i=1}^{K_s} \log(p_i(c))
\]  

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 will be completed by integration test in the complete LPG in the last section of the paper. For that purpose, with 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 right foot (RF). A similar study can be conducted on the three other end-effectors.

The learning process is made from a set of 20000 points sampled uniformly in the configuration space. The KDE of this set is represented on the first row of Fig. 2. 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. 2 represents the colour 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.
Fig. 3. Evolution of the KL divergence between the KDE distribution and GMMs of different sizes for the four end-effectors of the HRP-2 robot.

Fig. 4. Illustration of the probability density distribution of the CoM w.r.t. the right foot frame of TALOS, projected along the three axis X,Y,Z. The first row corresponds to the ground truth distribution estimated through KDE (20000 points). The second row depicts the learned GMM with 4 Gaussian kernels in the mixture. The axes have the same scale than in Fig 3.

However, this approximation is conservative with respect to the support and the level sets of the original distribution.

Fig. 5 shows the results of this validation protocol for phases with two, three and four contacts. First, the CoM reachability volume decreases with the number of contacts for both real and approximated distributions, which is expected: with more contacts, less degrees of freedom are available to freely move the CoM. Second, it appears that in all scenarios, our approximations of the CoM distributions cover a larger region than the real distributions. However, this is not a limitation as our optimal control formulation tends to move the CoM toward the highest probability regions which coincide with the real distributions.

IV. CENTROIDAL WRENCH CONE APPROXIMATION

As mentioned in Section II-D, the linear and angular momentum variations must lies in the CWC, which is defined by (7) as the Minkowski sum of Lorentz cones. In general, there is no analytically formulation of such Minkowski sum. As classical approximations of the CWC involves many linear inequalities that are hardly tractable by the OCP, we propose here a more efficient approximation by a single quadratic inequality.

A. State of the art

A first attempt has been proposed to compute analytically the supporting area in the context of static equilibrium from Lorentz contact cones [48]. Nevertheless, this method is limited to very specific cases called “tame stances”. To handle any scenario, it has been suggested to compute a linear version...
forces and the friction coefficient. With a more geometric view, for example, for the 3D Coulomb cone described in (all the developments apply in any dimension larger than 3). In its generic form, a Lorentz cone is easily represented by its normal direction \( d \). The projection is then \( \mathcal{P}_\Pi(y) = y - (y^\top d) d \). The ellipsoid \( E^5 \) can be represented by its center \( b \in \Pi \) and a symmetric positive (SDP) matrix \( Q \) (\( E^5 \) is the spectral ellipsoid of \( Q \)): 
\[
\mathcal{K}^6 = \{ y \in \mathbb{R}^6, \| y - (y^\top d) d - b \|_Q \leq y^\top d \} 
\]  
(16)

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

Our goal is to find the best outer Lorentz approximation \( \mathcal{K}_{\Pi}^6 \) of the CWC \( \mathcal{K}_6 \) using the generic form (16), i.e. to find the direction \( d \) and SDP matrix \( Q \) such that \( \mathcal{K}_{\Pi}^6 \subseteq \mathcal{K}_6^6 \) and \( \mathcal{K}_{\Pi}^6 \) is minimal (the center \( b \) being null at the optimum). This is equivalent to minimize the spectral radius of \( Q \) so that a sufficiently-large family of rays of \( \mathcal{K}_6^6 \) are inside the resulting outer approximation. This statement can be translated into the following optimization problem:
\[
\begin{align*}
\min_{Q \succeq 0, d \in \mathbb{R}^6} & \quad \det(Q) \\
\text{s.t.} & \quad \lambda_i \in \mathcal{K}_6^6(Q, d), i = 1, \ldots, N \\
& \quad \|d\| = 1 \\
& \quad d^\top Q d = 1 
\end{align*}
\]  
(17a)

where \((\lambda_i)_{i=1,N}\) is a family of rays of \( \mathcal{K}_6^6 \) (typically obtained by concatenation of regular rays of the 3D contact cones \( \mathcal{K}_{K_i}^3 \)). The cost (17a) induces the minimization of the area of the section, with (17b) 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 \( \mathcal{K}_6^6 \) parametrized by \( Q_o \) and \( d_o \). 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 [37] (i.e. 4 rays per contact cone). Here we have the advantage that the complexity of problem (17) typically

3This family of rays span a linear approximation of \( \mathcal{K}_6^6 \) which is typically handled by the double-description approach [37].

of the CWC by replacing contact cones with their linear approximations [37]. The set-membership constraint (6) is then reduced to a set of linear inequalities thanks to the double-description property of linear cones [25].

Most of current approaches now rely on the double-description of the CWC [25]. Yet, the calculus of the double-description is numerically unstable for 3 contacts and more [38]. In addition, the implicit description leads to high number of inequalities (about 50 inequalities with 2 contacts, more than 100 with 3 contacts) which depends on the contact placements, thereby increasing the dimensionality of the global problem.

On the contrary, we propose a conic approximation composed of a single quadratic inequality, no matter the number of contacts is. This approximation is composed of an outer approximation of the CWC which enables us to obtain a robust inner approximation. We first detail a systematic procedure to compute an outer (optimistic) approximation of the CWC with a Lorentz cone. This approximation is possible under a mild assumption on the contact point positions. Based on this outer approximation, we then easily deduce an inner Lorentz cone of the CWC using theoretical properties of the CWC. This inner (approximate) approximation is then used in (9) as conic constraint on the control \( u_c \).

**B. Outer approximation**

To keep the description of the method simple, we directly work in the 6-dimension space \( \mathbb{R}^6 \) (all the developments apply in any dimension larger than 3). In its generic form, a Lorentz cone is classically defined as:
\[
\mathcal{K}^6 \overset{\text{def}}{=} \{ y = (\tau, \eta) \in \mathbb{R}^6, \| \tau \|_2 \leq \eta \} 
\]  
(14)

For example, for the 3D Coulomb cone described in Section [11-A], the component \( \eta \) corresponds to the tangential forces and the \( \tau \) variable is the normal force scaled by the friction coefficient. With a more geometric view, \( \mathcal{K}^6 \) can rather be represented by a hyper-plane \( \Pi \) intersecting the cone (the so-called conic section) and a 5-dimension ellipsoid \( E^5 \) in this hyper-plane:
\[
\mathcal{K}^6 = \{ y \in \mathbb{R}^6, \mathcal{P}_\Pi(y) \in E^5 \} 
\]  
(15)

with \( \mathcal{P}_\Pi(y) \) the normal projection of \( y \) in \( \Pi \). The conic section \( \Pi \) is easily represented by its normal direction \( d \). The projection is then \( \mathcal{P}_\Pi(y) = y - (y^\top d) d \). The ellipsoid \( E^5 \) can be represented by its center \( b \in \Pi \) and a symmetric positive (SDP) matrix \( Q \) (\( E^5 \) is the spectral ellipsoid of \( Q \)):
scales linearly with the number of rays while it induces a combinatorial when using the double description.

Nevertheless, (17) is hard to solve in its own. 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. 6: (i) we first find a suboptimal direction \( d \), (ii) then a suboptimal (noncentered) ellipsoid \( E^5 = (Q, b) \); (iii) this ellipsoid is used to compute the optimal direction \( d \) where the ellipsoid would be centered; (iv) the optimal ellipsoid is then obtained by optimizing the sole matrix \( Q \).

(i) Choosing a initial direction \( d \): We can chose the normal direction \( n \) by only considering the family of rays \( (\lambda_i) \) (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 \( \lambda_i \) 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 hyper-plane \( \Pi \) will not properly intersect the cone (i.e. the intersection of \( K^6_\alpha \) and \( \Pi \) is not an ellipsoid) [49]. We then search \( d \) 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):

\[
\min_{d \in \mathbb{R}^n} \frac{1}{2} \sum_i \|d - \lambda_i\|^2 \quad \text{s.t.} \quad \Lambda^T d > 0
\]

where \( \Lambda \) is the matrix where columns are the rays \( \lambda_k \) (see Fig. 6a).

(ii) Computing an outer ellipsoid on the plane: Any hyper-plane \( \Pi = \{ x \in \mathbb{R}^n, n^T x = \beta \} \) with \( \beta > 0 \) can be considered (we typically take \( \beta = 1 \)). The intersection of the rays \( (\lambda_i) \), with \( \Pi \) defines a family of points \( (p_i) = (\beta \lambda_i) \). The convex hull of \( (p_i) \) is the intersection of the linear inner approximation with \( \Pi \). We search \( E^5 \) as the minimum-volume ellipsoid that encloses the set of points \( (p_i) \), also called the Löwner-John ellipsoid [32], represented by its center \( b \) and spectral matrix \( Q \). The pair of parameters is obtained by the following second-order conic program (SOCP):

\[
\begin{align*}
\min_{b \in \mathbb{R}^n, Q \in \mathbb{R}^{n \times n}} & \ \det Q \\
\text{s.t.} & \ \enclose{circle}{Q} \succeq 0 \\
& \forall i = 1..N, \ \|Qp_i - b\| \leq 1
\end{align*}
\]

(iii) Choosing the optimal direction: As previously explained, the minimal outer approximation is found when the ellipsoid \( E^5 \) is centered on direction \( d \). We can directly obtained the optimal direction by considering the antipodal points of the initial ellipsoid \( E^5 \) (the opposite points on the ellipsoid corresponding to Eigen directions of \( Q \)). Consider the bisecting planes \( B_i \) defined from antipodal points \( (i = 1..5) \), see Fig. 6c, i.e. \( B_i \) is the hyper-plane containing the center of the cone and for which the pair of antipodal points are reflections. Then the optimal direction \( d^* \) is defined by the intersections of the 5 hyper-planes \( B_i \).

(iv) Computing the optimal ellipsoid: Finally, the minimal section is computed with the same SOCP [19]. We first define the intersecting plane \( \phi \) with normal \( d^* \) and level value \( \beta = 1 \). Then, we project the rays onto this plane and compute the minimal ellipsoid \( E^* \) defined by \( Q^* \) and \( b = 0 \) enclosing those projected points by imposing its center to be zero in the plane frame (see Fig. 6d).

C. Inner approximation

An inner ellipsoid can be directly obtained for the minimal outer ellipsoid with the guaranty to be strictly inside the convex-hull. For that, it is sufficient to divide the ellipsoid by 5 (the hyper-plane dimension). In addition, if the convex-hull is symmetric with respect to the center of the ellipsoid, it can be simply reduced by a factor \( \sqrt{5} \) [32]. Using this property, we obtain an inner approximation of \( K^6_\alpha \), denoted \( K^6_\alpha \) and having the same direction than \( K^6_\alpha \).

The proposed approximation is guaranty to strictly lie inside the CWC by construction. While it may lead to a certain conservatism –less centroidal wrench variations are allowed– the proposed approximation can also 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 \( \alpha \) can be chosen in the interval \([\frac{1}{5}, \frac{1}{\sqrt{5}}]\). Using the lower bound leads to theoretical guaranty, while choosing a greater value allows to exploit geometric properties of the contact set, like symmetries. Theoretically, the factor \( \alpha \) can be adjusted on the flight by a quick dichotomy in the range \([\frac{1}{5}, \frac{1}{\sqrt{5}}]\). In practice, we will see in the following that choosing an arbitrary fixed \( \alpha \) leads to effective results on the real robot.

D. Validation of the centroidal cone approximation

We now empirically validate this inner approximation with respect to both the real CWC and the linearized version of the CWC on the scenarios already used in Sec. III-C2. Given a contact configuration of the robot (i.e. contact placements and COM position), we uniformly sample values of the centroidal contact set, like symmetries. Theoretically, the factor \( \alpha \) can be chosen in the interval \([\frac{1}{5}, \frac{1}{\sqrt{5}}]\). In practice, we will see in the following that choosing an arbitrary fixed \( \alpha \) leads to effective results on the real robot.

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 experiment 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 4 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 \( \alpha \) 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 2D section (while it does in the 6D space).

A side result is obtained from comparing the cone resulting of 2 contacts to the cone resulting of 3 contacts. The CWC remarkably grows with the addition of a new contact. This goes in favor of multi-contact locomotion: adding contacts enable the robot to increase its dynamics capabilities while constraining more its kinematics.

V. Final formulation of the optimal control problem

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

A. Tailored optimal control problem

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

$$\min_{\mathbf{x}, \mathbf{u}} \sum_{s=1}^{S} \int_{t_s}^{t_s + \Delta t_s} \ell_s(\mathbf{x}, \mathbf{u}) + \sum_{i} \log(p_i(\mathbf{x})) \, dt$$

s.t. \( \forall t \) \( \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \) \quad (20b)

\( \forall t \) \( \mathbf{u} \in \mathcal{K} \) \quad (20c)

\( \mathbf{x}(0) = \mathbf{x}_0 \) \quad (20d)

\( \mathbf{x}(T) = (c_f, 0, 0), \dot{\mathbf{x}}(T) = 0 \) \quad (20e)

where the feasibility constraints (9d) is replaced by the additional log likelihood sum in the cost function, as explained in Sec. III. The control variable \( \mathbf{u} \) can be either \( \mathbf{u}_f \) (the contact forces with the ice-cream cone constraints for (20c)) or as \( \mathbf{u}_c \) (centroidal wrench with approximate quadratic CWC for (20c)), as explained in Sec. 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 i-th mixture of Gaussians \( p_i(\mathbf{c}) \) has been replaced by \( p_i(\mathbf{x}) \) to be generic. And the cost function is given by:

$$\ell_s(\mathbf{x}, \mathbf{u}) = w_c \| \dot{\mathbf{c}} \|^2 + w_L \| \dot{\mathbf{L}} \|^2$$

For all the experiments and robots presented in the next section, we use the same weighting in the cost function: \( w_c = w_L = 10 \). 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.

B. Efficient resolution: the multiple shooting approach

Problem (9) and (20) consider optimization variables \( \mathbf{x} \) and \( \mathbf{u} \) of infinite dimension and cannot be directly handled by a computer. Addressing these nominal problems requires the use of indirect methods like the Pontryagin’s maximum principle or the Hamilton-Jacobi-Bellman principles, in order to reformulate the optimization problem as an integration problem of an augmented system. Unfortunately, these indirect approaches cannot handle (20) due to the bilinear constraint (20c). In addition, it is hard to guess a correct initial value of the adjoint systems. Alternatively, “direct” approaches turn the initial infinite-dimensional problem into a finite-dimensional one by constraining the control or the state trajectories to live in an arbitrary basis function.

Various details of implementation should be considered to obtain an efficient resolution. The most important in our opinion is the way the pair \( (\mathbf{x}, \mathbf{u}) \) is handled. On the first hand, collocation [15, 37] explicitly represents both the state and the control variables. The collocation method then tends to match them at the collocation nodes. On the other hand, single shooting [21, 50] only explicit the control trajectory while the state is obtained by integration. In between, multiple shooting makes explicit the control trajectory along with some few state variables at given shooting nodes [51]. The multiple-shooting intervals are then made independent one from each other. Then state and control continuation are enforced at the shooting nodes.

Both collocation and multiple-shooting approaches can deal with unstable dynamics like the LIPM or centroidal dynamics model (unlike single shooting). To be really effective, collocation methods must rely on fine discretization grid in order to make the state trajectory consistent with the dynamics. This leads to a high dimensional problem, likely
to be difficult to solve with real-time aiming even on modern computers. For its part, the multiple-shooting method is able to work with coarser grid, leading to an underlying optimization problem of smaller dimension.

In the end, it appears that multiple-shooting is a well suited approach to solve in a sparse manner \(^{(20)}\) thanks to the problem structure. In addition, a multiple-shooting problem can be easily warm-started with a good initial guess of the state trajectory. This initial solution just needs to be consistent on the multiple-shooting interval, not necessarily on the interval bounds.

Our implementation of \(^{(20)}\) relies on the multiple-shooting optimal control framework MUSCOD-II \(^{(52)}\). MUSCOD-II provides internal routines for accurate integration and computation of sensitivities along with an efficient sparse sequential quadratic program (SQP) solver. In our experiments, we used the sparse solver OOQP \(^{(55)}\) as internal QP solver of the SQP. Finally, it is worth mentioning that MUSCOD-II has already been successively applied for multi-contact locomotion \(^{(12)}\).

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 then 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.

A. Description of the complete pipeline

Our locomotion framework is composed of three stages:

a) Contact sequence planning: Depending on the experiments, the contact sequences are either manually designed or automatically generated using the contact planner \(^{(50)}\). We also manually design the end-effector trajectories by using splines with zero acceleration and velocity at take off and landing instants.

b) 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 discontinue 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 differentiable along all the motions.

c) 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 \(^{(54)}\). In addition, the IK must track the end-effector trajectories. Optimal forces are also extracted from the OCP (if \(u_f\) is the control variable) and can be used as references to control the robot with an inverse dynamics low-level controller.

d) 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 have been done 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 7ms and 15ms 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 seconds of horizon length, allowing the solver to run at 20Hz. This matches the application needs \(^{(55)}\).

There exists no open-source software that would have allowed us to benchmark our method with respect existing works Time scores are given in some previous works. In \(^{(12)}\), 30 minutes are needed for some few steps. In \(^{(14)}\), 8 minutes are needed per iteration for long movements. In \(^{(19)}\), 100ms are needed per iterations for 5 steps, but to the cost of 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 ten minutes of computation. Whole-body optimization using MUSCOD-II \(^{(55)}\), \(^{(57)}\) requires hours of computation to generate biped gaits. In \(^{(40)}\), the solver needs 3 hours to generate multi-contact movements. Model-predictive control is targeted in \(^{(50)}\), \(^{(58)}\), while one step of optimization (with horizon length of 0.5s) implies 100ms of computation; however, the results are yet not realistic enough to generate locomotion movements 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.

B. Experiment 1 - long steps walking

In this first experiment, we aim to compare the influence of both types of controls \(u_c\) and \(u_f\) on the solution. For that purpose, we use a simple benchmark which consists in long step walking with a stride of 0.9m with the HRP-2 robot. This stride is quite huge for such a humanoid robot of 1.6m height. Then, starting from a resting position and ending to an other resting position, the solver has to find a crouching gait in order to satisfy kinematics feasibility constraints.
Fig. 8. Comparison of the state trajectories obtained with either the force-based OCP (simple and exact 3D cones, nonminimal parameters – $a_f$) or the motion-based OCP (approximate 6D cone, minimal 6D 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.

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 $u_f$. 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 $u_c$ better able to approximate it thanks to the more direct correlation between the centroidal variables and the resulting motion.

C. 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 1.4s and 0.2s respectively. The resulting motion is depicted in Fig. 10. 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. 9 shows two trajectories of the CoM projected in the right foot 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 right foot (depicted in Fig. 2). 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.

D. Experiment 3 - climbing up 15-cm high steps with handrail support

The experimental setup is another industrial stairs 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 1.8s and 0.4s 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 1.78m high humanoid robot weighting around 100kg. 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. 12.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an efficient approach to generate multi-contact motion for legged robots. For that aim, we first showed under which conditions the locomotion problem can be decoupled into two stages: first find a feasible centroidal trajectory, then track this centroidal trajectory at the whole-body level. This led us to introduce a generic optimal control formulation able to work both with the contact forces or with the centroidal wrench. In a second time, we proposed a generic way to handle feasibility constraints of the centroidal dynamics (or reduced models in general) inside the optimal control formulation as occupation measure. In particular, we suggest a learning procedure to approximate the occupation measure of the CoM with respect to the contact sequence. To work with the centroidal wrench as control input, we also introduced a conic approximation of the centroidal wrench cone leading to a single dimensional constraint. We experimentally validated all those contributions with several multi-contact experiments on the HRP-2 robot on real scenarios and also in simulation with the TALOS humanoid robot.

This work first shows that both formulations are able to deal with Receding Horizon thanks to computation times very low both for the centoidal wrench (near 0.2s for 25s of motion) and the force implementations (near 0.4s for 17s of motion). For any investigated scenario, the centroidal wrench formulation is largely faster than the formulation in contact forces. This is due to the dimensionality of the control which remains constant and equal to 6 in the first case. However, if one seeks for robustness in the locomotion pattern, one must additionally adjust the duration of each phase and also consider contact placements as free variables as suggested in [59]. In this precise case, the second formulation is much more suited as no computation of the CWC approximation is required between two iterations of the Multiple-Shooting algorithm. In addition, the feasibility constraint on the CoM is already adapted for such case thanks to the independence assumption.

Currently, we have only investigated the learning of the CoM feasibility constraint. As an extension of this work, a promising research area is to look at the complete proxy \( \mu(c, \dot{c}, \ddot{c}, L_c, \dot{L}_c) \) which will link all the centroidal state and its dynamics to the whole-body kinematics and dynamics constraints. Then, the cost function of the optimal centroidal problem will be only composed of the proxy term, the regularization terms currently contained in \( \ell_s \) will become
We warmly thank the Simulation and Optimization group at the Interdisciplinary Center for Scientific Computing (IWR) of Heidelberg University in Germany for providing the optimal control framework MUSCOD-II.

REFERENCES


