Learning Feasibility Constraints for Multi-contact Locomotion of Legged Robots
Justin Carpentier, Rohan Budhiraja, Nicolas Mansard

To cite this version:

HAL Id: hal-01526200
https://hal.laas.fr/hal-01526200
Submitted on 22 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 4.0 International License
Learning Feasibility Constraints for Multi-contact Locomotion of Legged Robots

Justin Carpentier, Rohan Budhiraja and Nicolas Mansard
Laboratoire d’Analyse et d’Architecture des Systèmes
Université de Toulouse
7 avenue du Colonel Roche
Toulouse, FRANCE
Email: justin.carpentier@laas.fr

Abstract—Relying on reduced models is nowadays a standard cunning to tackle the computational complexity of multi-contact locomotion. To be really effective, reduced models must respect some feasibility constraints in regards to the full model. However, such kind of constraints are either partially considered or just neglected inside the existing reduced problem formulation. This work presents a systematic approach to incorporate feasibility constraints inside trajectory optimization problems. In particular, we show how to learn the kinematic feasibility of the centre of mass to be achievable by the whole-body model. We validate the proposed method in the context of multi-contact locomotion: we perform two stairs climbing experiments on two humanoid robots, namely the HRP-2 robot and the new TALOS platform.

I. INTRODUCTION

Multi-contact locomotion of legged robots inside heterogeneous non-flat environments is a hard and challenging problem. One major difficulty of this problem lies in its computational complexity: finding the whole-body trajectory solving a locomotion task corresponds to a high dimensional mathematical problem with a complex structure, hardly tractable by modern computers and particularly the ones embedded in modern legged robots [27, 17].

1) Reduced models: To tackle those limitations, various strategies have been proposed in the literature. Most of them are based upon using reduced models: instead of working with the full dynamics, only a subpart is considered, covering the essential properties of the whole dynamics. In the context of bipedal locomotion, the most famous reduced model is the linear inverted pendulum model (LIPM) [15].

The locomotion is then reduced to the problem of finding a trajectory for the reduced model which will then drive the whole-body system. Starting with [16], various optimal control formulation have been proposed by the community, to either tackle the robustness problem [30], include viability conditions [25], allow altitude variations of the center of mass (CoM) [2], or also include foot placements as parameters of the problem [19].

However, LIPM-based methods are restricted to basic environments (flat ground, no obstacles) and cannot deal with more complex scenarios as non-coplanar contact cases, climbing stairs using handrail, etc. Considering non-coplanar contacts breaks the nice linearization leading to the LIPM model. A first approach to handle the non-linear dynamics was proposed in [14], however requiring technical and dedicated developments based on limiting assumption (e.g. prior knowledge of force distribution). In quite another vein, it has been proposed to simplify the whole-body optimization problem by e.g. assuming unconstrained torque capabilities [6]. Both approaches indeed boil down to optimizing the so-called centroidal dynamics [20] as reduced model. Direct resolution of the underlying optimal control problem based on multiple-shooting approach has been recently proposed [4], leading to 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 [13, 5, 2]. In most cases, the footstep sequence is assumed given, although some solvers are also able to discover it while optimizing the centroidal dynamics [19, 7], to the price of heavier computational costs.

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). Such constraints are difficult to express as solely function of the reduced model. For instance, the CoM trajectory must be achievable by the whole-body kinematics. These constraints can be tackled explicitly, by adding the corresponding whole-body variables in the optimization scheme [19, 6].
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 [32]. In most of previous works, proxy constraints are defined by some rough approximations (box constraints, elliptic bounds, etc) leading to a certain conservatism; or it is simply ignored inside the reduced problem formulation. Footstep limits have been encoded by hyper-plane based on a dataset of robot success and failure inside a dynamic simulator [23]. Similar constraints can be obtained by training a training a neural network [21]. In [32], similar bounds are obtained by trial and errors based on stability analysis of the whole-body system.

Constraining only the CoM position is not sufficient. It is also necessary to consider the constraints related to the contact forces [29] which must lie inside so-called friction cones, the capacity of robot to generate such value of angular momentum, etc. The main problem lies in the fact that it is hard to find analytic formulas to represent and express those constraints.

3) Contribution: In this paper, we propose a systematic approach to handle feasibility constraints in the context of trajectory optimization for reduced models, leading to efficient resolution on the real robot. The resulting constraint formulation could be employed in most of the optimal control solver based on centroidal dynamics [19] [7] [13], although we implemented it inside a multiple-shooting solver [4].

We first recall the equations of motion for both the whole-body and the centroidal models and define the resulting optimal control problem (OCP) for multi-contact locomotion in Section II. These definitions logically introduce the need of proxy constraint. Our main contribution, described in Section II, is to represent proxy constraints by occupancy measures, whose corresponding cost of transport is optimized in the OCP. In Section IV we propose a complete solution to learn the CoM feasibility constraint by off-line sampling the robot motion capabilities. Finally, the effectiveness of the approach is highlighted with two real experiments on the HRP-2 robot climbing stairs with or without using handrail and one in simulation with the TALOS humanoid robot [26] climbing stairs using handrail, reported in Section V.

The joint configuration \( \mathbf{q} \) and its time derivatives \( \dot{\mathbf{q}} \) to the torque actuation \( \mathbf{\tau} \) and the contact forces \( \mathbf{f}_k \):

\[
\begin{bmatrix}
    H_u \\
    H_a
\end{bmatrix}
\dot{\mathbf{q}} +
\begin{bmatrix}
    \mathbf{b}_u \\
    \mathbf{b}_a
\end{bmatrix}
= 
\begin{bmatrix}
    \mathbf{g}_u \\
    \mathbf{g}_a
\end{bmatrix}
+ \begin{bmatrix}
    0_b \\
    \mathbf{\tau}_a
\end{bmatrix} + 
\sum_{k=1}^{K}
\begin{bmatrix}
    J_{k,u}^T \\
    J_{k,a}^T
\end{bmatrix}
\mathbf{f}_k
\]

where subscripts \( u \) and \( a \) stands for the under-actuated and actuated parts respectively, \( H \) is the generalized mass matrix, \( b \) covers centrifugal and Coriolis effects, \( g \) is generalized gravity vector and \( J_k \) is the Jacobian of the contact \( k \).

The 6 first rows of (1) represent the under-actuated dynamics of the robot, also called the centroidal dynamics [20]. It coincides with the Newton-Euler equations of motion which links the joint configuration \( \mathbf{q} \) and its time derivatives \( \dot{\mathbf{q}}, \ddot{\mathbf{q}} \) to the torque actuation \( \mathbf{\tau} \) and the contact forces \( \mathbf{f}_k \):

\[
\begin{bmatrix}
    H_u \\
    H_a
\end{bmatrix}
\ddot{\mathbf{q}} +
\begin{bmatrix}
    \mathbf{b}_u \\
    \mathbf{b}_a
\end{bmatrix}
= 
\begin{bmatrix}
    \mathbf{g}_u \\
    \mathbf{g}_a
\end{bmatrix}
+ \begin{bmatrix}
    0_b \\
    \mathbf{\tau}_a
\end{bmatrix} + 
\sum_{k=1}^{K}
\begin{bmatrix}
    J_{k,u}^T \\
    J_{k,a}^T
\end{bmatrix}
\mathbf{f}_k
\]

where subscripts \( u \) and \( a \) stands for the under-actuated and actuated parts respectively, \( H \) is the generalized mass matrix, \( b \) covers centrifugal and Coriolis effects, \( g \) is generalized gravity vector and \( J_k \) is the Jacobian of the contact \( k \).

The 6 first rows of (1) represent the under-actuated dynamics of the robot, also called the centroidal dynamics [20]. It coincides with the Newton-Euler equations of motion which links the joint configuration \( \mathbf{q} \) and its time derivatives \( \dot{\mathbf{q}}, \ddot{\mathbf{q}} \) to the torque actuation \( \mathbf{\tau} \) and the contact forces \( \mathbf{f}_k \):

\[
\begin{bmatrix}
    H_u \\
    H_a
\end{bmatrix}
\ddot{\mathbf{q}} +
\begin{bmatrix}
    \mathbf{b}_u \\
    \mathbf{b}_a
\end{bmatrix}
= 
\begin{bmatrix}
    \mathbf{g}_u \\
    \mathbf{g}_a
\end{bmatrix}
+ \begin{bmatrix}
    0_b \\
    \mathbf{\tau}_a
\end{bmatrix} + 
\sum_{k=1}^{K}
\begin{bmatrix}
    J_{k,u}^T \\
    J_{k,a}^T
\end{bmatrix}
\mathbf{f}_k
\]

II. MULTI-CONTACT LOCOMOTION

This section reviews the centroidal dynamics and its links with the whole-body model. We then describe the generic optimal control problem used to compute the centroidal trajectories, based on [4]. Although this section contains mostly known materials, we believe that the clean formulation of the dynamics decoupling is a contribution per se. It is indeed a prerequisite to the introduction of proxy constraints in the next section.

A. Contact model

The interaction between a robot and the environment is defined through a set of contact points \( \mathbf{p}_k \in \mathbb{R}^3, k = 1...K \). 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 \( \mathbf{p}_k \) is defined a contact force \( \mathbf{f}_k \). In the case of unilateral contacts, \( \mathbf{f}_k \) must lie inside a 3-dimensional quadratic friction cone \( K_k \), characterized by a positive friction coefficient \( \mu_k \). Fig. 1 depicts humanoids robots making contact with the environment.

We only consider here rigid contact interaction (contacting bodies are fixed) which is a reasonable assumption for modern 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.

Computing automatically the contact sequence is a difficult problem [3], but efficient contact planners now exist to compute it in a short amount of time [10] [28].

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 the Lagrange equation of motion, which links the joint configuration \( \mathbf{q} \) and its time derivatives \( \dot{\mathbf{q}}, \ddot{\mathbf{q}} \) to the torque actuation \( \mathbf{\tau} \) and the contact forces \( \mathbf{f}_k \):

\[
\begin{bmatrix}
    H_u \\
    H_a
\end{bmatrix}
\ddot{\mathbf{q}} +
\begin{bmatrix}
    \mathbf{b}_u \\
    \mathbf{b}_a
\end{bmatrix}
= 
\begin{bmatrix}
    \mathbf{g}_u \\
    \mathbf{g}_a
\end{bmatrix}
+ \begin{bmatrix}
    0_b \\
    \mathbf{\tau}_a
\end{bmatrix} + 
\sum_{k=1}^{K}
\begin{bmatrix}
    J_{k,u}^T \\
    J_{k,a}^T
\end{bmatrix}
\mathbf{f}_k
\]

where subscripts \( u \) and \( a \) stands for the under-actuated and actuated parts respectively, \( H \) is the generalized mass matrix, \( b \) covers centrifugal and Coriolis effects, \( g \) is generalized gravity vector and \( J_k \) is the Jacobian of the contact \( k \).

The \( 6 \) first rows of (1) represent the under-actuated dynamics of the robot, also called the centroidal dynamics [20]. It coincides with the Newton-Euler equations of motion which links the variation of the whole-system linear and angular momentum to the contact forces. Denoting by \( \mathbf{L}_c \) the angular momentum, the centroidal dynamics is:

\[
\mathbf{m}\ddot{\mathbf{c}} = \sum_k \mathbf{f}_k + \mathbf{mg}
\]

\[
\mathbf{L}_c = \sum_k (\mathbf{p}_k - \mathbf{c}) \times \mathbf{f}_k
\]

where \( \mathbf{g} \triangleq (0,0,-9.81) \) is the gravity vector and the operator \( \times \) denotes the cross product.

The \( n \) last rows of (1) are the classic Lagrange dynamics of a fixed robot manipulator making contacts, which is both linear in the torque and force inputs.

C. Partial decoupling of centroidal and manipulator dynamics

Under the assumption of sufficient torque, it is possible to split up the problem of whole-body locomotion into two successive (hierarchical) stages: a) find the contact force trajectories driving the the centroidal dynamics; b) recover the required joint torque trajectory from the centroidal trajectory and the end-effector trajectories using for instance a second
order inverse kinematics/dynamics solver. This assumption is reasonable given motorization of current legged robots, that are sufficient to generate high torques. We discuss the resulting limits in the conclusion.

To ensure the feasibility of the first stage with respect to the second stage, some conditions must be met by the first stage:

i) the forces must remain inside the friction cones in case of unilateral contacts;
ii) the centroidal dynamics must be kinematically and dynamically feasible by the system;

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 through the centroidal mapping:

\[
\begin{bmatrix}
    h \\
    L_c
\end{bmatrix} = A_g(q) \dot{q},
\]

with \( A_g \) the so-called centroidal momentum matrix [20].

D. Centroidal optimal control formulation

From the previous assumptions, the problem of multi-contact locomotion can be reduced to the following formulation:

From a given contact sequence and an initial centroidal state, find a feasible centroidal trajectory, satisfying the Newton-Euler dynamics, respecting the contact constraints and leading to a viable state.

This formulation can be directly transcribed as an optimal control problem of the form:

\[
\min_{\mathbf{x}, \mathbf{u}} \sum_{s=1}^{S} \int_{t_s}^{t_{s} + \Delta t_s} \ell_s(x, u) \, dt
\]

s.t. \( \forall t \), \( \dot{x} = f(x, u) \) \hspace{1cm} (3b)

\( \forall t \), \( u \in \mathcal{K} \) \hspace{1cm} (3c)

\( \forall t \), \( \exists (q, \dot{q}, \ddot{q}) \) s.t. \( x, \dot{x} \) is feasible \hspace{1cm} (3d)

\( x(0) = (c_0, 0, 0) \) \hspace{1cm} (3e)

\( x(T) \) is viable \hspace{1cm} (3f)

where \( s \) is the index of the contact phase, \( \mathbf{x} \) and \( \mathbf{u} \) 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 (3b) and (3c) enforce consistent dynamics with respect to the contact model. Eq. (3d) 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 how it can be replaced by an automatic proxy constraint. Constraint (3e) constrains the trajectory to start with a given state (typically estimated by the sensor of the real robot). Terminal constraint (3f) is difficult to exactly represent [31] and is replaced in practice by zero terminal movement \( \ddot{c}(T) = L_c(T) = 0 \) and \( x(T) = (c^*, 0, 0) \). Finally, \( \ell_s \) is the cost function which enforces the smoothness of both the state and control trajectories. Various \( \ell_s \) can be discussed and implemented. The one used in the experiments requires some additional definitions and is given in Section IV.

III. Feasibility of the centroidal problem

This section reports the main contribution of the paper. We present a mathematical coding of the feasibility constraints as probability measures. We then discuss the interest of this representation with respect to more-classical set-membership and show how it can be used to efficiently implement (3d) in the OCP. This section introduces the abstract definitions, that next section section uses to build the complete implementation.

A. Mathematical representation of feasibility constraints

Our objective is to efficiently implement the feasibility constraint (3d) 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 [6]. 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 [11, 7] to encode maximal step size in biped walking, or in [5] 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 [23, 21, 32].

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 is likely to fail. Consequently, the set is often arbitrarily reduced to improve the robustness of the whole-body solution.

B. Proxy as occupation measure

In this paper, we rather state that the proxy is best implemented by the occupation measure over \( x, \dot{x} \).

Consider a state trajectory \( \mathbf{x} \). With (3d), we want to maximize the likelihood that the inverse-kinematics solver converges on a trajectory \( \mathbf{q} \) such that \( \mathbf{x} \) is the image of
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.

IV. 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 \( c(t) \). Handling this sole constraint first is a proper way of validating our approach. It is also interesting in practice, as the feasibility of the CoM is the most limiting constraint for humanoid robots. 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.

A. 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”. \( p_i(c) \) 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 greatly simplifies 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.

### B. 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) \(^{[22]}\). 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.

### C. 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 \( 10 \times N_{\text{samples}} \) 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) \(^{[1]}\). 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 \(^{[8]}\).

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 \(^{[12]}\). 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.

### D. 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 effectors in contact with the environment.

### E. Proposed optimal control formulation

We can now express the complete formulation of the cost function \( \ell_s(x, u) \).

\[
\ell_s(x, u) = w_x \| x \|^2 - \sum_{i=1}^{K_c} \log(p_i(c))
\]  

where the first term\(^1\) enforces a smooth trajectory, the second term is the cost of transport for the approximate CoM occupancy measure, and \( w_x \) weights their relative importance. The first term can similarly be interpreted as a weak formulation of the occupancy measures for the second order terms \( (\dot{c} \text{ and } L_c) \) and their derivatives, through centred Gaussian measures (i.e. no prior on occupancy distribution). If the complete occupancy measure \( \mu_o \) is available, the first term would become useless.

### V. Results

We first illustrate the learning procedure exposed in Sec. IV on the HRP-2 robot. We then present the complete pipeline we used to achieve both real and simulated motions on the two different robots. After that, we show two real experiments of multi-contact locomotion with the HRP-2 robot inside an environment similar to what can be found in the industry. Finally, we end this section by showing the versatility of the approach with the simulation of a multi-contact motion on another humanoid robot. We refer the reader to the accompanying video for a better viewing of the multi-contact experiments\(^2\).

#### A. Illustration of the learning procedure

We validate the proposed learning approach on the HRP-2 robot. For that purpose, 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 plane. In other words, this means that the distribution cannot be approximated by a

\(^1\)\( \| x \| \) is a function of \( x \) and \( u \) through \( x = f(x, u) \)

\(^2\)available at https://youtu.be/7hiL6DpMAA
single normal distribution but must be composed of several distributions. 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. However, this approximation is conservative with respect to the support and the level sets of the original distribution.

Fig. 3 highlights the experimental procedure suggested in Sec. IV-C 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 7 kernels in the mixture. In other words, it is sufficient to take a GMM of size 7 to represent the CoM distribution in the foot frames. For the right and left grippers, it is a little bit different. The KL-divergence first decreases and then increases from 14 kernels. This behaviour can be explained by the fact that the EM algorithm does not optimize the KL divergence but the likelihood of observation (expectation). We chose to represent the CoM distribution w.r.t. the grippers with a GMM of size 14.

A similar study has been done on the TALOS humanoid robot, which is larger than HRP-2 and as different leg and arm kinematics. The distributions for the right foot of TALOS robot, which is larger than HRP-2 and as different leg and arm kinematics, and can be used as references to control the robot. During execution, the reference posture is tracked as well as the reference foot forces using the robot low-level control system (named HRP “stabilizer”).

Computing the 25s of motion takes 42 iterations of the multiple-shooting algorithm, costing about 8s in total. In average, each iteration takes approximately 0.2s for 25s of motion. About 70% of the computation time is spent solving the underlying quadratic program of the multiple-shooting algorithm and other 20% are dedicated to the numerical integration of the dynamics together with the computations of sensitivities (derivatives).

Fig. 5 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. close to the maxima of the PDF. On the contrary, if we do not add the log-pdf term, the CoM tends to be infeasible.

It is worth to mention that maximizing the stay of the CoM in the most feasible region leads to a crouching walk. This may not be desirable for aesthetics. However, in the case of HRP-2, this crouch of the legs must exist to avoid singularities which are not tolerated by the low-level controller.
with a guardrail. The steps have a length of industrial stairs made of four 15-cm high steps and equipped with a guardrail support:

2) **Experiment 2 - climbing up 15-cm high steps with guardrail support:** The experimental setup is another guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing. We reproduce the climbing stairs with guardrail scenario, but this time with the TAOS robot in simulation. Compared to HRP-2, TAOS 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 was asked to achieve multi-contact stairs climbing.

We have defined our proxy to be an occupancy measure over the whole centroidal state and contact forces, although only the measure over the CoM was approximated. Learning the CoM kinematic feasibility is only a first stage. We plan to extend the approach by learning the constraints related to the other centroidal variables. This may severely improve the quality of the whole-body behavior. The methodology would also nicely apply to learning the terminal viability constraint. As discussed at the end of Section IV, we believe that introducing the occupancy measure over all centroidal variables would reduce the locomotion problem to a simple optimal control problem composed of a single cost function, with only initial constraints.

Finally, our methodology goes beyond trajectory optimization. We have introduced proxies as a way to ensure consistency between two sets of motion models (centroidal model and whole-body model). The same consistency is desirable between the centroidal pattern generator and the upstream contact planner, whose motion models could be similarly learned by statistical sampling.

**Acknowledgments**

We warmly thank the Simulation and Optimization of Heidelberg University for providing the optimal control framework MUSCOD-II. This work is supported by the European Research Council (ERC) through the [Actanthrope project](#)(ERC grant agreement 340050), the RoboCom++ FLAG-ERA JTC 2016 proposal and the French National Research Agency through the Loco3D project (ANR grant agreement ANR-16-CE33-0003).

**References**


In IEEE-RAS Int. Conf. on Humanoid Robotics (ICHR), 2014.


2003.