



**HAL**  
open science

## **C-CROC: Continuous and Convex Resolution of Centroidal dynamic trajectories for legged robots in multi-contact scenarios**

Pierre Fernbach, Steve Tonneau, Olivier Stasse, Justin Carpentier, Michel Taïx

► **To cite this version:**

Pierre Fernbach, Steve Tonneau, Olivier Stasse, Justin Carpentier, Michel Taïx. C-CROC: Continuous and Convex Resolution of Centroidal dynamic trajectories for legged robots in multi-contact scenarios. 2018. hal-01894869v1

**HAL Id: hal-01894869**

**<https://laas.hal.science/hal-01894869v1>**

Preprint submitted on 12 Oct 2018 (v1), last revised 20 Feb 2020 (v4)

**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.

# C-CROC: Continuous and Convex Resolution of Centroidal dynamic trajectories for legged robots in multi-contact scenarios

Pierre Fernbach, Steve Tonneau, Olivier Stasse, Justin Carpentier and Michel Taïx

**Abstract**—Synthesizing legged locomotion requires planning one or several steps ahead (literally): when and where, and with which effector should the next contact(s) be created between the robot and the environment? Validating a contact candidate implies a *minima* the resolution of a slow, non-linear optimization problem, to demonstrate that a Center Of Mass (COM) trajectory, compatible with the contact transition constraints, exists.

We propose a conservative reformulation of this trajectory generation problem as a convex 3D linear program, CROC. It results from the observation that if the COM trajectory is a polynomial with only one free variable coefficient, the non-linearity of the problem disappears. This has two consequences. On the positive side, in terms of computation times CROC outperforms the state of the art by at least one order of magnitude, and allows to consider interactive applications (with a planning time roughly equal to the motion time). On the negative side, in our experiments our approach finds a majority of the feasible trajectories found by a non-linear solver, but not all of them. Still, we demonstrate that the solution space covered by CROC is large enough to achieve the automated planning of a large variety of locomotion tasks for different robots, demonstrated in simulation and on the real HRP-2 robot, several of which were rarely seen before.

Another significant contribution is the introduction of a Bezier curve representation of the problem, which guarantees that the constraints of the COM trajectory are verified continuously, and not only at discrete points as traditionally done. This formulation is lossless, and results in more robust trajectories. It is not restricted to CROC, but could rather be integrated with any method from the state of the art.

**Index Terms**—Multi contact locomotion, centroidal dynamics, Humanoid robots, legged robots, motion planning

## I. INTRODUCTION

ONE long standing challenge in the domain of legged robotics is the proposition of a generic method, able to automatically synthesize motions for arbitrary robots in arbitrary environments. Resolving this issue is required to achieve a long term objective: the deployment of autonomous legged robots, able to navigate safely among unknown environments, outside of their research laboratories.

The term “multi-contact motion” has been proposed to distinguish this problem from the gaited locomotion problem [1], [2], because in this context no assumption can be made regarding the nature of the environment, or the contacts that will be created with it. In the multi contact case, the open problem of controlling a robot while satisfying dynamic and geometric constraints is made harder by the combinatorial

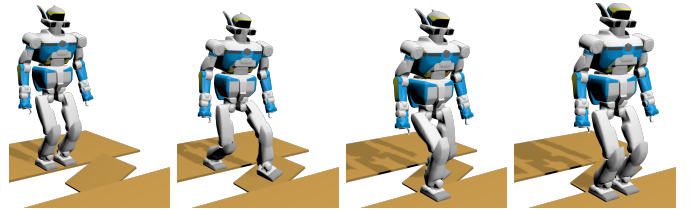


Fig. 1: An instance of the transition feasibility problem: can we guarantee that the contact sequence shown in this picture can be used to produce a feasible motion for the robot? To address this issue in this example we need to account for 7 different contact phases (including phases where the effector is flying, not displayed here).

aspect introduced by the choice (among an infinity of possibilities), of when and a where to create a contact between the robot and the environment, and with which effector. So far, this non-linear problem has remained out of reach of any existing method.

However, an increasing number of contributions consider the multi-contact problem, roughly following one of the two apparently different options: a) decompose the problem into a sequence of smaller problems, easier to solve [3]–[6]. In this case the difficulty is to find a formulation of the smaller problems equivalent to the original one, which results so far in approximations; b) Tackle the initial problem entirely, but in a computationally efficient way, through a reduction of the dimensionality, also obtained through approximations [7]–[9].

Both approaches have obtained significant successes, and while the authors lay in the former family of methods [10], [11], the objective of this paper is not to claim that one prevails. We rather claim that despite being different in spirit, those approaches face the same fundamental challenge: how to make sure that the solution computed using a reformulation of the multi-contact problem provides a straightforward solution to the original problem? As an example, both families of approaches propose contributions that rely on a model-based approach called the *centroidal model*, which only considers the dynamics of the Center Of Mass of the robot, rather than the whole-body dynamics. This model introduces approximations regarding the geometric constraints that lie on the robot, and also regarding the angular momentum variation induced by the motion of the rigid bodies that compose the robot. The question is then to determine whether it is possible to

78 formulate additional constraints on the centroidal dynamics,  
79 that would take into account the whole-body constraints.

80 Finding what we call the “reduction properties”: formal  
81 theorems or empirical properties that will prove the validity of  
82 the problem decomposition or approximation, is the original  
83 scientific issue that we propose to tackle.

84 In particular, in this work, we consider what we call the  
85 **transition feasibility** problem: given two states of the robot,  
86 can we guarantee that there exists (or not) a dynamically  
87 and kinematically consistent motion that connects these two  
88 states (Figure 1)? Being able to address efficiently this issue  
89 is desirable in the context of the authors’ framework, but  
90 not only, as the objective is to provide additional guaran-  
91 tees to the centroidal model, and to improve significantly  
92 its computational efficiency. From an applicative point of  
93 view, its resolution would also allow to address the N-step  
94 capturability problem [12]–[14]: given the current state of the  
95 robot, determine whether it will be able to come to a stop  
96 without falling in at most  $N$  steps ( $N \geq 0$ ). This issue is  
97 very important to guarantee the safety of the robot and its  
98 surroundings.

#### 99 A. The transition feasibility in a divide and conquer context

100 Over the last few years, we have proposed a methodology  
101 to tackle the multi-contact motion problem, which relies on  
102 its decomposition into three sub-problems solved sequentially  
103 (Figure 2). This approach follows a “divide and conquer”  
104 pattern, with the idea that three sub-problems should be ad-  
105 dressed in a sequentially independent fashion:  $\mathcal{P}_1$ , the planning  
106 of a trajectory for the root of the robot,  $\mathcal{P}_2$  the generation  
107 of a discrete contact sequence along the root’s trajectory  
108 and  $\mathcal{P}_3$  the generation of a whole-body motion from this  
109 contact sequence. We have proposed several contributions  
110 to each sub-problems [15]–[17], and built a prototype that  
111 demonstrated its capability to find solutions for various robots  
112 and environments, with interactive computation times (a few  
113 seconds of computation for several steps of motion).

114 The decoupling between each sub-problem allows to break  
115 the complexity, and comes with a cost that is the introduction  
116 of a feasibility problem: each sub-problem must be solved in  
117 the feasibility domain of the next sub-problems: ie. there must  
118 exist a sequence of contacts (problem  $\mathcal{P}_2$ ) that can follow the  
119 root’s trajectory found (solution of  $\mathcal{P}_1$ ), and similarly there  
120 must exist a feasible whole-body motion (problem  $\mathcal{P}_3$ ) from  
121 the computed contact sequence (solution of  $\mathcal{P}_2$ ). The latter  
122 problem is an instance of the transition feasibility problem  
123 that we address in this paper (The former was considered in  
124 [15]).

125 It is important to observe that in this context, establishing  
126 the transition feasibility as fast as possible is crucial:  $\mathcal{P}_2$  is  
127 a combinatorial problem, which implies that many contact  
128 sequences (thousands) must possibly be tried before finding  
129 a feasible contact sequence.

130 Recent contributions have proposed centroidal trajectory  
131 generation methods that could theoretically be used to answer  
132 the transition feasibility problem [18]–[20]. However, because  
133 of the combinatorial aspect of contact planning, the computa-  
134 tional time required by these methods is too important to use

a trial-and-error approach to verify the feasibility. Caron et al.  
recently proposed a computationally efficient method [21], but  
its application range is restricted to single-contact to single-  
contact transitions.

The work that is the closest to the present paper is the  
one of Ponton et al. [22]. By integrating the dynamic con-  
straints inside a mixed-integer programming problem [8], they  
addressed the transition feasibility problem at the contact  
planning level. However the constraints are only approximated  
through a convex relaxation (convex approximation is also  
done in [23]), and mixed-integer approaches remain subject to  
combinatorial explosion. The main difference between their  
formulation and the method presented in this paper lies in  
the fact that the presented method uses conservative dynamics  
constraints rather than approximated ones, and is also more  
computationally efficient.

#### B. Contribution

In this paper, as we tackle the transition feasibility issue,  
we also complete a framework able to automatically generate  
dynamic, collision free and multi-contact whole-body motions  
for a legged robot in complex environments. This framework  
has been presented in previous work: [15] [16] for  $\mathcal{P}_1$  and  
 $\mathcal{P}_2$ , and [17] [18] for  $\mathcal{P}_3$ . The framework is thus not directly  
a contribution of this paper.

The main contribution is the formulation of a **transition  
feasibility** criterion, able to test if there exists a kinematically  
and dynamically valid motion that connects two states of the  
robot, called CROC (which stands for Convex Resolution of  
Centroidal dynamic trajectories). Thanks to a conservative and  
convex reformulation of the problem, this is achieved in a  
fraction of the computational cost required by standard non-  
linear solvers. This method also produces a feasible CoM tra-  
jectory. This trajectory can be used as a valuable initial guess  
by a non-conservative non-linear solver to converge towards  
an optimal solution. Noticeably, this formulation is, along  
with [24], one of the few **able to continuously guarantee  
that the computed trajectories respect the constraints of  
the problem**, when other approaches require to discretize the  
trajectory and check punctually the constraints.

Thanks to this criterion, we can provide strong guarantees  
that the computed contact sequence will lead to a feasible  
whole body motion. This results in a major technical con-  
tribution, as we obtain and demonstrate a framework able  
to automatically and robustly generate complex motions, in  
simulation and on the real HRP-2 robot.

In the following section we recall the formal definition of  
the problem. In section III we present our approach for the  
feasibility criterion.

We then present our framework in section IV. Finally, we  
present our experimental results in section V.

#### C. Situation of the contribution with respect to the authors previous work

The present paper is an extension of an IROS conference  
paper [25], where we propose a convex optimization method

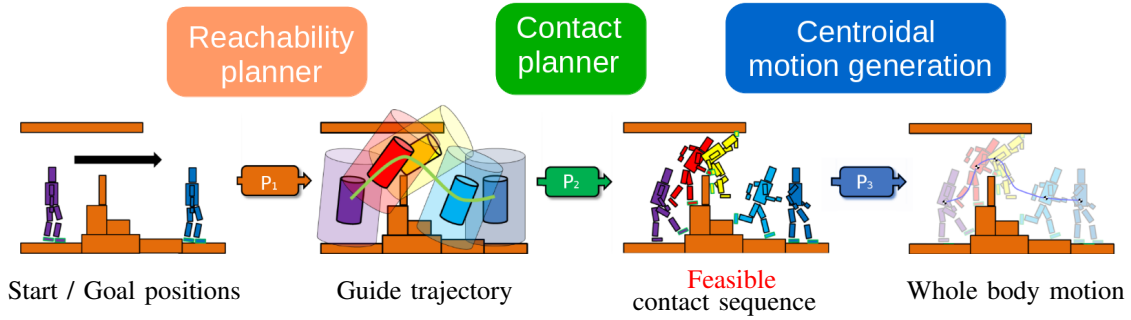


Fig. 2: Complete framework overview of our decoupled approach. In this work we only focus on addressing the transition feasibility problem, from  $\mathcal{P}_2$  to  $\mathcal{P}_3$ .

191 to solve the transition feasibility problem. Our previous formu-  
 192 lation, as others in the community, is limited by the necessity  
 193 to use of the double description method [26], an unstable  
 194 mean to compute the linear constraints that apply to the  
 195 problem [18], which allows for fast computation times. As  
 196 for all existing methods, it also requires a discretization of the  
 197 solution trajectory, such that the constraints of the problem  
 198 are only checked at specific instants. In this paper we also  
 199 propose a continuous formulation of the problem. It removes  
 200 the need for discretization, and is fast enough to avoid using  
 201 the double description method. We advocate for its adoption  
 202 for any centroidal generation method.

203 Sections II and III present important similarities with respect  
 204 to [25]. The novelty appears from section III-D, where we  
 205 present a continuous formulation able to deal with contact  
 206 switch during the trajectory.

207 The other sections of the paper are also novel. These novel-  
 208 ties include the completion of our experimental framework,  
 209 which allows us to validate our method on several experiments.  
 210 We also provide an empirical analysis of the performance of  
 211 our method with respect to a non-linear solver, in terms of  
 212 success rate and computational time.

## 213 II. PROBLEM DEFINITION

214 We define the transition feasibility problem as follows.  
 215 Given two configurations of a robot; given the contact loca-  
 216 tions associated to these two configurations; given the position,  
 217 velocity and acceleration of the Center Of Mass of the robot  
 218 at these two configurations; can we guarantee that there exists  
 219 a **feasible** motion that connects the two configurations? A  
 220 feasible motion should respect the kinematic constraints of  
 221 the robot, as well as the dynamics expressed at its Center Of  
 222 Mass. Depending on the use case, some constraints may be  
 223 removed (for instance if the end configuration is unknown, or  
 224 the problem is simply to put the robot to a stop).

225 Thus, in this work we define the transition feasibility  
 226 problem with respect to the centroidal dynamics of a robot,  
 227 as now commonly done [27], [19], [18]. In this section we  
 228 provide some formal definitions that are used in the rest of  
 229 the paper.

### A. Contact sequence and state

230 A legged motion can be discretized into a sequence of  
 231 contact phases, each differing by exactly one contact creation  
 232 or removal. Each contact phase defines a number of active  
 233 contacts, and their locations remain constant during the phase  
 234 (for instance when walking, the contact sequence is gaited as  
 235 it follows a periodic pattern: both feet in contact, left foot in  
 236 contact, both feet in contact, right foot in contact...). Thus,  
 237 each contact phase constrains kinematically and dynamically  
 238 the motion of the robot.  
 239

240 We define a state  $\mathbf{x} = (\mathbf{c}, \dot{\mathbf{c}}, \ddot{\mathbf{c}}) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$  as  
 241 the triplet describing a Center Of Mass (COM) position,  
 242 velocity and acceleration. To indicate that a state is compatible  
 243 with the dynamic and kinematic constraints associated with  
 244 a contact phase  $p \in \mathbb{N}$ , we use the superscript notation  
 245  $\mathbf{x}^{\{p\}} = (\mathbf{c}^{\{p\}}, \dot{\mathbf{c}}^{\{p\}}, \ddot{\mathbf{c}}^{\{p\}})$ .

246 Given two states  $\mathbf{x}_s^{\{p\}}$  and  $\mathbf{x}_g^{\{q\}}$  with  $q \geq p$ , the transition  
 247 feasibility problem consists in determining whether there ex-  
 248 ists a feasible trajectory  $\mathbf{c}(t), t \in \mathbb{R}^+$  of duration  $T \in \mathbb{R}^+$ ,  
 249 which connects exactly  $\mathbf{x}_s^{\{p\}}$  and  $\mathbf{x}_g^{\{q\}}$ .

### B. Centroidal dynamic constraints on $\mathbf{c}(t)$

250 For a contact phase  $\{p\}$ , for any  $t \in [0, T]$  the centroidal  
 251 dynamic constraints are given by the Newton-Euler equations.  
 252 These constraints form a convex cone (or polytope), which can  
 253 be expressed under two different formulations, theoretically  
 254 equivalent [28]–[30], but really different in practice. In this  
 255 paper we present and discuss both formulations.  
 256

257 1) *Equality constraint representation (or force formula-*  
 258 *tion)*: The Newton-Euler equations are:

$$\begin{bmatrix} m(\ddot{\mathbf{c}} - \mathbf{g}) \\ m\mathbf{c} \times (\ddot{\mathbf{c}} - \mathbf{g}) + \dot{\mathbf{L}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \dots & \mathbf{I}_3 \\ \hat{\mathbf{p}}_1 & \dots & \hat{\mathbf{p}}_{nc} \end{bmatrix} \mathbf{f} \quad (1)$$

259 Where :

- 260 •  $m$  is the total mass of the robot;
- 261 •  $nc$  is the number of contact points;
- 262 •  $\mathbf{p}_i \in \mathbb{R}^3, 0 \leq i \leq nc$  is the location of the  $i$ -th contact  
 263 point;
- 264 •  $\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{nc}]^T \in \mathbb{R}^{3nc}$  is the stacked vector of  
 265 contact forces applied at each contact point;
- 266 •  $\mathbf{g} = [0 \ 0 \ -9.81]^T$  is the gravity vector;

- $\dot{\mathbf{L}} \in \mathbb{R}^3$  is the derivative of the angular momentum (expressed at  $\mathbf{c}$ ).
- $\mathbf{p}_i$  denotes the skew-symmetric matrix of  $\mathbf{p}_i$ .

The contact forces are further constrained to lie in their so-called friction cone, which we conservatively linearize with four generating rays. Thus  $\mathbf{f}$  has the form  $\mathbf{f} = \mathbf{V}\boldsymbol{\beta}$ , where  $\mathbf{V} \in \mathbb{R}^{3nc \times 4nc}$  is the matrix containing the diagonally stacked generating rays of the friction cone of each contact point and  $\boldsymbol{\beta} \in \mathbb{R}^{4nc}$  is a positive vector variable.

This formulation has the disadvantage of introducing a large number of variables associated to the contact forces (one vector  $\boldsymbol{\beta}$  for each instant where the constraints are verified).

2) *Inequality constraint representation (or Double Description formulation)*: Because the set of admissible contact forces is a polytope, it is possible to use an equivalent “face representation” of the constraints that apply to the center of mass and angular momentum. With this formulation, the force variables disappear:

$$\mathbf{H}^{\{p\}} \underbrace{\begin{bmatrix} m(\ddot{\mathbf{c}} - \mathbf{g}) \\ m\mathbf{c} \times (\ddot{\mathbf{c}} - \mathbf{g}) + \dot{\mathbf{L}} \end{bmatrix}}_{\mathbf{w}} \leq \mathbf{h}^{\{p\}} \quad (2)$$

where  $\mathbf{H}^{\{p\}}$  and  $\mathbf{h}^{\{p\}}$  are respectively a matrix and a vector defined by the contact points of the phase and their friction coefficients.

With this formulation, the dimension of the problem is greatly reduced. However, the computation of the matrices  $\mathbf{H}^{\{p\}}$  and  $\mathbf{h}^{\{p\}}$  is a non-trivial operation called the double description method [26]. It is computationally expensive, and subject to occasional failures.

In the following theoretical sections, we will use the inequality formulation because we believe our contribution is more intuitive with this representation. In terms of implementation the equality formulation is more reliable but slower. However we show that under our formulation the computation times remain in the same order of magnitude in both cases.

3) *The dynamic constraints are not convex*: Because of the cross product between  $\mathbf{c}$  and  $\ddot{\mathbf{c}}$ , the constraints are not linear, and the issue of finding a trajectory satisfying them in the general case is a non-convex problem.

### C. Centroidal kinematic constraints on $\mathbf{c}(t)$

Each active contact creates kinematic constraints on  $\mathbf{c}(t)$ . We use linear constraints to represent these constraints depending on the 6D positions of each active contact frames. They give us a necessary but not sufficient condition for kinematic feasibility (evaluated and discussed in section V-A5). We refer the reader to [31] for the computation of these constraints. We write them  $\mathbf{K}^{\{p\}}\mathbf{c} \leq \mathbf{k}^{\{p\}}$  for phase  $\{p\}$ .

## III. CONVEX FORMULATION OF THE TRANSITION PROBLEM

As previously proposed [25], in order to determine the existence of a valid centroidal trajectory  $\mathbf{c}(t)$ , we formulate the problem as a convex one by getting rid of the non-linear constraints induced by the cross product  $\mathbf{c} \times \ddot{\mathbf{c}}$ . To achieve this we impose a conservative condition on  $\mathbf{c}(t)$ .

However, a significant contribution with respect to [25] and other contributions is a continuous reformulation of the problem, which guarantees that the resulting trajectory is always valid. Indeed, traditionally the constraints are only verified at specific points of the trajectory, using a discretization step that must be carefully calibrated to avoid an explosion in the number of variables and constraints, while guaranteeing that the constraints won't be violated in between.

### A. Reformulation of $\mathbf{c}(t)$ as a Bezier curve

Let us assume that  $\mathbf{c}(t)$  is described by an arbitrary polynomial of degree  $n$  of unknown duration  $T$ . In such case, without loss of generality,  $\mathbf{c}(t)$  is equivalently defined as a constrained Bezier curve of the same degree  $n$ :

$$\mathbf{c}(t) = \sum_{i=0}^n B_i^n(t/T)\mathbf{P}_i \quad (3)$$

where the  $B_i^n$  are the Bernstein polynoms and the  $\mathbf{P}_i$  are the control points.

With this formulation we can easily constrain the initial or final position, velocity or any other derivatives by setting the value of the control points. To connect exactly two states  $\mathbf{x}_s = (\mathbf{c}_s, \dot{\mathbf{c}}_s, \ddot{\mathbf{c}}_s)$  and  $\mathbf{x}_g = (\mathbf{c}_g, \dot{\mathbf{c}}_g, \ddot{\mathbf{c}}_g)$  we thus need at least 6 control points to ensure that the following constraints are verified:

- $\mathbf{P}_0 = \mathbf{c}_s$  and  $\mathbf{P}_n = \mathbf{c}_g$  guarantee that the trajectory starts and ends at the desired locations;
- $\mathbf{P}_1 = \frac{\dot{\mathbf{c}}_s/n}{T} + \mathbf{P}_0$  and  $\mathbf{P}_{n-1} = \mathbf{P}_n - \frac{\dot{\mathbf{c}}_g/n}{T}$  guarantee that the trajectory initial and final velocities are respected;
- $\mathbf{P}_2 = \frac{\ddot{\mathbf{c}}_s/(n(n-1))}{T^2} + 2\mathbf{P}_1 - \mathbf{P}_0$  and  $\mathbf{P}_{n-2} = \frac{\ddot{\mathbf{c}}_g/(n(n-1))}{T^2} + 2\mathbf{P}_{n-1} - \mathbf{P}_n$  guarantee that the initial and final accelerations are respected.

Depending on the considered problem, some constraints on the boundary positions, velocities or accelerations can be removed, without changing the validity of our approach. For instance, if the objective is simply to put the robot to a stop, the end velocities and accelerations can be set to zero, while the end position is left unconstrained. We can also extend this to any degree and add constraints on initial or final jerk or higher derivatives and automatically compute the position of the control points with a symbolic calculus script such as the one that we provide at the url <sup>1</sup>. We only need to compute the equation of the control points once and for all so we do not need to compute them at runtime. In the following equations, we use a curve of degree 6 with the constraints on initial and final position, velocity and acceleration as described above, and the same reasoning applies to all cases.

### B. Conservative reformulation of the transition problem

We now constrain  $\mathbf{c}(t)$  to be a Bezier curve of degree  $n = 6$ . This is a conservative approximation of the transition problem as it does not cover the whole solution space.

As we already need 6 control points to ensure that we connect exactly the two states, this leaves a free control point  $\mathbf{P}_3 = \mathbf{y}$ :

<sup>1</sup><http://stevetonneau.fr/files/publications/iros18/derivate.py>

$$\mathbf{c}(t, \mathbf{y}) = \sum_{i \in \{0,1,2,4,5,6\}} B_i^6(t/T) \mathbf{P}_i + B_3^6(t/T) \mathbf{y} \quad (4)$$

368 In this case,  $\mathbf{y}$  and  $T$  are the only variables of the problem.  
 369 For the time being, we fix  $T$  to a constant value. We derive  
 370 twice to obtain  $\ddot{\mathbf{c}}(t)$ , and compute the cross product to get the  
 371 expression of  $\mathbf{w}(t)$  :

$$\mathbf{w}(t) = \begin{bmatrix} m(\ddot{\mathbf{c}} - \mathbf{g}) \\ m\mathbf{c} \times (\ddot{\mathbf{c}} - \mathbf{g}) + \dot{\mathbf{L}} \end{bmatrix} \quad (5)$$

372 The non-convexity of the problem disappears, because the  
 373 cross product of  $\mathbf{y}$  by itself is  $\mathbf{0}$ , and all other terms are  
 374 either constant or linear in  $\mathbf{y}$ .  $\mathbf{w}(t, \mathbf{y})$  is thus a six-dimensional  
 375 Bezier curve of degree  $2n - 3$  [32] (9 in this case) linearly  
 376 dependent of  $\mathbf{y}$ :

$$\mathbf{w}(t, \mathbf{y}) = \sum_{i \in \{0..9\}} B_i^9(t/T) \mathbf{w}_i(\mathbf{y}) + \dot{\mathbf{L}}(t) \quad (6)$$

377 where  $\mathbf{w}_i(\mathbf{y}) \in \mathbb{R}^6$  are the control points expressed as :

$$\mathbf{w}_i(\mathbf{y}) = \mathbf{w}_i^y \mathbf{y} + \mathbf{w}_i^s \quad (7)$$

378 The  $\mathbf{w}_i^y \in \mathbb{R}^{6 \times 3}$  and  $\mathbf{w}_i^s \in \mathbb{R}^6$  are constants that only  
 379 depend on the control points  $\mathbf{P}_i$  of  $\mathbf{c}(t)$  and of  $T$ .

380 In what follows, for the sake of simplicity, we assume  
 381  $\dot{\mathbf{L}}(t) = \mathbf{0}$ . This is not a limitation: if we express  $\dot{\mathbf{L}}(t)$  as  
 382 a polynomial in the problem the following reasoning stands.  
 383 One way to include  $\dot{\mathbf{L}}(t)$  is to represent it as a Bezier curve  
 384 with one or more free variables. However guaranteeing that we  
 385 can generate a whole-body motion that tracks a variable  $\dot{\mathbf{L}}(t)$   
 386 requires additional information on the whole-body motion,  
 387 which we leave for future work [19], [33].

388 **The existence of a valid trajectory  $\mathbf{c}(t)$  can thus be**  
 389 **determined by solving a convex problem.**

### 390 C. Application for a motion with no contact switch

391 We first consider the case where  $p = q = 1$ .

392 1) *Continuous formulation:* Using the fact that a Bezier  
 393 curve is comprised in the convex hull of its control points, and  
 394 assuming that the start and goal states are feasible (otherwise  
 395 the problem has no solution), we only need to find a  $\mathbf{y}$   
 396 such that  $\mathbf{y}$  satisfy the kinematic constraints and the control  
 397 points of  $\mathbf{w}(t, \mathbf{y})$  satisfy the dynamic constraints of the contact  
 398 phase (Figure 3). In this case, the whole trajectory necessarily  
 399 satisfies the constraints everywhere, as they form a convex  
 400 set. This problem is thus a linear Feasibility Problem (FP) in  
 401 3 dimensions:

$$\begin{aligned} & \text{find } \mathbf{y} \\ & \text{s. t. } \mathbf{K}^{\{p\}} \mathbf{y} \leq \mathbf{k}^{\{p\}} \\ & (m\mathbf{H}^{\{p\}} \mathbf{w}_i^y) \mathbf{y} \leq \mathbf{h}^{\{p\}} + m\mathbf{H}^{\{p\}} \left( \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} - \mathbf{w}_i^s \right) \quad , \forall i \end{aligned} \quad (8)$$

402 Constraining the control points of  $\mathbf{w}(t)$  to satisfy the  
 403 constraints of the trajectory is *a priori* a conservative approach  
 404 that further constrains the solution space (we will see that this

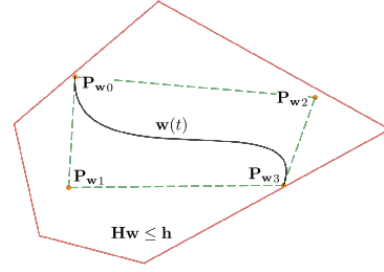


Fig. 3: A bezier curve is comprised in the convex hull of its control points. In this abstract view, the red polygon represents the 6D constraints on  $\mathbf{w}(t)$ . If the control points of  $\mathbf{w}(t)$  satisfy the constraints, then the complete curve satisfies the constraints.

405 limitation can be easily overcome). However, this approach  
 406 allows for a continuous solution to the problem and guarantees  
 407 that the trajectory is entirely valid.

408 2) *Discrete formulation:* Alternatively, we can remove the  
 409 constraint on the control points of  $\mathbf{w}(t)$ , and use a classical  
 410 discretized approach to verify that some of the points of  $\mathbf{w}(t)$   
 411 satisfy the constraints. This approach is less conservative,  
 412 although it increases the dimensionality of the problem, and  
 413 introduces the risk that the constraints be violated between  
 414 two discretization steps. Using a discretization step  $\Delta t$ , we  
 415 discretize  $\mathbf{w}(t, \mathbf{y})$  over  $T$  as follows :

$$\mathbf{w}(j\Delta t, \mathbf{y}) = \mathbf{w}_j^y \mathbf{y} + \mathbf{w}_j^s \quad (9)$$

416 Where  $\mathbf{w}_j^y, \mathbf{w}_j^s$  are constants given by  $\mathbf{P}_{\{0,1,2,4,5,6\}}$ , the  
 417 total duration  $T$  and the time step  $j\Delta t$ .  $j$  belongs to the phase  
 418 set  $J^{\{p\}} : \{j \in \mathbb{N} : 0 \leq j\Delta t \leq T^{\{p\}}\}$ . We can now rewrite  
 419 inequality (2) expressed at the discretization point  $j\Delta t$ :

$$\underbrace{(m\mathbf{H}^{\{p\}} \mathbf{w}_j^y) \mathbf{y}}_{\mathbf{U}_j^{\{p\}}} \leq \underbrace{\mathbf{h}^{\{p\}} + m\mathbf{H}^{\{p\}} \left( \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} - \mathbf{w}_j^s \right)}_{\mathbf{u}_j^{\{p\}}} \quad (10)$$

420 Thus we rewrite FP (8) in a discretized form :

$$\begin{aligned} & \text{find } \mathbf{y} \\ & \text{s. t. } \underbrace{\begin{bmatrix} \mathbf{K}^{\{p\}} \mathbf{c}_j^y \\ \mathbf{U}_j^{\{p\}} \end{bmatrix}}_{\mathbf{E}_j^{\{p\}}} \mathbf{y} \leq \underbrace{\begin{bmatrix} \mathbf{k}^{\{p\}} - \mathbf{K}^{\{p\}} \mathbf{c}_j^s \\ \mathbf{u}_j^{\{p\}} \end{bmatrix}}_{\mathbf{e}_j^{\{p\}}} \quad \forall j \in J^{\{p\}} \end{aligned} \quad (11)$$

### 421 D. Application to a motion with one contact switch

422 We now consider the case where  $q = p + 1$ . In this case we  
 423 define  $T^{\{p\}}$  and  $T^{\{q\}}$  as the time spent in each phase, such  
 424 that  $T = T^{\{p\}} + T^{\{q\}}$ .

425 When a contact switch occurs during a motion, the constraints  
 426 applied to the CoM trajectory change at the switching  
 427 time  $t = T^{\{p\}}$ . When  $t < T^{\{p\}}$ , the constraints of phase  
 428  $\{p\}$  must be applied and conversely, the constraints of phase  
 429  $\{q\}$  must be applied and when  $t > T^{\{p\}}$ . At  $t = T^{\{p\}}$ , the  
 430 constraints of both phases must be applied.

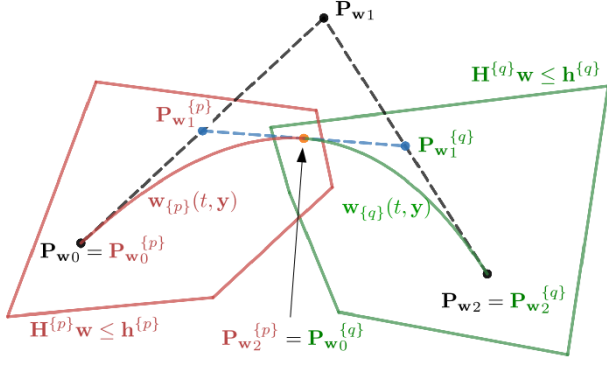


Fig. 4: Example of curve decomposition with the De Casteljau algorithm. The original curve comprises 3 control points (black). It is decomposed into two curves comprising the same number of control points each (3). We can then constrain the control points of the first curve (red) to lie in the first set of constraints, and similarly constrain the control points of the second curve (green) to lie in the second set of constraints. As a result, if the constraints can be satisfied, the connecting control point of both curves satisfies both set of constraints, and we obtain the guarantee that each sub-curve satisfies its respective set of constraints. Interestingly, the control points of the sub-curves are constrained to belong to their respective cones, but those of the original curve can lie outside of the constraints.

431 1) *Continuous formulation:* In this case, since  $w(t)$  spans  
 432 2 distinct sets of linear inequalities, the convex hull of its  
 433 control points is not guaranteed to lie in the constraint set.  
 434 The key idea, and a main contribution with respect to the  
 435 work of Lengagne et al. [24], is to fall back to the case  
 436 where no contact switch occurs, by considering two curves  
 437 that continuously connect at the switching time  $T^{\{p\}}$ . We use  
 438 the De Casteljau algorithm to divide the original curve into two  
 439 curves  $c(t, y)$ , each curve being subject to the constraints of  
 440 their respective contact phase (Figure 4). The result is thus the  
 441 expression of the control points of two Bezier curves  $c_{\{p\}}(t, y)$   
 442 and  $c_{\{q\}}(t, y)$  with the same degree as the original curve, such  
 443 that :

$$\begin{cases} c_{\{p\}}(t, y) = c(t, y) & \forall t \in [0; T^{\{p\}}] \\ c_{\{q\}}(t, y) = c(t, y) & \forall t \in [T^{\{p\}}; T] \end{cases} \quad (12)$$

444 The De Casteljau decomposition guarantees that  
 445  $c_{\{p\}}(T^{\{p\}}, y) = c_{\{q\}}(T^{\{p\}}, y)$ , and that the composition of  
 446 the curves is infinitely differentiable ( $\mathcal{C}^\infty$ ), as it is strictly  
 447 equivalent to  $c(t, y)$ . The control points of the new curves  
 448 are linearly dependent on the control points of the original  
 449 un-split curve, and thus have the form :

$$c_{\{z\}}(t, y) = \sum_{i=0}^n B_i^n(t/T^{\{z\}}) P_i^{\{z\}}(y) \quad \forall z \in \{p, q\} \quad (13)$$

450 where  $P_i^{\{z\}}$  has the form:

$$P_i^{\{z\}}(y) = P_i^{y\{z\}} y + P_i^{s\{z\}} \quad (14)$$

with  $P_i^{y\{z\}}$  and  $P_i^{s\{z\}}$  constants.

It follows that  $w_{\{p\}}(t, y)$  and  $w_{\{q\}}(t, y)$  are also linearly dependent of  $y$ :

$$w_{\{z\}}(t, y) = \sum_{i=0}^n B_i^n(t/T) w_i^{\{z\}}(y) \quad (15)$$

$$\text{with } w_i^{\{z\}}(y) = w_i^{y\{z\}} y + w_i^{s\{z\}} \quad , \forall z \in \{p, q\}$$

Finally the constraints of (8) can be rewritten to deal with the contact switch. The kinematics constraints expressed at each control points are written:

$$\underbrace{\mathbf{K}^{\{z\}} P_i^{y\{z\}}}_{\mathbf{A}_i^{\{z\}}} y \leq \underbrace{\mathbf{k}^{\{z\}} + \mathbf{K}^{\{z\}} P_i^{s\{z\}}}_{\mathbf{a}_i^{\{z\}}}, \forall i, \forall z \in \{p, q\} \quad (16)$$

and the dynamic constraints:

$$\underbrace{(m \mathbf{H}^{\{z\}} w_j^{y\{z\}})}_{\mathbf{D}_j^{\{z\}}} y \leq \underbrace{\mathbf{h}^{\{z\}} + m \mathbf{H}^{\{z\}} \left( \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} - w_j^{s\{z\}} \right)}_{\mathbf{d}_j^{\{z\}}}, \quad (17)$$

$\forall j, \forall z \in \{p, q\}$

We can then stack the constraints:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_0^{\{p\}} \\ \vdots \\ \mathbf{A}_n^{\{p\}} \\ \mathbf{A}_0^{\{q\}} \\ \vdots \\ \mathbf{A}_n^{\{q\}} \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} \mathbf{a}_0^{\{p\}} \\ \vdots \\ \mathbf{a}_n^{\{p\}} \\ \mathbf{a}_0^{\{q\}} \\ \vdots \\ \mathbf{a}_n^{\{q\}} \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} \mathbf{D}_0^{\{p\}} \\ \vdots \\ \mathbf{D}_{2n-3}^{\{p\}} \\ \mathbf{D}_0^{\{q\}} \\ \vdots \\ \mathbf{D}_{2n-3}^{\{q\}} \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}_0^{\{p\}} \\ \vdots \\ \mathbf{d}_{2n-3}^{\{p\}} \\ \mathbf{d}_0^{\{q\}} \\ \vdots \\ \mathbf{d}_{2n-3}^{\{q\}} \end{bmatrix} \quad (18)$$

We recall that in our case  $n = 6$ . Finally, we can rewrite FP (8) with a contact switch as:

$$\begin{aligned} & \text{find } y \\ & \text{s. t. } \mathbf{A}y \leq \mathbf{a} \\ & \quad \mathbf{D}y \leq \mathbf{d} \end{aligned} \quad (19)$$

This boils down to check if each control point of each split curves satisfies the constraints of the current contact phase.

2) *Discrete formulation:* The discrete formulation of the problem is more straightforward: the formulation remains the same, with the only difference that the constraints that must be verified at each discretized point change at  $t = T^{\{p\}}$  and  $t > T^{\{p\}}$ . We thus have 3 sets of constraints in this case: two for each phase, plus one for the transition time  $t = T^{\{p\}}$  where the constraints of both phases apply. We define  $J^{\{q\}} : \{j \in \mathbb{N}, T^{\{p\}} \leq j\Delta t \leq T^{\{q\}}\}$  and obtain the following FP:

$$\begin{aligned} & \text{find } y \\ & \text{s. t. } \mathbf{E}_j^{\{z\}} y \leq \mathbf{e}_j^{\{z\}} \quad , \forall j \in J^{\{z\}}, \forall z \in \{p, q\} \end{aligned} \quad (20)$$

472 *E. General case*

473 In the general case, the same idea will apply. In the contin-  
 474 uous case, we use the De Casteljau algorithm to split  $c(t)$  into  
 475 as many curves as required, thus falling back to a formulation  
 476 with no contact switches. In the discrete case, we assign  
 477 the appropriate constraints for each discretized time step.  
 478 While these decompositions appear mathematically heavy,  
 479 from a programming point of view, they can be automatically  
 480 generated, and thus are in fact simple to implement.

481 In our experiments, we only consider three consecutive  
 482 phases (which correspond to one step), and solve a new  
 483 problem for each subsequent set of phases. We call one such  
 484 convex problem “CROC”, which stands for *Convex Resolution*  
 485 *of Centroidal dynamic trajectories*.

486 *F. Non-conservative continuous formulation*

487 The presented continuous formulation is more conservative  
 488 than the discretized one. Constraining the control points to  
 489 lie inside the constraint set prevents the generation of curves  
 490 such as the one shown in Figure 5. In particular, it is not  
 491 possible for the curve to lie exactly on the constraint set, except  
 492 for the start and end points (because the other control points  
 493 are never reached by definition of a Bezier curve). However,  
 494 coming back to the De Casteljau algorithm, one can make an  
 495 interesting observation. Figure 4 illustrates the fact that while  
 496 the control points of the sub-curves all lie in their respective  
 497 constraint set, one control point of the original curve lies  
 498 outside both sets.

499 When a curve is split, the formulation of the constraints  
 500 changes: they no longer apply to the control points of the  
 501 original curve, but to the control points of the sub-curves. The  
 502 former are no longer constrained to lie in the constraint set  
 503 (although they depend on the control points of the sub-curves).  
 504 In particular, it is then possible to assign the control points of  
 505 the sub-curves exactly on a boundary of the constraint set, and  
 506 as a result the original curve will lie partially on the boundary  
 507 of the constraint set, without crossing it. If the curve is split  
 508 an infinite number of times, it is straightforward to see that  
 509 the original curve can span entirely its original definition set.

510 The price to pay is that the number of constraints increases  
 511 with the number of curve divisions: a curve of degree  $s$  split  
 512  $b$  times comprises  $(s + 1) * (b + 1)$  constraints. The higher the  
 513 number of splits, the more constraints to address. A parallel  
 514 can be made with the discretized approach: the lower the  
 515 discretization step is, the higher the number of constraints is.

516 We believe that a deeper analysis of the pros and cons of  
 517 using a continuous formulation, not only in the case of CROC,  
 518 but with any other formulation of the problem, requires a  
 519 significant amount of research, and thus will be discussed in  
 520 a future paper. In this paper, we only divide the curve at the  
 521 transition points, and we show that this is in practice sufficient  
 522 to perform as well as with the discretized approach, while  
 523 ensuring comparable time performances.

524 *G. Cost function and additional constraints*

525 As the transition feasibility problem is addressed by CROC,  
 526 a feasible COM trajectory is computed. It is possible to

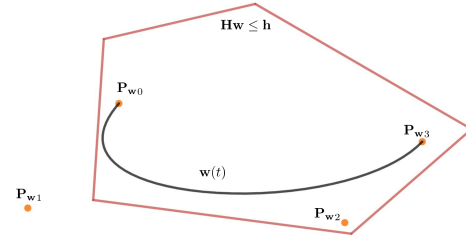


Fig. 5: The curve  $w(t)$  belongs entirely to the convex bound-  
 aries (red), while a control point  $P_{w1}$  lies outside of them.

527 optimize this trajectory to minimize a given cost function  
 528  $l(y)$ , either linear or quadratic. In the latter case problem (19)  
 529 then becomes a Quadratic Program (QP). One can for instance  
 530 minimize the integral of the squared acceleration norm or the  
 531 angular momentum. This cost function is irrelevant to solve  
 532 the transition feasibility problem, but it can be later used as a  
 533 reference COM trajectory for a whole-body motion generator,  
 534 or as an initial guess for a non-linear solver as discussed in  
 535 Section V-A4. The main interest of using a non-linear solver  
 536 with the input of CROC is that the trajectory can then be  
 537 refined globally (while the authors advise to use CROC with  
 538 at most 3 contact phases), at the cost of a higher computational  
 539 burden. Figure 6 provides a trajectory computed with CROC  
 540 and the same trajectory refined with a non-linear solver as an  
 541 illustration of the typical differences of both approaches.

542 The formulation also allows to add inequality constraints  
 543 on  $c$  and any of its derivatives by rewriting the expression of  
 544 the control points of the desired curve as done in equation  
 545 (14). Here again, these constraints can either be verified con-  
 546 tinuously on the concerned control points, or in a discretized  
 547 fashion. In any case, they take the form

$$Oy \leq o \tag{21}$$

548 We use such constraints to impose bounds on the velocity  
 549 and acceleration of the center of mass or on the angular  
 550 momentum variation. The most generic form of our problem  
 551 is thus the generic QP:

$$\begin{aligned} &\text{find } y \\ &\text{min } l(y) \\ &\text{s. t. } Ay \leq a \\ &\quad Dy \leq d \\ &\quad Oy \leq o \end{aligned} \tag{22}$$

552 In our experiments we set constraints on the acceleration  
 553 and velocity and minimize the squared acceleration norm as  
 554 a cost  $l$ . In the remainder of the paper “CROC” refers to the  
 555 problem( 22).

556 *H. Time sampling*

557 To remain convex, we choose not to include the duration  
 558 of each phase  $T^{p}, T^{p+1}$  and  $T^{p+2}$  as variables of  
 559 CROC. We rather sample various combinations of times and  
 560 solve the corresponding QPs in sequence until a solution is



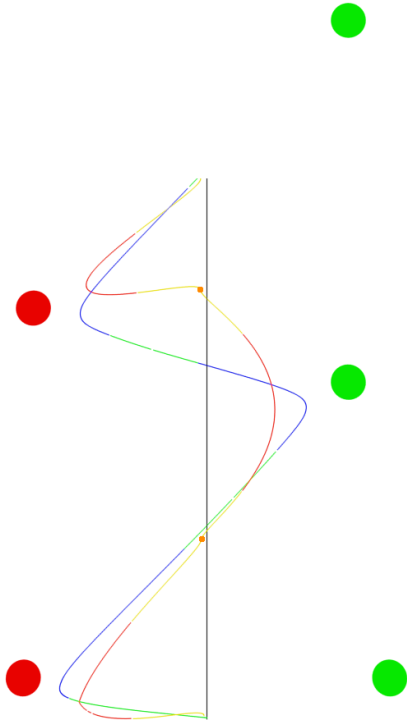


Fig. 6: Example of centroidal trajectories generated with CROC and a non linear-solver (bird eye view), in a case of bipedal walking. The red and green circles represent the contact positions of the (respectively) left and right feet centers over time. The yellow and orange (respectively related to single and double support phases) curve is the curve obtained through the concatenation of curves computed with CROC. The blue and green (respectively related to single and double support phases) curve is obtained through optimization of the latter curve with a non linear solver. The orange squares represent the constrained COM positions resulting from the contact planning phase, which are ignored by the non-linear solver to produce smoother motions.

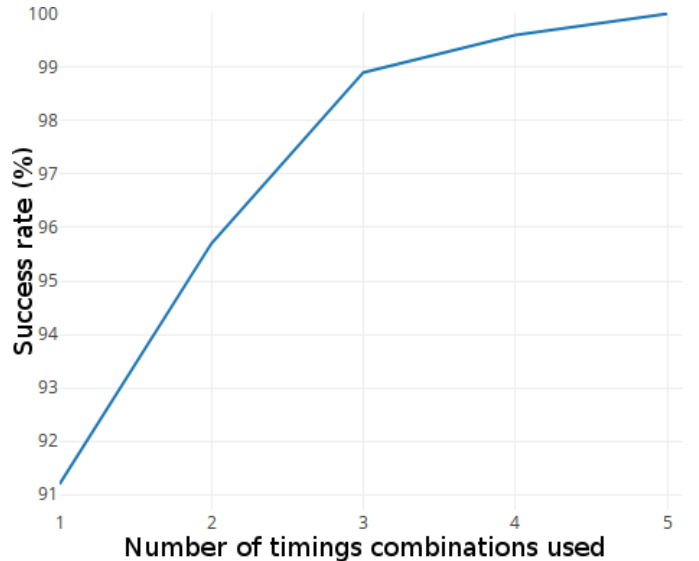


Fig. 7: Evolution of the success rate of CROC according to the number of timings combinations used. Tested on various scenario with coplanar and non-coplanar contacts and with a bipedal and a quadrupedal robots.

combinations are enough to reach 99% of success but that two additional combinations are required to reach exactly 100%.

$T^{\{p\}}$	timings (s)		Success rate (%)
	$T^{\{p+1\}}$	$T^{\{p+2\}}$	
1	0.8	0.8	91.2
1	0.75	0.9	89.2
0.8	0.8	0.9	88.3
0.7	0.5	0.85	77.7
1.2	0.6	1.1	70.8

TABLE I: Success rate with the five used timings combinations.

found. In theory, this would mean that we need to sample an infinity of combinations in order to be complete. However, we pragmatically reduce this number and give up on the completeness while maintaining a high success rate as follows. We sampled a time for each duration phase  $T^{\{z\}}$  by choosing a value between 0.1 and 2 seconds for phases without end-effector motion and between 0.5 and 2 seconds for phases with end-effector motion, with increments of 50ms. For a sequence of three phases with one phase with end-effector motion, this gives a total of 43320 possible combinations. We tested CROC with all these combinations on various problems : with HRP-2 or HyQ robots on flat and non-coplanar surfaces, for several thousands of states.

Upon analysis of the results of the convergence of the QPs, we found out that we can use a small list of timings combinations (5 in our case, shown in table I) that covers 100% of the success cases for all the robots and scenarios tested. We thus solve a maximum of 5 QPs for each validation. Figure 7 shows the evolution of the success rate according to the number of timings combinations used. We observe that 3

#### IV. EXPERIMENTAL FRAMEWORK

Figure 8 shows the complete framework used for our experiments, implemented with the Humanoid Path Planner [34] framework. The inputs are an initial (respectively goal) position and orientation for the root of the robot, as well as a set of bounds on the velocities and acceleration applying to the COM and the end-effector. The output is a dynamically consistent and collision free whole-body motion which can be played on a real robot as shown in section V.

In this paper, we only modify the contact generation method by adding CROC as a feasibility criterion. The other methods and used as black boxes and thus only briefly introduced, with a reference to their respective publications.

##### A. RB-RRT kinodynamic planner

The first block generates a rough guide trajectory<sup>2</sup> for the root of the robot  $\mathbf{x}(t)_{planning}$ . RB-RRT is a planning method based on the sampling-based RRT algorithm, which plans a guide trajectory for the geometric center of a simplified model

<sup>2</sup>This guide is followed exactly to solve  $\mathcal{P}_2$ , but ignored when solving  $\mathcal{P}_3$ .

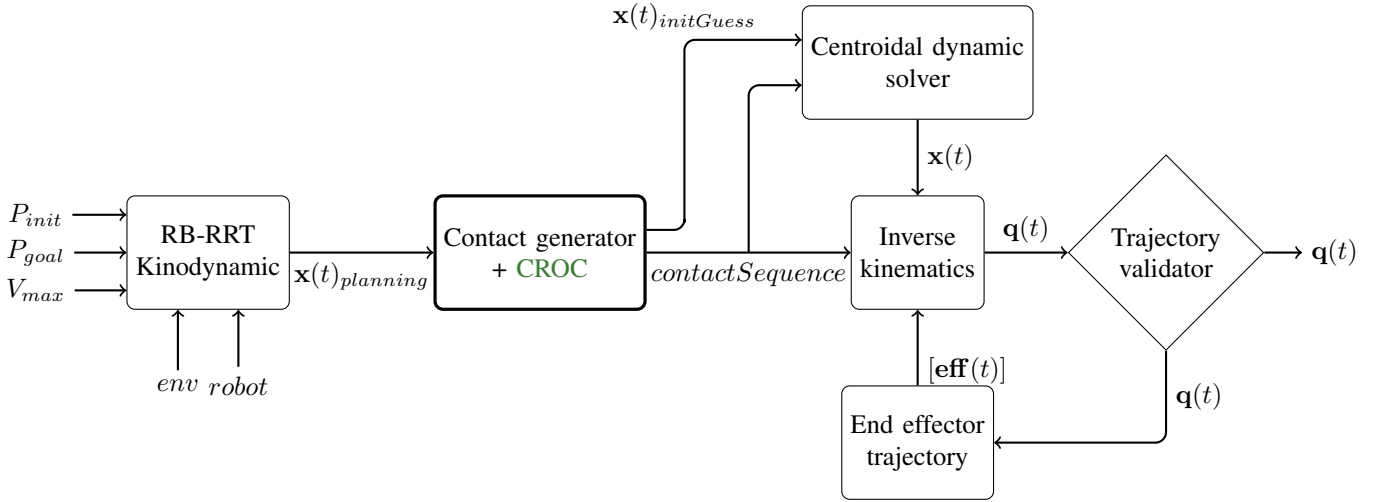


Fig. 8: Complete experimental framework.

601 of the robot. It thus solves a problem of lower dimension  
 602 than planning in the configuration space of the real robot. The  
 603 goal of this method is to find a trajectory for the root of the  
 604 robot which will allow contact creation. This block was first  
 605 presented in [15] and later extended to a kinodynamic version  
 606 in [16], which is the one we use.

#### 607 B. Contact generator with CROC as a feasibility criterion

608 The *contact generator* block computes a contact sequence,  
 609 as a list of whole body postures along the discretized guide  
 610 trajectory  $\mathbf{x}(t)_{planning}$ . It also generates an initial guess of the  
 611 timing of each contact phase. This method was also introduced  
 612 in [15]. CROC is integrated as a feasibility criterion within  
 613 this contact generator. More precisely it is used as a filter to  
 614 determine which transitions are unfeasible and discard them  
 615 during the planning in order to produce contact sequence  
 616 containing only feasible transitions. The integration of CROC  
 617 to this pipeline provides strong guarantees that the computed  
 618 contact sequence will lead to a feasible CoM trajectory and  
 619 thus that the centroidal dynamic solver will converge with this  
 620 contact sequence as input.

621 A byproduct of this test is a feasible CoM trajectory between  
 622 each adjacent contact phases ( $\mathbf{x}(t)_{initGuess}$ ). This trajectory,  
 623 not optimal, is used as a warm-start for a non-linear solver  
 624 which will use it to compute a more optimal trajectory. The  
 625 three different trajectories found in the framework of the figure  
 626 8 are shown in the figure 6,  $\mathbf{x}(t)_{planning}$  is represented in  
 627 black,  $\mathbf{x}(t)_{initGuess}$  in yellow and orange and  $\mathbf{x}(t)$  in green  
 628 and blue.

#### 629 C. Centroidal dynamic solver

630 The *centroidal dynamic solver* block was proposed in [18],  
 631 it takes as input the contact sequence found by the previous  
 632 block, along with an initial guess of the timing of each phases  
 633 and an initial guess of the CoM trajectory. The output of this  
 634 block is a CoM trajectory that respects the centroidal dynamics  
 635 of the robot  $\mathbf{x}(t)$  and minimize a tailored cost function. This

method solves an optimal control problem with a multiple-  
 shooting algorithm implemented in MUSCOD-II [35].

#### D. Inverse kinematics

636 Finally, the whole-body motion  $\mathbf{q}(t)$  is generated with a  
 637 second order Inverse Kinematics solver, similar to [36]. This  
 method takes as input a reference trajectory for the CoM, as  
 641 well as references for the trajectories of the end-effectors.  
 642

#### E. End-effector trajectory

643 In order to automatically generate valid end-effector tra-  
 644 jectories for complex and constrained scenarios, we use a  
 645 dedicated block. The trajectories computed are such that  
 646 the whole limb is collision free and respect the kinematic  
 647 constraint. The trajectories are represented as Bezier curves  
 648 constrained to have a null initial and final velocity, acceleration  
 649 and jerk and which respect velocity, acceleration and jerk  
 650 bounds along the whole trajectory. In order to guarantee that  
 651 the whole surface of the effector creates or breaks the contact  
 652 at the same instant the curves are also constrained to have a  
 653 velocity orthogonal to the contact surface for a small time step  
 654 at the beginning and the end of the trajectory.  
 655

656 The positions of the control points of this Bezier curve  
 657 are computed as the solution of a QP optimization method,  
 658 which is called iteratively to find a compromise between a  
 659 reference optimal trajectory and a collision free one, provided  
 660 by a probabilistic planner. This planner computes a geometric  
 661 path for the moving limb that respects all the kinematic and  
 662 collision constraints but which may present discontinuities in  
 663 velocity and higher derivatives and do not respect the dynamic  
 664 constraints described in the previous paragraph. This path is  
 665 then used in the cost function of our optimization method in  
 666 order to produce a trajectory as smooth as possible and without  
 667 any useless motion while being collision free and respecting  
 668 all the kinematics and dynamics constraints.

## V. RESULTS

### A. Performances of CROC

Computing the success rate of our method is a hard task because we do not have any way to determine the "ground truth" feasibility of a transition (ie. there does not exist any method able to determine in finite time whether there exists a valid centroidal trajectory between the two states). Still, as the goal is to solve the contact planning problem ( $\mathcal{P}_2$ ) in the feasibility domain of the centroidal motion generation problem ( $\mathcal{P}_3$ ), we do not need to compare our method against the ground truth but only against the non linear solver used by  $\mathcal{P}_3$ , presented in section IV-C.

In the table II we show the success rate and the computation time of our method. We compare the method presented in this paper (with the continuous formulation) against the non linear solver with a naive warm start, and the non linear solver with the solution of our method as a warm start when it is available. This last method is considered as our ground truth.

The methods were tested with randomly generated sequences of 3 contact phases such that:

- both initial and final contact phases are in static equilibrium
- both initial and final contact phases have the same number of effectors in contact, between two and four
- there is exactly one contact repositioning between both initial and final contact phases and no other contact variation
- the intermediate contact phase is not required to be in static equilibrium.

We considered two kind of scenarios. In the first case we only sample contact phases with coplanar contacts. In the second case we sampled truly random contacts, which lead to contact phases with non-coplanar contacts and contact sequences that require complex motions. The results are shown in the table II.

All the benchmarks were run on a single core of an Intel Xeon CPU E5-1630 v3 at 3.7Ghz. The QP problems are solved with QuadProg, and the FP problems with GLPK [37].

Method	Coplanar success (%)	Non-coplanar success (%)	Total time (ms)
CROC (DD)	88.4	57.2	3.93
CROC (force)	88.4	57.2	13.01
OCP	100	94.1	$\simeq 150$
OCP (warm start)	100	100	$\simeq 130$

TABLE II: Comparison between CROC and a non linear solver for randomly generated contact sequences of three contact phases. The two first methods are the ones presented in this paper, with the continuous formulation and using either the inequality representation of the dynamic constraints (DD) or the equality representation (force). These methods are compared with the non linear solver presented in [18], either with their naive warm start (OCP) or with the solution found by CROC as a warm start when available (OCP warm start). This last method is used as a "ground truth" for computing the success rate.

Formulation	Metric	Number of contacts		
		2	3	4
DD	DD time (ms)	3.56	14.88	28.16
	Total time (ms)	4.19	16.18	37.41
Force	Total time (ms)	13.01	25.28	49.65

TABLE III: Comparison between the computation times required to generate and solve the FP<sup>3</sup> defined by CROC using either the Double Description (DD) or the Force formulation.

1) *How conservative is our CROC?:* Because of its conservative reformulation, CROC does not cover the whole solution space. As expected, our method find less solutions than the non linear solver used. In the coplanar case, CROC almost finds 90% of the solutions. In the non-coplanar case the centroidal trajectory may be required to present several changes of direction and/or to be really close of the constraints, which explains the difference of success rate between the two cases. However, even in such cases CROC still finds the majority of the solutions.

2) *Computation time:* As claimed in the introduction, CROC is about two order of magnitude faster than the non-linear solver that we are using for the centroidal motion generation. Thanks to this efficiency, it is realistic to use our method during the contact planning to evaluate hundreds of candidate transitions.

For the inequality representation with the double description method, the computation time allocated to solve the QP of equation (22) is extremely fast with  $50\mu s$  on average. The computation time of CROC, which comprises the time required to solve the QP and the time required to compute all the constraints matrices of equation (18) is around  $400\mu s$ . The total time also includes the time required by the double description method. In some use cases, the same contact phases may be used several times and the double description method only needs to be computed once per contact phase, thus the time required for the double description may be factorized.

The major difference between the two representations lies in the dimension of the variables and the constraints of the problem, which is greater in the case of the force formulation. As shown in Table III the computation times between the double description and the force formulations remain in the same order of magnitude for 2 to 4 contacts, with an advantage for the double description. However this advantage reduces as the number of contacts increase. Indeed, while the computation time for the force formulation doubles at each additional contact, the time grows cubically with the Double Description (DD) formulation.

3) *Comparison between continuous and discretized formulation:* Table IV compares four variants of CROC: the discretized version presented in [25] with three different values of number of discretization points per phases and the continuous version presented in this paper. The experimental protocol is the same as in the previous sub-section.

<sup>3</sup>QP and FP give similar times for the DD formulation, while the FP is much more efficient in the Force formulation. This is only an implementation problem, since GLPK exploits the sparsity of the problem while QuadProg does not.

Method	Coplanar		Non-coplanar		Total time (ms)
	Success (%)	Invalid solutions (%)	Success (%)	Invalid solutions (%)	
D (3 pts)	89.7	10.6	61.4	19.7	0.20
D (7 pts)	89.7	6.7	60.6	9.3	0.37
D (15 pts)	89.1	4.2	60.6	6.9	0.75
C	88.4	0	57.2	0	0.41

TABLE IV: Comparison between the method CROC with the discrete formulation (D), with varying number of discretization points, and the continuous formulation (C) presented in this paper. The "ground truth" used to compute the success rate is the non linear solver of [18].

752 The third and fifth columns of Table IV show the percentage  
753 of solutions found that were not dynamically valid. These  
754 tests were made by evaluating the dynamic constraints with  
755 a really small discretization step on the centroidal trajectory  
756 found. Only the discretized version of CROC ([25]) can  
757 find such invalid solutions, and depending of the number  
758 of discretization points used it can reach a non negligible  
759 value. This issue is common to all methods that relies on  
760 discretization and this results emphasizes the fact that we need  
761 a continuous method, able to check exactly whether the whole  
762 trajectory is valid.

763 The drawback of using the continuous formulation proposed  
764 in the section III is that it is more conservative than the  
765 discretized formulation. However, according to our results,  
766 the discretized version found a solution while the continuous  
767 version did not converge only 5.7 % of the times. This number  
768 is similar to the percentage of invalid solutions computed  
769 with the discretized approach, and thus appears favorable.  
770 Moreover, in the section III-D1 we proposed to only split the  
771 trajectory in one curve for each contact phases but it's possible  
772 to split the trajectory in an arbitrary number of curves, as long  
773 as each curve is entirely contained in one contact phases, as  
774 detailed in section III-F. By increasing the number of split  
775 curves, we can further reduce the loss of solutions.

776 4) *Using CROC to warm start a non linear solver:* Choosing  
777 an initial guess for the non linear solver of a trajectory  
778 generation method is essential but may be challenging for  
779 multi-contact motions. The quality of this initial guess has  
780 a significant influence on the convergence of the non linear  
781 solver. For the trajectory generation method used in our  
782 framework, [18] proposed a naive initial guess of the centroidal  
783 trajectory based solely on the position of the contact points and  
784 a predefined height.

785 Interestingly, Table II suggests that the solution set spanned  
786 by CROC is not strictly included in the one spanned by  
787 this non linear solver with this naive initial guess. Using the  
788 solution of CROC to warm start the non linear solver can thus  
789 help it to converge and increase it's success rate. As shown in  
790 Table II, this increase only appears for the non-coplanar case  
791 because the naive initial guess used is always close to a valid  
792 solution in the coplanar case. We expect that the importance  
793 of the initial guess will grow if the contact sequences do not  
794 allow static equilibrium configurations at the contact phases,  
795 and will check this hypothesis in the future.

796 Moreover, by using the solution of CROC to warm start the

797 non linear solver we measured a reduction of the number of  
798 iterations required to converge of 20% on average, reducing  
799 the total computation time (ie it is faster to use CROC than  
800 the non-linear solver, even if CROC fails, than using the non-  
801 linear solver directly).

802 5) *Validity of our kinematic constraints:* As explained in the  
803 section II-C, our representation of the kinematics constraints  
804 is a necessary but not sufficient approximation. In order to  
805 evaluate the accuracy of this approximation, for each feasible  
806 transition found by CROC between random configurations, we  
807 tested explicitly the kinematic feasibility of the centroidal tra-  
808 jectory with an inverse kinematic. This tests showed that 17.5  
809 % of the trajectories found by CROC were not kinematically  
810 valid. This shows that our approximation of the kinematic  
811 constraints is not sufficient. However, this is not a limitation  
812 of our formulation. Indeed, any other linear representation of  
813 the kinematic constraints could be incorporated in our method.

814 Moreover, by doing the same tests without any kinematic  
815 constraints we found a total of 72.3 % of kinematically  
816 unfeasible trajectories, this results show the interest of our  
817 kinematic constraints approximation to greatly improve the  
818 feasibility of the trajectories found by CROC.

## B. Experimental results

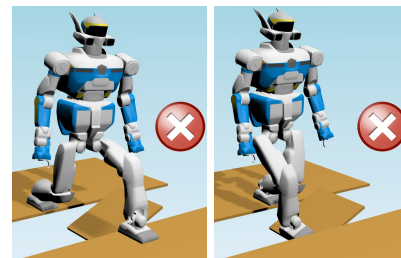


Fig. 9: Unfeasible stepping strategies invalidated by CROC.

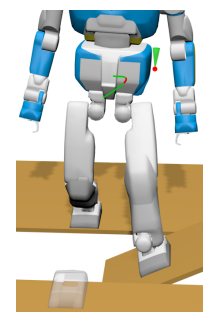


Fig. 10: 3D solution space for CROC (green polytope). The red point is a solution that generates the displayed trajectory.

820 The complete experimental framework presented in the  
821 previous section was tested on several scenarios in semi struc-  
822 tured environments, each scenario showing specific features or  
823 difficulties. We insist that the only manual inputs given to our  
824 framework were an initial and a goal position for the root of  
825 the robot. Most of the obtained motions are demonstrated in  
826 the companion video. They were validated either in a dynamics  
827 simulator or on the real robot.

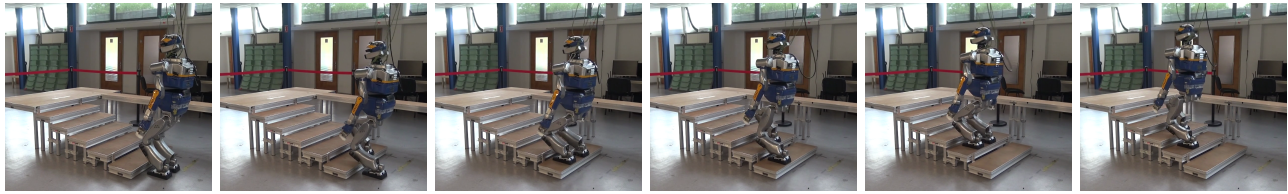


Fig. 11: Snapshots of the motion for the 10cm stairs, the complete motion is shown in the companion video.

828 1) *Inclined platform crossing*: This scenario requires the  
 829 robot to go from one flat platform to the other by taking a step  
 830 in an inclined platform (Figure 1). The scenario is designed  
 831 such that no quasi-static solution exists to the problem, and  
 832 is truly multi-contact in the sense that part of the motion  
 833 occurs entirely on non-flat ground. CROC allows to invalidate  
 834 unfeasible contact sequences that would involve directly taking  
 835 a step on the final platform, or take a step with the right foot  
 836 first (Figure 9). It rather allows to find a solution where the  
 837 left foot is used to step on the inclined platform Figure 1),  
 838 which leads to a feasible whole-body motion demonstrated in  
 839 the companion video.

840 Additionally, CROC also allows to ensure that the left foot  
 841 is positioned in such a way that the problem becomes feasible,  
 842 which is not trivial considering the size of the solution space  
 843 for the chosen step position (Figure 10).

844 2) *10 cm high steps*: This experimental setup is an indus-  
 845 trial set of stairs shown in Figure 11 and 12(a). It consists of  
 846 six 10 cm high and 30 cm long steps. This experiment was  
 847 done with the HRP-2 robot. All the valid contact sequences  
 848 produced contains at least 13 contact phases as the robot is  
 849 kinematically constrained to put both feet on each step.

850 The complete motion is shown in the companion video. The  
 851 crouching walk seen is required to avoid singularities in the  
 852 knee of the extending leg, which are not tolerated by the low-  
 853 level controller.

854 An example of unfeasible contact sequence filtered out by  
 855 our feasibility criterion is depicted on Figure 13. All three  
 856 configurations in this sequence are valid (ie. respect kinematics  
 857 and dynamics constraints) but there isn't any valid centroidal  
 858 trajectory between the last two configurations. Our feasibility  
 859 criterion will filter out this kind of contact transitions during  
 860 contact planning.

861 3) *15 cm high steps with handrail*: This other set of stairs  
 862 is composed of four 15 cm high steps and equipped with  
 863 a handrail. The contact sequence is shown in Figure 12(b)  
 864 and snapshots of the motion are shown in Figure 14. This  
 865 is a typical multi-contact problem, showing a acyclic contact  
 866 sequence with non co-planar contact surfaces. The problem  
 867 was already solved in a previous work [17], but the input  
 868 contact sequence and effector trajectories had to be manually  
 869 selected from a large number of trials. In this paper, the only  
 870 input is a root goal position at the top of the stairs.

871 4) *Flat surface with ground level obstacles*: This exper-  
 872 imental setup consists of a flat floor with obstacles, shown  
 873 in Figure 12(c) and (d). In (c) there is only one obstacle  
 874 in front of the robot's initial position, in (d) we add smaller  
 875 obstacles on the floor. This scenario shows that our planner is  
 876 able to compute a valid guide root trajectory that avoid bigger

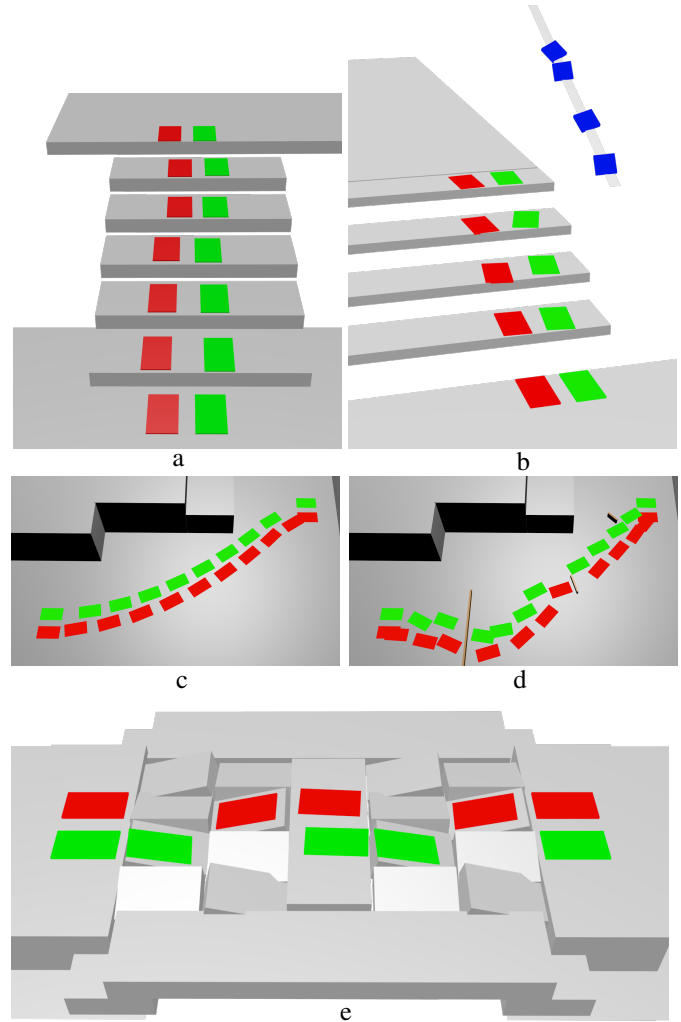


Fig. 12: Examples of contact sequences found with our framework. The color patches represent the planned contact location: green for right foot, red for left foot, blue for right hand.

877 obstacles and that our contact planner is able to avoid collision  
 878 with smaller obstacles on the ground.

879 The difficulty of this scenario lies on the generation of  
 880 collision free feet trajectories. Indeed, some obstacles are small  
 881 enough to permit the feet to pass over the obstacles, but others  
 882 are too high and require a lateral motion of the feet to avoid  
 883 them. As shown in Figure 15 our method presented briefly in  
 884 section IV-E is able to find such trajectories automatically.

885 5) *Uneven platforms*: This setup consists of 30 cm long  
 886 and 20 cm wide platforms, oriented of  $15^\circ$  around either the  $x$

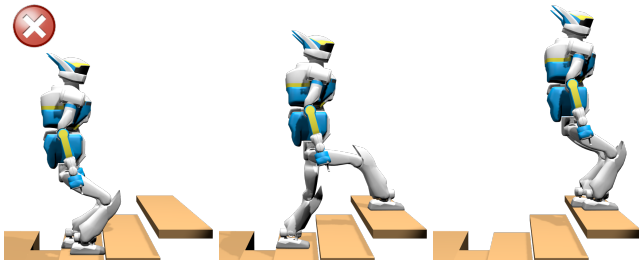


Fig. 13: Example of unfeasible contact transition detected by CROC and rejected during contact planning

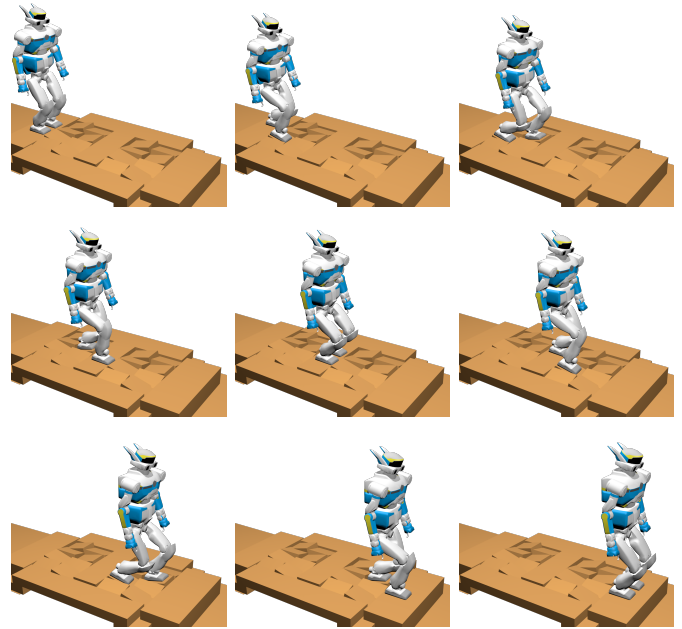


Fig. 16: A feasible contact sequence computed with our contact planner and CROC on uneven platforms.

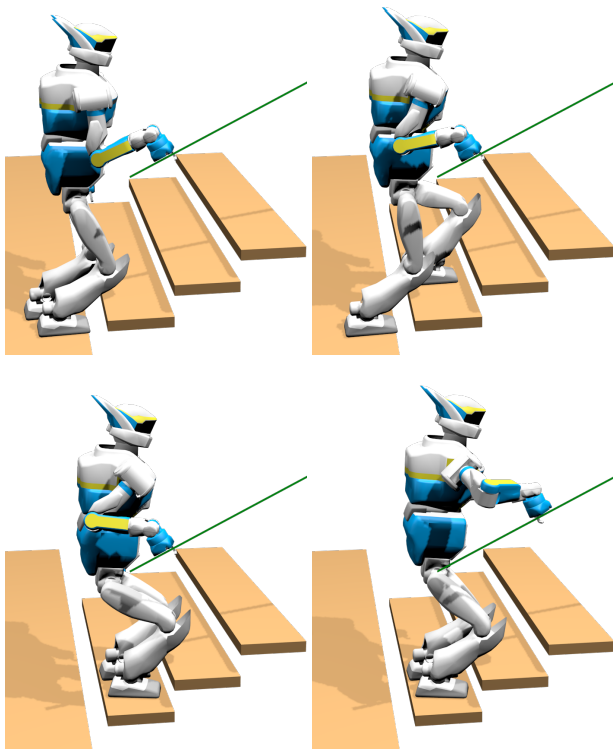


Fig. 14: A feasible multi-contact sequence for a stair climbing with handrail support on the HRP-2 robot automatically computed with our contact planner and CROC.

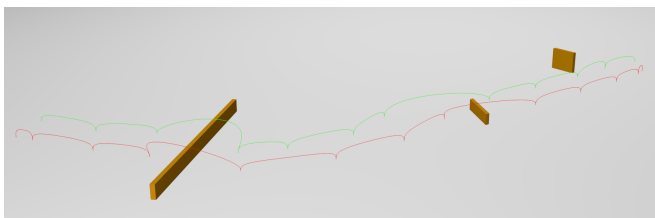


Fig. 15: Feet trajectories computed for scenario with ground level obstacles. Green for right foot and red for left foot.

this, there is really few collision free candidates positions for the feet. The probability of finding a contact position which leads to a collision-free configuration while maintaining the equilibrium is extremely small for this setup.

The contact sequence found is shown in Figure 12(e), snapshots of the motion are shown in figure 16 and a motion for this scenario is shown in the companion video. These motions have been validated in the dynamic simulator OpenHRP.

The Figure 17 shows two examples of unfeasible contact sequence filtered out by CROC in this scenario.

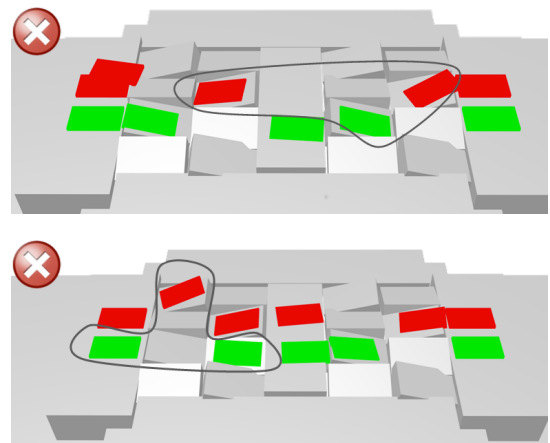


Fig. 17: Examples of unfeasible contact sequences filtered out by CROC. There doesn't exist any valid centroidal trajectory for the contact transitions encircled in black.

887 or  $y$  axis. This scenario is particularly difficult for the contact  
 888 planner because of all the possible collisions generated by the  
 889 feet. We recall that the feet of HRP-2 are 24 cm long for 14  
 890 cm wide, which means that the platforms of this setup are only  
 891 a few centimeters bigger than the feet of the robot. Because of

892  
 893  
 894  
 895  
 896  
 897  
 898  
 899  
 900  
 901

Scenario	Method	success (%) <sup>4</sup>	time (s)	Contact planning		Centroidal trajectory generation success (%)
				n. of candidates (avg.)	n. of configurations (avg.)	
Walk (flat)	Without CROC	100	0.58	8.2	6.3	98
	With CROC	100	0.63	21.9	7.0	100
Stairs (3 steps)	Without CROC	100	0.61	24.4	6.1	52
	With CROC	94	0.82	87.3	7.3	100
Stairs (handrail)	Without CROC	98	1.24	144.3	11.6	31
	With CROC	84	1.57	322.6	13.2	100
platforms	Without CROC	47	1.84	319.2	9.3	15
	With CROC	32	2.43	969.6	9.8	100

TABLE V: Evaluation of the feasibility of the contact plans found with or without CROC as a feasibility criterion. The *Contact Planning* column shows the success rate of the contact planner (ie when it successfully reaches the goal root’s position with a contact sequence), the computation time required, the average number of contact candidates evaluated per runs, and the average number of configurations in contact in the solution. The last column shows the success rate of the centroidal trajectory generation method with the contact sequence found by the planner.

C. Benchmarks

1) *Using CROC as a feasibility criterion:* In order to quantify the improvement of our contact planner from the use of CROC as the feasibility criterion, we used the following test procedure: for some of the scenarios presented in the previous section, we tried to solve the problem using our framework with and without using CROC as a feasibility criterion during the contact planning. We then measured the success rate of the contact planner in both cases, and when it succeeded we tried the centroidal trajectory generation with the contact plan found and measured the success rate of this step. The results are shown in Table V.

In the walking on flat floor scenario, CROC brings only a marginal improvement to our contact planner because our previously used heuristics were sufficient in this case to provide a feasible contact plan most of the time. However, in all the other cases the results empirically prove the main claim of this paper: using CROC as a feasibility criterion during the contact generation greatly increases the success rate of the centroidal trajectory generation because it produce contact plans with only feasible transitions. Another expected result is that there isn’t any ‘false positive’ found by our method: when CROC converges, the non linear solver always converges for the same transition.

The trade-off is a small increase of the computation time required by the contact generator. This is explained partly by the addition of the time required to run CROC for each candidates, but mostly by the fact than we need to evaluate a lot more candidates before we find a valid one (ie. which lead to a feasible transition). This is shown in the column 5 of Table V, which provides the average number of contact candidates evaluated during a run of the contact planner. Another drawback is a decrease of the success rate of the contact generator, explained by the fact that it can get stuck with only unfeasible candidates. But this decrease is only virtual because without CROC the planner could find unfeasible contacts sequences which count as success for the contact planning, while with

<sup>4</sup>The contact planner uses some approximations that may result in failures during the planning. When this occurs in general one can simply restart the planner until a solution is actually found. Thus the success rate is only indicative here. The relevant information is rather the success rate of the trajectory generation.

CROC all success of the contact planning are feasible contact sequences.

2) *Benchmarks of the complete framework:* Table VI shows a benchmark of the performances of the complete motion planning framework presented in section IV. We recall that this framework take as input only an initial and goal position for the center of the robot and produce as output a whole body motion. We observe that the success rate is close to 100% except for complex scenarios where it is still above 80% in the worst case. The main cause of failure in our current implementation of the framework is the inverse kinematics that may produce whole-body motions that do not respect the kinematic constraints or that are in self-collision. Concerning the computation time, in most of the cases we achieve interactive performances (ie. the computation time is smaller than the motion duration). In the worst case the computation time is greater than the motion duration, but only by a small margin.

As shown in Figure 18, the inverse kinematics method is currently the bottleneck of our framework and takes more than 60% of the total computation time, as it requires several iterations to generate collision-free trajectories.

Scenario	Motion duration (s)	Total time (s)	Success (%)
Walk (3 steps)	7.7	4.43	100
Walk with obstacles	55.02	51.5	99.3
Uneven platforms	14.94	17.83	83.5
Stairs	16.23	12.56	90.5
Stairs with handrail	23.13	18.09	88.05

TABLE VI: Performance analysis of the complete motion planning framework presented in section IV, without the time required to compute collision free end-effector trajectory. *Motion duration* is the average duration of the solution, *total time* is the average computation time required to compute the motion. *Success* is the success rate of the complete framework.

VI. CONCLUSION

In this paper we introduce a continuous, accurate and efficient formulation of the centroidal dynamics of a legged robot, named *CROC*. Our method guarantees that it can compute valid centroidal trajectories that do not require discretization, nor use approximation or relaxation of the dynamic constraints. This formulation is convex yet conservative, but not

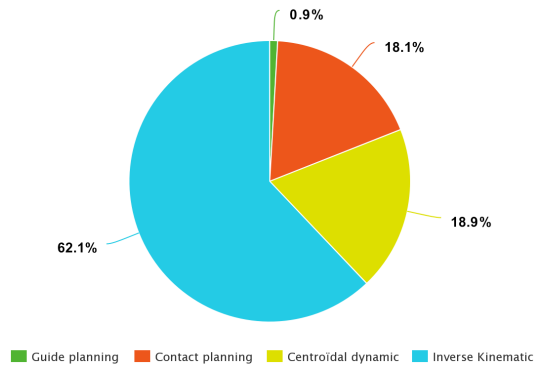


Fig. 18: Division of the computation time among the different methods of the motion planning framework.

limited to quasi-static motions. To our knowledge, this is the first method to combine all these properties.

Thanks to the computational efficiency of our method, requiring only a few milliseconds to solve the centroidal dynamic problem with three contact phases, we can use this method as a feasibility criterion during contact planning. The interest of this feasibility criterion have been shown both qualitatively and empirically, our results show that all the contact plans produced with CROC as a feasibility criterion lead to feasible centroidal dynamic problems. We also show that without using this feasibility criterion, the contact planner find unfeasible contact sequences with a high probability on complex scenarios.

Moreover, the centroidal trajectory produced by CROC can be used to warm-start a non linear solver, resulting in the improvement on the convergence rate and computation time of the non linear solver by comparison to the naive initial guess previously used.

Thanks to the continuous formulation proposed in this paper, we have the guarantee that the whole centroidal trajectory is valid, by opposition to the discretized methods of the state of the art that only guarantee that the discretized points of the trajectory are valid. We showed that the discretization may lead to a non negligible amount of invalid solutions where the trajectory is invalid between two valid discretization points, which emphasizes the interest of a continuous formulation. We believe that this continuous formulation of the constraints on the centroidal trajectory may be useful for all state-of-the-art methods, convex or non-linear. We leave the study of the feasibility and the interest of this application to a future work.

Finally, the feasibility criterion proposed in this paper permits us to complete our locomotion planning framework [11]. In this paper we showed that our framework is able to produce indifferently simple walking motions and multi-contact motions (ie. with non coplanar contacts and acyclic behaviors). These motions were validated in simulation or on the robot HRP-2. We also showed empirically that our framework presents a success rate close to 100% and present interactive computation times (the time required to compute a motion is smaller than the duration of this motion) in the studied scenarios, expect for the most complex scenario where the computation time is approximately 20% greater than the

duration of the motion, but still remain in the same order of magnitude. We believe that with an optimization of the implementation, interactive performances could be achieved even in the worst cases.

For future work we would like to try more complex motions on the real robotic platform, but we are currently limited by the capabilities of our low level controller.

#### A. Handling whole-body approximations and uncertainties

The remaining source of approximation is shared with all centroidal-based methods, and comes from the whole-body constraints (joint limits, angular momentum and torques), which are only approximated or ignored in the current formulation. One solution could be to alternate centroidal optimization with whole-body optimization as other approaches do [19], however for the transition feasibility problem, this approach would result in an increased computational burden that is not compatible with the combinatorial aspect of the search. One way to improve the quality of this approximation is to integrate torque constraints [38], [39]. Expressing such constraints at the CoM level is considered for future work.

#### B. Application to 0 and 1 step capturability

The N-Step capturability problem consists in determining the ability of a robot (in a given state) to come to a stop (ie. null velocity and acceleration) without falling by taking at most N steps. It is used to detect and prevent fall.

We can easily change the constraints on  $c(t)$  defined in subsection III-A to remove the constraint on  $c_g$  and constrain ( $\dot{c}_g = 0, \ddot{c}_g = 0$ ). With this set of constraints, the feasibility of FP (8) determines the 0-Step capturability. Similarly, FP (19) determines the 1-Step capturability.

For future work we would like to empirically determine the accuracy of our method with respect to this problem, using a framework similar to [14].

#### SOURCE CODE

Code available (C++/python) under a BSD-2 license: [https://gitlab.com/stonneau/bezier\\_COM\\_traj](https://gitlab.com/stonneau/bezier_COM_traj)

#### ACKNOWLEDGMENT

Supports: ANR LOCO3D ANR-16-CE33-0003, ERC Actanthrope ERC-2013-ADG, H2020 Memmo ICT-780684.

#### REFERENCES

- [1] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point," in *2003 IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, Sep. 2003.
- [2] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006.
- [3] T. Bretl, "Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem," *The Int. Journal of Robot. Research (IJRR)*, vol. 25, no. 4, pp. 317–342, Apr. 2006. [Online]. Available: <http://dx.doi.org/10.1177/0278364906063979>
- [4] A. Escande, A. Kheddar, and S. Miossec, "Planning contact points for humanoid robots," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 428 – 442, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889013000213>



- 1064 [5] M. X. Grey, A. D. Ames, and C. K. Liu, "Footstep and motion planning  
1065 in semi-structured environments using randomized possibility graphs,"  
1066 in *2017 IEEE International Conference on Robotics and Automation  
1067 (ICRA)*, May 2017, pp. 4747–4753.
- 1068 [6] P. Kaiser, C. Mandery, A. Boltres, and T. Asfour, "Affordance-based  
1069 multi-contact whole-body pose sequence planning for humanoid robots  
1070 in unknown environments," in *IEEE International Conference on  
1071 Robotics and Automation*, 2018.
- 1072 [7] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behav-  
1073 iors through contact-invariant optimization," *ACM Trans. on Graph.*,  
1074 vol. 31, no. 4, pp. 43:1–43:8, 2012.
- 1075 [8] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with  
1076 mixed-integer convex optimization," in *Humanoid Robots (Humanoids),  
1077 14th IEEE-RAS Int. Conf. on*, Madrid, Spain, 2014.
- 1078 [9] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait  
1079 and Trajectory Optimization for Legged Systems through Phase-based  
1080 End-Effector Parameterization," *IEEE Robotics and Automation Letters*,  
1081 pp. 1–1, 2018. [Online]. Available: [http://ieeexplore.ieee.org/document/  
1082 8283570/](http://ieeexplore.ieee.org/document/8283570/)
- 1083 [10] S. Tonneau, A. D. Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard,  
1084 "An efficient acyclic contact planner for multiped robots," vol. 34, no. 3,  
1085 June 2018, pp. 586–601.
- 1086 [11] J. Carpentier, A. Del Prete, S. Tonneau, T. Flayols, F. Forget,  
1087 A. Mifsud, K. Giraud, D. Atchuthan, P. Fernbach, R. Budhiraja,  
1088 M. Geisert, J. Solà, O. Stasse, and N. Mansard, "Multi-contact  
1089 Locomotion of Legged Robots in Complex Environments – The  
1090 Loco3D project," in *RSS Workshop on Challenges in Dynamic Legged  
1091 Locomotion*, Boston, United States, Jul. 2017, p. 3p. [Online]. Available:  
1092 <https://hal.laas.fr/hal-01543060>
- 1093 [12] T. Koolen, T. de Boer, J. R. Rebula, A. Goswami, and J. E. Pratt,  
1094 "Capturability-based analysis and control of legged locomotion, part 1:  
1095 Theory and application to three simple gait models," *I. J. Robotics  
1096 Res.*, vol. 31, no. 9, pp. 1094–1113, 2012. [Online]. Available:  
1097 <https://doi.org/10.1177/0278364912452673>
- 1098 [13] J. E. Pratt, T. Koolen, T. de Boer, J. R. Rebula, S. Cotton, J. Carff,  
1099 M. Johnson, and P. D. Neuhau, "Capturability-based analysis and  
1100 control of legged locomotion, part 2: Application to m2v2, a lower-body  
1101 humanoid," *I. J. Robotics Res.*, vol. 31, no. 10, pp. 1117–1133, 2012.  
1102 [Online]. Available: <https://doi.org/10.1177/0278364912452762>
- 1103 [14] A. Del Prete, S. Tonneau, and N. Mansard, "Zero Step Capturability for  
1104 Legged Robots in Multi Contact," *Accepted on IEEE Trans on Robotics*,  
1105 2018. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01574687>
- 1106 [15] S. Tonneau, A. D. Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard,  
1107 "An efficient acyclic contact planner for multiped robots," *IEEE Trans-  
1108 actions on Robotics*, vol. 34, no. 3, pp. 586–601, June 2018.
- 1109 [16] P. Fernbach, S. Tonneau, A. D. Prete, and M. Taïx, "A kinodynamic  
1110 steering-method for legged multi-contact locomotion," in *IEEE/RSJ  
1111 International Conference on Intelligent Robots and Systems (IROS)*, Sept  
1112 2017, pp. 3701–3707.
- 1113 [17] J. Carpentier, R. Budhiraja, and N. Mansard, "Learning Feasibility  
1114 Constraints for Multi-contact Locomotion of Legged Robots," in  
1115 *Robotics: Science and Systems*, Cambridge, MA, United States, Jul.  
1116 2017. [Online]. Available: <https://hal.laas.fr/hal-01526200>
- 1117 [18] J. Carpentier and N. Mansard, "Multi-contact locomotion of legged  
1118 robots," *Rapport LAAS n 17172*. <https://hal.laas.fr/hal-01520248>. *Con-  
1119 ditionally accepted for IEEE Trans. on Robotics*, 2017.
- 1120 [19] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory gener-  
1121 ation for multi-contact momentum-control," in *Humanoid Robots  
1122 (Humanoids), 15th IEEE-RAS Int. Conf. on*, Nov. 2015.
- 1123 [20] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning  
1124 with centroidal dynamics and full kinematics," in *Humanoid Robots  
1125 (Humanoids), 14th IEEE-RAS Int. Conf. on*, Madrid, Spain, 2014, pp.  
1126 295–302.
- 1127 [21] S. Caron, A. Escande, L. Lanari, and B. Mallein, "Capturability-based  
1128 analysis, optimization and control of 3d bipedal walking," Jan.  
1129 2018, submitted. [Online]. Available: [https://hal.archives-ouvertes.fr/  
1130 hal-01689331](https://hal.archives-ouvertes.fr/hal-01689331)
- 1131 [22] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of  
1132 humanoid momentum dynamics for multi-contact motion generation,"  
1133 in *Proceedings of the 2016 IEEE-RAS International Conference on  
1134 Humanoid Robots*, 2016.
- 1135 [23] G. Mesesan, J. Engelsberger, C. Ott, and A. Albu-Schffer, "Convex prop-  
1136 erties of center-of-mass trajectories for locomotion based on divergent  
1137 component of motion," *IEEE Robotics and Automation Letters*, vol. 3,  
1138 no. 4, pp. 3449–3456, Oct 2018.
- [24] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of  
1139 whole-body optimal dynamic multi-contact motions," *The International  
1140 Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.  
1141
- [25] P. Fernbach, S. Tonneau, and M. Taïx, "Croc: Convex resolution of  
1142 centroidal dynamics trajectories to provide a feasibility criterion for the  
1143 multi contact planning problem," in *IEEE/RSJ International Conference  
1144 on Intelligent Robots and Systems (IROS)*, 2018.  
1145
- [26] K. Fukuda and A. Prodon, *Double description method revisited*. Berlin,  
1146 Heidelberg: Springer Berlin Heidelberg, 1996, pp. 91–111.  
1147
- [27] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a  
1148 humanoid robot," *Autonomous Robots*, vol. 35, no. 2, pp. 161–176, Oct  
1149 2013.  
1150
- [28] Z. Qiu, A. Escande, A. Micaelli, and T. Robert, "Human motions  
1151 analysis and simulation based on a general criterion of stability," in  
1152 *Int. Symposium on Digital Human Modeling*, 2011.  
1153
- [29] S. Caron, Q.-C. Pham, and Y. Nakamura, "Leveraging Cone Double  
1154 Description for Multi-contact Stability of Humanoids with Applications  
1155 to Statics and Dynamics," in *Robotics, Science and Systems (RSS)*, 2015.  
1156
- [30] A. Del Prete, S. Tonneau, and N. Mansard, "Fast Algorithms to Test Rob-  
1157 ust Static Equilibrium for Legged Robots," in *2016 IEEE International  
1158 Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden,  
1159 2016.  
1160
- [31] S. Tonneau, A. D. Prete, J. Pettré, and N. Mansard, "2PAC: Two Point  
1161 Attractors for Center of Mass Trajectories in Multi Contact Scenarios,"  
1162 Sep. 2017, accepted with major revisions for *Trans. on Graphics*.  
1163 [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01609055>
- [32] J. M. ([https://math.stackexchange.com/users/305862/jean marie](https://math.stackexchange.com/users/305862/jean%20marie)), "Is the  
1164 cross product of two bezier curves a bezier curve?" *Mathematics Stack  
1165 Exchange*, uRL:<https://math.stackexchange.com/q/2228976> (version:  
1166 2017-12-10). [Online]. Available: [https://math.stackexchange.com/q/  
1167 2228976](https://math.stackexchange.com/q/2228976)  
1168
- [33] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An  
1169 efficient optimal planning and control framework for quadrupedal loco-  
1170 motion," in *Robotics and Automation (ICRA), 2017 IEEE International  
1171 Conference on*. IEEE, 2017, pp. 93–100.  
1172
- [34] J. Mirabel, S. Tonneau, P. Fernbach, A. K. Seppälä, M. Campana,  
1173 N. Mansard, and F. Lamiroux, "Hpp: A new software for constrained  
1174 motion planning," in *2016 IEEE/RSJ International Conference on Intel-  
1175 ligent Robots and Systems (IROS)*, Oct 2016, pp. 383–389.  
1176
- [35] "An efficient multiple shooting based reduced sqp strategy for  
1177 large-scale dynamic process optimization. part 1: theoretical aspects,"  
1178 *Computers and Chemical Engineering*, vol. 27, no. 2, pp. 157 – 166,  
1179 2003. [Online]. Available: [http://www.sciencedirect.com/science/article/  
1180 pii/S0098135402001588](http://www.sciencedirect.com/science/article/pii/S0098135402001588)  
1181
- [36] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soares, and J. Y.  
1182 Fourquet, "Dynamic whole-body motion generation under rigid contacts  
1183 and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29,  
1184 no. 2, pp. 346–362, April 2013.  
1185
- [37] A. Makhorin, "Glpk (gnu linear programming kit)," [http://www.gnu.  
1186 org/software/glpk/glpk.html](http://www.gnu.org/software/glpk/glpk.html), 2008.  
1187
- [38] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. G. Caldwell, and  
1188 C. Semini, "Application of wrench based feasibility analysis to the online  
1189 trajectory optimization of legged robots," *IEEE Robotics and Automation  
1190 Letters*, pp. 1–1, 2018.  
1191
- [39] V. Samy, S. Caron, K. Bouyarmane, and A. Kheddar, "Post-impact  
1192 adaptive compliance for humanoid falls using predictive control of a  
1193 reduced model," in *2017 IEEE-RAS 17th International Conference on  
1194 Humanoid Robotics (Humanoids)*, Nov 2017, pp. 655–660.  
1195  
1196