



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. 2019. hal-01894869v2

HAL Id: hal-01894869

<https://laas.hal.science/hal-01894869v2>

Preprint submitted on 8 Aug 2019 (v2), 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 Taix

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 (Convex Resolution Of Centroidal dynamic trajectories). 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

THIS paper is concerned with the issue of planning multi-contact motions for legged robots in human environments.

The term “multi-contact motion” has been proposed to distinguish the problem from the gaited locomotion one [1], [2]. Gaited motions result from the contact interactions created and broken periodically between the end effectors and a flat terrain. The multi-contact problem is more general as it can include non horizontal contacts, and is not restricted to a cyclic strategy. This results in a combinatorial problem in the choice of the contacts being created. It also requires a more complex formulation of the dynamics that govern the motion. This non-linear problem remains open to this date.

Steve Tonneau is at the University of Edinburgh, Scotland.

The other authors are with LAAS-CNRS / Université de Toulouse, France. e-mail: pfernba@laas.fr

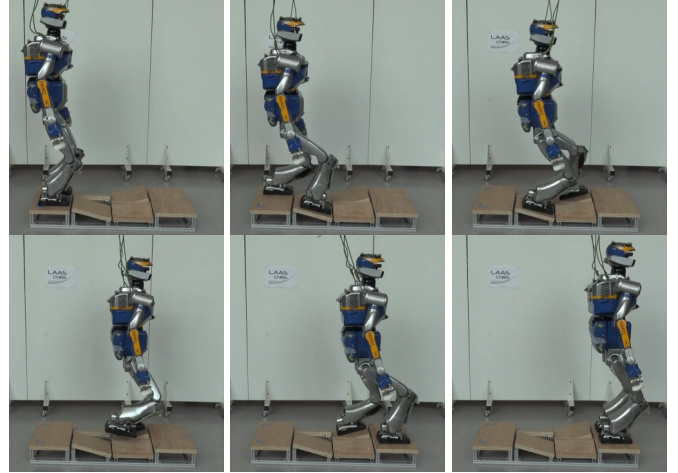


Fig. 1: An instance of the transition feasibility problem: can we guarantee that the contact sequence shown in this picture can be used to produce a feasible motion for the robot? To address this issue in this example we need to account for 9 different contact phases (including phases where the effector is flying, as displayed in the fourth image).

One key issue of multi-contact locomotion consists in choosing contact locations such that the contacts can be broken or created at a given time without violating dynamic or geometric constraints. To tackle this issue one option is to simultaneously optimize the contact locations and the motion of the robot. The problem is non-linear, though promising results have been obtained using approximations [3]–[6]. Such approximations include ignoring collision avoidance or considering a point-mass model.

The present paper lies in the continuity of an alternate approach that decomposes the problem into a sequence of smaller ones [7]–[11]. In such approaches, the computation of a contact plan is achieved prior to the motion generation. This simplifies the problem, but introduces the question of the validity (feasibility) of the contact plan.

Those approaches thus 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 formulate additional constraints on the centroidal dynamics, that would take into account the whole-body constraints.

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

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

A. The transition feasibility in a divide and conquer context

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

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

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

Recent contributions have proposed centroidal trajectory generation methods that could theoretically be used to answer the transition feasibility problem [18]–[20]. However, because of the combinatorial aspect of contact planning, the computational 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 constraints inside a mixed-integer programming problem [4], 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. Contributions

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 trajectory. 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 a computed contact sequence will lead to a feasible whole body motion. This also results in a major technical contribution, as we obtain and demonstrate a framework able to automatically and robustly generate complex motions, both in simulation and on the real HRP-2 robot. The framework is an extension of our previous works: [15] [16] for \mathcal{P}_1 and \mathcal{P}_2 , and [17] [18] for \mathcal{P}_3 .

This paper is organized as following. In section II, we recall the formal definition of the problem. The main contribution of the paper is presented in section III. We then introduce our complete framework in section V. Finally, we present our experimental results in section VI.

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 introduce a convex optimization method

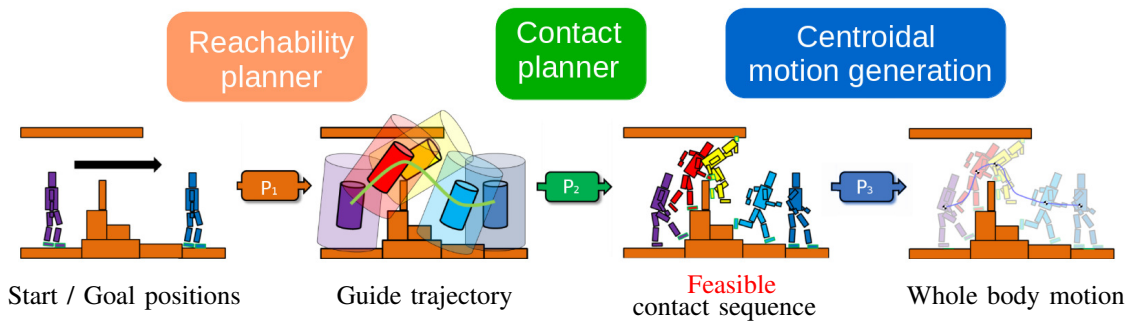


Fig. 2: Complete framework overview of our decoupled approach. In this work we only focus on addressing the transition feasibility problem, from \mathcal{P}_2 to \mathcal{P}_3 .

182 to solve the transition feasibility problem. Our previous formu-
 183 lation, as others in the community, is limited by the necessity
 184 to use of the double description method [26], an unstable
 185 mean to compute the linear constraints that apply to the
 186 problem [18], which allows for fast computations. As for
 187 all existing methods, it also requires a discretization of the
 188 solution trajectory, such that the constraints of the problem are
 189 only checked at specific instants. This behavior is unsafe as
 190 the trajectory between each discretization point is unchecked
 191 and may not respect the constraints, as illustrated in Figure 3.
 192 In [25], we proposed a continuous formulation of the problem
 193 in the simple case of a motion without contact changes,
 194 where the trajectory is linearly constrained. In this paper, we
 195 extend this continuous formulation to the general setting of
 196 a motion with any number of contact transitions. As we will
 197 see, this extension is not trivial as it requires handling the
 198 non-convex constraint of belonging to a union of polytopes. In
 199 addition, this formulation removes the need for discretization
 200 of the centroidal trajectory, guaranteeing that the constraints
 201 are respected along the whole trajectory. This continuous
 202 formulation is also fast enough to avoid using the double
 203 description method. The computational gain results from the
 204 lower number of variables required to satisfy the constraints.

205 We advocate for its adoption for any centroidal generation
 method.

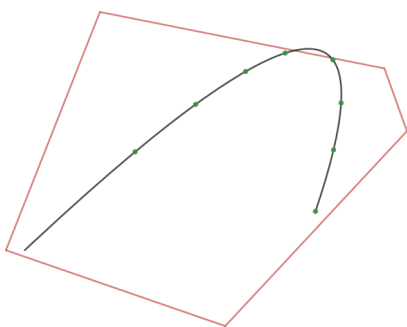


Fig. 3: Example of an invalid solution found by a discretized method. The red lines represent the constraints, while the black curve is the solution and the green dots are the discretization points. All the discretization points satisfy the constraints while the curve clearly violates them.

207 Sections II and III present important similarities with respect
 208 to [25]. The novelty appears from section III-D, where we
 209 present a continuous formulation able to deal with contact
 210 switching during the trajectory.

211 The other sections of the paper are also novel. These
 212 novelties include the completion of our experimental frame-
 213 work, which enables us to validate our method on several
 214 experiments on the real robot. We also provide an empirical
 215 analysis of the performances of our method with respect to a
 216 state-of-the-art nonlinear solver, in terms of success rate and
 217 computation times.

218 II. PROBLEM DEFINITION

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

230 Thus, in this work we define the transition feasibility
 231 problem with respect to the centroidal dynamics of a robot, as
 232 now commonly done in the legged robotics community [27],
 233 [19], [18]. In this section we provide some formal definitions
 234 that are used in the rest of the paper.

235 A. Contact sequence and state

236 A legged motion can be discretized into a sequence of
 237 contact phases. Each contact phase defines a number of
 238 active contacts, and their locations remain constant during
 239 the phase. Thus, each contact phase constrains kinematically
 240 and dynamically the motion of the robot. Within a contact
 241 sequence, each adjacent contact phase differs by exactly one
 242 contact creation or removal (for instance when walking, the
 243 contact sequence is gaited and alternates simple and double
 244 support phases). The considered contact surfaces are assumed
 245 to be rectangular (4 extreme points on each foot) for humanoid
 246 robots, and punctual for quadrupedal robots.

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 the triplet describing the COM position, velocity and acceleration. To indicate that a state is compatible with the dynamic and kinematic constraints associated with a contact phase $p \in \mathbb{N}$, we use the superscript notation $\mathbf{x}^{\{p\}} = (\mathbf{c}^{\{p\}}, \dot{\mathbf{c}}^{\{p\}}, \ddot{\mathbf{c}}^{\{p\}})$.

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

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

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

1) *Equality constraint representation (or force formulation)*: The Newton-Euler equations are:

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

Where :

- m is the total mass of the robot;
- nc is the number of contact points;
- $\mathbf{p}_i \in \mathbb{R}^3, 1 \leq i \leq nc$ is the location of the i -th contact point;¹
- $\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{nc}]^T \in \mathbb{R}^{3nc}$ is the stacked vector of contact forces applied at each contact point;
- $\mathbf{g} = [0 \ 0 \ -9.81]^T$ is the gravity vector;
- $\dot{\mathbf{L}} \in \mathbb{R}^3$ is the time derivative of the angular momentum (expressed at \mathbf{c}).
- $\hat{\mathbf{p}}_i$ 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 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 applies both to the center of mass and the angular momentum quantities. With this formulation, the force variables disappear:

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

where \mathbf{H} and \mathbf{h} are respectively a matrix and a vector defined by the position of the contact points, their normal and their

¹As commonly done, in the case of rectangular contacts (like most robot’s feet) we define a contact point at each vertex of the rectangle.

friction coefficients. As this matrix and vector are uniquely defined for a contact phase, we note them with the superscript $\{p\}$ for a contact phase p .

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 with both formulations.

3) *The dynamic constraints are not convex*: Because of the cross product between \mathbf{c} and $\ddot{\mathbf{c}}$ in the equations (1) and (2), the constraints are bi-linear, leading to a non-convex problem to solve.

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

Each active contact also introduces a kinematic constraint on $\mathbf{c}(t)$, depending of the placement of the end-effectors of the robot. We use a linear constraint formulation to represent this constraint depending on the 6D positions of each active contact frames. They give us a necessary but not sufficient condition for the kinematic feasibility (evaluated and discussed in section IV-G). We refer the reader to [31] for the computation of these constraints. For a given contact phase $\{p\}$ this constraint can be expressed as :

$$\mathbf{K}^{\{p\}} \mathbf{c} \leq \mathbf{k}^{\{p\}} \quad (3)$$

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 operation $\mathbf{c} \times \ddot{\mathbf{c}}$. 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 will not be violated in between. Figure 3 illustrates the violation of the constraints.

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 :

$$\mathbf{c}(t) = \sum_{i=0}^n B_i^n(t/T) \mathbf{P}_i \quad (4)$$

343 where the B_i^n are the Bernstein polynomials and the \mathbf{P}_i are
344 the control points.

345 With this formulation we can easily constrain the initial
346 or final positions, velocity or any other derivatives by setting
347 the value of the control points. To exactly connect two states
348 $\mathbf{x}_s = (\mathbf{c}_s, \dot{\mathbf{c}}_s, \ddot{\mathbf{c}}_s)$ and $\mathbf{x}_g = (\mathbf{c}_g, \dot{\mathbf{c}}_g, \ddot{\mathbf{c}}_g)$, we thus need at least
349 6 control points to ensure that the following constraints are
350 verified:

- 351 • $\mathbf{P}_0 = \mathbf{c}_s$ and $\mathbf{P}_n = \mathbf{c}_g$ guarantee that the trajectory starts
352 and ends at the desired locations;
- 353 • $\mathbf{P}_1 = \frac{\dot{\mathbf{c}}_s T}{n} + \mathbf{P}_0$ and $\mathbf{P}_{n-1} = \mathbf{P}_n - \frac{\dot{\mathbf{c}}_g T}{n}$ guarantee that
354 the trajectory initial and final velocities are respected;
- 355 • $\mathbf{P}_2 = \frac{\ddot{\mathbf{c}}_s T^2}{n(n-1)} + 2\mathbf{P}_1 - \mathbf{P}_0$ and
356 $\mathbf{P}_{n-2} = \frac{\ddot{\mathbf{c}}_g T^2}{n(n-1)} + 2\mathbf{P}_{n-1} - \mathbf{P}_n$ guarantee that the initial
357 and final accelerations are respected.

358 Depending on the considered problem, some constraints
359 on the boundary positions, velocities or accelerations can be
360 removed, without changing the validity of our approach. For
361 instance, if the objective is simply to put the robot to a stop,
362 the end velocities and accelerations can be set to zero, while
363 the end position is left unconstrained. We can also extend this
364 to any degree and add constraints on initial or final jerk or
365 higher derivatives and automatically compute the position of
366 the control points with a symbolic calculus script such as the
367 one that we provide at the url ². We only need to compute the
368 equation of the control points once and for all so we do not
369 need to compute them at runtime. In the following equations,
370 we use a curve of degree 6 with the constraints on initial and
371 final position, velocity and acceleration as described above,
372 and the same reasoning applies to all cases.

373 B. Conservative reformulation of the transition problem

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

377 As we already need 6 control points to ensure that we
378 connect exactly the two states, this leaves a free control point
379 $\mathbf{P}_3 = \mathbf{y}$:

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

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

$$\mathbf{w}(t) = \begin{bmatrix} m(\ddot{\mathbf{c}} - \mathbf{g}) \\ m\mathbf{c} \times (\ddot{\mathbf{c}} - \mathbf{g}) + \dot{\mathbf{L}} \end{bmatrix} \quad (6)$$

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

$$\mathbf{w}(t, \mathbf{y}) = \sum_{i \in \{0..9\}} B_i^9(t/T) \mathbf{P}_{\mathbf{w}i}(\mathbf{y}) + \dot{\mathbf{L}}(t) \quad (7)$$

389 where $\mathbf{P}_{\mathbf{w}i}(\mathbf{y}) \in \mathbb{R}^6$ are the control points of $\mathbf{w}(t, \mathbf{y})$
390 expressed as :

$$\mathbf{P}_{\mathbf{w}i}(\mathbf{y}) = \mathbf{P}_{\mathbf{w}i}^y \mathbf{y} + \mathbf{P}_{\mathbf{w}i}^s \quad (8)$$

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

393 In what follows, for the sake of simplicity, we assume
394 $\dot{\mathbf{L}}(t) = \mathbf{0}$. This is not a limitation: if we express $\dot{\mathbf{L}}(t)$ as
395 a polynomial in the problem the following reasoning stands.
396 One way to include $\dot{\mathbf{L}}(t)$ is to represent it as a Bezier curve
397 with one or more free variables. However guaranteeing that we
398 can generate a whole-body motion that tracks a variable $\dot{\mathbf{L}}(t)$
399 requires additional information on the whole-body motion,
400 which we leave as future work [19], [33], [34].

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

403 C. Application for a motion with no contact switch

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

405 1) *Discrete formulation:* Using a discretization step Δt , we
406 discretize $\mathbf{c}(t, \mathbf{y})$ and $\mathbf{w}(t, \mathbf{y})$ over T as follows:

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

407 Where \mathbf{c}_j^y , \mathbf{c}_j^s , \mathbf{w}_j^y and \mathbf{w}_j^s are constants given by
408 $\mathbf{P}_{\{0,1,2,4,5,6\}}$, the total duration T and the time step $j\Delta t$. j
409 belongs to the phase set $J^{\{p\}} : \{j \in \mathbb{N} : 0 \leq j\Delta t \leq T^{\{p\}}\}$.
410 Given these expressions, we can replace $\mathbf{w}(t)$ in (2) by its
411 value at each discretization point $j\Delta t$:

$$\mathbf{H}^{\{p\}} \mathbf{w}_j^y \mathbf{y} \leq \mathbf{h}^{\{p\}} - \mathbf{H}^{\{p\}} \mathbf{w}_j^s \quad (10)$$

412 By proceeding similarly for the kinematic constraint (3), we
413 can formulate the following linear feasibility problem (FP) in
414 3 dimensions:

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

415 With this discrete formulation the number of constraints in
416 the problem is proportional to the number of discretization
417 points. Moreover, the constraints are verified only at the
418 discretization points, which leaves a risk that a part of the
419 solution trajectory between two discretization points does not
420 satisfy the constraints of the problem (Figure 3). Choosing the
421 number of discretization steps is thus a compromise between
422 the computation time (which depends on the number of
423 constraints) and the risk of finding a solution partially invalid.
424 This is a well-known issue when relying on discretization
425 methods.

²<http://stevetonneau.fr/files/publications/irosl8/derivate.py>

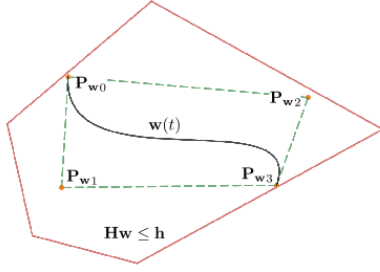


Fig. 4: A bezier curve is comprised in the convex hull of its control points. In this abstract view, the red polygon represents the 6D constraints on $\mathbf{w}(t)$. If the control points $\mathbf{P}_{\mathbf{w}i}$ of $\mathbf{w}(t)$ satisfy the constraints, then the complete curve satisfies the constraints.

2) *Continuous formulation:* Alternatively, in [25] we proposed a continuous formulation of this problem, only valid for the case where no contact transition occurs. We recall this formulation below as it is fundamental for the following section.

Using the fact that a Bezier curve is comprised in the convex hull of its control points, the main idea of this formulation is to express the kinematic constraints (3) on the control points \mathbf{P}_i of $\mathbf{c}(t, \mathbf{y})$ and the dynamic constraints (2) on the control points $\mathbf{P}_{\mathbf{w}i}(\mathbf{y})$ of $\mathbf{w}(t, \mathbf{y})$ (see Figure 4). Constraining the control points of $\mathbf{w}(t, \mathbf{y})$ to satisfy the constraints of the trajectory is *a priori* a conservative approach that further constrains the solution space (we will see that this limitation can be easily overcome). However, this approach allows for a continuous solution to the problem and guarantees that the trajectory is entirely valid.

Assuming that the start and goal states are feasible (otherwise the problem has no solution), for the kinematic constraints we only need to find a \mathbf{y} that satisfies the constraints. For the dynamic constraints all the control points $\mathbf{P}_{\mathbf{w}i}(\mathbf{y})$ must satisfy the equation (2), given the expression (8) we can express the dynamic constraints as follow:

$$\mathbf{H}^{\{p\}} \mathbf{P}_{\mathbf{w}i}^y \mathbf{y} \leq \mathbf{h}^{\{p\}} - \mathbf{H}^{\{p\}} \mathbf{P}_{\mathbf{w}i}^s, \forall i \in [0, 2n - 3] \quad (12)$$

Finally, we can reformulate the discretized Linear Feasibility Problem (11) in a continuous fashion:

$$\begin{aligned} \text{find } & \mathbf{y} \\ \text{s. t. } & \mathbf{K}^{\{p\}} \mathbf{y} \leq \mathbf{k}^{\{p\}} \\ & \mathbf{H}^{\{p\}} \mathbf{P}_{\mathbf{w}i}^y \mathbf{y} \leq \mathbf{h}^{\{p\}} - \mathbf{H}^{\{p\}} \mathbf{P}_{\mathbf{w}i}^s, \forall i \end{aligned} \quad (13)$$

In this case, the whole trajectory necessarily satisfies the constraints everywhere, as they form a convex set.

D. Application to a motion with one contact switch

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

When a contact switch occurs during a motion, the constraints applied to the COM trajectory change at the switching time $t = T^{\{p\}}$. When $t < T^{\{p\}}$, the constraints of phase

$\{p\}$ must be applied and conversely, the constraints of phase $\{q\}$ must be applied and when $t > T^{\{p\}}$. At $t = T^{\{p\}}$, the constraints of both phases must be applied.

1) *Discrete formulation:* Adapting the discretized FP (11) to this case is 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: one for each of the two phases, plus one for the transition time $t = T^{\{p\}}$ where the constraints of both phases apply. We define $J^{\{q\}} : \{j \in \mathbb{N}, T^{\{q-1\}} \leq j\Delta t \leq T^{\{q\}}\}$ and obtain the following FP:

$$\begin{aligned} \text{find } & \mathbf{y} \\ \text{s. t. } & \mathbf{E}_j^{\{z\}} \mathbf{y} \leq \mathbf{e}_j^{\{z\}}, \forall j \in J^{\{z\}}, \forall z \in \{p, q\} \end{aligned} \quad (14)$$

2) *Continuous formulation:* In this case, since $\mathbf{w}(t)$ spans 2 distinct sets of linear inequalities, the convex hull of its control points is not guaranteed to lie in the constraint set. The key idea, unlike Lengagne et al. [24], is to fall back to the case where no contact switch occurs, by considering two curves that continuously connect at the switching time $T^{\{p\}}$. A similar approach has been proposed before, in the context of UAVs [35], with the difference that in our case the continuity of the trajectory is guaranteed by the De Casteljau decomposition algorithm. This algorithm divides the original curve into two curves $\mathbf{c}(t, \mathbf{y})$, each curve being subject to the constraints of their respective contact phase (see Figure 5). The result is thus the expression of the control points of two Bezier curves $\mathbf{c}_{\{p\}}(t, \mathbf{y})$ and $\mathbf{c}_{\{q\}}(t, \mathbf{y})$ with the same degree as the original curve, such that :

$$\begin{cases} \mathbf{c}_{\{p\}}(t, \mathbf{y}) = \mathbf{c}(t, \mathbf{y}) & \forall t \in [0; T^{\{p\}}] \\ \mathbf{c}_{\{q\}}(t, \mathbf{y}) = \mathbf{c}(t, \mathbf{y}) & \forall t \in [T^{\{p\}}; T] \end{cases} \quad (15)$$

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

$$\mathbf{c}_{\{z\}}(t, \mathbf{y}) = \sum_{i=0}^n B_i^n(t/T^{\{z\}}) \mathbf{P}_i^{\{z\}}(\mathbf{y}) \quad \forall z \in \{p, q\} \quad (16)$$

where the $\mathbf{P}_i^{\{z\}}(\mathbf{y})$ have the form:

$$\mathbf{P}_i^{\{z\}}(\mathbf{y}) = \mathbf{P}_i^{y\{z\}} \mathbf{y} + \mathbf{P}_i^{s\{z\}} \quad (17)$$

with $\mathbf{P}_i^{y\{z\}}$ and $\mathbf{P}_i^{s\{z\}}$ constants.

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

$$\mathbf{w}_{\{z\}}(t, \mathbf{y}) = \sum_{j=0}^{2n-3} B_j^{2n-3}(t/T^{\{z\}}) \mathbf{P}_{\mathbf{w}j}^{\{z\}}(\mathbf{y}) \quad (18)$$

with $\mathbf{P}_{\mathbf{w}j}^{\{z\}}(\mathbf{y}) = \mathbf{P}_{\mathbf{w}j}^{y\{z\}} \mathbf{y} + \mathbf{P}_{\mathbf{w}j}^{s\{z\}}$, $\forall z \in \{p, q\}$

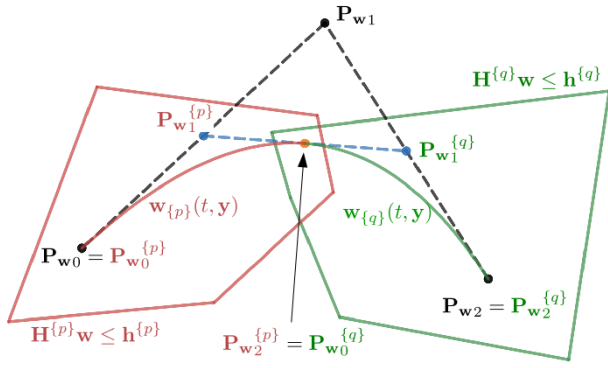


Fig. 5: Example of curve decomposition with the De Casteljau algorithm. The original curve comprises 3 control points (black). It is decomposed into two curves comprising the same number of control points each (3). We can then constrain the control points of the first curve (red) to lie in the first 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.

Finally the constraints of (13) can be rewritten to deal with the contact switches. The kinematic constraints expressed at each control points are written:

$$\underbrace{\mathbf{K}^{\{z\}} \mathbf{P}_i^{y\{z\}}}_{\mathbf{A}_i^{\{z\}}} \mathbf{y} \leq \underbrace{\mathbf{k}^{\{z\}} + \mathbf{K}^{\{z\}} \mathbf{P}_i^{s\{z\}}}_{\mathbf{a}_i^{\{z\}}}, \forall i, \forall z \in \{p, q\} \quad (19)$$

and the dynamic constraints:

$$\underbrace{(\mathbf{H}^{\{z\}} \mathbf{P}_{w_j}^{y\{z\}})}_{\mathbf{D}_j^{\{z\}}} \mathbf{y} \leq \underbrace{\mathbf{h}^{\{z\}} - \mathbf{H}^{\{z\}} \mathbf{P}_{w_j}^{s\{z\}}}_{\mathbf{d}_j^{\{z\}}}, \quad (20)$$

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

We can then stack the constraints:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_0^{\{p\}} \\ \vdots \\ \mathbf{A}_n^{\{p\}} \\ \mathbf{A}_0^{\{q\}} \\ \vdots \\ \mathbf{A}_n^{\{q\}} \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} \mathbf{a}_0^{\{p\}} \\ \vdots \\ \mathbf{a}_n^{\{p\}} \\ \mathbf{a}_0^{\{q\}} \\ \vdots \\ \mathbf{a}_n^{\{q\}} \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} \mathbf{D}_0^{\{p\}} \\ \vdots \\ \mathbf{D}_{2n-3}^{\{p\}} \\ \mathbf{D}_0^{\{q\}} \\ \vdots \\ \mathbf{D}_{2n-3}^{\{q\}} \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}_0^{\{p\}} \\ \vdots \\ \mathbf{d}_{2n-3}^{\{p\}} \\ \mathbf{d}_0^{\{q\}} \\ \vdots \\ \mathbf{d}_{2n-3}^{\{q\}} \end{bmatrix} \quad (21)$$

We recall that in our case $n = 6$. Finally, we can rewrite FP (13) with a contact switch as:

$$\begin{aligned} & \text{find } \mathbf{y} \\ & \text{s. t. } \mathbf{A}\mathbf{y} \leq \mathbf{a} \\ & \quad \mathbf{D}\mathbf{y} \leq \mathbf{d} \end{aligned} \quad (22)$$

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

E. General case

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

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

F. Non-conservative continuous formulation

The presented continuous formulation is more conservative than the discretized one. Constraining the control points to lie inside the constraint set prevents from the generation of curves such as the one illustrated in Figure 6.

However, by relying on the De Casteljau algorithm, it is possible to continuously satisfy the constraints while considering control points outside of the constraint set. Indeed when a curve is split, the constraints no longer apply to the control points of the original curve, but to the control points of the sub-curves. This is illustrated in Figure 5. If the curve is split an infinite number of times, it is straightforward to see that the original curve can span entirely its original definition set as the position of the control points converge to the original curve as the number of split increase.

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

We believe that a deeper analysis of the pros and cons of using a continuous formulation, not only in the case of CROC, but with any other formulation of the problem, requires a significant amount of research, and thus will be discussed in a future work. In this paper, we only divide the curve at the transition points, and we show in our experiments that this is already sufficient to perform similarly to the discretized approaches, while ensuring comparable time performances.

G. Cost function and additional constraints

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

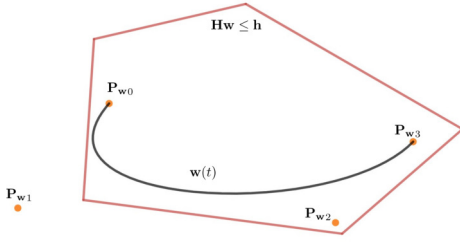


Fig. 6: The curve $w(t)$ belongs entirely to the convex boundaries (red), while a control point P_{w1} lies outside of them.

553 optimize this trajectory to minimize a given cost function $l(y)$,
 554 either linear or quadratic. In the latter case the FP problem (22)
 555 then becomes a Quadratic Program (QP). One can for instance
 556 minimize the integral of the squared acceleration norm or the
 557 angular momentum. This cost function is irrelevant to solve
 558 the transition feasibility problem, but it can be later used as a
 559 reference COM trajectory for a whole-body motion generator,
 560 or as an initial guess for a nonlinear solver as discussed in
 561 Section IV-F.

562 The formulation also allows to add inequality constraints
 563 on c and any of its derivatives by rewriting the expression of
 564 the control points of the desired curve as done in equation
 565 (17). Here again, these constraints can either be verified con-
 566 tinuously on the concerned control points, or in a discretized
 567 fashion. In any case, they take the form:

$$Oy \leq o \quad (23)$$

568 We use such constraints to impose bounds on the velocity
 569 and acceleration of the center of mass or on the angular
 570 momentum variation. The most generic form of our continuous
 571 problem is thus the following QP:

$$\begin{aligned} \text{find } & y \\ \text{min } & l(y) \\ \text{s. t. } & Ay \leq a \\ & Dy \leq d \\ & Oy \leq o \end{aligned} \quad (24)$$

572 In our experiments we set constraints on the acceleration
 573 and velocity and minimize the squared acceleration norm as
 574 a cost l . In the remainder of the paper “CROC” refers to
 575 this generic QP. If nothing is specified, by default CROC
 576 refers to the continuous formulation and with the inequalities
 577 representation of the dynamic constraints, as in the QP (24).

578 H. Time sampling

579 In the previous sections, in order to remain convex when
 580 computing $w(t)$ (equation (6)) we assumed that the duration
 581 of each phase $T^{\{p\}}$, $T^{\{p+1\}}$ and $T^{\{p+2\}}$ was given.

582 Time can be reintroduced in the problem using a bi-level
 583 optimization approach [36]. However, in this work we choose a
 584 more pragmatic offline-sampling approach to compute relevant
 585 timing candidates, which turns out to be lossless among all of
 586 our experiment set.

To achieve this, we consider a large variety of instances
 of the transition problem. We first consider all the scenarios
 demonstrated in Section VI (for HRP-2 and HyQ), from which
 we extract instances of the transition problem. We secondly
 generate random scenarios (Figure 9). We randomly allocate
 initial and end velocities for the center of mass along the
 direction of motion, between 0 and 1.5 m.s^{-1} .

For a total of 10 000 instances of the transition problem, we
 sample various combinations of times, solve the corresponding
 QPs and check whether a solution is found. In theory, this
 would mean that we need to sample an infinity of time com-
 binations 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
 combinations are enough to reach 99% of success but that two
 additional combinations are required to reach exactly 100%.

The number 100% may appear large. Intuitively however, it
 seems to highlight the fact that the accuracy of the transition
 times are not that important for the considered feasibility
 problem. Indeed $T^{\{p\}}$ constrains the COM trajectory to lie
 in the intersection of two contact phase constraints at this
 precise time. However this intersection is in general of a
 significant volume. As a result the COM trajectory will belong
 to the intersection for a large time window, which results in a
 significant slack in the selection of time.

We recall that here, we are only concerned in finding
 feasible times. For instance, typical double support times when
 walking on flat ground are closer to 0.2 seconds than 1 second
 for $T^{\{p\}}$ in dynamical cases. However 0.2 seconds is not
 feasible when starting from a null velocity. In both cases the
 interval between 0.8 and 1.2 seconds is almost always feasible
 in our experiments, which explains why such timings were
 selected for $T^{\{p\}}$. As such, table I should **not** be considered
 as a table giving optimal contact time durations, but rather one
 maximizing feasibility over our set of problems.

IV. PERFORMANCES OF CROC

A. CROC vs a nonlinear solver

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

$T\{p\}$	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.

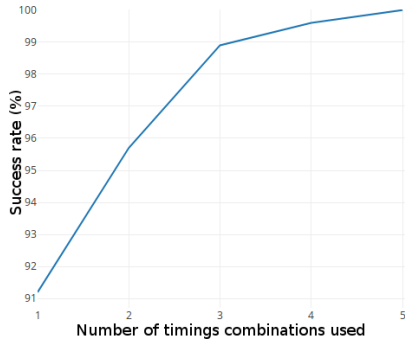


Fig. 7: Evolution of the success rate of CROC according to the number of timings combinations used. Tested on various scenarios with coplanar and non-coplanar contacts and with a bipedal and a quadrupedal robots.

643 method able to determine in finite time whether there exists a
 644 valid centroidal trajectory between the two states). We choose
 645 to compare the relative success rate of CROC with respect to
 646 a state-of-the-art non-linear formulation of the same problem
 647 [18], which is reported to give similar results to the one from
 648 Ponton et al. [22].

649 Both approaches share similar formulations in terms of
 650 kinematic constraints. Conversely the nonlinear solver does
 651 not use the conservative formulation of CROC that makes the
 652 problem convex, and thus is able to explore a larger part of
 653 the solution space, and thus to find a “more optimal” solution
 654 of a given locomotion problem.

655 From a practical point of view, the nonlinear solver is also
 656 able to tackle motion synthesis problems over large sequences
 657 of contacts. While CROC only interpolates trajectories over
 658 two waypoints given by the planner, the non-linear solver
 659 is able to ignore the waypoints to find a better solution
 660 (Figure 8). This locality is, in our experience, the main source
 661 of difference between the trajectories computed by CROC
 662 and the nonlinear solver. This difference is what ultimately
 663 motivates the use of the nonlinear solver to refine the trajec-
 664 tories obtained by CROC in Section V-C, at a stage where
 665 the contact sequence is fixed and the combinatorial is not a
 666 problem anymore.

667 B. Comparison benchmarks

668 The scenarios used in our benchmarks consist of randomly
 669 generated sequences of 3 contact phases such that:

- 670 • both initial and final contact phases are in static equilib-
 671 rium
- 672 • both initial and final contact phases have the same effec-
 673 tors in contact, between two and four

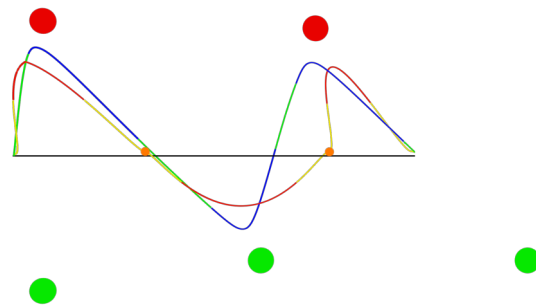


Fig. 8: Example of centroidal trajectories generated with CROC and a nonlinear solver (bird eye view), in a case of bipedal walking. The red and green circles represent the contact positions of the (respectively) left and right feet centers over time. The red and yellow (respectively related to single 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 nonlinear solver. The orange circles represent the constrained COM positions resulting from the contact planning phase, which are ignored by the nonlinear solver to produce smoother motions.

- 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.

679 These benchmarks thus only consider the case of a “repositioning”
 680 of an end-effector but as explained in section III-E
 681 this is our main use case for CROC as it encompasses the
 682 only two other possible cases (creating a contact or breaking
 683 a contact) and because this is the kind of contact sequences
 684 produced by our contact-planner.

685 For this benchmark we considered two kind of scenarios.
 686 In the first case, we only sample contact phases with coplanar
 687 contacts. In the second case, we sample truly random contacts,
 688 which lead to contact phases with non-coplanar contacts
 689 and contact sequences that require complex motions. Some
 690 examples of randomly generated scenarios are shown in Figure
 691 9.

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

695 The first benchmark compares four different methods: both
 696 discrete³ and continuous formulation of CROC presented in
 697 this paper (using the inequality representation of the con-
 698 straints), the nonlinear resolution proposed in [18] and the
 699 same nonlinear method but initialized with the solution found
 700 by CROC when available. As we compare the relative success
 701 rate between the methods, we only consider the scenarios
 702 where at least one of the method finds a solution when
 703 computing the percentage of success. The results are shown
 704 in table II.

³with 7 discretization points per contact phases, which corresponds to a time step of approximately 100ms.

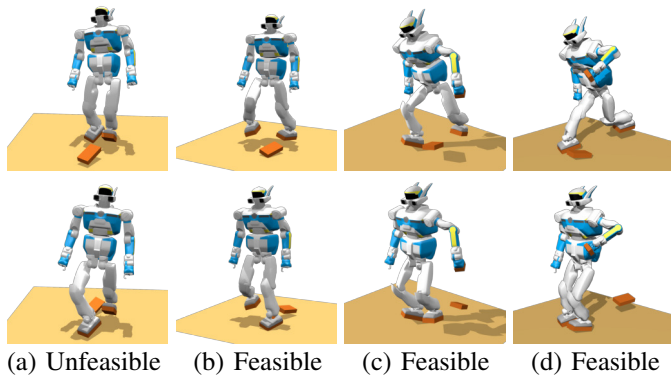


Fig. 9: Examples of random contact transitions used for benchmarking. Top row : initial configuration, bottom row : final configuration. (a) and (b) only have both feet in contact, (c) and (d) have both feet and the left hand in contact. All the displayed configurations are in static equilibrium, but the intermediate configuration with one less contact (not shown) is not constrained to be in static equilibrium. None of the methods found a solution for the transition (a), the other transitions were successfully solved.

Method	Coplanar success (%)	Non-coplanar success (%)	Total time (ms)
CROC (discrete)	89.7	60.6	3.89
CROC (continuous)	88.4	57.2	3.93
Non-linear	100	94.1	≈ 150
N-L with init guess	100	100	≈ 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 either the discrete³ or continuous formulation and using the inequality representation of the dynamic constraints. These methods are compared with the non linear solver presented in [18], either with their naive initial guess (Non-linear) or with the solution found by CROC as an initial guess when available (N-L with init guess). The percentages on the "success" columns only consider the scenario where at least one method found a solution.

C. How conservative is CROC?

Because of its conservative reformulation, CROC does not cover the whole solution space. As expected, our method finds less solutions than the nonlinear solver. 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 inflexion points and/or to be really close of the constraints, which cannot be represented using a single variable control point for the trajectory. This explains the difference of success rates between the two cases. However, even in such complex cases CROC still finds around 60% of the solutions.

D. Computation time

As claimed in the introduction, CROC is about two order of magnitude faster than the nonlinear 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 (24) 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 (21) is around $400\mu s$. The total time in table II also includes the time required by the double description method. However, in some 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.

1) Comparison with the equality representation: Table III shows the difference in computation time between the inequality and equality formulation, with a varying number of contacts.

Formulation	Metric	Number of contacts		
		2	3	4
Double-Description	DD time (ms)	3.52	14.88	28.16
	Total time (ms)	3.93	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 FP⁴ defined by CROC using either the Double Description (DD) or the Force formulation.

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.

E. Comparing the continuous and discretized formulations

The results of Table II confirm that the continuous formulation presented in section III-C2 is conservative with respect to the discrete formulation. However, these results show only a marginal difference of success rate between the discrete and continuous formulation of CROC (1 – 4%). This can first be explained by the fact that the De Casteljaou decomposition allows for the control point y to lie outside of the constraints (Figure 5), thus making the method less restrictive. We propose a second explanation, which is only intuitive (thus not a claim): the remaining missing solutions are necessarily those that will result in the curve lying close to the constraint boundaries. The discretized approach will theoretically find them, but

⁴QP 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.

the chances of finding a trajectory partially outside of the constraint sets are much higher in this case (Figure 3).

Moreover, in section III-D2 we proposed to only split the trajectory in one curve for each contact phases but it is possible to split the trajectory in an arbitrary number of curves, as long as each curve is entirely contained in one contact phases, as detailed in section III-F. By increasing the number of split curves, we can further reduce the loss of solutions.

1) *Invalid solutions of the discretized methods:* Again, the major drawback of a discretized approach is that the portions of the curve in-between two discretization points are never checked and could violate the constraints (Figure 3).

In order to measure this risk four variants of CROC were compared with the same randomly generated contact sequence as before: the discretized version with three different values of number of discretization points per phases and the continuous version presented in this paper. The four variants use the inequality representation of the dynamic constraints. Then, for each centroidal trajectory found as a solution, the dynamic constraints were verified with a really small discretization step. If the constraints were not satisfied for at least one point of the trajectory, we count this solution as "invalid".

Method	Invalid solutions (%)		Computation time (ms)
	Coplanar	Non-coplanar	
Discrete (3 pts)	10.6	19.7	0.20
Discrete (7 pts)	6.7	9.3	0.37
Discrete (15 pts)	4.2	6.9	0.75
Continuous	0	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.

Table IV shows that the percentage of invalid solutions found by the discrete methods is non negligible. Obviously, as the number of discretization points increase this percentage decreases. As shown in equation (11) the number of constraints in the discretized LP problem is proportional to the number of discretization points. Thus the number of discretization points used is a complex parameter to tune, as it is a compromise between the computation time and the risk of finding invalid solutions. This issue is common to all methods that rely on discretization. It emphasizes the fact that we need a continuous method, able to check exactly whether the whole trajectory is valid with a fixed number of constraints in the problem.

2) *Computational advantage of the continuous formulation:* Depending on the discretization, the continuous formulation can be slower or faster to compute. However, to reach less than 5 % of false positive trajectories with the discretized approach, table IV shows that the continuous formulation is actually faster.

F. Using CROC to initialize a non linear solver

Choosing an initial guess for the nonlinear solver of a trajectory generation method is essential but may be challenging for multi-contact motions. The quality of this initial guess has a significant influence on the convergence of the nonlinear

solver. For the nonlinear method considered in this section [18] proposed a naive initial guess of the centroidal trajectory based solely on the position of the contact points.

Interestingly, Table II suggests that the solution set spanned by CROC is not strictly included in the one spanned by this nonlinear solver with this naive initial guess. Using the solution of CROC to initialize the nonlinear solver can thus help it to converge and increase its success rate. As shown in Table II, this improvement only appears for the non-coplanar case because the naive initial guess used is always close to a valid solution in the coplanar case. We expect that the importance of the initial guess will grow if the contact sequences do not allow static equilibrium configurations at the contact phases, and will check this hypothesis in the future.

Moreover, by using the solution of CROC to initialize the nonlinear solver we measured a reduction of the number of iterations required to converge of 20% on average, reducing the total computation time (ie. it is faster to use CROC and then the non-linear solver than using the non-linear solver directly).

G. Validity of our kinematic constraints

As explained in the section II-C, our representation of the kinematics constraints is a necessary but not sufficient approximation. In order to evaluate the accuracy of this approximation, for each feasible transition found by CROC between random configurations, we tested explicitly the kinematic feasibility of the centroidal trajectory with an inverse kinematic. This tests showed that 17.5 % of the trajectories found by CROC were not kinematically valid. This shows that our approximation of the kinematic constraints is not sufficient. However, this is not a limitation of CROC, but rather of the formulation of the kinematic constraints, which we hope to improve in the future.

Moreover, by doing the same tests without any kinematic constraints we found a total of 72.3 % of kinematically unfeasible trajectories. This results show the interest of our kinematic constraints approximation to improve the feasibility of the trajectories found by CROC.

V. EXPERIMENTAL FRAMEWORK

Figure 10 shows the complete framework used for our experiments, implemented with the Humanoid Path Planner [38] 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, and a complete representation of the 3D environment. The output is a dynamically consistent and collision free whole-body motion which can be played on a real robot as shown in section VI.

In this paper, we modify the contact generation method by adding CROC as a feasibility criterion, and connect all the pieces of the framework together. These other pieces are used as black boxes and thus only briefly introduced, with a reference to their respective publications.

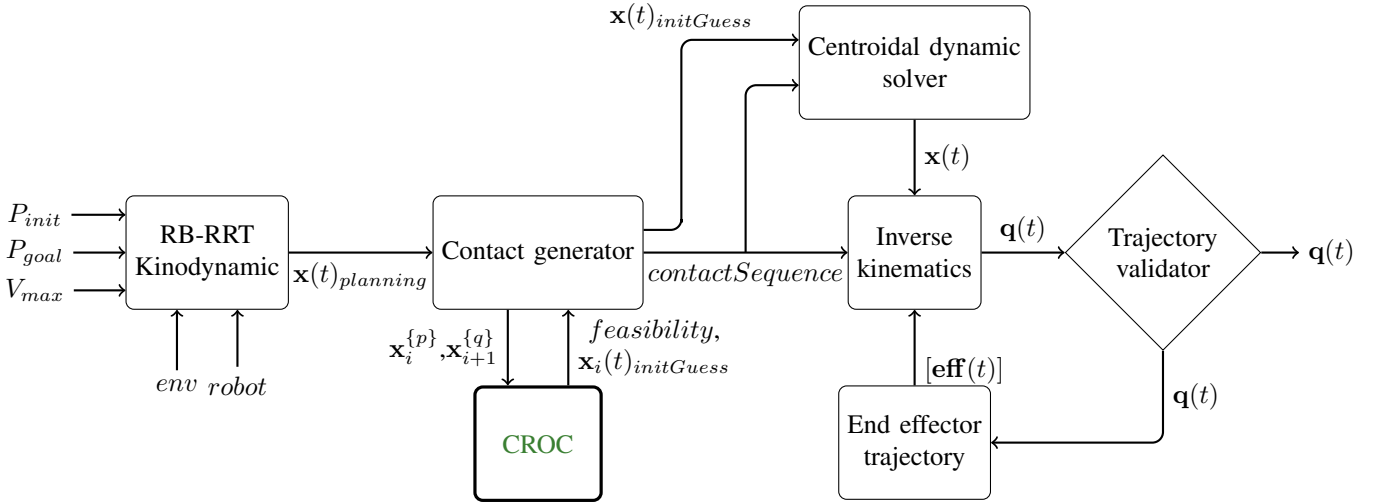


Fig. 10: Complete experimental framework.

A. RB-RRT kinodynamic planner

The first block generates a rough guide trajectory⁵ for the root of the robot $\mathbf{x}(t)_{planning}$. It thus solves the sub-problem \mathcal{P}_1 defined in Figure 2. 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 of the robot. It thus solves a problem of lower dimension than planning in the configuration space of the real robot. The goal of this method is to find a trajectory for the root of the robot which will allow contact creation. This block was first presented in [15] and later extended to a kinodynamic version in [16], which is the one we use.

B. Contact generator with CROC as a feasibility criterion

The *contact generator* block computes a contact sequence, as a list of whole body postures along the discretized guide trajectory $\mathbf{x}(t)_{planning}$. This block solves the sub-problem \mathcal{P}_2 defined in Figure 2. It also generates an initial guess of the timing of each contact phase. This method was also introduced in [15].

CROC is used as a feasibility criterion by this contact generator. More precisely it is used as a filter to determine which transitions are unfeasible and discard them during the planning in order to produce contact sequence containing only feasible transitions. CROC will thus be called for each contact transitions considered by the contact generator ($\mathbf{x}_i^{\{p\}}$ and $\mathbf{x}_{i+1}^{\{q\}}$ in Figure 10) and output the feasibility of the given contact transition. The integration of CROC to this pipeline provides strong guarantees that the computed contact sequence will lead to a feasible CoM trajectory and thus that the centroidal dynamics solver will converge with this contact sequence as input.

A byproduct of the feasibility test made with CROC is a feasible CoM trajectory between each adjacent contact phases ($\mathbf{x}(t)_{initGuess}$). This trajectory, not optimal, is used as an

initial guess for a non-linear solver which will use it to compute an optimal trajectory.

C. Centroidal dynamics solver

The *centroidal dynamics solver* block was proposed in [18], it takes as input the contact sequence found by the previous block, along with an initial guess of the timing of each phases and an initial guess of the CoM trajectory. The output of this block is a CoM trajectory that respects the centroidal dynamics of the robot $\mathbf{x}(t)$ and minimizes a tailored cost function. This method solves an optimal control problem with a multiple-shooting algorithm implemented in MUSCOD-II [39].

The main interest of using a non-linear solver with the input of CROC is that the trajectory can then be refined globally (while the authors advise to use CROC with at most 3 contact phases), at the cost of a higher computational burden. CROC and the non linear solver are thus complementary: CROC does not provide an optimum, but is computationally efficient, which allows it to be used with a trial-and-error approach (ie. trying to solve problems that we dont know if a solution exist, until we find a solvable problem). Conversely, the non-linear solver is too computationally expensive to be used with a trial-and-error approach, but will in general propose a trajectory with a better optimum with respect to the optimized cost function. The proposed framework is designed to call this non-linear solver only once, with a problem that is known to have a solution.

The three different trajectories found in the framework of Figure 10 are shown in Figure 8, $\mathbf{x}(t)_{planning}$ is represented in black, $\mathbf{x}(t)_{initGuess}$ in yellow and orange and $\mathbf{x}(t)$ in green and blue. This figure shows a trajectory computed with CROC and the same trajectory refined with a non-linear solver as an illustration of the typical differences of both approaches.

D. Inverse kinematics

The whole-body motion $\mathbf{q}(t)$ is generated with a second order Inverse Kinematics solver, similar to [40]. This method

⁵This guide is followed exactly to solve \mathcal{P}_2 , but ignored at \mathcal{P}_3 .

930 takes as input a reference trajectory for the CoM, as well as
931 references for the trajectories of the end-effectors.

932 E. End-effector trajectory

933 In order to automatically generate valid end-effector tra-
934 jectories for complex and constrained scenarios, we use a
935 dedicated block. The trajectories computed are such that
936 the whole limb is collision free and respect the kinematic
937 constraints. The trajectories are represented as Bezier curves
938 constrained to have a null initial and final velocity, acceleration
939 and jerk and which respect velocity, acceleration and jerk
940 bounds along the whole trajectory. In order to guarantee that
941 the whole surface of the effector creates or breaks the contact
942 at the same instant the curves are also constrained to have a
943 velocity orthogonal to the contact surface for a small time step
944 at the beginning and the end of the trajectory.

945 The positions of the control points of this Bezier curve
946 are computed as the solution of a QP optimization method,
947 using a cost function that defines a compromise between a
948 reference optimal trajectory and a collision free one, provided
949 by a probabilistic planner. This planner computes a geometric
950 path for the moving limb that respects all the kinematic and
951 collision constraints. However it may present discontinuities
952 in velocity and higher derivatives and does not respect the
953 dynamic constraints described in the previous paragraph.
954 Moreover, as with any sampling-based method this path will
955 not be optimal. Because of this, this geometric path will not
956 be used directly as the end-effector trajectory but will be used
957 inside the cost function of our optimization method.

958 Then several iterations are made between this optimization
959 method and the inverse kinematics method, producing trajecto-
960 ries for the end-effectors and checking if the resulting whole-
961 body motion is valid. If not, the weight of the cost associated
962 to the solution of the geometric planner is increased at each
963 iteration, until a valid motion is found.

964 VI. EXPERIMENTAL RESULTS

965 A. Experimental scenario

966 The complete experimental framework presented in the
967 previous section was tested on several locomotion scenarios in
968 semi structured environments, each scenario showing specific
969 features or difficulties. We insist that the only manual inputs
970 given to our framework were an initial and a goal position
971 for the root of the robot. Most of the obtained motions are
972 demonstrated in the companion video. They were validated
973 either in a dynamics simulator or on the real robot.

974 1) *Inclined platform crossing*: This scenario requires the
975 robot to go from one flat platform to the other by taking a step
976 on an inclined platform (Figure 12). The scenario is designed
977 such that no quasi-static solution exists to the problem, and is
978 truly multi-contact for two reasons: firstly part of the motion
979 occurs entirely on non-flat ground; secondly the problem is
980 unfeasible if the right foot is the one selected to go first on the
981 platform. CROC then allows to invalidate unfeasible contact
982 sequences that would involve directly taking a step on the final
983 platform, or take a step with the right foot first (Figure 13). It
984 rather allows to find a solution where the left foot is used to

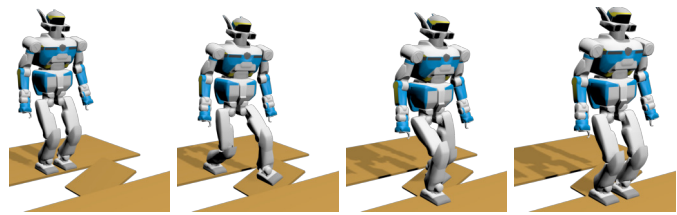


Fig. 12: Platform crossing scenario: no quasi-static solution exists for the flying phase where the left foot is on the inclined platform.

985 step on the inclined platform (Figure 12). A feasible whole-
986 body motion is demonstrated in the companion video.

987 Additionally, CROC also ensures that the left foot is posi-
988 tioned in such a way that the problem becomes feasible, which
989 is not trivial considering the size of the solution space for the
990 chosen step position (Figure 19(a)).

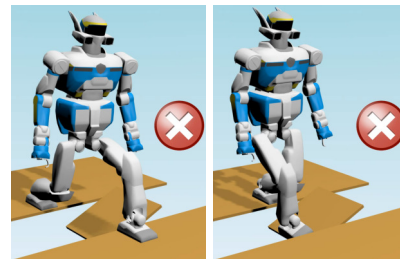


Fig. 13: Unfeasible stepping strategies invalidated by CROC.

991 2) *10 cm high steps*: This experimental setup is an indus-
992 trial set of stairs shown in Figure 11 and 18(a). It consists of
993 six 10 cm high and 30 cm long steps. This experiment was
994 done with the HRP-2 robot. All the valid contact sequences
995 produced contain at least 13 contact phases as the robot is
996 kinematically constrained to put both feet on each step.

997 The complete motion is shown in the companion video. The
998 crouching walk seen is required to avoid singularities in the
999 knee of the extending leg, which are not tolerated by the low-
1000 level controller.

1001 An example of unfeasible contact sequence filtered out by
1002 our feasibility criterion is depicted on Figure 14. All three
1003 configurations in this sequence are valid (ie. respect kinematics
1004 and dynamics constraints) but there isn't any valid centroidal
1005 trajectory between the last two configurations. Our feasibility
1006 criterion will filter out this kind of contact transitions during
1007 contact planning.

1008 3) *15 cm high steps with handrail*: This other set of stairs
1009 is composed of four 15 cm high steps and equipped with
1010 a handrail. The contact sequence is shown in Figure 18(b)
1011 and snapshots of the motion are shown in Figure 15. This is
1012 a typical multi-contact problem, showing an acyclic contact
1013 sequence with non co-planar contact surfaces. The problem
1014 was already solved in a previous work [17], but the input
1015 contact sequence and effector trajectories had to be manually
1016 selected from a large number of trials. In this paper, the only
1017 input is a root goal position at the top of the stairs.

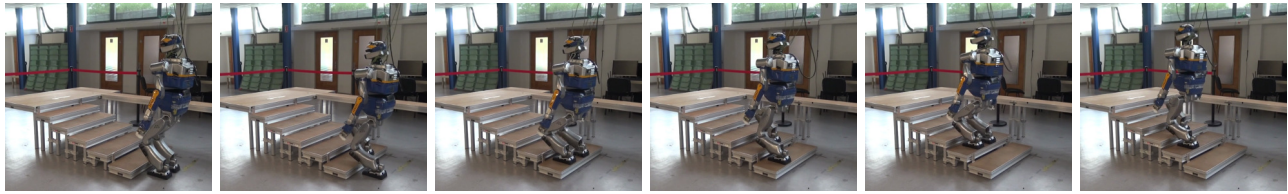


Fig. 11: Snapshots of the motion for the 10cm stairs, the complete motion is shown in the companion video.

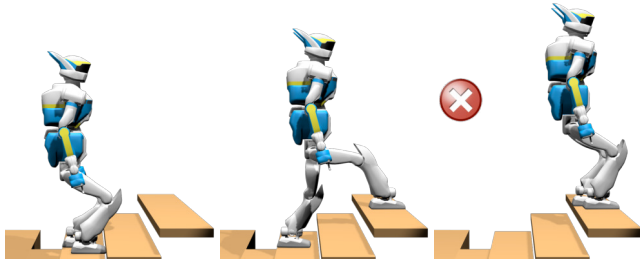


Fig. 14: Example of unfeasible contact transition detected by CROC and rejected during contact planning

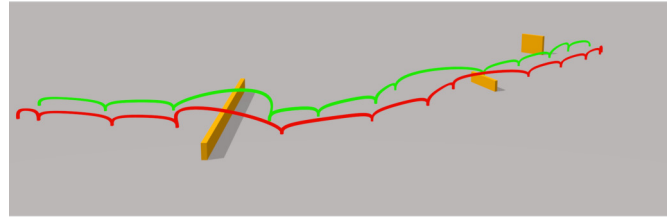


Fig. 16: Feet trajectories computed for scenario with ground level obstacles. Green for right foot and red for left foot.

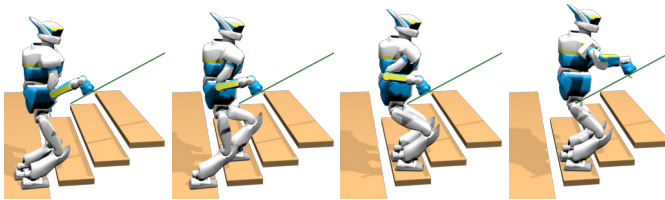


Fig. 15: A feasible multi-contact sequence for a stair climbing with handrail support on the HRP-2 robot automatically computed with our contact planner and CROC.

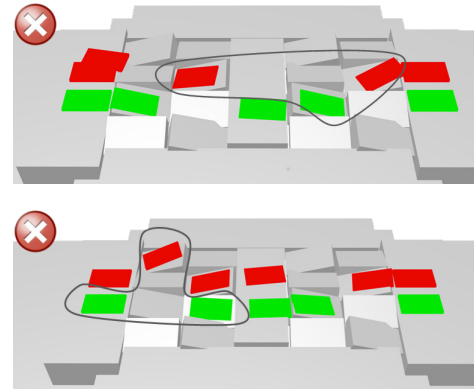


Fig. 17: Examples of unfeasible contact sequences filtered out by CROC. There doesn't exist any valid centroidal trajectory for the contact transitions encircled in black.

1018 A example of centroidal trajectory found by CROC for one
 1019 contact transition in this scenario is shown in Figure 19(b).

1020 4) *Flat surface with ground level obstacles*: This exper-
 1021 imental setup consists of a flat floor with obstacles, shown
 1022 in Figure 18(c) and (d). In (c) there is only one obstacle
 1023 in front of the robot's initial position, in (d) we add smaller
 1024 obstacles on the floor. This scenario shows that our planner
 1025 is able to compute a valid guide root trajectory that avoids
 1026 bigger obstacles and that our contact planner is able to avoid
 1027 collision with smaller obstacles on the ground.

1028 The difficulty of this scenario lies on the generation of
 1029 collision free feet trajectories. Indeed, some obstacles are small
 1030 enough to permit the feet to pass over the obstacles, but others
 1031 are too high and require a lateral motion of the feet to avoid
 1032 them. As shown in Figure 16 our method presented briefly in
 1033 section V-E is able to find such trajectories automatically.

1034 5) *Uneven platforms*: This setup consists of 30 cm long
 1035 and 20 cm wide platforms, oriented of 15° around either the x
 1036 or y axis. This scenario is particularly difficult for the contact
 1037 planner because of all the possible collisions generated by the
 1038 feet. We recall that the feet of HRP-2 are 24 cm long for 14
 1039 cm wide, which means that the platforms of this setup are only
 1040 a few centimeters bigger than the feet of the robot. Because of
 1041 this, there is really few collision free candidates positions for

1042 the feet. The probability of finding a contact position which
 1043 leads to a collision-free configuration while maintaining the
 1044 equilibrium is extremely small for this setup.

1045 The contact sequence found is shown in Figure 18(e),
 1046 snapshots of the motion are shown in Figure 1 and a motion for
 1047 this scenario is shown in the companion video. These motions
 1048 have been validated on the real robot.

1049 The Figure 17 shows two examples of unfeasible contact
 1050 sequence filtered out by CROC in this scenario.

1051 6) *Quadrupedal between inclined planes*: The quadrupedal
 1052 robot HyQ navigates between two planes inclined at 45° .
 1053 Figure 19(c) shows the the centroidal trajectory found by
 1054 CROC in this scenario for one contact transition. This scenario
 1055 shows that our method may be adapted to any type of legged
 1056 robot.

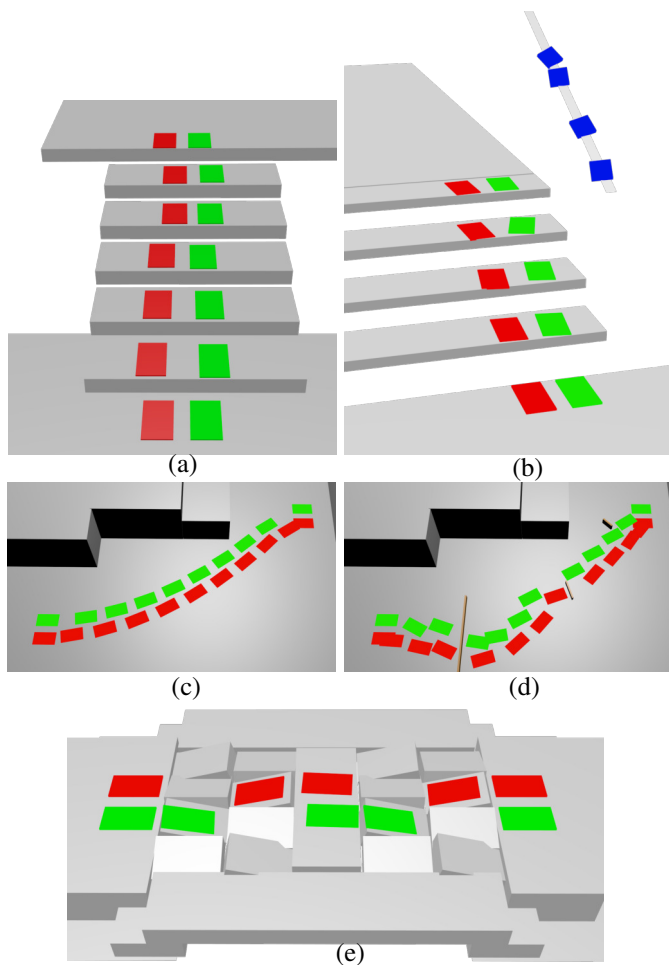


Fig. 18: Examples of contact sequences found with our framework. The color patches represent the planned contact location: green for right foot, red for left foot, blue for right hand.

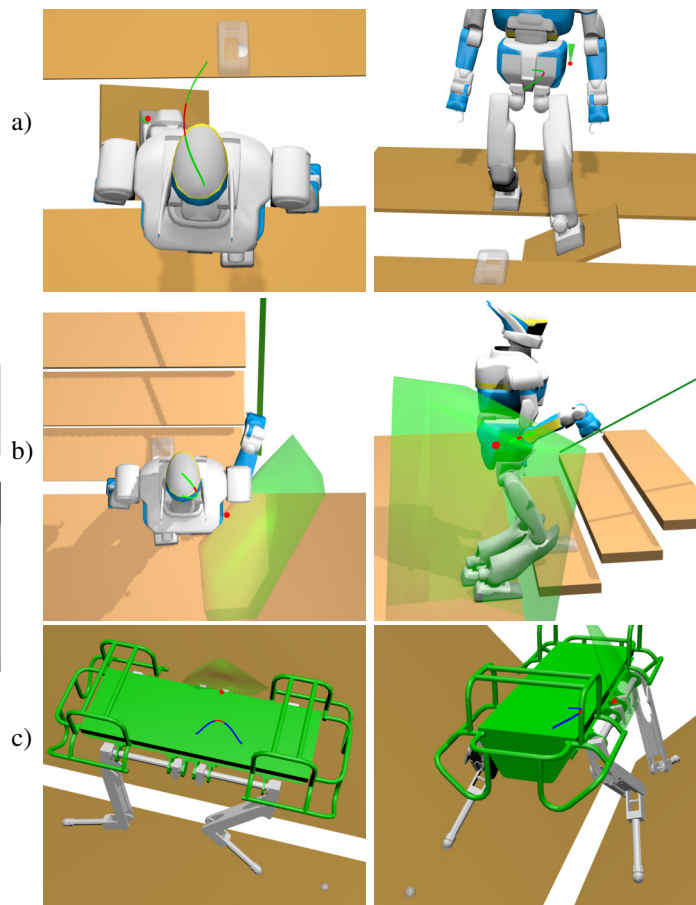


Fig. 19: Examples of centroidal trajectories found by our method. Green polytopes : valid position of y that verifies the constraints of the problem (24), red sphere : solution found for y for a given cost function (minimum of the squared acceleration norm). The red part of the trajectory is for the phase with $n_c - 1$ active contacts. The next contact is shown in transparency.

1057 **B. Benchmarks**

1058 1) *Using CROC as a feasibility criterion:* In order to
 1059 quantify the improvement of our contact planner from the use
 1060 of CROC as a feasibility criterion, we used the following test
 1061 procedure: for some of the scenarios presented in the previous
 1062 section, we tried to solve the problem using our framework
 1063 with and without using CROC as a feasibility criterion during
 1064 the contact planning. We then measured the success rate of
 1065 the centroidal dynamic solver with the contact plan found.
 1066 The results are shown in Table V.

1067 In the walking on flat floor scenario, CROC brings only
 1068 a marginal improvement to our contact planner because our
 1069 previously used heuristics were sufficient in this case to
 1070 provide a feasible contact plan most of the time. However, in
 1071 all the other cases the results empirically prove the main claim
 1072 of this paper: using CROC as a feasibility criterion during the
 1073 contact generation greatly increases the success rate of the
 1074 centroidal trajectory generation because it produces contact
 1075 plans with only feasible transitions. Another expected result
 1076 is that there isn't any "false positive" found by our method:
 1077 when CROC converges, the non linear solver always converges

for the same transition.

1078 The trade-off is an increase of the computation time required
 1079 by the contact generator, from a few percents to nearly the
 1080 double. This is explained partly by the addition of the time
 1081 required to run CROC for each candidates, but mostly by the
 1082 fact that we need to evaluate a lot more candidates before
 1083 we find a valid one (ie. which lead to a feasible transition).
 1084 This is shown in the column 4 of Table V, which provides
 1085 the average number of contact candidates evaluated during the
 1086 contact planning phase. Actually, depending on the scenario
 1087 considered, only 7 to 16% of the total "contact planning"
 1088 computation time is spent solving CROC problems. The rest of
 1089 the time is mostly spent by projection methods and collision
 1090 tests. This shows that our formulation is fast enough to be
 1091 used inside a contact planner, without too much impact on its
 1092 computation time.
 1093

1094 2) *Benchmarks of the complete framework:* Table VI shows
 1095 a benchmark of the performances of the complete motion
 1096 planning framework presented in section V. We recall that
 1097 this framework take as input only an initial and goal position

Scenario	Method	Contact planning		Centroidal dynamic solver success (%)
		time (s)	Evaluated candidates (avg.)	
Walk (flat)	Without CROC	0.58	8.2	98
	With CROC	0.63	21.9	100
Stairs (3 steps)	Without CROC	0.61	24.4	52
	With CROC	0.87	92.9	100
Stairs (handrail)	Without CROC	1.26	147.2	31
	With CROC	1.87	384.0	100
Uneven platforms	Without CROC	3.91	679.1	15
	With CROC	7.59	3030	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 computation time required by the contact planning phase and the average number of contact candidates evaluated during this phase. The last column shows the success rate of the centroidal trajectory generation method with the contact sequence found by the planner. Each scenario have been run 100 times.

1098 for the center of the robot and produce as output a whole body
1099 motion.

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
Stairs	16.23	12.56	90.5
Stairs with handrail	23.13	18.09	88.05
Uneven platforms	14.94	17.83	83.5

TABLE VI: Performance analysis of the complete motion planning framework presented in section V, 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.

1100 We observe that the success rate is close to 100% except
1101 for complex scenarios where it is still above 80% in the worst
1102 case. The main cause of failure in our current implemen-
1103 tation of the framework is the inverse kinematics that may
1104 produce whole-body motions that do not respect the kine-
1105 matic constraints or that are in self-collision. Concerning the
1106 computation time, in most of the cases we achieve interactive
1107 performances (ie. the computation time is smaller than the
1108 motion duration). In the worst case the computation time is
1109 greater than the motion duration, but only by a small margin.

1110 As shown in Figure 20, the inverse kinematics method is
1111 currently the bottleneck of our framework and takes more than
1112 60% of the total computation time.

1113 VII. CONCLUSION

1114 In this paper we introduce a continuous, accurate and effi-
1115 cient formulation of the centroidal dynamics of a legged robot,
1116 named *CROC*. Our method guarantees that it can compute
1117 valid centroidal trajectories that do not require discretization,
1118 nor use approximation or relaxation of the dynamic con-
1119 straints. This formulation is convex yet conservative, but not
1120 limited to quasi-static motions. To our knowledge, this is the
1121 first method to combine all these properties.

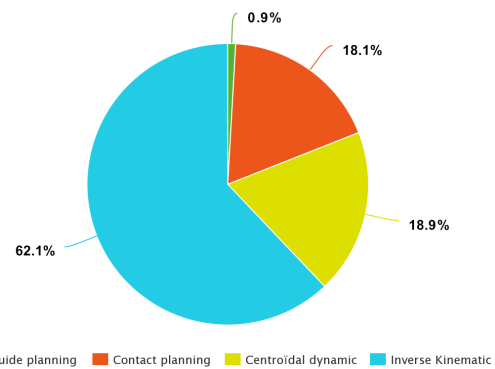


Fig. 20: Division of the computation time among the different methods of the motion planning framework.

1122 Thanks to the computational efficiency of our method,
1123 requiring only a few milliseconds to solve the centroidal
1124 dynamic problem with three contact phases, we can use this
1125 method as a feasibility criterion during contact planning. The
1126 interest of this feasibility criterion has been demonstrated both
1127 qualitatively and empirically. Our results show that all the
1128 contact plans produced with CROC as a feasibility criterion
1129 lead to feasible centroidal dynamics problems. We also show
1130 that without using this feasibility criterion, the contact planner
1131 finds unfeasible contact sequences with a high probability on
1132 complex scenarios.

1133 Moreover, the centroidal trajectory produced by CROC can
1134 be used to provide a relevant initial guess to a non linear
1135 solver, resulting in the improvement on the convergence rate
1136 and computation time of the non linear solver by comparison
1137 to the naive initial guess previously used.

1138 Thanks to the continuous formulation proposed in this pa-
1139 per, we have the guarantee that the whole centroidal trajectory
1140 is valid, by opposition to the discretized methods of the state
1141 of the art that only guarantee that the discretized points of the
1142 trajectory are valid. We showed that the discretization may
1143 lead to a non negligible amount of invalid solutions where the
1144 trajectory is invalid between two valid discretization points,
1145 which emphasizes the interest of a continuous formulation.
1146 We believe that this continuous formulation of the constraints
1147 on the centroidal trajectory may be useful for all state-of-the-
1148 art methods, convex or non-linear. We leave the study of the
1149 feasibility and the interest of this application to a future work.

1150 Finally, the feasibility criterion proposed in this paper
1151 permits us to complete our locomotion planning framework
1152 [41]. In this paper we showed that our framework is able
1153 to produce indifferently simple walking motions and multi-
1154 contact motions (ie. with non coplanar contacts and acyclic
1155 behaviors). These motions were validated in simulation or
1156 on the robot HRP-2. We also showed empirically that our
1157 framework presents a success rate close to 100% and present
1158 interactive computation times (the time required to compute
1159 a motion is smaller than the duration of this motion) in the
1160 studied scenarios, except for the most complex scenario where
1161 the computation time is approximately 20% greater than the
1162 duration of the motion, but still remain in the same order
1163 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 [42], [43]. 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 (13) determines the 0-Step capturability. Similarly, FP (22) 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://github.com/humanoid-path-planner/hpp-bezier-com-traj>

ACKNOWLEDGMENT

Supports: ANR LOCO3D ANR-16-CE33-0003, ERC Actantrophe 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] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. on Graph.*, vol. 31, no. 4, pp. 43:1–43:8, 2012.
- [4] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Humanoid Robots (Humanoids)*, *14th IEEE-RAS Int. Conf. on*, Madrid, Spain, 2014.
- [5] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The Int. Journal of Robot. Research (IJRR)*, vol. 33, no. 1, pp. 69–81, Jan. 2014.
- [6] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Systems through Phase-based End-Effector Parameterization," *IEEE Robotics and Automation Letters*, pp. 1–1, 2018. [Online]. Available: <http://ieeexplore.ieee.org/document/8283570/>
- [7] 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>
- [8] K. Hauser, T. Bretl, and J.-C. Latombe, "Non-gaited humanoid locomotion planning," in *Humanoid Robots, 2005 5th IEEE-RAS Int. Conf. on*, 2005, pp. 7–12.
- [9] 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>
- [10] M. X. Grey, A. D. Ames, and C. K. Liu, "Footstep and motion planning in semi-unstructured environments using randomized possibility graphs," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4747–4753.
- [11] P. Kaiser, C. Mandery, A. Boltres, and T. Asfour, "Affordance-based multi-contact whole-body pose sequence planning for humanoid robots in unknown environments," in *IEEE International Conference on Robotics and Automation*, 2018.
- [12] T. Koolen, T. de Boer, J. R. Rebuta, A. Goswami, and J. E. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *I. J. Robotics Res.*, vol. 31, no. 9, pp. 1094–1113, 2012. [Online]. Available: <https://doi.org/10.1177/0278364912452673>
- [13] J. E. Pratt, T. Koolen, T. de Boer, J. R. Rebuta, S. Cotton, J. Carff, M. Johnson, and P. D. Neuhuis, "Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid," *I. J. Robotics Res.*, vol. 31, no. 10, pp. 1117–1133, 2012. [Online]. Available: <https://doi.org/10.1177/0278364912452762>
- [14] A. Del Prete, S. Tonneau, and N. Mansard, "Zero Step Capturability for Legged Robots in Multi Contact," *Accepted on IEEE Trans on Robotics*, 2018. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01574687>
- [15] S. Tonneau, A. D. Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, June 2018.
- [16] P. Fernbach, S. Tonneau, A. D. Prete, and M. Taïx, "A kinodynamic steering-method for legged multi-contact locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 3701–3707.
- [17] J. Carpentier, R. Budhiraja, and N. Mansard, "Learning Feasibility Constraints for Multi-contact Locomotion of Legged Robots," in *Robotics: Science and Systems*, Cambridge, MA, United States, Jul. 2017. [Online]. Available: <https://hal.laas.fr/hal-01526200>
- [18] J. Carpentier and N. Mansard, "Multi-contact locomotion of legged robots," *Rapport LAAS n 17172*. <https://hal.laas.fr/hal-01520248>. Conditionally accepted for *IEEE Trans. on Robotics*, 2017.
- [19] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum-control," in *Humanoid Robots (Humanoids)*, *15h IEEE-RAS Int. Conf. on*, Nov. 2015.
- [20] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *Humanoid Robots (Humanoids)*, *14th IEEE-RAS Int. Conf. on*, Madrid, Spain, 2014, pp. 295–302.
- [21] S. Caron, A. Escande, L. Lanari, and B. Mallein, "Capturability-based analysis, optimization and control of 3d bipedal walking," jan 2018, submitted. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01689331>
- [22] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots*, 2016.
- [23] G. Mesesan, J. Engelsberger, C. Ott, and A. Albu-Schffer, "Convex properties of center-of-mass trajectories for locomotion based on divergent component of motion," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3449–3456, Oct 2018.
- [24] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.
- [25] P. Fernbach, S. Tonneau, and M. Taïx, "Croc: Convex resolution of centroidal dynamics trajectories to provide a feasibility criterion for the multi contact planning problem," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

- 1295 [26] K. Fukuda and A. Prodon, *Double description method revisited*. Berlin,
1296 Heidelberg: Springer Berlin Heidelberg, 1996, pp. 91–111.
- 1297 [27] D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a
1298 humanoid robot,” *Autonomous Robots*, vol. 35, no. 2, pp. 161–176, Oct
1299 2013.
- 1300 [28] Z. Qiu, A. Escande, A. Micaelli, and T. Robert, “Human motions
1301 analysis and simulation based on a general criterion of stability,” in
1302 *Int. Symposium on Digital Human Modeling*, 2011.
- 1303 [29] S. Caron, Q.-C. Pham, and Y. Nakamura, “Leveraging Cone Double
1304 Description for Multi-contact Stability of Humanoids with Applications
1305 to Statics and Dynamics,” in *Robotics, Science and Systems (RSS)*, 2015.
- 1306 [30] A. Del Prete, S. Tonneau, and N. Mansard, “Fast Algorithms to Test Robust
1307 Static Equilibrium for Legged Robots,” in *2016 IEEE International
1308 Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden,
1309 2016.
- 1310 [31] S. Tonneau, A. D. Prete, J. Pettré, and N. Mansard, “2PAC: Two Point
1311 Attractors for Center of Mass Trajectories in Multi Contact Scenarios,”
1312 Sep. 2017, accepted with major revisions for *Trans. on Graphics*.
1313 [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01609055>
- 1314 [32] J. M. ([https://math.stackexchange.com/users/305862/jean marie](https://math.stackexchange.com/users/305862/jean%20marie)), “Is the
1315 cross product of two bezier curves a bezier curve?” *Mathematics Stack
1316 Exchange*, uRL:<https://math.stackexchange.com/q/2228976> (version:
1317 2017-12-10). [Online]. Available: <https://math.stackexchange.com/q/2228976>
- 1318 [33] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, “An
1319 efficient optimal planning and control framework for quadrupedal loco-
1320 motion,” in *Robotics and Automation (ICRA), 2017 IEEE International
1321 Conference on*. IEEE, 2017, pp. 93–100.
- 1322 [34] R. Budhiraja, J. Carpentier, and N. Mansard, “Dynamics consensus
1323 between centroidal and whole-body models for locomotion of legged
1324 robots,” in *ICRA 2019-IEEE International Conference on Robotics and
1325 Automation*, 2019.
- 1326 [35] F. Gao, W. Wu, Y. Lin, and S. Shen, “Online safe trajectory gener-
1327 ation for quadrotors using fast marching method and bernstein basis
1328 polynomial,” in *2018 IEEE International Conference on Robotics and
1329 Automation (ICRA)*, May 2018, pp. 344–351.
- 1330 [36] W. Sun, G. Tang, and K. Hauser, “Fast UAV trajectory optimization
1331 using bilevel optimization with analytical gradients,” *CoRR*, vol.
1332 abs/1811.10753, 2018. [Online]. Available: [http://arxiv.org/abs/1811.
1333 10753](http://arxiv.org/abs/1811.10753)
- 1334 [37] A. Makhorin, “Glpk (gnu linear programming kit),” [http://www.gnu.
1335 org/software/glpk/glpk.html](http://www.gnu.org/software/glpk/glpk.html), 2008.
- 1336 [38] J. Mirabel, S. Tonneau, P. Fernbach, A. K. Seppälä, M. Campana,
1337 N. Mansard, and F. Lamiroux, “Hpp: A new software for constrained
1338 motion planning,” in *2016 IEEE/RSJ International Conference on Intel-
1339 ligent Robots and Systems (IROS)*, Oct 2016, pp. 383–389.
- 1340 [39] D. B. Leineweber, I. Bauer, H. G. Bock, and J. P. Schlöder,
1341 “An efficient multiple shooting based reduced sqp strategy for
1342 large-scale dynamic process optimization. part 1: theoretical aspects,”
1343 *Computers and Chemical Engineering*, vol. 27, no. 2, pp. 157 – 166,
1344 2003. [Online]. Available: [http://www.sciencedirect.com/science/article/
1345 pii/S0098135402001588](http://www.sciencedirect.com/science/article/pii/S0098135402001588)
- 1346 [40] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soares, and J. Y.
1347 Fourquet, “Dynamic whole-body motion generation under rigid contacts
1348 and other unilateral constraints,” *IEEE Transactions on Robotics*, vol. 29,
1349 no. 2, pp. 346–362, April 2013.
- 1350 [41] J. Carpentier, A. Del Prete, S. Tonneau, T. Flayols, F. Forget,
1351 A. Mifsud, K. Giraud, D. Atchuthan, P. Fernbach, R. Budhiraja,
1352 M. Geisert, J. Solà, O. Stasse, and N. Mansard, “Multi-contact
1353 Locomotion of Legged Robots in Complex Environments – The
1354 Loco3D project,” in *RSS Workshop on Challenges in Dynamic Legged
1355 Locomotion*, Boston, United States, Jul. 2017, p. 3p. [Online]. Available:
1356 <https://hal.laas.fr/hal-01543060>
- 1357 [42] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. G. Caldwell, and
1358 C. Semini, “Application of wrench based feasibility analysis to the online
1359 trajectory optimization of legged robots,” *IEEE Robotics and Automation
1360 Letters*, pp. 1–1, 2018.
- 1361 [43] V. Samy, S. Caron, K. Bouyarmane, and A. Kheddar, “Post-impact
1362 adaptive compliance for humanoid falls using predictive control of a
1363 reduced model,” in *2017 IEEE-RAS 17th International Conference on
1364 Humanoid Robotics (Humanoids)*, Nov 2017, pp. 655–660.
- 1365