A Versatile and Efficient Pattern Generator for Generalized Legged Locomotion

Justin Carpentier, Steve Tonneau, Maximilien Naveau, Olivier Stasse, Nicolas Mansard

A Versatile and Efficient Pattern Generator for Generalized Legged Locomotion

Justin Carpentier, Steve Tonneau, Maximilien Naveau, Olivier Stasse, Nicolas Mansard

***Abstract*— This paper presents a generic and efficient approach to generate dynamically consistent motions for under-actuated systems like humanoid or quadruped robots. The main contribution is a walking pattern generator, able to compute a stable trajectory of the center of mass of the robot along with the angular momentum, for any given configuration of contacts (e.g. on uneven, sloppy or slippery terrain, or with closed-gripper). Unlike existing methods, our solver is fast enough to be applied as a model-predictive controller. We then integrate this pattern generator in a complete framework: an acyclic contact planner is first used to automatically compute the contact sequence from a 3D model of the environment and a desired final posture; a stable walking pattern is then computed by the proposed solver; a dynamically-stable whole-body trajectory is finally obtained using a second-order hierarchical inverse kinematics. The implementation of the whole pipeline is fast enough to plan a step while the previous one is executed. The interest of the method is demonstrated by real experiments on the HRP-2 robot, by performing long-step walking and climbing a staircase with handrail support.**

I. Introduction

To use humanoid robots in factories or in disaster scenarios, a key practical issue is locomotion. For instance, in the aircraft industry, factories include a lot of stairs with steps of height between 20 to 30 cm, with a slope above 27 degrees. The large height implies the natural strategy to use the handrail on the side of the stairs. Despite its apparent simplicity, multi-contact locomotion in a generic way remains an open problem. This paper proposes a complete pipeline to automatically compute and execute such movements on a real robot, using as only inputs the 3D representation of the environment and a desired final posture.

The first versatile methods able to demonstrate effective multi-contact locomotion on a human-size humanoid robot were optimizing over the whole robot actuation on a relatively small time window [1]. Since then, multiple contact locomotion and manipulation have been demonstrated many times on such robots [2], [3], [4]. The major difficulty lies in the time to compute a feasible solution. Similarly to what was done for bipedal locomotion on flat floor [5], a natural way to simplify this problem is to only consider the dynamics of the robot through its momenta, namely its center of mass (COM) and angular momentum. Given a sequence of contacts and their timings, the problem is to compute the trajectory of the COM along with the angular momentum that would result in a dynamically stable whole body trajectory of the robot. Following some recent contributions to the problem [6], [7], [8], [9], we first propose an efficient way to compute such a trajectory. Our computation time are sufficiently low to enable online computation and model-predictive control (MPC). We also show how this solver can be connected to an efficient contact planner to automatically generate a feasible whole-body trajectory without any help from a human operator.

Section II recalls the fundamental basis to model the problem and introduces our notations. Both are used in Section III to cast a tailored optimal control problem which can be efficiently solved using a multiple shooting approach. We postpone the presentation of the state of the art to this section, so that we have the technical background to discuss it in detail. Section IV describes a planner that computes the contact sequence. The computed contacts are then used in Section V to bring the real humanoid robot HRP-2 atop of a staircase using the handrail, and various other movements demonstrate the interest of our approach.

The authors are with CNRS, LAAS, 7 avenue du colonel Roche, and Univ. de Toulouse, F-31400 Toulouse, France. email: {firstname.surname}@laas.fr

II. Dynamic model

A. Model of the whole body

We consider a free-floating based system composed of $6 + n$ degrees of freedom (DOF). Its configuration vector $\boldsymbol{q} \in \mathcal{Q} \stackrel{\text{def}}{=} SE(3) \times \mathbb{R}^n$ can be split in two parts: $M = (R, \boldsymbol{p}) \in SE(3)$ characterizes the placement of a given link of the robot relatively to the inertial frame (e.g. the center of the robot pelvis); and $\boldsymbol{q}_a \in \mathbb{R}^n$ is the configuration vector of the joints. The first and second time derivatives of $\boldsymbol{q}$ are denoted by $\dot{\boldsymbol{q}} = (\boldsymbol{v}, \boldsymbol{\omega}, \dot{\boldsymbol{q}}_a)$ and $\ddot{\boldsymbol{q}} = (\dot{\boldsymbol{v}}, \dot{\boldsymbol{\omega}}, \ddot{\boldsymbol{q}}_a)$ where $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are respectively the linear and angular velocity of the arbitrary free-floating base.

Given a set of K contacts $\mathcal{I} \subset SE(3)^K$, the Lagrangian dynamics of the polyarticulated system reads:

$$H_{\boldsymbol{q}}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + b_{\boldsymbol{q}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = g_{\boldsymbol{q}}(\boldsymbol{q}) + S^T \boldsymbol{\tau}_q + \sum_{k=1}^{K} J_k^T(\boldsymbol{q}) \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{\tau}_i \end{bmatrix}, \quad (1)$$

where $H_{\boldsymbol{q}}$ is the mass matrix, $b_{\boldsymbol{q}}$ is the centrifugal and Coriolis effects, $g_{\boldsymbol{q}}$ is generalized gravity vector, $S = \begin{bmatrix} 0_{n\times 6} & I_{n\times n} \end{bmatrix}$ distributes the joint-torque vector $\boldsymbol{\tau}_q$ on the articulation, J_k is the Jacobian of contact k and $\boldsymbol{f}_k$ and $\boldsymbol{\tau}_k$ are the force and torque applied at the contact k.

B. Contact model

We assume that each contact k corresponds to a rigid interface (i.e. no relative motion, only forces) between one body of the robot and the environment. Each contact is associated to a placement $M_k = (R_k, \boldsymbol{p}_k) \in SE(3)$ ie. the

position $\boldsymbol{p}_k$ of an arbitrary reference point of the contacting body in the world and the orientation R_k of this body.

The interface is defined by a finite set of contact points where only forces (no torques) are exerted. For instance, the contact of a rectangular foot with the ground is represented by the four corner contact points of the foot. Each force is typically constrained to stay within a friction (quadratic "ice-cream") cone defined by the friction coefficient μ.

Rather than considering for contact k the collection of all these forces, we only consider the resulting wrench: the linear force $\boldsymbol{f}_k$ and the torque $\boldsymbol{\tau}_k$ about $\boldsymbol{p}_k$. The wrench $(\boldsymbol{f}_k, \boldsymbol{\tau}_k)$ is constrained to be in a 6D conic set $\mathcal{K}_k$, obtained as the Minkowski sum of the cones of the contact points [10]. Considering $(\boldsymbol{f}_k, \boldsymbol{\tau}_k) \in \mathcal{K}_k$ is equivalent to considering all the forces of the interface in their 3D cones. $\boldsymbol{\phi} = (\boldsymbol{f}_1, \boldsymbol{\tau}_1, ..., \boldsymbol{f}_K, \boldsymbol{\tau}_K)$ is the concatenation of all contact wrenches, $\mathcal{K}$ is the Cartesian product of the 6D contact cones.

C. The under-actuated dynamics

The contact forces and torques influence the variations of linear and angular momenta. We denote by $\boldsymbol{h}$ the linear momentum and $\mathcal{L}$ the angular momentum around the COM of the robot (once more expressed in $\mathcal{F}_0$). Denoting by $\boldsymbol{c}$ the COM, the linear momentum is simply $\boldsymbol{h} = m\dot{\boldsymbol{c}}$ with m the total mass of the robot. The contact forces and torques modify the momentum according to the Newton-Euler law:

$$\dot{\boldsymbol{h}} = \sum_{k=0}^{K} \boldsymbol{f}_k + m\boldsymbol{g} \tag{2a}$$

$$\dot{\mathcal{L}} = \sum_{k} (\boldsymbol{p}_k - \boldsymbol{c}) \times \boldsymbol{f}_k + \boldsymbol{\tau}_k, \tag{2b}$$

with $\boldsymbol{p}_k$ the "center" of contact k around which $\boldsymbol{\tau}_k$ is expressed and $\boldsymbol{g} = (0, 0, -9.81)$ is the gravity vector.

These two equations simply correspond to the first six rows of (1), but expressed around the COM instead of the robot root. Let the dynamics of the free-floating base (first six rows) be denoted by index b and the dynamics of the actuated segment (n last rows) be denoted by index a.

$$\begin{bmatrix} H_b \\ H_a \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} b_b \\ b_a \end{bmatrix} = \begin{bmatrix} g_b \\ g_a \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_6 \\ \boldsymbol{\tau}_q \end{bmatrix} + \sum_{k=1}^{K} \begin{bmatrix} J_{k,b}^T \\ J_{k,a}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{\tau}_k \end{bmatrix} \tag{3}$$

The b rows corresponds to the total wrench applied on the robot, but expressed around the robot reference base instead of the COM. Following this observation, $J_{k,b}$ has a particular shape $J_{k,b} = \begin{bmatrix} I & (\boldsymbol{p}_k - \boldsymbol{b})\times \\ 0 & I \end{bmatrix} = {}^{b}X_k$ with $\boldsymbol{b}$ the position of the base and $\boldsymbol{v}\times$ corresponding to the skew matrix operator.

The total momentum of the system expressed around $\boldsymbol{c}$ is obtained by multiplying the first six rows of (3) by ${}^{c}X_b^*$:

$$\begin{bmatrix} \boldsymbol{h} \\ \mathcal{L} \end{bmatrix} = {}^{\boldsymbol{c}}X_b^* \, H_b \, \dot{\boldsymbol{q}}, \quad {}^{\boldsymbol{c}}X_b^* = \begin{bmatrix} I & 0 \\ (\boldsymbol{b} - \boldsymbol{c})\times & I \end{bmatrix} \tag{4}$$

D. Sequence of contacts and phases

In the following, we consider a sequence of contact configurations, i.e. an ordered collection of S contact sets $\{\mathcal{I}^1, ...\mathcal{I}^S\}$. Each contact set $\mathcal{I}^s$ corresponds to the *phase* s of the movement. Inside a phase, all the contacts of $\mathcal{I}^s$ are constants (according to the contact model). We denote the phase by exponent s: K^s is the number of active contacts during phase s; $\mathcal{K}_k^s$ is the 6D friction cone of contact k during phase s; etc. For instance a walking sequence may be first both feet on the ground, then only one foot on the floor, etc.

The duration of the phase is typically specified by a time interval $[\Delta\underline{t}^s, \Delta\bar{t}^s]$ of the minimum to maximum duration.

The number of contacts typically varies at each change of phase, thus the dimension of $\boldsymbol{\phi}$ varies too. In practice, two solutions might be considered. It is possible to consider that the wrenches of all possible contact bodies are contained in $\boldsymbol{\phi}$ while a binary variable α_k^s specifies if the contact k is active during phase s: α_k^s is added in the dynamic equations as a factor of every instance of $\boldsymbol{f}_k$ and $\boldsymbol{\tau}_k$ to nullify the effect of inactive contacts [8]. The interest of this solution is that the dynamics keep a constant dimension during all the movement, which simplifies the implementation of any control method. Alternatively, it is possible to specifically handle the variation of dimension of $\boldsymbol{\phi}$ in the implementation. The advantage is that there is no artificial "zeros" in the dynamics and the implementation is more efficient [9]. In the implementation of our method, we have chosen the second solution. In the following, we will abusively neglect the change of dimension to keep the presentation simple.

III. The optimal control problem

The dynamics (2) corresponds to the difficult part to control on a humanoid robot. It is not directly controllable by the joint torques $\boldsymbol{\tau}_q$ but only indirectly by the contact wrenches $\boldsymbol{\phi}$. It is also the part of the dynamics that is unstable (because of the cross-product with $\boldsymbol{c}$ on the second line, that will grow exponentially if something goes wrong). On the other hand, if the robot has enough torque (which current high-performance humanoid robots usually have) it will always be possible to find $\boldsymbol{\tau}_q$ to satisfy the actuated part of the dynamics if (2) is satisfied.

Walking pattern generators (WPG) therefore focus on (2) (or on a reformulation of it) to find a valid trajectory of $\boldsymbol{c}$ and $\mathcal{L}$ satisfying the contact constraints $\mathcal{K}$. In this direction, a very classical hypothesis to keep the problem simple is to assume that all the contacts are on a flat ground where slippage is impossible ($\mu = +\infty$), but recent contributions have been proposed to get rid of this hypothesis in a satisfactory manner [9], [8], [7]. In this section, we propose an original formulation of a WPG that is able to handle any distribution of contacts with interactive capabilities (ie. computing one step is faster than executing one).

In a first time, we present an optimal control problem (OCP) under a generic form that represents the problem of computing walking patterns. This form is not suitable for efficient resolution but can be seen as a generic template that covers several previous WPG. We then propose a new formulation that makes an interesting trade off between efficiency and generality. The last part of the section shows how the solution to this problem can be efficiently computed using a particular direct approach.

A. The generic optimal control problem

We consider the central dynamics (2) along a finite-time trajectory. The state of the problem is composed of the COM position and the momentum. We denote it by $\boldsymbol{x} = (\boldsymbol{c}, \boldsymbol{h}, \mathcal{L})$. The control of this dynamic system is $\boldsymbol{u} = \boldsymbol{\phi}$ the contact wrench. Eq. (2) can be easily reformulated as

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = F_x \boldsymbol{x} + F_u(\boldsymbol{x})\boldsymbol{u} \tag{5}$$

where F_x and $F_u(\boldsymbol{x})$ are two matrices easily deduced from (2). We denote by $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{u}}$ the state and control trajectories.

Starting with a sequence of contacts, we are interested in computing a feasible trajectory for the under-actuated dynamics, satisfying the Newton-Euler equations, path and terminal constraints. This can be achieved by setting the following OCP over all the sequence:

$$\min_{\substack{\underline{\boldsymbol{x}}=(\boldsymbol{c},\boldsymbol{h},\mathcal{L}),\\ \underline{\boldsymbol{u}}=\boldsymbol{\phi}}} \quad \sum_{s=1}^{S} \int_{t_s}^{t_s+\Delta t_s} \ell_s(\boldsymbol{x}, \boldsymbol{u})\, dt \tag{6a}$$

$$s.t. \quad \forall t \quad \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{6b}$$

$$\forall t \quad \boldsymbol{\phi} \in \mathcal{K} \tag{6c}$$

$$\forall t \quad \mathcal{L} \in \mathcal{B}_{\mathcal{L}} \tag{6d}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{6e}$$

$$\boldsymbol{x}(T) \in \mathcal{X}_* \tag{6f}$$

where $t_{s+1} = t_s + \Delta t_s$ is the start time of the phase s with $t_0 = 0$. Constraints (6b) and (6c) enforce consistent dynamics with respect to the contact model. Constraint (6d) imposes some bounds on the angular momentum (or its variations). Constraint (6e) constrains the trajectory to start with a given state (typically estimated by the sensor of the real robot) while (6f) typically enforces a viable terminal state [11]. The cost (6a) is typically decoupled in $\ell_x(\boldsymbol{x}) + \ell_u(\boldsymbol{u})$ whose parameters may vary according to the phase. ℓ_x is generally used to smooth the state trajectory while ℓ_u tends to equally distribute the forces, producing a more dynamic movement. The resulting control is stable as soon as ℓ_x involves the L-2 norm of one time derivative of $\boldsymbol{c}$ [11].

B. Previous formulations

Problem (6) is a difficult problem to solve in its generic form. In particular, it seems hard to find a closed-form expression of the viable states $\mathcal{X}_*$, or an equivalent form suitable for numerical resolution. Similarly, there is no evidence of what could be some realistic bounds $\mathcal{B}_{\mathcal{L}}$ (that would very likely depends on the configuration of the joints $\boldsymbol{q}_a$). In the following we list some of the main WPG methods and show how they correspond to some specific choices of the generic template (6).

1) Walking patterns in 2D: In addition to the previous remarks, another difficulty is the bilinear form of the dynamics (5). When the contacts are all taken on a same plane, a clever reformulation of the dynamics makes it linear [5], by neglecting the dynamics of both the COM altitude and the angular momentum. In that case, $\mathcal{K}$ boils down to the constraint of the zero-momentum point (ZMP) to lie in the support polygon.

Kajita et al. [5] did not explicitly check the constraint (6c); in exchange, ℓ_u is used to keep the control trajectory close to a reference trajectory provided a priori. Similarly, (6f) is not checked; in exchange, ℓ_x tends to stabilize the robot at the end of the trajectory by minimizing the jerk of the COM. These three simplifications turns (6) into a simple unconstrained problem of linear-quadratic regulation that is implicitly solved by integrating the corresponding Riccati equation.

The LQR was reformulated into an explicit OCP [12], directly solved as quadratic program. The OCP formulation makes it possible to explicitly handle inequality constraints: (6c) is then explicitly checked under its ZMP form. A modification of this OCP is proposed in [13] where (6f) is nicely approximated by the capturability constraint, which constrains the COM position and its first time derivative in case of planar contacts.

2) Walking patterns in 3D: An iterative scheme is proposed in [14] that can be written as an implicit optimization scheme whose cost function is the distance to a given COM trajectory and given forces distributions. The resulting forces satisfies (6c) by construction of the solution. There is no condition on the angular momentum (6d) neither on the viability of the final state (6f), however the reference trajectory enforced by the cost function is very likely to play the same role.

In [6], (6c) is explicitly handled (using the classic linear approximation of the quadratic cones). As in [7], (6f) is indirectly handled by minimizing the jerk. No condition (6d) on the angular momentum is considered. Additionally, the proposed cost function maximizes the robustness of the computed forces $\boldsymbol{\phi}$ and minimizes the execution time. Finally, constraints are added to represent the limitation of the robot kinematics.

In [7], $\dot{\mathcal{L}}$ is null by construction of the solution. Moreover, (6c) is supposed to always hold by hypothesis and is not checked, while (6f) is not considered but tends to be enforced by minimizing the norm of the jerk of the COM, like in [5]. These assumptions result in an (bilinear)-constrained quadratic program that is solved by a dedicated numerical method.

In [8], (6c) is handled under a simple closed form solution, while (6f) is not considered. To stabilize the resolution, the cost function tends to stay close to an initial trajectory of both the COM and the angular momentum, computed beforehand from a kinematic path. Consequently, (6d) is not considered either (as it will simply stay close to the initial guess).

In [9], the conic constraint is directly handled. The angular momentum is treated through the orientation of the system ($\mathcal{L} \approx \tilde{I}\boldsymbol{\omega} + \boldsymbol{\tau}_{\mathcal{L}}$, with $\tilde{I}$ the compound (rigid) inertia of the robot and $\boldsymbol{\tau}_{\mathcal{L}} = H_a \dot{\boldsymbol{q}}_a$ the angular momentum due to the internal gesticulation). $\tilde{I}\boldsymbol{\omega}$ is kept low by penalizing the large rotation $\boldsymbol{\omega}$ but $\boldsymbol{\tau}_{\mathcal{L}}$ is unlimited, resulting in (6d) not being checked. The viability (6f) is not checked neither, but like previously, it is approximately handled by minimizing

the derivatives of the state in the cost function (however the first derivatives instead of the third), while a reference trajectory of the COM is provided to keep a nice behavior of the numerical scheme. Additionally, constraints are added to represent the kinematic limits of the whole body.

3) Computing the contact placements: When considering an explicit OCP formulation, additional static variables can be added to the problem. Typically, the contact placements given as invariant in (6) might be computed at the same time. This was first proposed in [12] for a 2D WPG, and similarly used in [13] and other works by the same authors. In both cases, the contact placements are unlimited or similarly limited to a convex compact set. The problem becomes much harder when the contacts might be taken among a discrete set of placements. In [15], the problem was formulated has a mixed-integer program (i.e. having both continuous and discrete variables) in case of flat contacts, and solved using an interior-point solver to handle the discrete constraints. In [16], the same problem is handled using a dedicated solver relying on a continuation heuristic and illustrates with the animation of virtual avatars.

C. The tailored formulation

As mentioned earlier, template (6) is hard to implement as such. We propose a new instance of problem (6) able to compute a walking pattern for arbitrary 3D contacts without providing any reference state or control trajectory. Furthermore, we want the resulting trajectory to be smooth and feasible under the other whole-body constraints, and the method to be fast enough to deal with interactive capabilities.

1) Formulation: We suggest the following OCP:

$$\min_{\substack{\underline{\boldsymbol{x}}=(\boldsymbol{c},\boldsymbol{h},\mathcal{L}),\\ \underline{\boldsymbol{u}}=\boldsymbol{\phi}}} \sum_{s=1}^{S} \int_{t_s}^{t_s+\Delta t_s} \ell_h(\boldsymbol{x}) + \ell_\kappa(\boldsymbol{x}) + \ell_{\mathcal{L}}(\dot{\boldsymbol{x}}) + \ell_\phi(\boldsymbol{u})\, dt \quad (7a)$$

$$s.t. \quad \forall t \quad \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \quad (7b)$$

$$\forall t \quad \boldsymbol{\phi} \in \mathcal{K} \quad (7c)$$

$$\boldsymbol{x}(0) = (\boldsymbol{c}_0, \mathbf{0}, \mathbf{0}) \quad (7d)$$

$$\boldsymbol{x}(T) = (\boldsymbol{c}^*, \mathbf{0}, \mathbf{0}) \quad (7e)$$

$$\dot{\boldsymbol{h}}(0) = \dot{\mathcal{L}}(0) = \dot{\boldsymbol{h}}(T) = \dot{\mathcal{L}}(T) = \mathbf{0} \quad (7f)$$

where $\ell_h(x) = \lambda_h ||\boldsymbol{h}||^2$, $\ell_{\mathcal{L}}(\dot{\boldsymbol{x}}) = \lambda_{\mathcal{L}} ||\dot{\mathcal{L}}||^2$, $\ell_\phi(\boldsymbol{u}) = ||\boldsymbol{\phi}||^2$ and $\ell_\kappa(\boldsymbol{x}) = \sum_{k=1}^{K} \kappa(\boldsymbol{c}, \boldsymbol{p}_k)$ takes care of robot kinematic limits. It corresponds to an exponential barrier on the distance between the COM and the contact points:

$$\kappa(\boldsymbol{c}, \boldsymbol{p}_k) = \exp(\|\boldsymbol{c} - \boldsymbol{p}_k\| - u_b) + \exp(-\|\boldsymbol{c} - \boldsymbol{p}_k\| + l_b) \quad (8)$$

where u_b, l_b are the arbitrary upper and lower bounds. Additionally, the weight λ_h is adapted depending on the phase: for support phases involving large displacement (like a large movement of the swing foot), the weight is divided by 10 with respect to its nominal value.

2) Comments: Compared to the template (6), this OCP literally takes into account the actuation constraint (6c). We replaced the viability constraint (6f) by an easier formulation to reach a stable rest state at a given COM position.

While the trajectory of the COM is easy to draw, the shape of the angular momentum seems really hard to guess. However, neglecting it [9] or constraining it to zero [7] or to an a-priori guess [8] are not satisfactory solutions either. We propose to relax (6d) by penalizing the variations of the angular momentum quantity. Following [8] we also tried to penalize the deviation of the angular momentum from a reference trajectory (or to $\mathbf{0}$), but it did not improve the results obtained, thus we did not keep it. Finally, (7f) are initial and terminal constraints which ensure the under-actuated dynamics to be at rest at both ends.

We additionally enforced a constraint representing the kinematic limits, in the spirit of [16], [9]. Like in [9] and contrary to [16] we used a simple elliptic region to represent the reachability region. However, contrary to [9], we integrated this constraint as a smooth exponential barrier. We indeed noticed in practice that a hard constraint or a more aggressive log barrier tend to confuse the numerical solver.

On top of this, the proposed cost function manages a good trade-off between the dynamics of the trajectory and its smoothness.

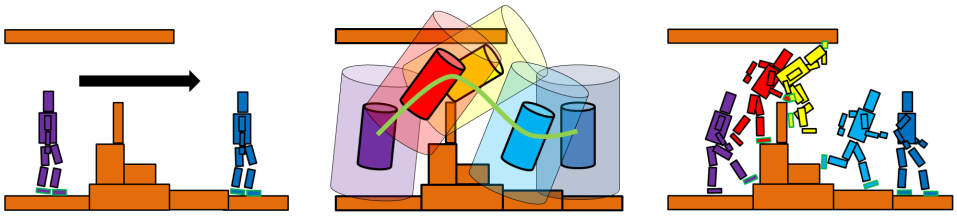
3) Additional variables: The phase durations Δt_s are also treated as variables, to be chosen in a specific interval. In addition, it would be straightforward to compute the contact placements in the same OCP for little additional cost.

D. The multiple shooting approach

Problems (6) and (7) consider variables of infinite dimension and cannot be directly handled by a computer. Addressing these nominal problems requires the use of indirect methods like the Pontryagin's maximum principle or dynamic programming, to reformulate the optimization problem as an integration problem of an augmented system. Unfortunately, these indirect approaches cannot handle (7) due to the bilinear constraint (7b). Alternatively, "direct" approaches turn the initial infinite-dimensional problem into a finite-dimensional one by constraining the control or the state trajectories to live in an arbitrary basis function.

Various details of implementation should be chosen to obtain an efficient resolution. The most important in our opinion is the way the pair $(\underline{\boldsymbol{x}}, \underline{\boldsymbol{u}})$ is handled. We refer to [17] for more details on the aforementioned methods. Collocation [6], [16] explicitly represents the state variable while the control is obtained from the state trajectory by inverting the system dynamics. On the other hand, single shooting [18], [7] explicitly represents the control trajectory while the state is obtained by integration. In between, multiple shooting makes explicit the control trajectory along with some few state variables at given shooting nodes.

Focusing on problem (7), the dynamics (2) is numerically quite stable: collocation, which tends to be robust to unstable dynamics, becomes unnecessary. On the contrary, it is relatively easy to build a good initial guess of the state trajectory, while guessing the control trajectory is a more complex affair. So, multiple-shooting and collocation are suitable while single shooting would be difficult to initialize. By elimination, multiple shooting is the best option.



(a) Initial request (b) Reachability planning (c) Contact sequence

Fig. 1: Contact planner overview. (a) A request between an initial and a final configuration. (b) A path is planned for the root of the robot with the reachability condition, abstracted with cylinders: the inner one must avoid collision while the outer one must be in collision. (c) The root path is discretized and extended into a sequence of static equilibrium configurations.

IV. Acyclic contact planning

To automatically generate an input contact sequence S for the pattern generator, we use a contact planner [19]. We recall its principle and detail modifications brought to it in the present work. An improved version of our planner, dedicated to robotics applications, has recently been submitted [20].

A. General principle

Our objective is to describe the motion between initial and goal postures with a discrete sequence of configurations in static equilibrium, thus in contact. To tackle this issue efficiently, we use a decoupled approach: we first compute a low-dimensional path for the root of the robot, then we compute a contact sequence along this path (Fig. 1).

When planning a root path, we check efficiently that a root configuration can lead to a whole-body configuration in static equilibrium with the "reachability condition". Informally, it verifies that a root configuration is "close, but not too close": close to allow contact creation (obstacles are in the reachable workspace – outer cylinders in Fig. 1), not too close to avoid collisions (a scaling of the root is collision-free – inner cylinders in Fig. 1). Any sampling-based planner can be used to compute a path of such configurations. Instead of the PRM [21] used in our previous work, we use a bi-RRT [22], which allows efficient online requests, compatible with the MPC capabilities of our pattern generator.

Once a root path has been computed, the whole-body static equilibrium configurations are generated along the discretized path. Between two successive configurations, they verify that one contact is created or broken at most. The generation can be biased with user-defined heuristics [23]. In this work we implement two heuristics to account for the limitations of the torque capabilities of the HRP-2 robot:

- The orientations of the feet are constrained to be parallel to the direction of motion;
- Contact generation is biased towards high manipulability limb configurations [24].

B. Kinematic interpolation of the contact sequence

The contact planner generates a collision free root path, but it does not provide a continuous path for the limbs. We thus introduce limb-RRT, a local interpolation method that computes a collision-free limb path between two successive configurations of the sequence, given the root path. The limb-RRT considers the following inputs:

- A kinematic chain l composed of n joints (here, $n = 6$). The origin of l is the geometrical root of the robot.
- $\boldsymbol{q}_0^l$ Initial configuration of limb l
- $\boldsymbol{q}_1^l$ Goal configuration of limb l
- $\boldsymbol{q}^r(t) : [0,1] \longrightarrow SE(3)$ a normalized interpolation path for the root of the robot.

It outputs $\boldsymbol{q}^l(t) = [0,1] \longrightarrow \mathbb{R}^6$, a collision-free path.

To take into account the root trajectory during the planning, we use a bi-RRT where configurations have an extra dimension $t \in [0,1]$, used to randomly sample root configurations in $\boldsymbol{q}^r(t)$. The graph is ordered according to t to ensure continuity of the root positions (an edge from a to b only exists if $t_a < t_b$).

The distance between two configurations is computed based on the n joint values, weighted by the length of the sub-kinematic chain they support. For instance for the robot arm, the three joints of the shoulder have a weight of 1, the two joints of the elbow a weight of $\frac{l_{\text{arm}} - (l_{\text{forearm}} + l_{\text{hand}})}{l_{\text{arm}}}$ and so on. The limb-RRT can directly consider bounds on joint velocities, and return the total time necessary to perform the motion. Optionally, it can also be bounded to find a solution respecting a time window.

V. Experimental Results

Two main experiments carried out on the HRP-2 robot are presented. The first experiment concerns the generation of a classic walking motion: it is a unitary test but it is important to properly understand the behaviour of our solver compared to classic WPG. The second experiment is the climbing stairs scenario depicted in Fig. 2, where the robot has to make use of the handrail to help its ascension of the stairs. We additionally show a standing motion not yet executed on the robot but demonstrating the versatility of our approach.

A. Experimental setup

All the computations were performed offline on a Intel Xeon(R) CPU E3-1240 v3 @ 3.40GHz. The contact planner is open-source and available at `https://github.com/humanoid-path-planner`. The OCP is solved using the proprietary software MUSCOD-II provided by the Interdisciplinary Center for Scientific Computing (IWR) of Heidelberg University. This software offers an OCP toolkit

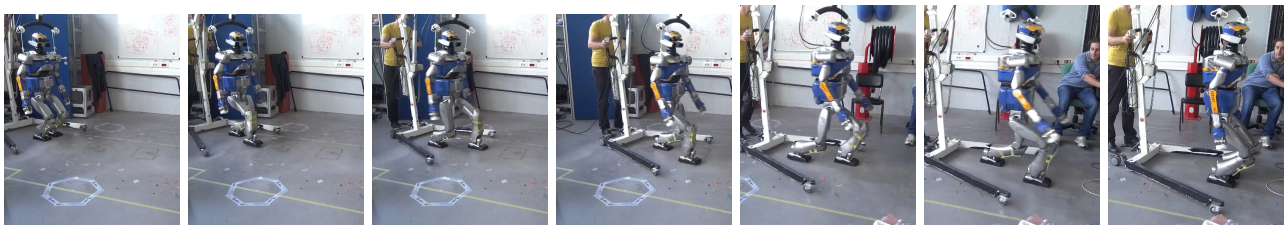


Fig. 2: Exp. 1: Walking in straight line with large stride of 100cm.

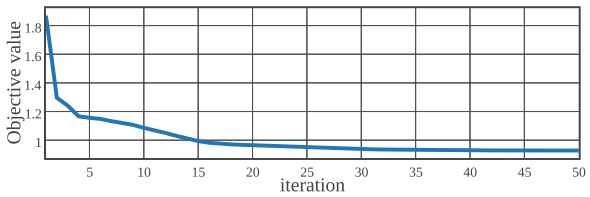


Fig. 3: Exp. 1 - Evolution of the cost function.

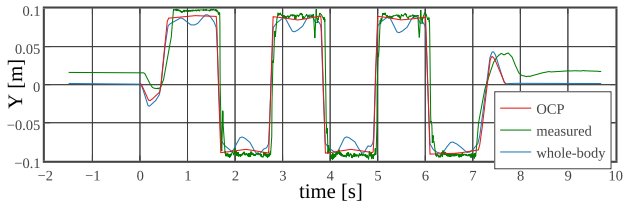


Fig. 4: Exp. 1 - ZMP trajectories obtained from the OCP, the multi-body dynamics and the measurements.

(e.g. integration and numerical-differentiation routines) along with an efficient sparse sequential-quadratic-program solver. The whole-body trajectory is obtained from the contact sequence and the COM and angular-momentum trajectory using a second-order inverse kinematics. The typical tasks where the tracking of the contact placements, the tracking of the COM position and an additional posture task to keep the configuration close to the planned postures. The computed trajectories are then executed by the real robot. For the walking experiments, we used a closed-loop control provided with the robot to stabilise the movements of the rubber bush inside each foot [25]. The stabiliser was not used for the climbing scenario as it is not able to handle hand contacts.

B. Experiment 1: large stride on a flat ground

In this first experiment, a sequence of cyclic contacts is manually generated with 100 cm long stride (a very large step compared to the 1.60 m height of the HRP-2 robot). The timings are fixed (single support: 1.0 s; double support: 0.1 s) with a total duration of 8.2 s. We then compute a feasible COM trajectory using the proposed OCP. The foot trajectories are a collection of splines connecting the desired contact placements and ensuring a zero velocity and acceleration during take-off and landing of the foot. The experiment is summarized by Fig. 2 to 4.

Fig. 3 reports the numerical behaviour of the OCP solver. A near optimal solution (i.e. KKT tolerance below 10^{-6}) is obtained in 4 s after 50 iterations of the multiple shooting algorithm. The objective value decreases rapidly in the beginning, and slows down its progression as the algorithm tries to fulfil the path constraints. After a feasible solution is found, every new iteration (i.e. what is computed during one iteration of a MPC) lasts 40 ms. The overall movement is depicted in Fig. 2 and in the accompanying video.

Fig. 4 shows the ZMP trajectory on the Y axis resulting from the OCP, compared to the estimation coming from force sensors measurement. The ZMP is very similar to what could be obtained by a classic WPG with assumption of flat contact. The proper tracking on the real robot shows the dynamic consistency of the output of the OCP.

C. Experiment 2 : climbing stairs equipped with a handrail

In the climbing scenario, the contact sequence given by the planner is no more cyclic and takes around 1s to be computed. The computation of a feasible trajectory to climb one stair is done in less than 5.5s after 85 iterations.

Fig. 6 illustrates both the forces computed by the solver and the forces exerted on the real robot. The simulated and measured forces do not match exactly but they have similar variations. In both cases, we observe that the robot makes use of its right hand either for pulling or pushing. The oscillations in the forces response are mainly due to the presence of a flexibility part in the robot's feet and to the compliance of the handrail. These two external disturbances are not considered in our framework.

D. Simulated motion

A standing-up motion was also planned, where the robot exploits the proximal environment in order to stand up. For this movement, the sequence of contact configurations is nontrivial and would have been difficult to build manually. Fig. 7 and the companion video illustrate the motion.

VI. Conclusion and Future Works

In this paper, we have proposed an original formulation to efficiently generate walking pattern for any multi-contact scenarios. The walking pattern generator is written as a nonlinear optimal control problem, which can be solved very efficiently, leading to interactive capabilities. While building an efficient problem, we kept the formulation as generic as possible, by comparing and justifying every technical choice with respect to several other formulations proposed in the state of the art. The few arbitrary modelling choices that were taken are clearly exhibited and might be the topic of future research. For example, a future direction may be to express more clever bounds on the angular momentum variations.

We have also shown how to integrate this WPG in a complete application, by using a contact planner to compute the reference contact sequence. The dynamically-consistent

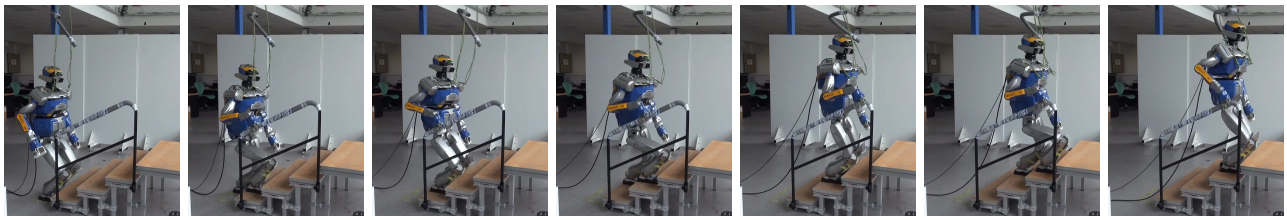


Fig. 5: Exp.2 - Climbing the stairs of 15cm height by using the handrail.

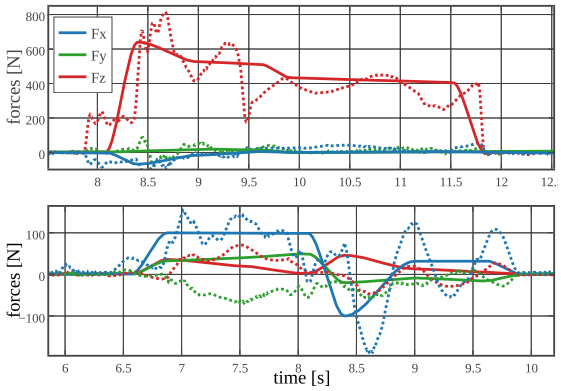


Fig. 6: Exp. 2 - Reference (solid line) and measured (dotted line) forces acting on the right foot (on top) and hand (on bottom) during one contact phase.

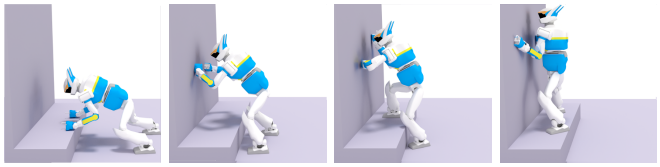


Fig. 7: Exp.3 - The robot is standing up thanks to wall contacts.

whole-body trajectory is finally obtained by performing a second-order inverse kinematics using the COM and angular momentum reference of the pattern generator. The complete pipeline is interactive on the demonstrated examples and in most of the classic scenarios that a robot could meet in a factory. The maturity of the approach was demonstrated with two real executions with the HRP-2 robot and another example in simulation.

Acknowlegment

This work is supported by the European Research Council through the Actanthrope project (ERC Grant Agreement 340050), the European project KOROIBOT (FP7 Grant Agreement 611909), the French National Research Agency project ENTRACTE (ANR Grant Agreement 13-CORD-002-01) and the national PSPC-Romeo 2 project.

References

[1] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *Int. Journal of Robotics Research (IJRR)*, vol. 32, no. 9-10, 2013.

[2] S. Noda, M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Generating whole-body motion keep away from joint torque, contact force, contact moment limitations enabling steep climbing with a real humanoid robot," in *ICRA*, 2014.

[3] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot," in *ICRA*, 2015.

[4] M. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing," in *ICHR*, 2014.

[5] 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 *ICRA*, 2003.

[6] Z. Qiu, A. Escande, A. Micaelli, and T. Robert, "Human motions analysis and simulation based on a general criterion of stability," in *International Symposium on Digital Human Modeling*, 2011.

[7] N. Perrin, D. Lau, and V. Padois, "Effective generation of dynamically balanced locomotion with multiple non-coplanar contacts," in *Int. Symp. on Robotics Research (ISRR)*, 2015.

[8] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *ICHR*, 2015.

[9] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres, and K. Mombaur, "Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations," 2015.

[10] S. Caron, Q.-C. Pham, and Y. Nakamura, "Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas," in *ICRA*, 2015.

[11] P.-B. Wieber, "Viability and predictive control for safe locomotion," in *ICHR*, 2008.

[12] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Advanced Robotics*, vol. 24, no. 5-6, 2010.

[13] A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Whole body motion controller with long-term balance constraints," in *ICHR*, 2014.

[14] H. Hirukawa, S. Hattori, S. Kajita, K. Harada, K. Kaneko, F. Kanehiro, M. Morisawa, and S. Nakaoka, "A pattern generator of humanoid robots walking on a rough terrain," in *ICRA*, 2007.

[15] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *ICHR*, 2014.

[16] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 43, 2012.

[17] M. Diehl, H. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions in Biomechanics and Robotics*, vol. 340, 2005.

[18] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *ICRA*, 2014.

[19] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon, and J. Pettre, "A reachability-based planner for sequences of acyclic contacts in cluttered environments," in *ISRR*, 2015.

[20] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots." submitted to IJRR – (hal id: 01267345v2), 2016.

[21] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, 1996.

[22] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *Int. Journal of Robotics Research (IJRR)*, vol. 20, no. 5, 2001.

[23] S. Tonneau, J. Pettré, and F. Multon, "Using task efficient contact configurations to animate creatures in arbitrary environments," *Computers & Graphics*, vol. 45, 2014.

[24] T. Yoshikawa, "Manipulability of robotic mechanisms," *The Int. Journal of Robot. Research (IJRR)*, vol. 4, no. 2, 1985.

[25] Y. Mikami, T. Moulard, E. Yoshida, and G. Venture, "Identification of hrp-2 foot's dynamics," in *IROS*, 2014.