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, that is 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 previously proposed method, our solver is efficient enough to be applied as a model-predictive controller. We then propose to integrate this versatile walking pattern generator in a complete application: 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 real-time. The interest of the method is demonstrated by real experiments on the HRP-2 robot, by performing long-step walking and by climbing a staircase with handrail support.**

I. Introduction

To introduce humanoid robots in factories or in disaster environment a key practical problem is the locomotion capability. In fact, factories such as the ones in the aircraft industry have a lot of stairs with steps of height between 20 to 30 cm with a slope above 27 degrees. The quite large size of those staircases implies the natural strategy to use the handrail on the side of the stairs. Despite its apparent simplicity, the problem of multi-contact locomotion in a generic manner is still open. This paper proposes a complete pipeline to automatically compute and execute such movements on a real robot, using as only inputs the 3D map of the environment and a desired final posture.

The first versatile methods able to demonstrate effective multi-contact locomotion on human-size humanoid robot were optimizing over the whole robot actuation on a relatively small time window [1]. Since then, multiple contacts locomotion and manipulation have been demonstrated multiple times on humanoid 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 the correspondent timings, the problem is to compute the trajectory of the COM along with the angular momentum that would result in a dynamically stable trajectory of the robot's whole body. 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 complete stable trajectory of the robot's whole body without any action of 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 {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: $M = (R, p) \in SE(3)$ (represented by a homogeneous matrix $M \in \mathbb{R}^{4\times4}$) characterizes the placement of a given link of the robot relatively to the inertial frame; 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 (typically, the center of the robot pelvis).

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

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

where $M_{\boldsymbol{q}}$ is the generalized inertia matrix, $b_{\boldsymbol{q}}$ is the centrifugal and Coriolis effects, $g_{\boldsymbol{q}}$ is the action of the gravity field, $S = \begin{bmatrix} 0_{n\times6} & I_{n\times n} \end{bmatrix}$ is a selection matrix which distributes the joint-torque vector $\boldsymbol{\tau}_q$ over the configuration space, J_k is the Jacobian of the contact k and $\boldsymbol{f}_k$ and $\boldsymbol{\tau}_k$ are the force and torque applied at the contact k.

In the following, we consider that the quantities $\boldsymbol{v}$, $\boldsymbol{\omega}$, $\boldsymbol{f}_k$, $\boldsymbol{\tau}_k$ are all represented in a common arbitrary inertial frame $\mathcal{F}_0$. We indistinctly design the Euclidean vector and its numerical representation. Similarly, the matrices $M_{\boldsymbol{q}}$ and J_k are written in the same coordinate system.

B. Contact model

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

The interface is defined by a finite set of contact points where only forces (no torques) are exerted. For example, the contact of a rectangular foot with the ground is represented by four contact points corresponding to the four corners 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, i.e. 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.

We denote by $\boldsymbol{\phi} = (\boldsymbol{f}_1, \boldsymbol{\tau}_1, ..., \boldsymbol{\tau}_K)$ the concatenation of all the contact wrenches, and by $\mathcal{K}$ the Cartesian product of the 6D contact cones.

C. The under-actuated dynamics

The contact forces and torques are related to the change 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 3D 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.

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

The b rows corresponds to the total wrench applied to the robot, but expressed around the robot basis 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 robot basis and $\boldsymbol{v}\times$ the skew matrix associated to any vector $\boldsymbol{v}$.

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

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

Eqs. (2) are obtained considering the velocity of the COM instead of the velocity of the base in the configuration velocity.

D. Sequence of contacts and phases

In the following, we will 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 example a walking sequence would be first both feet touching 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 (with $\Delta\underline{t}^s = \Delta\bar{t}^s$ when the duration is specified and $\Delta\underline{t}^s = 0$, $\Delta\bar{t}^s = +\infty$ when no prior information is available).

The number of contacts typically varies at each change of phase, and therefore the dimension of $\boldsymbol{\phi}$. 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 keeps 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 real-time capabilities.

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 $t_0 = 0$ and $t_K = T$). Constraints (6b) and (6c) enforce a consistent dynamics with respect to the contact model. Constraint (6d) imposes some bounds on the angular momentum (or its variation). Constraint (6e) imposes the trajectory to start to a given state (typically estimated by the sensor of the real robot). Constraint (6f) typically imposes the terminal state to be viable [11]. The cost (6a) is typically decoupled $\ell_s(\boldsymbol{x}, \boldsymbol{u}) = \ell_x(\boldsymbol{x}) + \ell_u(\boldsymbol{u})$ whose parameters may vary depending on the phase. ℓ_x is generally used to regularize and to smooth the state trajectory while ℓ_u tends to minimize the forces, then producing a more dynamic movement. The resulting control is stable as soon as ℓ_x comprehend the integral of the norm of one derivative of $\boldsymbol{c}$ [12].

B. Previous formulations

Problem (6) is a difficult problem to solve in its generic form. It seems in particular difficult to find a closed-form 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 be in the support polygon.

Kajita et al. [5] did not explicitly check the constraint (6c) is not explicitly checked; 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 [13], 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 [14] where (6f) is nicely approximated by the capturability constraint, which is a linear constraint on the COM and its first time derivative in case of 2D contacts.

2) Walking patterns in 3D: An iterative scheme is proposed in [15] 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 $\boldsymbol{\tau}_{\mathcal{L}} = A\dot{\boldsymbol{q}}_a$ the angular momentum due to the motion of the segment, i.e. the gesticulation of the whole body). $\tilde{I}\boldsymbol{\omega}$ 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 placement of the contact, that are given as invariant in (6), might be computed at the same time. This was first proposed in [13] for a 2D WPG, and similarly used in [14] and other works by the same authors. Similarly, it was proposed in [8] to include it in the proposed 3D WPG, but this feature was not implemented nor demonstrated. In both cases, the placements of the contacts 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 [16], the problem was formulated has a mixed-integer program (i.e. having both continuous and discrete variables) in case of flat contact, and solved using an interior-point solver to handle the discrete constraints. In [17], the same problem is handled using a dedicated solver relying on a continuation heuristic, and demonstrated for animating the motion of virtual avatars.

C. The tailored formulation

Our objective in proposing a new implementation of a problem following the template (6) is to have a reliable solution to compute a walking pattern for arbitrary 3D contacts without providing reference state or control trajectories. Additionally, we want the resulting trajectory to be smooth and to be easily executed by the robot under the other whole-body constraints, and to be computationally compatible with real-time capabilities.

1) Formulation: The proposed OCP is as follow:

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

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

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

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

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

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

where $\ell_h(x) = \lambda_h ||h||^2$, $\ell_{\mathcal{L}}(\dot{\boldsymbol{x}}) = \lambda_{\mathcal{L}} ||\dot{\mathcal{L}}||^2$, $\ell_\phi(\boldsymbol{u}) = ||\boldsymbol{\phi}||^2$ and $\ell_\kappa(\boldsymbol{x}) = \sum_{k=1}^{K} \kappa(\boldsymbol{c}, \boldsymbol{p}_k)$ represents the kinematic limits of the robot whole body by setting 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 phase 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 have reach a stable rest state at a given COM position.

While the trajectory of the center of mass position 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] is not satisfactory either. We propose to relax (6d) by penalizing the variation of the momentum. Following [8] we also tried to penalize the deviation of the angular momentum from a reference trajectory (or to 0), but this did not bring any additional value and we therefore did not keep it.

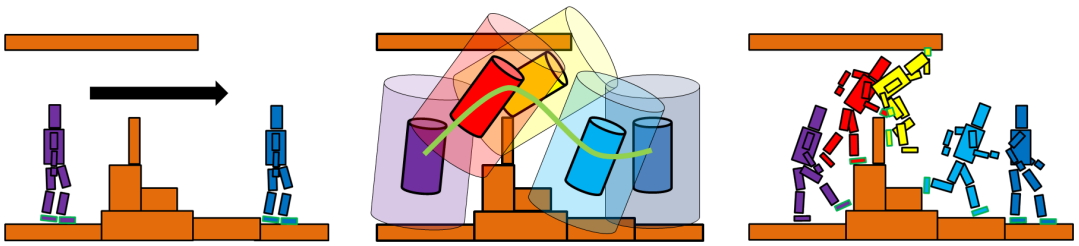
We additionally enforced a constraint representing the kinematic limits, in the spirit of [17], [9]. Like in [9] and contrary to [17] we used a simple elliptic region to represent the reachability region. However, contrary to [9], we integrated this constraint as a smooth exponential barrier. We indeed 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. Additionally, 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



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

Fig. 1: General principle of the contact planner. (a) The request is made by defining an initial and a final posture. (b) A central path is planned with the abstract shape, where the inside part of the shape should avoid collision while the outside part must stay in collision. (c) The central path is refined to compute a complete contact sequence.

indirect results like Pontryagins maximum principle or the dynamic programming, to reformulate the optimization problem as an integration problem of an augmented system. It is unlikely to solve (7) this way due to the bilinear constraint (7b). Alternatively, "direct" approaches turn the initial infinite-dimension problem into a finite-dimensional one by sampling the trajectory over an arbitrary function basis.

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. Collocation [6], [17] represents explicitly 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] represents explicitly the control trajectory while the state is obtained by integration. In between, multiple shooting represents explicitly the control trajectory along with some few state variables at given time instants.

Consider specifically the problem (7). The dynamics (2) is numerically quite stable: collocation, which tends to be quite robust to unstable dynamics, is not necessary. On the other hand, it is relatively easy to build a good initial guess of the state trajectory, while a guess of the control trajectory is more complex to have: single shooting would be difficult to initialize, but both multiple-shooting and collocation are suitable. By elimination, multiple shooting is the best option. We refer to [19] for more details on the aforementioned methods.

IV. The acyclic contacts planning problem

In this section, we quickly recall the basis of the contact planner presented in [20] and present the modifications that were added to the planner to integrate it with the proposed pattern generator.

A. General principle

A contact planner is an algorithm that connects two given initial and final postures by computing a sequence of contacts, that is to say a finite list of postures, each of them in a static equilibrium in contact with the environment, where each posture differs from the previous one by only one contact (i.e. one contact is removed or one new contact is created with a limb of the robot that was previously collision free). Existing contact planners can be separated in two classes: motion-before-contact planners and contact-before-motion planner. We proposed in [20] a very efficient motion-before-contact planner, i.e. it first computes a valid motion before computing explicitly the contacts that make this motion valid.

The main principle of our contact planner is depicted on Fig. 1. We recall it here to be self-contained. The interested reader is referred to the original paper for more details and a discussion about the interest of this approach versus the state of the art.

B. Reachability-based planner

In a first stage, a general path is planned such that it is very likely to find a valid sequence of contacts. We introduced an abstract shape that represents the robot with few DOF. This shape is composed of a inner part, that is typically a hull of the torso and must avoid collision; and of an outer part that corresponds to the reachable region of the end-effectors of limbs and must stay in collision. The reachable regions are computed off line to be the convex hull of a regular sampling. The idea is that if the inner part is collision free and the outer part collides with the environment, it is very likely to find a whole-body posture of the robot with only contacts at the robot end effectors. Depending of the size of the inner hull, this abstract condition might be a necessary condition or a sufficient condition for finding a posture in contact. The size of the shape is then adjusted off-line on a training environment to maximize the number of true positives while minimizing the number of false positives.

In our previous work, our reachability based planner was implemented as a probabilistic roadmap (PRM – [21]), so as to allow multiple queries on the same scene. This implies a long off-line initialization phase. In this new implementation, our planner is implemented as a variation of the bidirectional rapidly-exploring random tree (bi-RRT) algorithm [22], allowing efficient real time online requests, that are more in line with the MPC capabilities of our pattern generator.

C. Computing the contacts

When a general movement is obtained that connects the initial posture to the final one, the contacts to actuate this motion are computed. The contact sequence is built incrementally from the initial posture, by advancing the root of the robot along the planned path until the current contact set cannot results in a collision-free posture in static equilibrium. A contact is then heuristically broken then created again, by optimizing a comfort measure named EFORT [23], based on manipulability ellipsoids. Each computed posture ensures static equilibrium, by using the validation algorithm proposed in [6].

In the original implementation, the postures of the sequence were simply randomly shot (using a precomputed pseudorandom generator) while no robustness was taken into account in checking equilibrium. This might result in some undesired configurations that would be very unlikely to be executed on a real robot. While the proper solution to this problem is to implement a robust validation of the equilibrium, we rather rely here on some simple heuristics to account for the limitations of the real HRP-2 torque capabilities:

- The orientation of the feet is constrained to be projected in the direction of the motion;
- The contact generation is also biased towards limb configurations with high manipulability, as opposed to configurations maximizing EFORT.

D. Computation of the trajectory of the limbs

In [20], we stopped after computing a valid contact sequence. We additionally propose here a methods, the limb-RRT planner, that computes collision-free limb path. More precisely, the limb-RRT computes a collision-free path connecting two given initial and final configurations (taken from two successive postures of the sequence), for a 6-DOF limb attached to a moving root. The motion of the root is an invariant of this problem.

The limb-RRT considers the following inputs:

- A kinematic chain l comprising $n = 6$ joints. 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 $\underline{\boldsymbol{q}^l} = [\boldsymbol{q}_0^l, \boldsymbol{q}_i^l ..., \boldsymbol{q}_1^l]$, a collision free sequence of configurations for the limb, such that $i \in [0,1]$ corresponds to a configuration for root position $\boldsymbol{q}^r(i)$.

The difficulty here is to take into account the moving root inside an efficient bi-RRT algorithms. The proposed method is generic and would work for any n. It is is implemented by planning explicitly with the time, i.e. the sampling is performed in dimension $n+1$ with the compound variable $(\boldsymbol{q}, t)$. The extra parameter $t \in [0,1]$ is used to randomly sample root positions in $\boldsymbol{q}^0(t)$. The graph is ordered according to t to ensure continuity of the root positions (i.e. an edge from a to b does not exists if $t_a \geq t_b$). For the bi-RRT, this means that new configuration $\boldsymbol{q}_i^l$ is connected to a node from the end set $\boldsymbol{q}_j^l$ if and only if $i < j$. Conversely $\boldsymbol{q}_i^l$ is connected to a node from the start set $\boldsymbol{q}_j^l$ if and only if $i > j$. This prevents the insertion of a blocking node that would make impossible the connection of the two trees of the bi-RRT (resulting in a dead-lock).

To guide the search, we used the following heuristic: the distance between two configurations is computed only based on the n joint values, weighted by the length of the subkinematic 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. It returns the total time necessary to perform the motion. Optionally, it can also be bounded to find a solution respecting a time window, and fails if it does not succeed in doing so.

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 behavior of our solver compared to classical walking pattern generators. 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 two movements of running and standing up, that were not executed by the robot but help to demonstrate the versatility of the approach.

A. Experimental setup

All the computations (contact planning and WPG) were performed offline on a Intel Xeon(R) CPU E3-1240 v3 @ 3.40GHz. The contact planner is open-source and is available at `https://github.com/humanoid-path-planner`. The OCP is solved using the proprietary software MUSCOD provided by the Interdisciplinary Center for Scientific Computing (IWR) of Heidelberg University. This software offers a OCP toolkit (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 (i.e. computes the joint acceleration using the jacobian pseudoinverse). 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 (called the stabilizer) to stabilize the movements of the rubber bush inside each foot [24]. The stabilizer was not used for climbing the stairs, because it is not able to handle hand contacts.

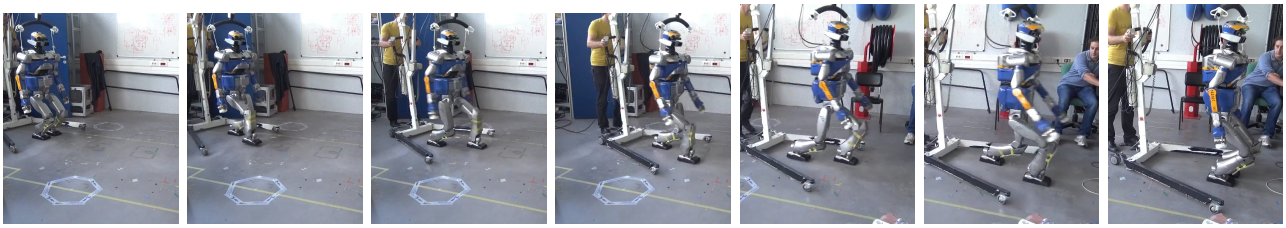


Fig. 2: Experiment 1: Walking in straight line with enjambment of 100cm.

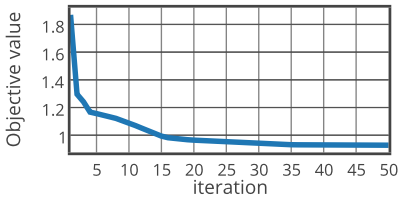


Fig. 3: Experiment 1: Evolution of the cost function along the iterations.

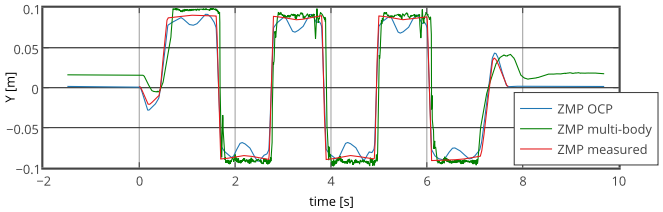


Fig. 4: Experiment 1: ZMP trajectories obtained from the OCP, the multi-body dynamics and the measurements.

B. Experiment 1: large enjambment on a flat ground

In this first experiment, a sequence of cyclic contacts is manually generated with enjambment of 100 cm. The timings are fixed (single support: 1.0 s; double support: 0.1 s). The total duration of the trajectory is 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 configuration given by the contact sequence 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 behavior 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 function decreases rapidly in the beginning, and slows down its progression as the algorithm tries to fulfill 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 (see also the accompanying video). The steps are very large for the humanoid robot (which is 1.60 m tall).

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 classical 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 physical disturbances are not considered in our framework.

D. Other motion

We also report two movements in simulation to exhibit some additional aspect of the approach. A running movement was computed with a periodic contact sequence. It implies some ballistic phases that the solver properly manages to handle. The resulting COM trajectory is presented in Fig. 7.

A standing-up motion was 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 manually build. This motion is illustrated by Fig. 8. See also the companion video.

VI. Conclusion and Future Works

In this paper, we have proposed an original formulation to efficiently generate walking pattern generator for any multi-contact configurations. The walking pattern generator is written as a nonlinear optimal control problem, which can be solved very efficiently, leading to real-time 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 modeling 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 for the angular momentum.

We also shown how to integrate this walking pattern generator in a complete implementation, by using a contact planner to compute the reference contact sequence.

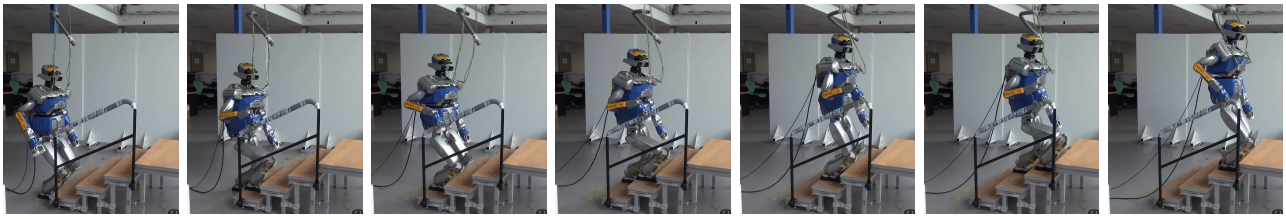


Fig. 5: Experiment 2: Climbing the stairs of 15cm height while using the handrail.

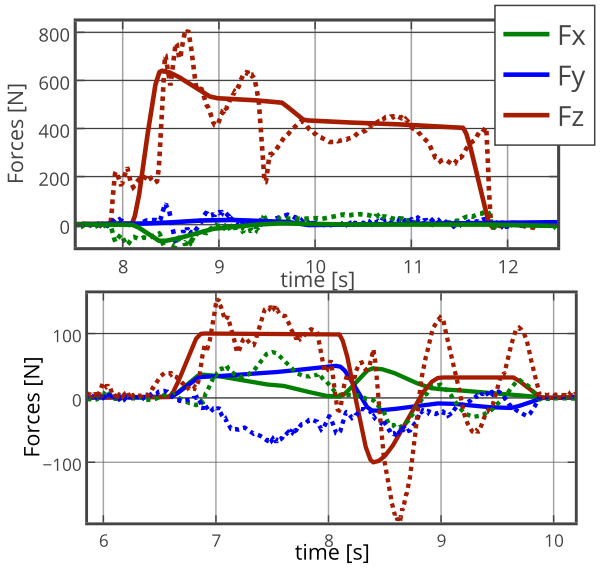


Fig. 6: Experiment 2: Reference (solid line) and measured (doted line) forces at the right foot (on top) and hand (on bottom) during one contact phase. The reference forces are properly tracked (even if some flexibility can be observed).

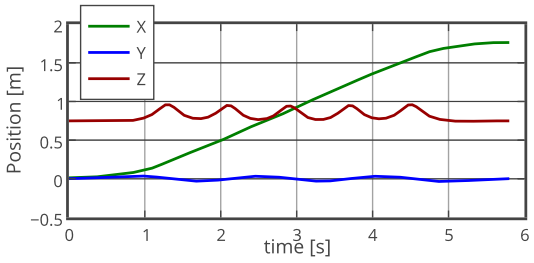


Fig. 7: Experiment 3: COM trajectory for the running scenario. One can recognize the ballistic shape of the Z component.

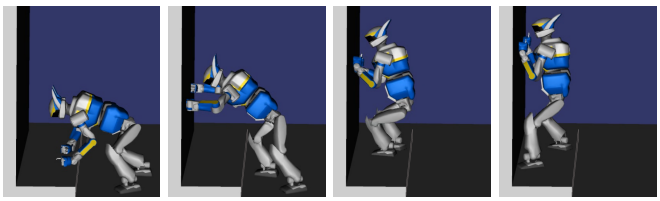


Fig. 8: Experiment 3: The robot is standing up helped by the contacts which it is making with the surrounding environment.

The dynamically-consistent whole-body trajectory is finally obtained by performing a second-order inverse kinematics using the COM references of the pattern generator and the posture and limbs movements from the planner. The complete pipeline is real-time on the demonstrated examples and in most of the classical situation that a robot could meet in a factory. The maturity of the approach was demonstrated by two real executions with the HRP-2 robot and another example in simulation.

Acknowlegment

The work is supported by the European Research Council (ERC) through the Actanthrope project (ERC-ADG 340050), the European project KOROIBOT (FP7-ICT-2013-10/611909) and the French National Research Agency (ANR) project ENTRACTE (13-CORD-002-01). We warmly want to thank Katja Mombaur and the Simulation and Optimization group at the Interdisciplinary Center for Scientific Computing (IWR) of Heidelberg University in Germany for providing the optimal control framework MUSCOD.

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, pp. 1104–1119, 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*, pp. 1775–1781, 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. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing," in *ICHR*, pp. 112–119, IEEE, 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*, (Taipe, Taiwan), September 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)*, (Sestri Levante, Italy), 2015.

[8] N. Rotella, A. Herzog, S. Schaal, and L. Righetti, "Humanoid momentum estimation using sensed contact wrenches," 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*, (Seattle, USA), May 2015.

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

[12] P.-B. Wieber, R. Tedrake, and S. Kuindersma, *Handbook of Robotics*, ch. Modeling and Control of Legged Robots. Springer, 2 ed., 2015.
[13] 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, pp. 719–737, 2010.
[14] A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Whole body motion controller with long-term balance constraints," in *ICHR*, 2014.
[15] 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*, (Roma, Italy), 2007.
[16] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *ICHR*, (Madrid, Spain), 2014.
[17] 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.
[18] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *ICRA*, (Hong-Kong, China), 2014.
[19] 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, pp. 65–93, 2005.
[20] 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 *Int. Symp. Robotics Research (ISRR)*, (Sestri Levante, Italy), September 2015.
[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, pp. 566–580, 1996.
[22] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *Int. Journal of Robotics Research (IJRR)*, vol. 20, no. 5.
[23] S. Tonneau, J. Pettré, and F. Multon, "Using task efficient contact configurations to animate creatures in arbitrary environments," *Computers & Graphics*, vol. 45, pp. 40–50, 2014.
[24] Y. Mikami, T. Moulard, E. Yoshida, and G. Venture, "Identification of hrp-2 foot's dynamics," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 927–932, Sept 2014.