Article

The International Journal of
Robotics Research
32(9-10) 1104–1119

DOI: 10.1177/0278364913478990
ijr.sagepub.com

Generation of whole-body optimal dynamic multi-contact motions

Sébastien Lengagne[1,2], Joris Vaillant[1,3], Eiichi Yoshida[1] and Abderrahmane Kheddar[1,3]

Abstract

We propose a method to plan optimal whole-body dynamic motion in multi-contact non-gaited transitions. Using a B-spline time parameterization for the active joints, we turn the motion-planning problem into a semi-infinite programming formulation that is solved by nonlinear optimization techniques. Our main contribution lies in producing constraint-satisfaction guaranteed motions for any time grid. Indeed, we use Taylor series expansion to approximate the dynamic and kinematic models over fixed successive time intervals, and transform the problem (constraints and cost functions) into time polynomials which coefficients are function of the optimization variables. The evaluation of the constraints turns then into computation of extrema (over each time interval) that are given to the solver. We also account for collisions and self-collisions constraints that have not a closed-form expression over the time. We address the problem of the balance within the optimization problem and demonstrate that generating whole-body multi-contact dynamic motion for complex tasks is possible and can be tractable, although still time consuming. We discuss thoroughly the planning of a sitting motion with the HRP-2 humanoid robot and assess our method with several other complex scenarios.

Keywords

Multi-contact motions, optimization, balance, Taylor series approximation, time-interval discretization, humanoid robots

1. Introduction

Planning dynamic motion for complex robotic systems such as humanoids, under various constraints, is known to be a complex problem in robotic research. For example, locomotion in cumbersome environments is non-gaited and approaches developed for biped cyclic locomotion do not apply. We propose a method that can generate gaited or non-gaited, optimal and dynamic multi-contact motions of the robotic system (in the sense of integrating a preview of what the following support contacts are).

Classical approaches usually solve the problem in two steps: the first step plans the motion of a reduced simple model (Kajita et al., 2003) that encompasses the main invariants of the motion (center of mass, centers of pressure, etc.); the second step consists of generating the whole-body dynamic motion that tracks at best the first step plan but where additional constraints (joint limits, torque limits, collision detection, etc.) are checked a posteriori. Hence, the first step assumes viability of the second step. In some typical tasks (e.g. walking) precaution are taken for the first step planning to generate very likely feasible plans. However, for more general tasks such as moving in cumbersome situations it is hard to both plan based on reduced/simple models and make plausible assumptions on the feasibility of the plan for the whole-body motion.

In this paper, we solve whole-body planning using nonlinear optimization techniques and a full-body dynamic model. Optimization techniques were used as early as the 1980s in robotic motion planning. The starting point of our work is the result obtained by Lee et al. (2005); Lee et al. proposed a method to generate motions for robots with redundant or exactly actuated closed chains. This method is based on the optimization of the parameters of the joint trajectories and computes analytically the model of the robot and its derivative. Their algorithm was illustrated with 2D simulation examples and we considered whether it is possible to extend the approach for the 3D complex case of multi-contact motions for humanoid robots. That is, how to impose unilateral constraints on the contact forces, ensure

[1]CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan
[2]Karlsruhe Institute of Technology, Institute for Anthropomatics, Humanoids and Intelligence Systems Lab, Karlsruhe, Germany
[3]CNRS-Université Montpellier 2, LIRMM, Interactive Digital Human group, Montpellier, France

Corresponding author:
Sébastien Lengagne, Karlsruhe Institute of Technology, Institute for Anthropomatics, Humanoids and Intelligence Systems Lab, Adenauerring 2, 76131 Karlsruhe, Germany.
Email: lengagne@gmail.com

balance, include collision avoidance and guarantee constraint satisfaction not only at sample time but also all along the motion whatever time-grid size is chosen? Our present work tackles these issues. The work by Lee et al. (2005) is certainly the root of our approach. However, we propose several novel contributions in order to generate complex and dynamic multi-contact motions for humanoid robots.

In Section 2, we present the related work about motion generation for humanoid robots and emphasize the lack of efficient methods to generate complex and dynamic multi-contact motions. Then, we present the mathematical model of the robot and the analytical formulation of the contact forces in Section 3. Section 4 is about formulating the motion generation problem as a nonlinear optimization. To solve this problem and produce motions with guaranteed constraint satisfaction all over the time, including non-desired collisions avoidance, Section 5 introduces our second contribution: the time-interval discretization. This discretization is based on Taylor series expansion approximation for the continuous state variables of the robot. Finally, in Section 6, we discuss the technical details and validate our method with several complex scenarios such as a sitting motion for the HRP-2 (Kaneko et al., 2004), where both hands, both feet and the waist are involved in contact transitions during motion.

2. Related work

Previous work proposed several approaches in order to generate sequences of contact stances, to compute optimal motions, to characterize its balance and to perform experiments of multi-contact motions.

Recently, there is an increasing interest in planning non-gaited motion where a humanoid robot is allowed to contact the environment with any possible part of its body to support motions in bulky and uneven spaces. For example, Bouyarmane et al. (2009); Bouyarmane and Kheddar (2012), Escande and Kheddar (2009); Escande et al. (2006), and Hauser et al. (2005); Hauser and Latombe (2010) developed a contact-before-motion planning algorithm to plan the sequence of contacts for such problems. Complex examples were illustrated on the HRP-2 robot (Escande and Kheddar, 2009; Escande et al., 2008). However, in these experiments, the motion between successive contacts, namely the transitions, was not fully dynamic. Our present contribution aims at filling this gap. That is, generating optimal dynamic motions in multi-contact transitions.

Optimal control methods are used to generate (periodic) complex motions such as one-legged hopping, walking (Mombaur et al., 2005) and running (Schultz and Mombaur, 2010). They are usually categorized into three different types (Diehl et al., 2006): (i) dynamic programming methods are based on the principle that the optimality of sub-arcs is restricted to small state dimensions; (ii) indirect methods use the Pontryagin's maximum principle, but have a small domain of convergence; (iii) direct methods that transform the infinite problem into a finite nonlinear programming problem (NLP) are commonly used for real systems such as humanoid robots.

The direct methods are composed of the direct single shooting, collocation and direct multiple shooting methods for which the control and/or the state variables are discretized and use the direct dynamic model of the robot. A fourth direct method uses parameterization. It finds a local solution of the optimal control through optimal values of the trajectories' parameters; as an example of parameterization B-splines are widely used, (cf. Lee et al.(2005) and Miossec et al. (2006)). Parameterization proved to be particularly efficient in robotics (Steinbach, 1997; von Stryk, 1998). In this case, an inverse dynamic model can be used that allows the motion to be decomposed into several independent parts, which can be parallelized using multi-thread software. Using state-of-the-art parameterization based methods, continuous time-dependent constraints are still discretized into a finite set of discrete constraints over a time grid. This kind of method was used by Arisumi et al. (2008) to realize the lifting motion of 23.4 kg with the HRP-2 humanoid (Kaneko et al., 2004). Each time an experiment crashed during trials, investigation of log data revealed that the torque constraint was violated in between two points of the chosen time grid, albeit the optimization process ended successfully. In that sense, this method is not *safe*, since the user cannot trust the results of the optimization process. At that time, the optimization problem was reformulated with smaller time grid, recomputed and re-experimented until success.

This observation motivates the work we performed and present here. Our new formulation, guarantees that, under model-match hypothesis, in the case of successful termination of the optimization process, the planned motion will not violate any constraints all along the time motion whatever discretization range is. This is the reason why we call it *safe*. One can use the time-interval discretization proposed by Lengagne et al. (2011c) but at the price of a substantially much longer computation time. We can also proceed in a recursive way by running the optimization using a time grid, and then compute the violation of the constraints using the time interval while penalizing the bounds of the constraints until no violation occurs as presented by Lengagne et al. (2011c). Yet, those methods cannot deal with continuous equality constraints that describe the constant position of a body during a contact stance.

Specific to humanoids robots, balance is a very challenging and a difficult issue. During walking motions, the balance of the robot is usually characterized by the zero moment point (ZMP) (Vukobratović and Borovac, 2004). Unfortunately, the use of the ZMP assumes that there are only co-planar contacts and cannot apply to general multi-contact motions. One can also consider the generalized ZMP (GZMP) (Harada et al., 2006) or the foot rotation index (Goswami, 1999). To characterize the balance of the robot, we rather monitor whether the contact wrench sum

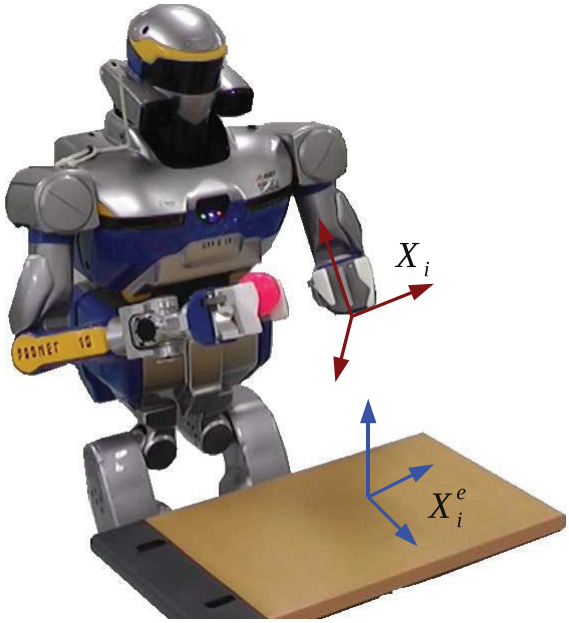


Fig. 1. Description of the contact of the left hand with the table. The frame $\mathbf{X}_i$ is linked to the body and must coincide the expected frame $\mathbf{X}_i^e$ on the table.

(CWS), due to gravity and dynamic effects, remains in the contact wrench cone (CWC) as presented by Hirukawa et al. (2006). From this definition of the balance, we propose an analytical formulation of the contact forces.

The use of reduced models during the optimization can lead to some constraints violation *a posteriori*. We better plan the motion with the most complete model and constraints possible in order to guarantee that whole-body motion is highly likely feasible. The choice of the full-body model impacts on the complexity of the computation, and hence on the computation time; but is the only way to ensure *a priori* a viable plan of the robot.

Collisions with the environment and the self-collisions are also critical constraints to handle during motion planning. However, collision avoidance requires the evaluation of the distance to obstacle. In order to have good performances of the optimization process, it is highly suggested to consider convex bounding shapes to describe the bodies of the robot such as the Sphere-Torus-Patch Bounding Volume (STP-BV) presented by Escande et al. (2007). This solution allows us to have continuous gradient of the distances and we are using it in our problem formulation. To ensure the collision avoidance over the entire motion's duration, some methods such as the continuous collision detection (Zhang et al., 2007) use Taylor models and temporal culling.

We assume a perfect full-body dynamic rigid model of the HRP-2 robot and a perfect knowledge of its environment. However, to perform real experiments, the last issue is subject to the control approach of the robot during multi-contact stances. Work by Hyon et al. (2007) and Stephens and Atkeson (2010) proposed a balance controller based on the control of the forces, whereas the work of Lee and Goswami (2012) was based on the control of both linear and angular momentum of the robot. Interestingly Sentis et al. (2010) presented a methodology to control the internal forces and center of mass (COM) during a multi-contact interaction between a humanoid robot and the environment using a virtual linkage model. This method was validated in simulations of manipulation tasks without locomotion, such as cleaning a board or manipulating a panel. Eventually, a controller used during locomotion of robots is presented by (Righetti et al., 2011); it uses the redundancy of the multi-contact motion to create optimal distribution of the contact force and was successfully applied to a quadruped robot during actual experiments. In our work, we assume that the robot's controller tracks the generated joint trajectories perfectly under little model uncertainties.

3. Multi-contact motion definition

In this section, we present the model and the constraints that are needed to generate a multi-contact motion.

3.1. Kinematics

First, we deal with the kinematic constraints. A contact is defined simply as the relative configuration of a given robot's link regarding the environment, excluding sliding. This can be defined with equality constraints involving two Cartesian frames: one on the contact body $\mathbf{X}_i$ and the other one on the environment at the expected desired position of the contact $\mathbf{X}_i^e$ (Figure 1), where $\mathbf{X} \in \mathbb{R}^6$ denotes the six-component contact vector of a frame position and orientation ($\mathbf{X} = [x, y, z, \theta_x, \theta_y, \theta_z]^{\mathrm{T}}$). In order to consider N_c contacts during a time interval $[\Delta t]$, we must ensure a set of n_i continuous equality constraint for each contacts as follows:

$$\begin{aligned} &\forall t \in [\Delta t], \forall i \in \{1, \ldots, N_c\}, \forall j \in \{1, \ldots, n_i\} \\ &\quad \mathrm{kin}_{i,j}(\mathbf{X}_i(t), \mathbf{X}_i^e) = 0 \end{aligned} \tag{3.1}$$

where n_i is the number of constraints for each type of contact. For a planar rigid contact there are three rotations and three positions constraints ($n_i = 6$). The general formulation of Equation 3.1 allows to consider planar, edge or point contacts. Moreover, we consider fully and partially defined contacts: the value of some components of the contact position and/or orientation can be unknown (but must stay constant). For example, for a planar rigid contact, one considers

$$\begin{aligned} &\forall t \in [\Delta t], \forall i \in \{1, \ldots, N_c\}, \forall j \in \{1, \ldots, 6\} \\ &\quad \begin{cases} \text{if } \kappa_{i,j} = 1 \rightarrow \mathbf{X}_{i,j}(t) = \mathbf{X}_{i,j}^e \\ \text{if } \kappa_{i,j} = 0 \rightarrow \frac{d}{dt}\mathbf{X}_{i,j}(t) = 0 \end{cases} \end{aligned} \tag{3.2}$$

where $\kappa_{i,j}$ is a Boolean attached to the jth component of the ith contact vector. This Boolean is equal to one if the component value is known and is equal to zero if the component value is unknown, in the case of partially defined

contacts. In the latter case, the actual value will be a result of the optimization process. Equations (3.1) will impose the kinematic behavior of the motion. The balance of the motion will depend on the dynamic of the motion as defined hereafter.

3.2. Inverse dynamic model

We plan motion with the full-body dynamic model of the humanoid robot:

$$\begin{bmatrix} \Gamma \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1(q) \\ \mathbf{M}_2(q) \end{bmatrix} \ddot{q} + \begin{bmatrix} \mathbb{H}_1(q,\dot{q}) \\ \mathbb{H}_2(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} \mathbf{J}_1^{\mathrm{T}}(q) \\ \mathbf{J}_2^{\mathrm{T}}(q) \end{bmatrix} \mathbb{F} \tag{3.3}$$

where $q(t) \in \mathbb{R}^{N_{dof}}$ is a vector containing the N_{dof} joint position $(q_i(t), \Gamma(t)), \in \mathbb{R}^{N_{dof}}$ is the vector of the joint torques, $\mathbb{F}(t)$ is the vector of the N_f active linear contact forces, $\mathbf{M}_1 \in \mathbb{R}^{N_{dof} \times N_{dof}}$ and $\mathbf{M}_2 \in \mathbb{R}^{6 \times N_{dof}}$ are the two components of the inertial matrix, $\mathbb{H}_1 \in \mathbb{R}^{N_{dof}}$ and $\mathbb{H}_2 \in \mathbb{R}^6$ the two components of vector due to gravity, centrifugal and Coriolis effects, $\mathbf{J}_1 \in \mathbb{R}^{N_{dof} \times 3N_f}$ and $\mathbf{J}_2 \in \mathbb{R}^{6 \times 3N_f}$ are the components of the Jacobian matrix. Equation (3.3) assumes that the first contact (from which the waist trajectory is computed, see Lengagne et al. (2010b)) is always maintained. This assumption is enforced by adding constraints on the balance of the robot as presented in Section 3.6. Equation (3.3) emphasizes the link between the contact forces and the joint torques. To know the joint torques, it is necessary to know the joint trajectories and the contact forces. Regarding the known joint trajectories, there is an infinite set of combination joint torques–contact forces, depending on the internal forces. In the following, we present our method to get rid of this indetermination and compute the joint torques and contact forces from the joint trajectories and a few additional variables, in a way that enforces the balance of the robot.

3.3. Balance

To characterize the balance of the motion, we constrain the CWS, due to gravity and dynamic effects, to remain within the CWC as presented by Hirukawa et al. (2006). In fact, we ensure that the contact forces that counterparts the dynamic effect $\mathbb{D}_2 = \mathbf{M}_2(q)\,\ddot{q} + \mathbb{H}_2(q,\dot{q})$ must be feasible, i.e. hold a desired contact state, prohibiting from any undesired sliding of the link on the surface or contact removal.

We define a set of N_f linear forces $\mathbb{F} = \{F_1, F_2, \ldots, F_{N_f}\}$ ($F_i \in \mathbb{R}^3$), for all of the contacting bodies, as presented in Figure 2 and their normal and tangent spaces. To avoid unexpected sliding or taking-off of any part of the robot, each linear contact force F_i must fulfill

$$\forall t \in [\Delta t], \forall i \in \{1, \ldots, N_f\} \quad \begin{cases} F_i^n(t) > 0 \\ ||F_i^t(t)||^2 \leq \mu_i^2 F_i^n(t)^2 \end{cases} \tag{3.4}$$

where F_i^n is the normal component of the contact force, F_i^t is the 2D vector of the tangent force and μ_i is the friction coefficient at contact i. For each contact forces i, those two constraints can be merge into the equivalent equation:

$$\begin{aligned} \mathrm{fric}(F_i(t)) = {} & F_{i,x}(t)^2 + F_{i,y}(t)^2 - \mathrm{sign}(F_{i,z}) \\ & \times \mu_i^2 F_{i,z}(t)^2 \leq 0 \end{aligned} \tag{3.5}$$

Note that we consider the z-axis as the normal axis at the contact space. Moreover, those contact forces must be counterparts to the dynamic effect by satisfying the second part of the dynamic Equation (3.3), that is

$$\mathbb{D}_2(q,\dot{q},\ddot{q}) + \mathbf{J}_2^{\mathrm{T}}(q)\,\mathbb{F} = 0 \tag{3.6}$$

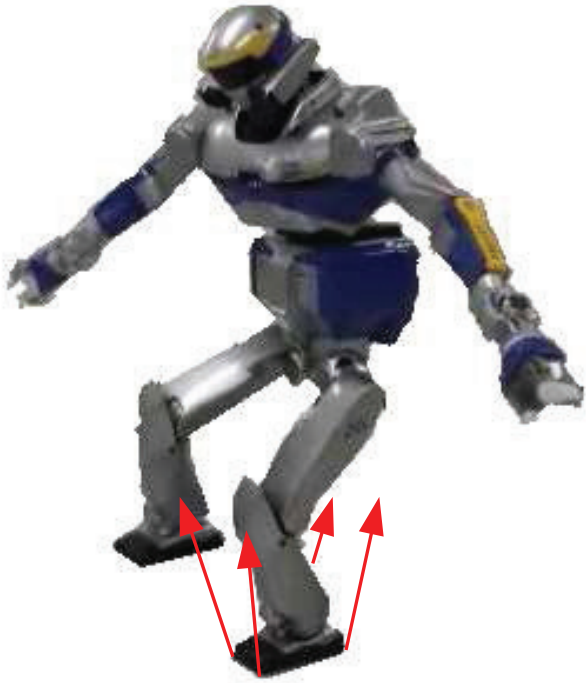


Fig. 2. Description of the contact forces for the left foot. We consider four forces: one for each corner of the foot sole.

3.4. Contact forces

In our previous work (Lengagne et al., 2010b), a 2D-model study of this problem revealed that Equation (3.6) cannot be satisfied if the contact forces are parameterized using B-spline functions. Hence, we compute the value of the contact forces $\mathbb{F} = \{F_1, F_2, \ldots, F_{N_f}\}$ from the joint trajectories that exactly counterpart the dynamics effects while fulfilling the constraint of Equation (3.4) as best as possible, i.e. the contact forces that solve

$$\begin{aligned} & \min \tfrac{1}{2} \textstyle\sum_i \beta_i (\alpha_i ||F_i^t||^2 + F_i^{n2}) \\ & \mathbb{D}_2(q,\dot{q},\ddot{q}) + \mathbf{J}_2^{\mathrm{T}}(q)\,\mathbb{F} = 0 \end{aligned} \tag{3.7}$$

where F_i^n is the normal component of the contact force F_i; F_i^t the 2D vector component of the tangent component of the contact force F_i; β_i and α_i are weight coefficients as detailed in Appendix A. The exact satisfaction of these constraints is left to the optimization solver. Appendix A shows

how to obtain the analytical formulation of the contact forces such as

$$F_i = \mathbb{W}_i^{-1}\left[\hat{P}_i A_i \; A_i\right]\Omega^{-1}\mathbb{D}_2 \tag{3.8}$$

with

$$\Omega = \sum_i \left(\begin{bmatrix} \hat{P}_i A_i \\ A_i \end{bmatrix} \mathbb{W}_i^{-1} \begin{bmatrix} \hat{P}_i A_i & A_i \end{bmatrix}\right) \tag{3.9}$$

and where:

- $\mathbb{W}_i = \text{diag}(\beta_i\alpha_i, \beta_i\alpha_i, \beta_i)$ is a weight matrix;
- $\beta_i > 0$ weights the contact forces i regarding the other forces;
- $\alpha_i \gg 1$ weights the tangential components regarding the tangential ones;
- P_i, A_i are the position and orientation of the contact forces.

The matrix Ω is a 6×6 matrix that we invert using the Gauss–Jordan algorithm, to compute the value of the contact forces $\mathbb{F}$.

3.5. Collision avoidance

During multi-contact motion transitions, there are parts of the robot for which we seek a sustained collision with the environment (desired motion supporting contacts), and the remaining part that should avoid (non-desired) collisions. Therefore, planning motion considers collision avoidance as a constraint. Let $\mathcal{L}_i$ be the ith link of the robot, $i \in [1, N_l]$ for a robot having N_l identified links or all of the sub-part that are convex composing all links, and let $\mathcal{O}_i$ be a set obstacles among N_o ones. Avoiding collision can be simply written as

$$d(\mathcal{L}_i, \mathcal{O}_j)(t) \geq \epsilon \tag{3.10}$$

where $d(X, Y)(t)$ is the distance separating body X from body Y, the pair i,j is the list of the defined links/obstacles to be checked and ϵ is a security margin. This constraint means that the separating distance between two any body must be kept positive (at ϵ margin) all along the motion. Self-collisions are enforced similarly:

$$d(\mathcal{L}_i, \mathcal{L}_j)(t) \geq \epsilon \quad i \neq j \text{ and } j \neq i-1 \tag{3.11}$$

There are several methods to compute the distance among pairs of geometrical objects. The simple the geometry the faster is the computation. This is the reason why in robotics we use bounding volume for the links (e.g. capsules, oriented bounding boxes, etc.). Since we are using the distance in the frame of an optimization problem, many solvers require computing the gradients of the constraints. In order to fulfill existence and continuity of the distance function gradients, while keeping fast computation, we need to have strictly convex bounding volume of the links. We devised in our previous papers such new bounding volume operator, called STP-BV that achieved bulged convex hulls so as the distance computation is fast and its gradient C^1 continuous (Escande et al., 2007; Benallegue et al., 2009). Note that computing the distance δ is the outcome of a local optimization process.

3.6. Constraints

We define a multi-contact motion by a sequence of multi-contact phases. One phase is an amount of time for which all the contacts hold (no creation or release of contacts). The transition between two phases is done when at least one contact is created or released.

The generation of the multi-contact motions is equivalent to finding the joint trajectories, the contact forces and the duration over each contact phase. This motion must validate a set of equality (Equation (3.12a)) and inequality (Equations (3.12b)–(3.12f)) continuous constraints:

$$\forall t \in [\Delta t], \forall i \in \{1, \ldots, N_c\}, \forall j \in \{1, \ldots, n_i\} \quad \text{kin}_{i,j}(\mathbf{X}_i(t), \mathbf{X}_i^e) = 0 \tag{3.12a}$$

$$\forall t \in [0, T], \forall i \in \{1, \ldots, N_{dof}\} \quad \underline{q_j} \leq q_j(t) \leq \overline{q}_j \tag{3.12b}$$

$$\forall t \in [0, T], \forall i \in \{1, \ldots, N_{dof}\} \quad \underline{\dot{q}_j} \leq \dot{q}_j(t) \leq \overline{\dot{q}_j} \tag{3.12c}$$

$$\forall t \in [0, T], \forall i \in \{1, \ldots, N_{dof}\} \quad \underline{\Gamma}_j \leq \Gamma_j(t) \leq \overline{\Gamma}_j \tag{3.12d}$$

$$\forall t \in [0, T], \forall i \in \{1, \ldots, N_f\} \quad \text{fric}(F_i(t)) \leq 0 \tag{3.12e}$$

$$\forall t \in [0, T], \forall i \in \{1, \cdots, N_d\} \quad \delta_i \geq \epsilon \tag{3.12f}$$

where $\delta_i = \text{sign}(d_i)\, d_i^2$ is the signed square distance between two bodies, this value is returned by the algorithm presented by Benallegue et al. (2009). One can also consider a set of k discrete constraints on any state x_k of the robot in order to specify desired property of the motion.

$$\forall k, \; z_k(x_k, t_k) \leq 0 \tag{3.13}$$

4. Problem formulation

In the previous section, we detailed the model and the constraints of a multi-contact motion. Now, we introduce the basics in terms of terminology, notation, and the general formulation of the nonlinear optimization problem.

4.1. Motion planning problem

The motion generation process solves the following optimization problem:

$$\begin{aligned} \min_{q(t), \Gamma(t), \mathbb{F}(t), \mathbf{T}_f} \; & C(q(t), \dot{q}(t), \ddot{q}(t), \Gamma(t), \mathbb{F}, \mathbf{T}_f) \\ \text{subject to} \\ & \{\Gamma, \mathbb{F}\} = \text{IDM}(q, \dot{q}, \ddot{q}) \\ & c_{eq}(q, \dot{q}, \ddot{q}, \Gamma, \mathbb{F}) = 0 \\ & c_{ineq}(q, \dot{q}, \ddot{q}, \Gamma, \mathbb{F}) \leq 0 \\ & c_{teq}(q(t_d), \dot{q}(t_d), \ddot{q}(t_d), \Gamma(t_d), \mathbb{F}(t_d)) \leq 0 \end{aligned} \tag{4.1}$$

This formalism is used by Miossec et al. (2006). Usually, the constraint $\{\Gamma, \mathbb{F}\} = \text{IDM}(q, \dot{q}, \ddot{q})$ is replaced by the direct dynamic model $\dot{x}(t) = f(x(t), u(t))$ where $x(t)$ is the state variable and $u(t)$ the control variables (Schultz and Mombaur, 2010), but both representations are mathematically equivalent and differ only from the implementation viewpoint. Thus, the optimal control problem (4.1) can be considered as an infinite programming (IP) problem that consists of finding the joint trajectories $q(t)$, the force coefficient β and the phase durations $\mathbf{T}_f$:

$$\begin{array}{ll} \underset{q(t),\beta,T_f}{\text{argmin}} & C(q(t), \beta, \mathbf{T}_f) \\ \forall i, \forall t \in [\Delta_i] & g_i(q(t), \beta, \mathbf{T}_f) \leq 0 \\ \forall j, \forall t \in [\Delta_j & h_j(q(t), \beta, \mathbf{T}_f) = 0 \\ \forall t_k \in \{t_1, t_2, \ldots\} & z_k(q(t_k), \beta, \mathbf{T}_f) \leq 0 \end{array} \tag{4.2}$$

where C is the cost function, which accounts for quantitative and/or qualitative robot motion in given application contexts. Well-known cost functions in robotics are: motion duration (Piazzi and Visioli, 1998; Bobrow, 1988), minimum torque (Breteler et al., 2001), global energy consumption (Miossec et al., 2006), jerk for smooth motion (Piazzi and Visioli, 2000), torque change (Uno et al., 1989), or any weighted combination of the above and more. In the equation, g, h and z are, respectively, the continuous inequality and equality and the discrete inequality constraints as presented in Section 3.6.

We voluntary omit the discrete constraints z, since they do not lead to any kind of problem in the state-of-the-art methods.

4.2. Semi-infinite programming

The problem (4.2) is an IP since the solution to be found is a set of continuous functions $q(t)$ that fulfill a set of continuous constraints all along the time t (both can be seen as infinite sets of discrete values). In order to make an IP computationally solvable, most of the methods (Reemtsen and Rückmann, 1998) reduce such complexity by defining a parameters set $\mathbf{P} \in \mathbb{R}^N$ that gives at least a parametric shape for the functions to be found (the optimal trajectories $q(t)$):

$$q(t) = f(\mathbf{P}, t) \tag{4.3}$$

Based on prior work in robotics (e.g. Lee et al., 2005), we chose to shape the joint trajectories with clamped uniform cubic B-spline curves. A B-spline function is the weighted sum of k-order basis functions defined by m number of control points:

$$\forall t \in [\Delta t] \qquad q(t) = \sum_{i=1}^{m} b_i^k(t) p_i \tag{4.4}$$

The basis functions $b_i^k(t)$ are computed using the Cox–de Boor recursion (de Boor, 1978) and the control points ρ_i are part of the parameters of the motion. This parameterization was already used for motion generation by Lee et al. (2005) and Miossec et al. (2006). Subsequently, the motion planning problem turns to finding the best parameter set $\mathbf{P} \in \mathbb{R}^N$ such as

$$\begin{array}{ll} \underset{\mathbf{P}}{\text{argmin}} & C(\mathbf{P}) \\ \forall i, \forall t \in [\Delta_i] & g_i(\mathbf{P}, t) \leq 0 \\ \forall j, \forall t \in [\Delta_j] & h_j(\mathbf{P}, t) = 0 \end{array} \tag{4.5}$$

This problem is a semi-infinite programming (SIP) (Reemtsen and Rückmann, 1998) since it deals with a finite set of optimization variables $\mathbf{P} \in \mathbb{R}^N$, which must satisfy a set of continuous constraints. The detail of the optimization parameter vector $\mathbf{P}$ in our case in presented in Section 6.1.

5. Dealing with continuous constraints

Enforcing joint position and velocity limits is straightforward, see Appendix B. For more complex constraints such as torques, we use a time-interval discretization based on a polynomial approximation.

5.1. Time-interval discretization

Most of the solvers cannot deal with continuous constraints and require a finite number of constraints. Although the time-grid discretization is widely used (Hettich and Kortanek, 1993; Reemtsen and Rückmann, 1998; von Stryk, 1993), it assumes (but does not guarantee) that if a constraint holds at times t_k and t_{k+1}, then it holds $\forall t \in [t_k, t_{k+1}]$ (Lengagne et al., 2011c). We experienced in our previous attempts (Arisumi et al., 2008; Lengagne et al., 2011c) and highlight that violations of the limits may happen, even when the optimization algorithm ended with success status. To solve this issue and generate motions in a safe way, the time-interval discretization (Lengagne et al., 2011c) replaces the continuous inequality constraints of the optimization problem by

$$\forall i, \forall [t] \in \mathbb{IT} \qquad \sup([g]_i(\mathbf{P}, [t])) \leq 0 \tag{5.1}$$

where $\mathbb{IT} = \{[t]_1, [t]_2, \cdots, [t]_{q-1}, [t]_q\}$ and $[t]_n = [t_{n-1}, t_n]$ is the time interval decomposition of the phase Δt. In our previous work (Lengagne et al., 2011c), we computed a conservative value of the extrema over each time interval $[t]_i$, and the optimization solver can produce a motion that guarantees all of the constraint validity over the entire motion duration. Unfortunately, this method has a huge computation time. In this paper, we consider the same time-interval discretization, but we rather use Taylor approximation, which reduces the computation time substantially and allows us to deal with continuous inequality and equality constraints.

5.2. Taylor approximation

5.2.1. Principle In order to perform a time-interval discretization (Equation (5.1)) that considers continuous inequality and equality constraints, we use a polynomial

Taylor approximation of the constraints. Taylor expansion was also used by Zhang et al. (2007) to tie the bounds of the bounding box hierarchies for collision detection.

The original computations (Lengagne et al., 2011c), based on interval analysis, do not keep track of the correlation between the different sub-functions which leads to an overestimation of the bounds (Zhang et al., 2007). To overcome this drawback, we define each function over a time interval $\Delta t = [t_s, t_e]$ as a n-order polynomial function:

$$\forall t \in [\Delta t] \qquad f(t) = \sum_{i=0}^{n} a_i(\mathbf{P}) \times t^i \tag{5.2}$$

where n is the order of the approximation, $\{a_0(\mathbf{P}), a_1(\mathbf{P}), \ldots, a_n(\mathbf{P})\} \in \mathbb{R}^{n+1}$ are the coefficient of the polynomial depending on the optimization parameters $\mathbf{P}$. Compared with previous work (Berz et al., 1998), we voluntary omit to compute the remaining error interval $[\epsilon]$, assuming that we choose the order n of the polynomial approximation so that we can neglect it (Lengagne et al., 2010a).

5.2.2. Inequality Since we get a polynomial function of the constraints (cf. Equation (5.2)), we want to know the extrema of this polynomial over the considered among of time $[\Delta t]$. Some work approximate the extrema of a polynomial based on the Inequality of the Means (de Alwis, 2004), but cannot find it for a given interval. Several methods find the extrema based on polynomial root finding of the derivatives of the original polynomial. Analytical approaches fail for polynomial having an order $n > 5$. In order to evaluate this extrema for any order n, we use the property of the B-splines explained in Appendix B. That is, given a polynomial, the function $f(t)$, and hence its extrema, are contained within the convex hull of the equivalent control points ρ_i. Therefore, we obtain

$$\forall t \in [\Delta_t], \underline{f} \leq f(t) \leq \overline{f} \Longrightarrow \forall i, \underline{f} \leq \rho_i \leq \overline{f} \tag{5.3}$$

with

$$[\rho_0, \rho_1, \ldots, \rho_n] = [a_0, a_1, \ldots, a_n] \times \mathbf{B}^{-1} \tag{5.4}$$

and where $\mathbf{B} \in \mathbb{R}^{(n+1)\times(n+1)}$ is a square matrix of the polynomial parameters of the n-order B-spline basis functions[1] $b_i^n(t)$ and relies on the time interval $[\Delta t]$. Using a time-scaled implementation of the polynomial, we can compute the inverse $\mathbf{B}^{-1}$ only once and use it all along the optimization process. As for the polynomial coefficient, the equivalent B-spline control point ρ_i rely on the optimization parameter: $\rho_i(\mathbf{P})$. Eventually, constraining the equivalent control point ρ_i to remain within two bounds enforces the continuous function to stay within those bounds.

5.2.3. Equality As shown in Section 3.6, multi-contact motions involve equality constraints that must hold all along the motion, or at some time intervals in order to define fully or partially the contacts. These constraints are the $h_j(\mathbf{P}, t)$ in Equation (4.5). Using Taylor approximations of h_j, we end up with the polynomial of Equation (5.2) and impose the coefficients $a_i(\mathbf{P})$ to fulfill the constraint

$$\forall i \in \{0, \ldots, N_e\} \quad a_i(\mathbf{P}) = 0 \tag{5.5}$$

The continuous equality constraint being of order N_e, $N_e \leq n$ is the order that we set for the solver to avoid an over-constrained formulation. As a result, the equality constraint is fulfilled only under small variation tolerance. The compromise between the precision and the optimization convergence rate is tuned heuristically with the choice of $N_e = 2$. Actually, this highlights a deeper fundamental problem linked to the choice of the joint trajectory parameterization for closed kinematics chains. There is certainly a venue for a deeper study specific to this issue.

5.3. Evaluation of collision avoidance constraints

We need to compute the minimum of the distance constraint for a range of time intervals (e.g. for $t \in [t_k, t_{k+1}]$). In contrast to the other constraints, the distance function does not have a closed-form formula nor an approximate one that allows it to be written as a polynomial in terms of the optimization variables. We therefore need to approach the evaluation of $\min_{t\in[t_k,t_{k+1}]} \delta(t)$ with a numerical method.

In our recent work (Lee et al., 2012), we devised a fast method, which evaluates the min distance along a given time interval. It combines the well-known Golden search method with the conservative advancement technique used in computer graphics simulation to find the first time of collision (see more details in Lee et al. (2012)). Yet, in Lee et al. (2012), we used capsules as bounding volume: not only they make fast computation of the distance, but the capsule shape formula allows fast computation of the conservative advancement bounds. However, the capsule is not a strictly convex shape. In this work, since we use the STP-BV bounding volume, we applied only the Golden search method to seek for the min value of the distance function on a given time interval with *a posteriori* check.

6. Experimental validation

6.1. Description of the motion

We validated our method by generating a sitting motion for the humanoid robot HRP-2 (Kaneko et al., 2004). This motion alternates contacts with both hands and feet to finally contact the waist of the robot with the chair, see Figure 4. The contact of the feet and of the waist are fully defined by their position and orientation on the ground, whereas the contact of the hand are partially defined, i.e. the hands are asked to contact the table whatever the exact position.

This motion was decomposed into 13 phases that start from the half-sitting motion and end when the robot has

both feet contacting the floor, the left hand contacting the table and the waist contacting the chair. Each phase is decomposed into six time intervals and involves nine control points per joints as illustrated on Figure 3. We ensure joint trajectory continuity by computing the three first control points of the next phase from the three last ones of the current phases. Hence, the Cartesian trajectories are continuous having smooth transitions between the contact phases.[2]

For one contacting body over every contact phase, we consider the same linear evolution of the coefficient β for all of its contact forces. We do not impose the continuity of the coefficient β over the whole motion duration. We set $\forall i, \alpha_i = 1000$, this allows us to avoid redundant and unnecessary optimization parameters. Eventually, the parameter vector $\mathbf{P} \in \mathbb{R}^n$ contains all of the control points, the linear coefficient of β and the phase durations.

During the sitting motion, the degrees of freedom of the head and of the wrists are set to a constant value, since they do not impact significantly the balance of the robot. Owing to the fragility of the forces sensor at the endpoints of the arms, we choose to contact the table with the wrist of the robot. We consider seventy different collisions (36 between the robot and its environment and 34 self-collisions). We also consider the constraints of the contact forces, the velocity and torque limits (the joint position limits are set by limits on the variables).

We are considering the following cost function C as the weighted sum of joint torques, joint jerks and motion duration, as presented:

$$C(q) = a \int_0^{\mathrm{T}} \sum_i \Gamma_i^2 dt + b \int_0^{\mathrm{T}} \sum_i \dddot{q}_i{}^2 dt + cT \qquad (6.1)$$

where $a = 1e-2$, $b = 1e-5$ and $c = 4$ are the value we set heuristically to have human-like walking motions with the HRP-2 (Lengagne et al., 2011a). We choose the order $n = 5$ for the polynomial approximation of the constraints. Eventually, the optimization gets 2,339 variables (1,883 free plus 556 constant variables) and 33,852 constraints.

6.2. Technical implementation

Our method was programmed in the C++ language and executed on the following hardware and software: CPU Intel(R) Xeon(R) E5620, 8 cores, 2.4 GHz, Cache 12 Mo: Linux Gentoo 3.0.6 64 bits.

We do not use the Lagrangian formalism of the IDM of Equation (3.3), but rather use a recursive formulation as in Lee et al. (2005); Park et al. (1995), which is fast and requires fewer operations (Khalil and Dombre, 2002).

Our software uses template classes for the model and specialized types to compute the derivative (Bendtsen and Stauning, 1996) and the Taylor form of the constraints and cost functions. The model computation needs a lot of creation and destruction of those types, which induces many memory allocations. To lower the memory allocations, we use the TC-Malloc library that gives better malloc performance than the glibc library.

Regarding the optimization solver, we believe[3] that it is better to use off-the-shelf available solvers since the optimization theory provides the fundamentals, but optimization is also a matter of tricks and recipes of tuning and numerical robustness. We also apply this policy by using the IPOPT solver (Wächter and Biegler, 2006). IPOPT is free; it handles large nonlinear optimization problems; it has a C++ interface, and was used in (Miossec et al., 2006; Lengagne et al., 2011c).

6.3. Experimental validation

The sequence of the contacts is the output of the method presented by Bouyarmane and Kheddar (2012), and is given as an input to our motion generation method. However, we formulate our problem to allow adjusting these contacts. Not all collisions and self-collisions are relevant to check and only pertinent ones are accounted for a given multi-contact motion problem. We performed several optimizations, in order to define the appropriate collisions and self-collisions that deserve to be considered. Often, first solutions are not achieved by the robot because they are too fast or result in high impacts or simply too risky to try. In these cases we add artificial constraints to drive the solution to our desired behavior or tune the weights of the cost function. The final collision-free optimization process spent 610 iterations and 3h 26mn 37s of computation time to generate the optimal motion on the entire sequence of contacts, i.e. the entire sitting motion starting from the initial posture to the final one.

Any experimental validations of this motion is successfully performed by the HRP-2 robot without exceeding its physical limits, without any unexpected collision or self-collisions and keeping its balance, see Figure 4 and Extension 1. The records of the time history of the feet contact forces and joint knee torques are given in Figures 5 and 6. Moreover, this experiment proved to be repeatable at will since it is often reproduced during VIP visits to our lab.

The cost function and the constraints are model-based. However, the ankle of the robot is equipped with a shock absorbing mechanism, which is equivalent to having a non-controllable flexible joint. This non-modeled flexible part produces small hopping of the foot at the beginning of the contacting phase. We think that this flexibility is also the cause of the release of the contact between the left hand and the table at the end of the motion (cf. Extension 1). Although we use only local joint PD control, without attitude or torque feedbacks, the motion was achieved with good balance within the physical limits of the robot. Yet, because of the existing discrepancies (e.g. we do not include flexibilities in the dynamic model, we might have different friction coefficients at contact, presence of small unexpected perturbations, etc.) joint torques differ from

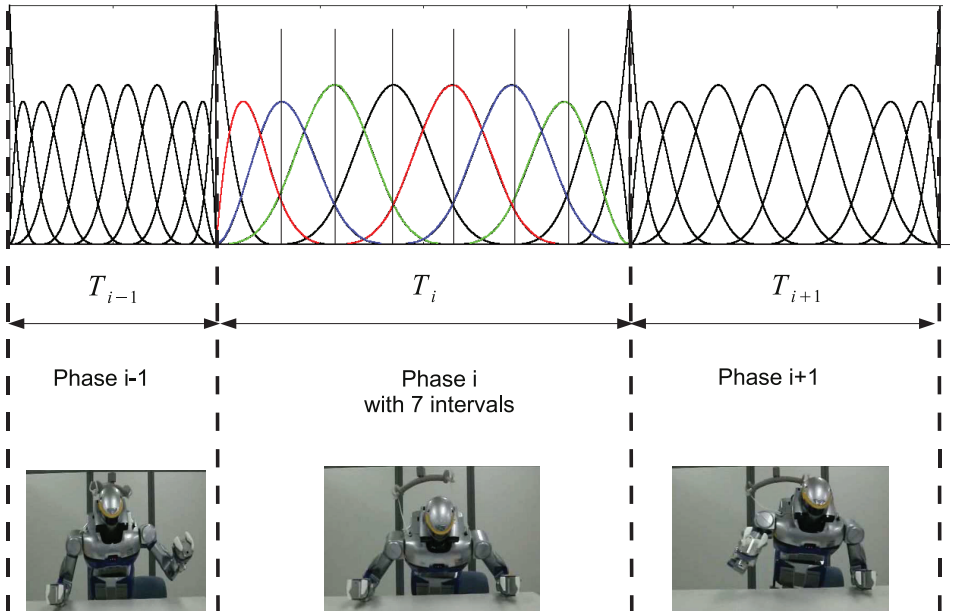


Fig. 3. Illustration of the decomposition of the motion into several phases and time intervals. At each time instant the joint trajectory relies only on four parameters.

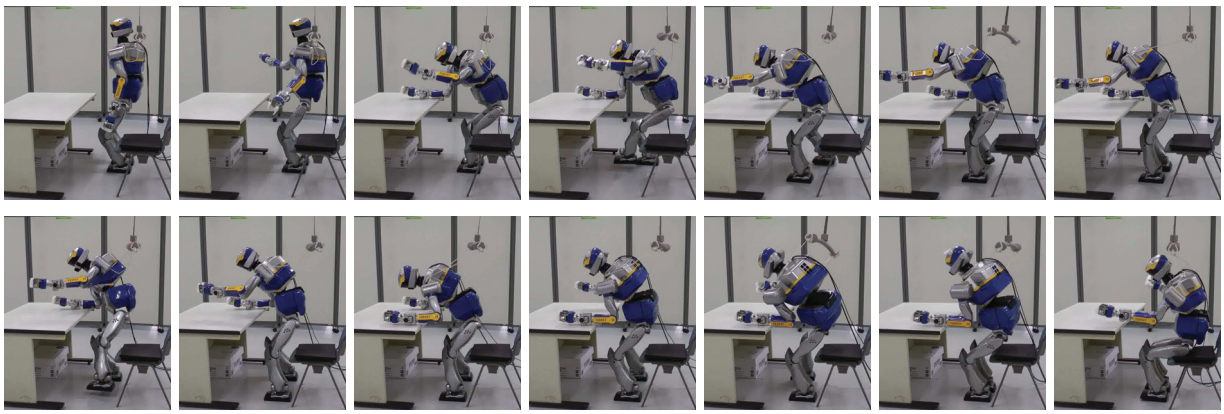


Fig. 4. Experimental validation of the sitting motion.

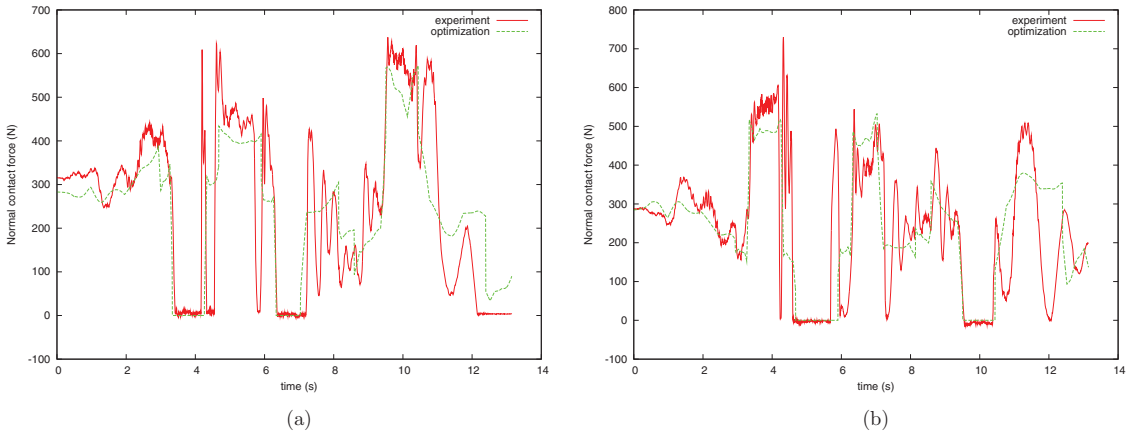


Fig. 5. Experimental validation of the sitting motion, plots of the feet contact forces: (a) left foot; (b) right foot.

the model-computed ones in the planning process; see, for example, the computed and actual torques of the knees in Figure 6. Despite this, contact forces shown in Figure 5 are quite similar to the computed ones, and this makes

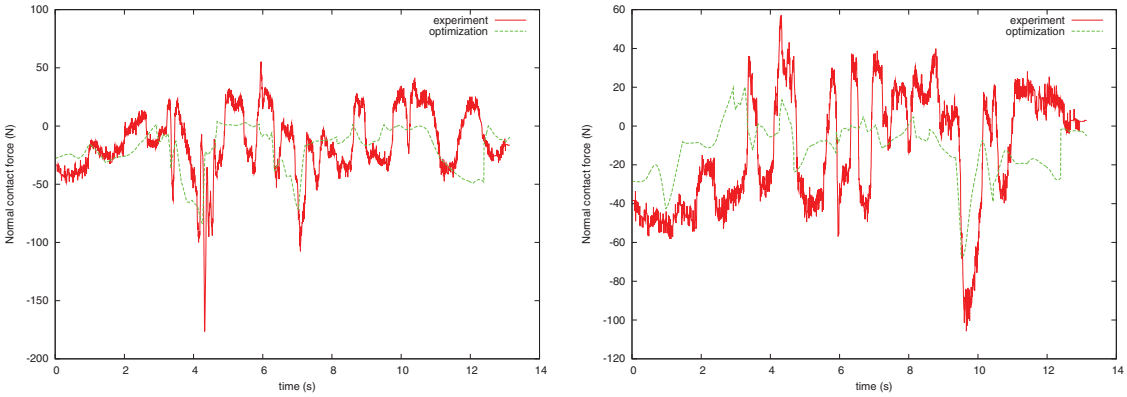


Fig. 6. Experimental validation of the sitting motion, plots of the knee-joint torques: (a) left foot; (b) right foot.

it possible to identify the contact and non-contact phases. Indeed, as far as the model used in the optimization does not differ much, induced errors can still be absorbed by the PD tracking controller. However, for considerable perturbations, discrepancies or uncertainties, re-planning is unavoidable.

In general, it does not make sense to include an error-tracking controller as part of the model-base optimization process. However, we need to distinguish between the low-level controller (that is based only on error tracking, which is assumed null in simulation, since no perturbation) and the controller strategy that may use robot states or contact forces, etc. A control strategy must be part of the model whereas the error tracking controller not.

In Figure 7, we present the evaluation of the constraint for the collision and self-collisions avoidance. In Figure 7(a), we see that the hips never contact the chair before the final contact stance at $t = 12$ s, the contact stance when both hands contact the table can easily be seen in Figure 7(b). Figures 7(c) and 7(d) represent two challenging self-collisions. It appears that self-collisions are very close to occur but will not (Note that $\delta(t)$ represents the signed square of the distance, hence, a distance of 1 mm will result in $\delta = 1 \times 10^{-6}$ on the plots.)

6.4. *Additional experiments*

The chair-sitting scenario is chosen for its complexity and also because we have a ground-truth on running it with different control strategies (see, e.g., Escande et al., 2008). We however assess our multi-contact motion generator with several other experiments involving the HRP-2 humanoid robot. In Lengagne et al. (2010a), we performed a kicking motion while making one hand stay still at a constant position, see snapshots Figure 8. We also generate a kicking motion with the right arm gripper uses a table to support the motion. The contact between the gripper and the table is put at the same position as the previous experiment in Figure 9. These experiments reveal also the difference in posture and force distribution when the humanoid's hand uses an additional contact support for the kicking.

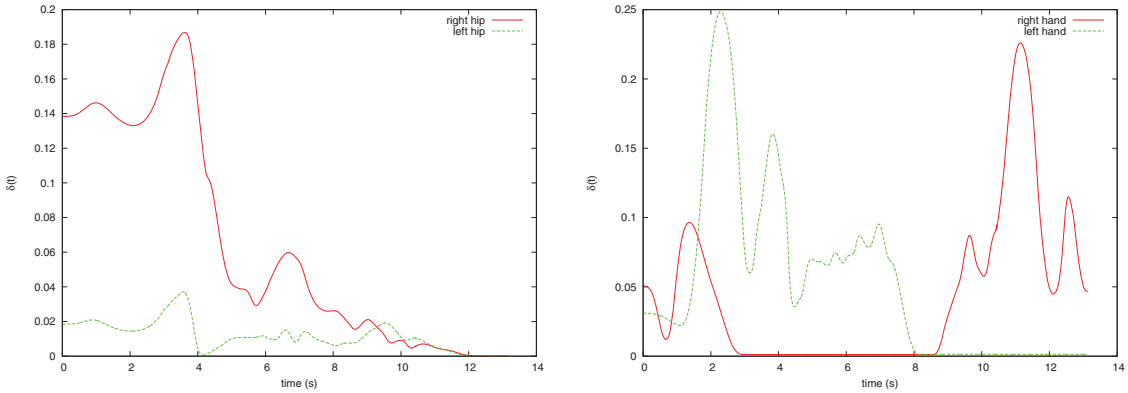
Using our method, we are certainly able to produce regular walking motions. Since there are some non-modeled flexibilities we consider additional constraints to account for the dynamic effects during motion (Lengagne et al., 2011b). Hence, we are able to generate normal walking (cf. Figure 11) and also a walking motion stepping on a 15 cm platform, see Figure 12 and (Lengagne et al., 2011b).

We are aware that more efficient state-of-the-art gaited walking pattern planners/controllers are able to generate this kind of walking motions. We believe that our method can be used to design whole-body trajectories for COM-COP reduced-model-based walking pattern generators. But, more importantly, our method is able to consider (without any change in the software framework) some additional constraints in order to reproduce leg impairments (Lengagne et al., 2011a) by setting a constant knee position to emulate a humanoid knee sprained and wearing a splint (Figure 13), or by setting maximal contact forces on one foot to emulate a broken or painful foot as illustrated by Figure 14.

We can also consider contact around the edge. We implemented a walking motion that reproduce the rotation around the heels and toes as presented in Figure 15. Eventually, we performed a throwing motion with the HRP-2 robot. Figure 16 shows the execution of the motion for which the foot steps were determined heuristically in order to mimic a baseball player throwing.

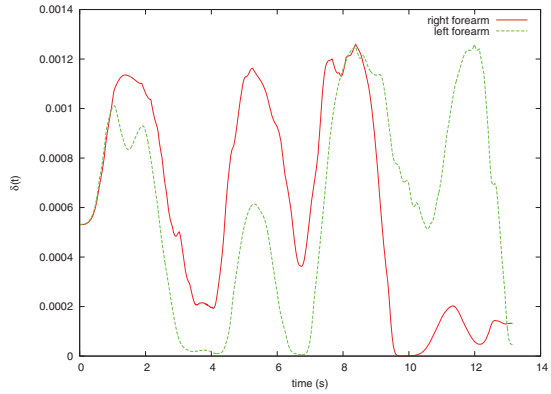
Note that as all of these experiments are made open-loop (i.e. without attitude of force feedbacks), we use only the local joint position PD controller when at least one hand is contacting the environment during the motion (Figures 9 and 10). Otherwise, for the walking motions we use the HRP-2 embedded stabilizer presented in Kajita et al. (2005) in order to deal with the non-modeled flexibilities.

We also used our method to generate motions for the HOAP-3 humanoid robot and for a simulated model of the human. We built a motion database for the HOAP-3 robot

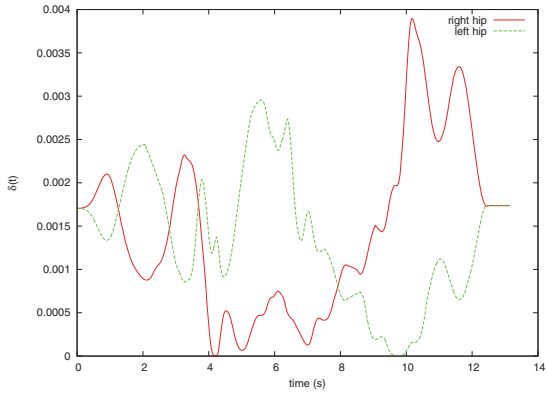


(a) chair–hips

(b) table–hands



(c) chest–forearms



(d) waist–hips

Fig. 7. Plots of the constraint δ for several collisions and self-collisions avoidance.

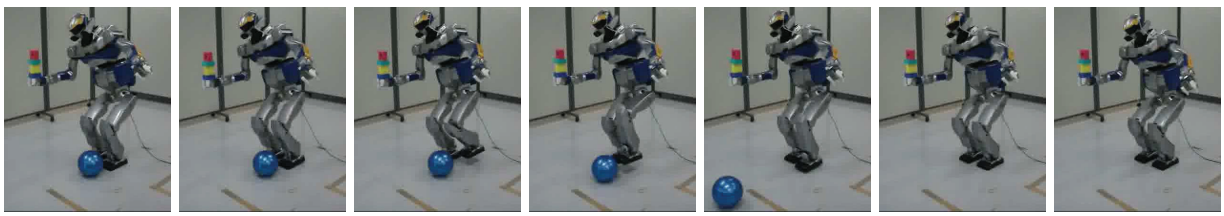


Fig. 8. The robot kicks the ball with the right gripper at a constant position.

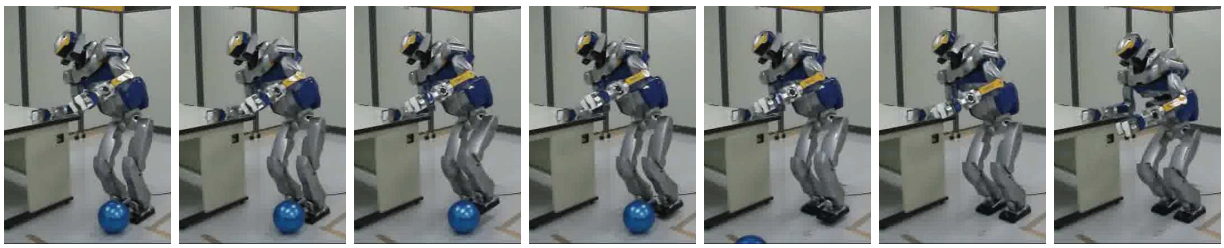


Fig. 9. The robot leans on a table in order to kick the ball.

that was used to perform walking motion during a functional magnetic resonance imaging (fMRI)-based robotic embodiment (Cohen et al., 2012). We extended this method to study the impact of passive or stimulated knee joints on the fatigue of the upper limbs of a paraplegic person during a sitting pivot transfer motion (lengagne et al., 2012). All of those experiments and simulations illustrate the versatility and efficiency of our method.

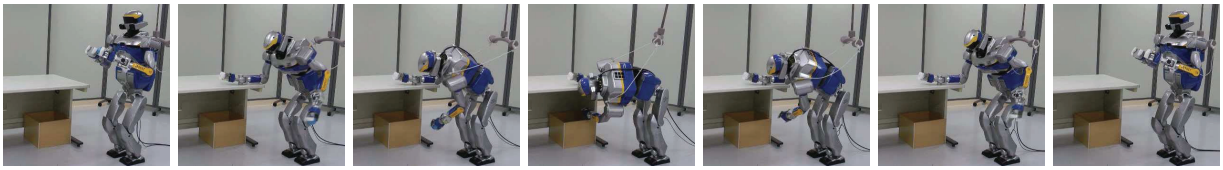


Fig. 10. With the right arm, the HRP-2 robot takes an additional support on the desk, which allows it to stably lean and put a ball into the trash box located under the desk.

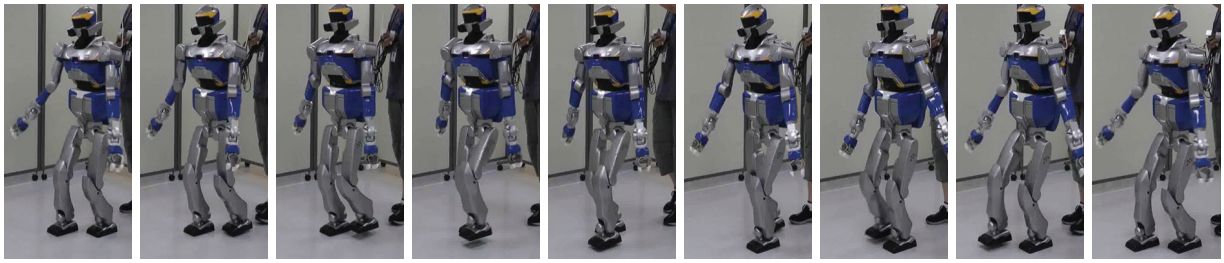


Fig. 11. Walking motion without additional constraints.

Fig. 12. The HRP-2 over crossing a 15 cm platform: the HRP-2 takes steps the platform with one foot and over-cross it with the other.

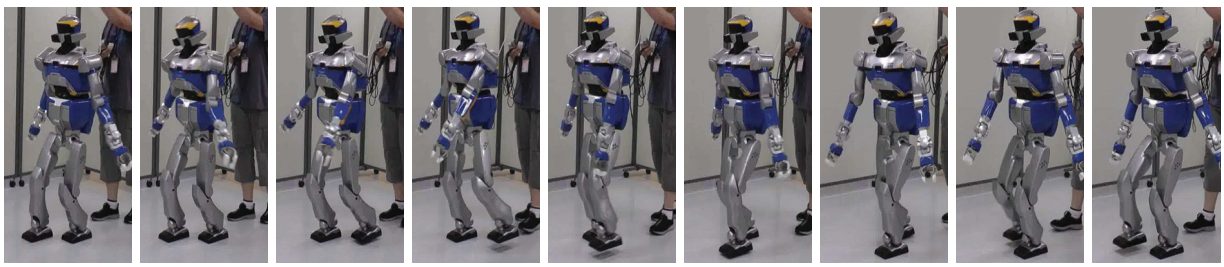


Fig. 13. Walking motion with a locked knee.

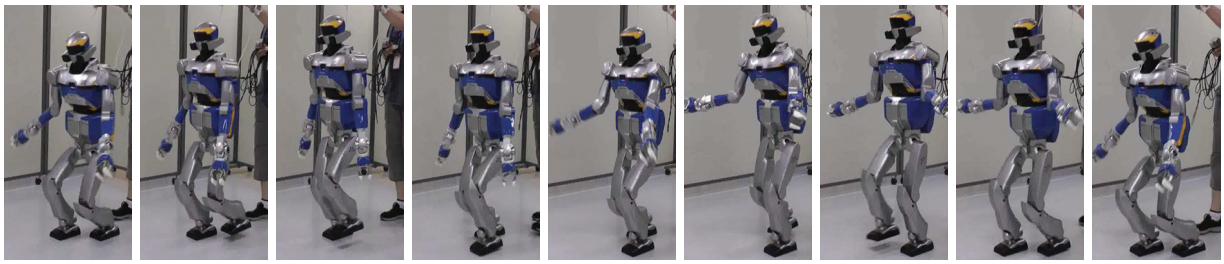


Fig. 14. Walking motion with a sore leg.

7. Discussions

We propose a versatile and effective method to generate dynamic and optimal multi-contact motions for humanoid robots. We highlight through several typical complex scenarios, its ability to generate complex motions that cannot be computed using existing geometric planning methods or task-function approaches.

Since we compute the entire motion over all of the phases during the same optimization process, we have an optimal motion for the considered sequence of contact stances. The generation of the sequence is based on quasi-static postures in the multi-contact planner that provide us with the contact stances. We believe that it is possible to merge the computation of the sequence of contact stances and the generation of the dynamic motion.

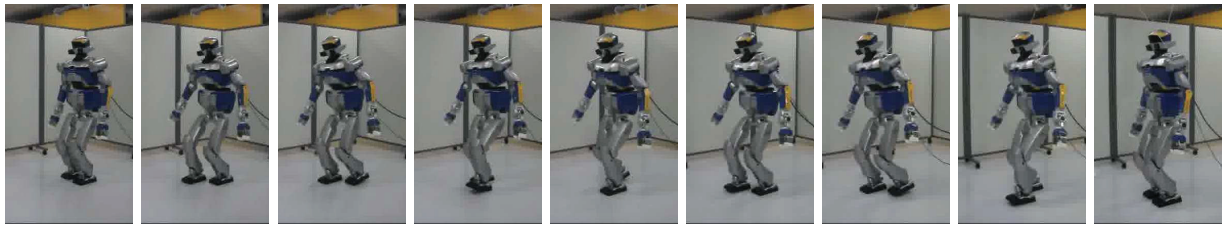


Fig. 15. The HRP-2 humanoid rotates around the edges during a walking motion.

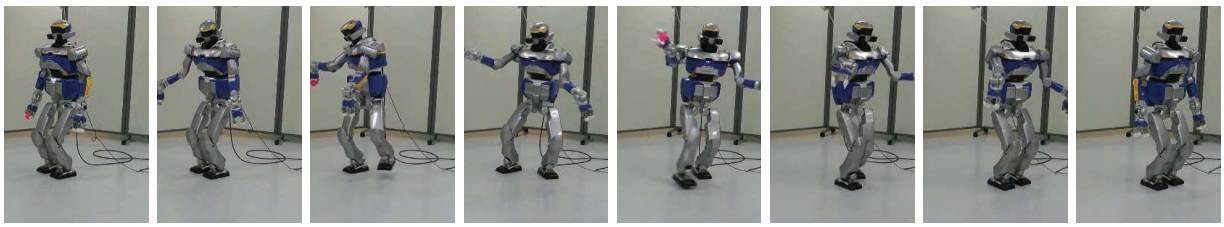


Fig. 16. The robot throws a ball mimicking a baseball player.

Our method still requires a large amount of computation time. To make it more attractive, we must decrease the computation time. There are several avenues to explore to facilitate this. The first trivial one is to reduce the horizon of the preview window by feeding the solver with only the upcoming new two of three transitions rather than giving the entire sequence of contacts. This option is interesting only after reaching good computation performances since the method can then be transformed into a model-based nonlinear preview controller.

We currently use the L-BFGS approximation of the Hessian provided by the IPOPT solver (Wächter and Biegler, 2006) that considers the Hessian matrix full. Indeed, regarding the decomposition of the motion into several phases and time interval, it appears that Hessian matrix is sparse which is not taken into account for the moment. An implementation of a sparse L-BFGS method should help to decrease the number of iterations, and hence the computation time. Another solution should be to include parallel architectures such as the general purpose graphics processing unit (GPGPU) in order to make use of parallel computation of the constraint and cost functions and of their gradients. It is possible to parallelize at least the computation of each time interval $[t]_i$ that are independent from each other. By considering these issues, it could be interesting to design a new solver that can use the computational efficiency of the GPGPU and exploit the sparse properties of the problem under study.

Albeit the presence of the flexible parts and assuming a perfectly known environment, the motion is currently performed without any balance controller. Future work should focus on the motion adaptation as presented by Lengagne et al. (2011c) and/or on a torque control law similar to Sentis et al. (2010) that can deal with uncertainties regarding the position and orientation of the contact points, i.e. on the position and dimension of the table and chair in our case.

8. Conclusion

In this paper, we solve multi-contact transitions motions using nonlinear optimization techniques. Computed trajectories guarantee constraint satisfaction all over the time motion whatever the time-grid size and accounts for all classical robotic constraints including non-desired collisions avoidance. We experimented the computed trajectories on complex multi-contact scenarios involving the HRP-2 humanoid robot. As far as we know, there is no existing method that can produce such complex multi-contact motions. This is mainly due to the difficulties to model the dynamic balance and unilateral forces, to deal efficiently with the continuous constraints and to integrate the collision and self-collision in a safe way. We presented an analytical formulation of the contact forces that is a counterpart to the dynamic effect and encompass the balance of the robot. To ensure the validity of the continuous constraints we make use of Taylor series expansion approximations for the models of the robot.

Our future work is now focused on building and improving performance and robustness strategies in using effectively such a method in close interchange with low-level controllers.

Notes

1. Note that the n-order basis functions used to evaluate extrema are different from the cubic basis function used to define the joint trajectories in Equation (4.4).
2. We neglect the impacts at contact creation because we constrain the landing speed at contact creations.
3. It is also very strongly suggested in all of the optimization textbooks (e.g. Gill et al., 1982, p. 5).

Funding

This work is partially supported by grants from the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft),

from the Japan Society for the Promotion of Science (JSPS; Grant-in-Aid for JSPS Fellows P09809 and for Scientific Research (B), 22300071, 2010), and from the RoboHow.Cog FP7 (see http://www.robohow.eu).

Acknowledgements

Authors would like to thank Dr Francois Keith and Pierre Gergondet for their help and assistance in making the simulation and experiments.

References

Arisumi H, Miossec S, Chardonnet JR and Yokoi K (2008) Dynamic lifting by whole body motion of human robots. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 668–675.

Benallegue M, Escande A, Miossec S and Kheddar A (2009) Fast C1 proximity queries using support mapping of sphere-torus-patches bounding volumes. In: *IEEE International Conference on Robotics and Automation, 2009 (ICRA '09)*, pp. 483–488.

Bendtsen C and Stauning O (1996) *FADBAD, a Flexible C++ Package for Automatic Differentation*.

Berz M, Hoffstätter G and Atter GH (1998) Computation and application of Taylor polynomials with interval remainder bounds. *Reliable Computing* 4: 83–97.

Bobrow JE (1988) Optimal robot plant planning using the minimum-time criterion. *IEEE Journal of Robotics and Automation* 4: 443–450.

Bouyarmane K, Escande A, Lamiraux F and Kheddar A (2009) Potential field guide for humanoid multicontacts acyclic motion planning. In: *IEEE International Conference on Robotics and Automation*.

Bouyarmane K and Kheddar A (2012) Humanoid robot locomotion and manipulation step planning. *Advanced Robotics (International Journal of the Robotics Society of Japan)* 26: 1099–1126.

Breteler MDK, Gielen SC and Meulenbroek RG (2001) End-point constraints in aiming movements: effects of approach angle and speed. *Biological Cybernetics* 85(1): 65–75.

Cohen O, Druon S, Lengagne S, et al. (2012) fMRI based robotic embodiment: a pilot study. In: *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*.

de Alwis T (2004) Maximizing or minimizing polynomials using algebraic inequalities. In: *Proceedings of the 9th Asian Technological Conference on Mathematics*, Singapore, pp. 88–97.

de Boor C (1978) *A Pratical Guide to Splines*, volume 27. New York: Springer-Verlag.

Diehl M, Bock HG, Diedam H and Wieber PB (2006) Fast direct multiple shooting algorithms for optimal robot control. In: Diehl M and Mombaur K (eds.), *Fast Motions in Biomechanics and Robotics* (*Lecture Notes in Control and Information Sciences*, vol. 340). Berlin: Springer, pp. 65–93.

Escande A and Kheddar A (2009) Contact planning for acyclic motion with tasks constraints. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*.

Escande A, Kheddar A and Miossec S (2006) Planning support contact-points for humanoid robots and experiments on HRP-2. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, pp. 2974–2979.

Escande A, Kheddar A, Miossec S and Garsault S (2008) Planning support contact-points for acyclic motions and experiments on HRP-2. In: Khatib O, Kumar V and Pappas G (eds.), *International Symposium on Experimental Robotics*, Athens, Greece (*Springer Tracts in Advanced Robotics*, number 54) Berlin: Springer-Verlag, pp. 293–302.

Escande A, Miossec S and Kheddar A (2007) Continuous gradient proximity distance for humanoids free-collision optimized-postures. In: *IEEE-RAS 7th International Conference on Humanoid Robots*.

Gill PE, Murray W and Wright MH (1982) *Practical optimization*. New York: Academic Press.

Goswami A (1999) Postural stability of biped robots and the foot-rotation indicator (FRI) point. *International Journal of Robotic Research* 18: 523–533.

Harada K, Kajita S, Kaneko K and Hirukawa H (2006) Dynamics and balance of a humanoid robot during manipulation tasks. *IEEE Transactions on Robotics* 22: 568–575.

Hauser K, Bretl T and Latombe JC (2005) Non-gaited humanoid locomotion planning. In: *2005 5th IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, pp. 7–12.

Hauser K and Latombe JC (2010) Multi-modal motion planning in non-expansive spaces. *The International Journal of Robotics Research* 29: 897–915.

Hettich R and Kortanek KO (1993) Semi-infinite programming: theory, methods, and applications. *SIAM Review* 35: 380–429.

Hirukawa H, Hattori S, Harada K, et al. (2006) A universal stability criterion of the foot contact of legged robots - adios ZMP. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1976–1983.

Hyon SH, Hale JG and Cheng G (2007) Full-body compliant human–humanoid interaction: balancing in the presence of unknown external forces. *IEEE Transactions on Robotics* 23: 884–898.

Kajita S, Kanehiro F, Kaneko K, et al. (2003) Biped walking pattern generation by using preview control of zero-moment point. In: *IEEE International Conference on Robotics and Automation*, volume 2, pp. 1620–1626.

Kajita S, Nagasaki T, Kaneko K, Yokoi K and Tanie K (2005) A running controller of humanoid biped HRP-2LR. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005 (ICRA 2005)*, pp. 616–622.

Kaneko K, Kanehiro F, Kajita S, et al. (2004) Humanoid robot HRP-2. In: *Proceedings 2004 IEEE International Conference on Robotics and Automation, 2004 (ICRA '04)*, vol. 2, pp. 1083–1090.

Khalil W and Dombre E (2002) *Modeling, Identification & Control of Robots*, 3rd edn. Paris: Hermes Sciences Europe.

Lee SH and Goswami A (2012) A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Journal of Autonomous Robots* 33: 399–414.

Lee SH, Kim J, Park F, Kim M and Bobrow JE (2005) Newton-type algorithms for dynamics-based robot movement optimization. *IEEE Transactions on Robotics* 21: 657–667.

Lee Y, Lengagne S, Kheddar A and Kim YJ (2012) Accurate evaluation of a distance function for optimization-based motion planning. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*.

Lengagne S, Jovic J, Pierella C, Fraisse P and Coste CA (2012) Generation of multi-contact motions with passive joints: Improvement of sitting pivot transfer strategy for paraplegics. In: *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*.

Lengagne S, Kheddar A, Druon S and Yoshida E (2011a) Emulating human leg impairments and disabilities in walking with humanoid robots. In: *IEEE International Conference on Robotics & Biomimetics*.

Lengagne S, Kheddar A and Yoshida E (2011b) Considering floatting contact and un-modeled effects for multi-contact motion generation. In: *Workshop on Humanoid Service Robot Navigation in Crowded and Dynamic Environments at the IEEE Humanoids Conference*.

Lengagne S, Mathieu P, Kheddar A and Yoshida E (2010a) Generation of dynamic motions under continuous constraints: Efficient computation using B-splines and Taylor polynomials. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Lengagne S, Mathieu P, Kheddar A and Yoshida E (2010b) Generation of dynamic multi-contact motions: 2D case studies. In: *IEEE-RAS Internationnal conference on Humanoid robots*.

Lengagne S, Ramdani N and Fraisse P (2011c) Planning and fast replanning safe motions for humanoid robots. *IEEE Transactions on Robotics* 27: 1095–1106.

Miossec S, Yokoi K and Kheddar A (2006) Development of a software for motion optimization of robots– application to the kick motion of the HRP-2 robot. In: *IEEE International Conference on Robotics and Biomimetics*, pp. 299–304.

Mombaur KD, Bock HG, Schlöder JP and Longman RW (2005) Open-loop stable solutions of periodic optimal control problems in robotics. *Zeitschrift für Angewandte Mathematik und Mechanik* 85: 499–515.

Park FC, Bobrow JE and Ploen SR (1995) A lie group formulation of robot dynamics. *The International Journal of Robotics Research* 14: 609–618.

Piazzi A and Visioli A (1998) Global minimum-time trajectory planning of mechanical manipulators using interval analysis. *International Journal of Control* 71: 631–652.

Piazzi A and Visioli A (2000) Global minimum-jerk trajectory planning of robot manipulators. *IEEE Transactions on Industrial Electronics* 47: 140–149.

Reemtsen R and Rückmann JJ (eds.) (1998) *Nonconvex Optimization and Its Applications: Semi-infinite Programming*. Kluwer Academic Publishers.

Righetti L, Buchli J, Mistry M and Schaal S (2011) Control of legged robots with optimal distribution of contact forces. In: *IEEE-RAS International Conference on Humanoid Robots*, pp. 318–324.

Schultz G and Mombaur KD (2010) Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics* 15: 783–792.

Sentis L, Park J and Khatib O (2010) Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Transactions on Robotics* 26: 483–501.

Steinbach MC (1997) *Optimal Motion Design using Inverse Dynamics*. Technical Report, Konrad-Zuse-Zentrum für Informationstechnik Berlin.

Stephens BJ and Atkeson CG (2010) Dynamic balance force control for compliant humanoid robots. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1248–1255.

Uno Y, Kawato M and Suzuki R (1989) Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cybernetics* 6(2): 89–101.

Von Stryk O (1993) Numerical solution of optimal control problems by direct collocation. In *Optimal Control* (*International Series of Numerical Mathematics*, vol. 111). Berlin: Springer, pp. 129–143.

Von Stryk O (1998) Optimal control of multibody systems in minimal coordinates. *Zeitschrift für Angewandte Mathematik und Mechanik* 78(S3): 1117–1120.

Vukobratović M and Borovac B (2004) Zero-moment point: thirty five years of its life. *International Journal of Humanoid Robotics* 1: 157–173.

Wächter A and Biegler LT (2006) On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming* 106: 22–57.

Zhang X, Redon S, Lee M and Kim YJ (2007) Continuous collision detection for articulated models using taylor models and temporal culling. *ACM Transactions on Graphics* 26(3): 15.

Appendix A: Analytical formulation of the contact forces

A.1. Problem

Knowing the joint trajectories, we analytically compute the contact forces in Equation (3.6), which can also be written as

$$\mathbb{F} = -(\mathbf{J}_2^{\mathrm{T}}(q))^{-1}\,\mathbb{D}_2 \tag{.1}$$

Since the matrix $\mathbf{J}_2$ is not necessarily square, the pseudo-inverse: $(\mathbf{J}_2^{\mathrm{T}})^{+} = (\mathbf{J}_2\mathbf{J}_2^{\mathrm{T}})^{-1}\,\mathbf{J}_2$ can be computed instead of $(\mathbf{J}_2^{\mathrm{T}})^{-1}$. By definition, the pseudo-inverse solves an equation system that has more degrees of freedom than equations, which is our case, by minimizing the Euclidean norm of the solution. Yet, our primary interest is not to minimize the Euclidean norm but to find a function $f(q, \mathbb{D}_2)$ that solves the contact forces which satisfy both Equation (3.6) and as much as possible Equation (3.4), that is

$$\mathbb{F} = f(q, \mathbb{D}_2) \tag{.2}$$

We propose to find the analytic solution for the contact forces that reflects the balance of the robot, cf. Equation (3.6), and the friction constraints, Equation (3.4), i.e. the analytic solution to the following problem:

$$\begin{gathered} \min \tfrac{1}{2} \textstyle\sum_i \beta_i(\alpha_i ||F_i^t||^2 + F_i^{n2}) \\ \textstyle\sum_i \left(\begin{bmatrix} \hat{P}_i A_i \\ A_i \end{bmatrix} [F_i] \right) + [\mathbb{D}_2] = 0 \end{gathered} \tag{.3}$$

In this formulation, we decompose $\mathbf{J}_2^{\mathrm{T}}$ through (i)$\hat{P}_i$, the screw operator of the contact position, and (ii) A_i the orientation of the contact framework; $[\mathbb{D}_2] = [M_x, M_y, M_z, F_x, F_y, F_z]^{\mathrm{T}}$ is the effort due to the free dynamics, β_i a coefficient to equilibrate (or not) the repartition of the forces, α_i is used to weight the tangential forces with regards to its corresponding normal force. The solution to this problem ensures the dynamic equality constraint Equation (3.6), but does not guarantee the feasibility of the contact forces that must be checked by the optimization solver. Note that defining $\forall i\ \beta_i = 1, \alpha_i = 1$ is equivalent to solve the pseudo-inverse problem.

A.2. Analytic solution

The resolution of this problem starts by writing the Lagrangian:

$$L = \sum_i \begin{bmatrix} \beta_i \alpha_i F_{x,i}^2 \\ \beta_i \alpha_i F_{y,i}^2 \\ \beta_i F_{z,i}^2 \end{bmatrix} + \left(\sum_i \begin{bmatrix} \hat{P}_i A_i \\ A_i \end{bmatrix} [F_i] + [\mathbb{D}_2] \right) [\lambda] \tag{.4}$$

Here, we assume that the z-axis is the normal direction of the contact forces and define $[\lambda]$ as the vector of the Lagrange multipliers. The optimal solution fulfills the optimality condition:

$$\frac{\partial L}{\partial (F_{\circ,i}, \lambda_j)} = 0 \tag{.5}$$

From the derivative with respect to F_i we have

$$F_i = -\mathbb{W}_i^{-1} \begin{bmatrix} \hat{P}_i A_i & A_i \end{bmatrix} [\lambda] \tag{.6}$$

with $\mathbb{W}_i = \text{diag}(\beta_i \alpha_i, \beta_i \alpha_i, \beta_i)$. Then, we replace the contact forces expression in the set of the equality constraint and we get

$$\Omega [\lambda] = - [\mathbb{D}_2] \tag{.7}$$

with

$$\Omega = \sum_i \left(\begin{bmatrix} \hat{P}_i A_i \\ A_i \end{bmatrix} \mathbb{W}_i^{-1} \begin{bmatrix} \hat{P}_i A_i & A_i \end{bmatrix} \right) \tag{.8}$$

where $\Omega \in \mathbb{R}^{6\times 6}$ is a 6×6 matrix that we can easily invert using the Gauss–Jordan algorithm to find the value of the Lagrange multipliers:

$$[\lambda] = -\Omega^{-1} [\mathbb{D}_2] \tag{.9}$$

Then, we put the Lagrange multipliers $[\lambda]$ in Equation (.6) to get the value of the contact forces. Eventually, we are able to compute the contact forces from the joint trajectories and, using Equation (3.3), we can compute the joint torques.

Appendix B: Properties of the B-splines

We briefly present the properties of the B-splines that are relevant during the motion generation process.

B.1. Control points

The first nicety is related to the computation of the derivative of the state variables regarding the optimization parameters. Since, at any time instant, the joint values rely only on four control points, we choose to define intervals so that, all along a time interval, the joint value relies only on four control points, see Figure 3. This avoids to compute the derivative regarding all the parameters since only a few of them have a non-null value.

B.2. Convex hull

B-splines ($q(t)$) is entirely contained in the convex hull of its control points; that yields

$$\begin{array}{lll} \text{if} & \forall i \in [1, m] & \underline{q} \le p_i \le \overline{q} \\ \text{then} & \forall t \in [\Delta t] & \underline{q} \le q(t) \le \overline{q} \end{array} \tag{.10}$$

We use this property to ensure the joint limits without implementing them as sets of continuous inequality constraints, but rather considering bounds directly on the optimization parameters.

B.3. Derivative of a B-spline

The time derivative of a B-spline with m control points is another B-splines parameterized with $m - 1$ control points. This is obtained by derivation of the Cox–de Boor recursion (de Boor, 1978) with respect to time t:

$$\dot{q}(t) = \sum_{i=1}^{m} \dot{b}_i^k(t) p_i = \sum_{i=1}^{m-1} b_i^{k-1}(t) r_i \tag{.11}$$

with

$$r_i = \frac{k}{u_{i+k+1} - u_{i+1}} (p_{i+1} - p_i) \tag{.12}$$

here k is the order of the B-spline basis functions, u_i is the ith component of the nodal vector as defined in (de Boor, 1978). It is then possible to obtain a system of $(m - 1)$ linear inequalities to impose joint speed limits, as we enforced joint position limits as bounds on the B-spline parameters. Thus, we add the following constraints:

$$\underline{\dot{q}} \le r_i \le \overline{\dot{q}} \tag{.13}$$

Extensions to upper derivatives follow in the same way.

B.4. Optimality

Hence, the obtained solution is very likely sub-optimal as there is no reciprocal to Equation (.10). We already raised this issue in Lengagne et al. (2010a). Yet, the parameterization of the joint trajectories (by any method) leads already to a sub-optimal motion and we believe that there is a venue for a thorough study on other possible parameterizations. Indeed, any kind of parameterization restricts the IP problem's solution for the chosen parameterization, which is *de facto* sub-optimal.

Appendix C: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Experimental Validations