Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints

Layale Saab, Oscar Efrain Ramos Ponce, François Keith, Nicolas Mansard, Philippe Souères, Jean-Yves Fourquet

▶ **To cite this version:**

Layale Saab, Oscar Efrain Ramos Ponce, François Keith, Nicolas Mansard, Philippe Souères, et al.. Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints. IEEE Transactions on Robotics, 2013, 29 (2), pp.346-362. 10.1109/TRO.2012.2234351 . lirmm-00831097

HAL Id: lirmm-00831097

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00831097

Submitted on 6 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints

Layale Saab, Oscar E. Ramos, François Keith, Nicolas Mansard, Philippe Souères, Jean-Yves Fourquet

***Abstract*—The most widely-used technique to generate whole-body motions on a humanoid robot accounting for various tasks and constraints is the inverse kinematics. Based on the task-function approach, this class of methods makes possible the coordination of the robot movements to execute several tasks in parallel and account for the sensor feedback, in real-time thanks to the low computation cost. To some extent, it also enables dealing with some of the robot constraints (e.g. joint limits or visibility) and managing the quasi-static balance of the robot. In order to fully use the whole range of possible motions, this paper proposes to extend the task-function approach to handle the full dynamics of the robot multi-body along with any constraint written as equality or inequality of the state and control variables. The definition of multiple objectives is made possible by ordering them inside a strict hierarchy. Several models of contact with the environment can be implemented in the framework. We propose a reduced formulation of the multiple rigid planar contact that keeps a low computation cost. The efficiency of this approach is illustrated by presenting several multi-contact dynamic motions in simulation and on the real HRP-2 robot.**

***Index Terms*—Humanoid robotics, redundant robots, dynamics, force control, contact modeling.**

I. Introduction

THE GENERATION of motion for humanoid robots is a challenging problem, due to the complexity of their tree-like structure and the instability of their bipedal posture [3]. Typical examples are shown in Fig. 1, with the HRP-2 robot using multiple non-coplanar contacts to perform a dynamic motion. These robots own a large number of degrees of freedom (DOF), typically more than 30. In return, they are subject to various sets of constraints (balance, contact, actuator limits), that reduce the space of possible motions. These constraints can typically be formulated as equalities (e.g. zero velocity at rigid-contact points [4]), and inequalities (e.g. joint position [5], velocity or torques bounds, obstacles [6]). Moreover, they are of relative importance (e.g. balance has to be considered more important than visibility [7]). In total, the motion has to be designed in a set that lives in the high-dimensional configuration space but is implicitly limited to a much smaller submanifold by the set of constraints. This makes the classical sampling methods [8], [9] more difficult to use than for a classical manipulator. The motion manifold cannot be sampled directly but by projection [10]. The connection process in high-dimension is costly [11] and and often fails due to the number of constraints.

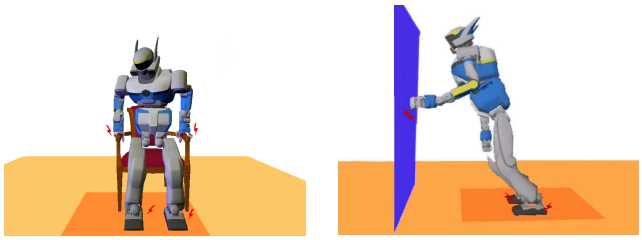


Fig. 1. Dynamic multi-contact motion with the HRP-2 model.

Rather than designing the motion at the whole-body level (configuration space), the task function approach [12], [13] proposes to design the motion in a space dedicated to the task to be performed. It is then easier to design the reference motion in the task space, and transcripting this reference from the task space to the whole-body level is only a numerical problem. This approach is versatile, since the same task is generally transposable from one robot or situation to another. It also eases the use of sensory feedback, since the sensory space is often a good task-space candidate [14], [15].

A task is a basic brick of motion, which can be combined sequentially [16] or simultaneously to a complex motion. Simultaneous execution can be achieved in two ways: by weighting, or by imposing a strict hierarchy. Coming from numerical optimization [17], this second solution was introduced in robotics by [18] and formalized for any number of tasks in [19], [20]. This approach is well fitted to cope with equality constraints. However, inequality constraints cannot be taken into account explicitly. Therefore, approximate solutions, such as potential field approaches [21], [7] or damping functions [5], [22] have been proposed to consider inequalities.

The transcription of the motion reference from the task space to the whole-body control is naturally written as a

Manuscript received July 19, 2012; accepted December 11, 2012. This paper was recommended for publication by Associate Editor Y. Choi and Editor B. J. Nelson upon evaluation of the reviewers comments. This paper was presented in part at the IEEE International Conference on Robotics and Automation in 2011 [1] and the IEEE/RSJ International Conference on Intelligent Robots and Systems in 2011 [2].

L. Saab was with the Laboratory for Analysis and Architecture of Systems, Centre National de la Recherche Scientifique, University of Toulouse, Toulouse 31077, France. She is now with EOS Innovation, Évry 91000, France (e-mail: layalesaab@gmail.com).

O. E. Ramos, N. Mansard, and P. Souères are with the Laboratory for Analysis and Architecture of Systems, Centre National de la Recherche Scientifique, University of Toulouse, Toulouse 31077, France (e-mail: oramos@gmail.com; nicolas.mansard@laas.fr; soueres@laas.fr).

F. Keith is with the Laboratoire dInformatique de Robotique et de Micro électronique de Montpellier, Centre National de la Recherche Scientifique, Montpellier 34095, France (e-mail: francois.keith@lirmm.fr).

J-Y. Fourquet is with the Laboratoire Génie de Production, de lÉcole Nationale d Ingénieurs de Tarbes, University of Toulouse, Tarbes 65016, France (e-mail: fourquet@enit.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2012.2234351

quadratic program (QP) [23]. A QP is composed of two layers, namely the constraint and the cost. It can be seen as a hierarchy of two levels, the constraint having priority over the cost. If only equality constraints are considered, the QP resolution corresponds to the inversion schemes [20], in the particular case of two levels. Inequalities can also be taken into account directly, as constraints, or in the cost function [24]. In [25], a method to extend the QP formulation to any number of priority levels is given. The solution of such a hierarchical problem is computed by solving a cascade of QP (or hierarchical QP). In [26], a dedicated solver has been proposed to obtain the solution at lower cost (in one step instead of a cascade).

All these works only consider the kinematics of the robot. On a humanoid robot, many constraints arise from the dynamics of the multi-body system. The formulation by task can be extended to compute the torque at the whole-body level from the reference motion expressed in the dedicated task space, also called *operational space* [27]. For a humanoid in contact, the motion is constrained to the submanifold of configurations that respects the contact model [28] as illustrated by Fig. 1. A review of the work in modeling and control of the dynamics of a set of bodies in contact is proposed in [29], [30]. The connection with inverse dynamics has been done in [31], [32]. Using these approaches, it is possible to take into account a hierarchy of tasks and constraints (or *stack* of tasks [33]), all written as equalities [34], [35]. In [36], a first solution to handle inequalities in the stack of tasks was proposed, but cannot set any inequality constraint on the contact forces. In [37], [38], the inverse-dynamics problem has been written as a QP, where the unilateral contact constraints, along with classical unilateral constraints (joint limits, etc) are explicitly considered. In that case, several tasks can be composed by setting relative weights, but a hierarchy of tasks is not possible.

In this paper, we propose a generic solution to take into account equalities and inequalities in a strict hierarchy to generate a dynamic motion. This solution is based on the similarities between inverse kinematics and inverse dynamics. In Section II, the inverse-kinematics scheme is recalled, written into a general form; the possibility of taking into account inequalities is then introduced using the solver [25], [26]. Then, putting the operational-space inverse dynamics under the same generic form, Section III uses the same hierarchical solver to take into account both dynamics and inequalities. This first solution deals with the robot in free space. In Section IV, contacts are introduced in the model and used in the resolution scheme. The contact model is generic and can be adapted to various situations (rigid contact, friction cone [39], elastic contact [40]). A solution is proposed in Section V to implement a reduced form of multiple plane/plane slidingless rigid contacts. In Section VI the connection is made with the zero-moment point (ZMP) contact criterion [41] classically used in humanoid robotics [42]. The generation is close to the real time (around 20ms per control cycle on a typical 30DOF robot). Some examples of complex motions involving non-coplanar contacts and their execution on the real robot are presented in Section VII.

II. Inverse Kinematics

A. The task-function approach

The task-function approach [13], or operational-space approach [27], [43], provides a mathematical framework to describe tasks in terms of specific output functions. The task function is a function from the configuration space to an arbitrary task space, chosen to ease the observation and the control of the motion with respect to the task to perform.

A task is defined by a triplet $(e, \dot{e}^*, Q)$, where e is the task function that maps the configuration space to the task space, $\dot{e}^*$ is the reference behavior expressed in the tangent space to the task space at e. Q is the differential mapping between the task space and the control space of the robot which verifies the relation:

$$\dot{e} + \mu = Qu \tag{1}$$

where u is the control in the configuration space and μ is the drift of the task. To compute a specific robot control u^* that performs the reference $\dot{e}^*$, any numerical inverse of Q can be used. The generic expression of the control law is then :

$$u^* = Q^{\#}(\dot{e}^* + \mu) + Pu_2 \tag{2}$$

In this expression, the first part performs the task, and the second part, modulated by the secondary control input u_2, expresses the redundancy of the task [18]. In the first term, $Q^{\#}$ is any reflexive generalized inverse of Q, often chosen to be the (Moore-Penrose) pseudoinverse Q^{+} [44] or a weighted inverse $Q^{\#W}$ [45] (see App. A). In the second term of (2), $P = I - Q^{\#}Q$ is the projector onto the null space of Q corresponding to $Q^{\#}$.

B. Hierarchy of tasks

The projector P is intrinsically related to the redundancy of the robot with respect to the task e. A secondary task $(e_2, \dot{e}_2^*, Q_2)$ can be executed using u_2 as a new control input. Introducing (2) in $\dot{e}_2 + \mu_2 = Q_2 u$ gives:

$$\dot{e}_2 + \tilde{\mu}_2 = \widetilde{Q_2} u_2 \tag{3}$$

with $\tilde{\mu}_2 = \mu_2 - Q_2 Q^{\#}(\dot{e}^* + \mu)$ and $\widetilde{Q_2} = Q_2 P$. This last equation fits the template (1), and can be solved using the generic expression (2) [20]:

$$u_2{}^* = \widetilde{Q_2}^{\#}(\dot{e}^* + \tilde{\mu_2}) + P_2 u_3 \tag{4}$$

where P_2 enables the propagation of the redundancy to a third task using the input u_3. By recurrence, this generic scheme can be extended to any arbitrary hierarchy of tasks.

C. Inverse kinematics formulation

In the inverse-kinematics problem, the control input u is simply the robot joint velocity $\dot{q}$. The differential map Q between the task and the control is the task Jacobian J. In that case, the drift $\mu = \frac{\partial e}{\partial t}$ is often null, and (1) is written:

$$\dot{e} = J\dot{q} \tag{5}$$

The simplest and most-often used solution is to choose $Q^{\#}$ to be the pseudoinverse Q^{+}, that gives the least Euclidean norm of both $\dot{q}$ and $\dot{e}^* - J\dot{q}$ [46], [47]. The control law is then:

$$\dot{q}^* = \dot{q}_1^* + P\dot{q}_2 \tag{6}$$

where $\dot{q}_1^* = J^+\dot{e}^*$. A typical reference behavior is an exponential decay of e to zero: $\dot{e}^* = -\lambda e, \lambda > 0$.

It may happen that J becomes singular, i.e. $rank(J) < r_0$, where r_0 is the nominal rank of J out of the singular configuration. Numerical problems can occur during the transition from the nominal situation to the singular one. To avoid these problems, the pseudoinverse is often approximated by the damped least-square $J^{\dagger}$ defined by [48], [49]:

$$J^{\dagger} = \begin{bmatrix} J \\ \eta I \end{bmatrix}^{+} \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{7}$$

where I is the identity matrix of proper size and η is a damping factor, chosen as an additional parameter of the control (typically, $\eta = 10^{-2}$ for a humanoid robot).

D. Projected inverse kinematics

Consider a secondary task $(e_2, \dot{e}_2^*, J_2)$. The template (3) is:

$$\dot{e}_2 - J_2\dot{q}_1^* = J_2P\dot{q}_2 \tag{8}$$

In this case, the differential map is the projected Jacobian $Q = J_2P$, and the drift is $\mu = -J_2\dot{q}_1^*$. The control input $\dot{q}_2^*$ is obtained once more by numerical inversion [20], [50]:

$$\dot{q}_2^* = (J_2P)^{+}(\dot{e}_2 - J_2\dot{q}^*) + P_2\dot{q}_3 \tag{9}$$

where P_2 is the projector into the null space of J_2P. The same scheme can be reproduced iteratively to take into account any number of tasks until P_i is null.

In general $rank(J_2P) \leq rank(J_2) \leq r_2$, where r_2 is the nominal rank of J_2. When the second inequality is strict, the singularity is said to be *kinematic*; when the first inequality is strict, the singularity is said to be *algorithmic* [51][1]. To avoid any numerical problem in the neighborhood of the singularity, a damped inverse can be used to invert J_2P.

E. Hierarchical quadratic program resolution

1) Generic formulation: When considering a single task, the solution obtained with the pseudoinverse (2) is known to be the optimal solution of the QP $\min_u ||Qu - \dot{e}^* - \mu||^2$. The great advantage of the QP formulation is that both linear equalities and inequalities can be considered, while the pseudoinverse-based schemes presented above cannot explicitly deal with inequalities. A quadratic program is composed of a quadratic cost function to be minimized while satisfying the set of constraints [52]. It can be seen as a two-level hierarchy, where the set of constraints has priority over the cost. Inequalities are set as the top-priority. The introduction of slack variables is a classical solution to handle an inequality at the second priority level [53]. In [25], it was proposed to use the slack variables to generalize the QP to more than two levels of hierarchy and thus to build a hierarchical quadratic problem (HQP) handling inequalities.

The HQP formulation is first recalled in a generic frame. A generic constraint k is defined by the linear map A_k and the two inequality bounds $(\underline{b}_k, \overline{b}_k)$, where $\underline{b}_k$ and $\overline{b}_k$ are respectively the lower and upper bounds on the reference behavior[2]. At level k, the cascade algorithm solving the hierarchy (A_k, b_k) is expressed by the following QP:

$$\begin{aligned} &\min_{u_k, w_k} ||w_k||^2 \\ \text{s.t.} \quad &\underline{\mathbf{b}}_{\mathbf{k-1}} \leq \mathbf{A_{k-1}}u_k + \mathbf{w}_{\mathbf{k-1}}^* \leq \overline{\mathbf{b}}_{\mathbf{k-1}} \\ &\underline{b}_k \leq A_k u_k + w_k \leq \overline{b}_k \end{aligned} \tag{10}$$

where $\mathbf{A_{k-1}}$, $(\underline{\mathbf{b}}_{\mathbf{k-1}}, \overline{\mathbf{b}}_{\mathbf{k-1}})$ are the constraints at all the previous levels from 1 to $k-1$ ($\mathbf{A_{k-1}} = (A_1, ..., A_{k-1})$), and A_k, $(\underline{b}_k, \overline{b}_k)$ is the constraint at level k.

The slack variable[3] w_k is used to add some freedom to the solver if no solution can be found when the constraint k is introduced under the $k-1$ previous constraints: w_k is variable and can be used by the solver to relax the last constraint A_k. On the other hand, $\mathbf{w}_{\mathbf{k-1}}^*$ is constant and set to the result of the previous optimization of the $k-1$ first QP (at each of the iterations of the cascade, $\mathbf{w}_{\mathbf{k-1}}^*$ is augmented with the optimal w_k^* by $\mathbf{w}_{\mathbf{k-1}}^* := (\mathbf{w}_{\mathbf{k-1}}^*, w_k^*)$). A solution to the *strict* $k-1$ constraint $\mathbf{A_{k-1}}$ is then always reached, even if the $slack$ constraint A_k is not feasible: this corresponds to the definition of the hierarchy.

A classical method to compute the solution of a QP or HQP relies on an active-search algorithm [52], [26] (see App. B), which implies iterative computations of the pseudoinverse of a subproblem of the initial QP. Since pseudoinverses are used, the classical numerical problems can occur in the neighborhood of singularities. Regularization methods that extend the damping inverse [49] used in robotics can be applied [54].

The method proposed above is generic and can be applied to any numerical problem written with a linear hierarchical structure. In that case, it is referred to as HQP (or cascade of QP) and denoted with the lexicographic order: $(i) \prec (ii) \prec (iii) \prec ...$ which means that the constraint (i) has the highest priority. In the following, we propose a solution to apply this formulation to invert kinematics and dynamics. The constraints are then the tasks defined above and the hierarchical solver is called a stack of tasks (SOT) or hierarchy of tasks.

2) Application to inverse kinematics: When considering a single task, inversion (6) corresponds to the optimal solution to the problem:

$$\min_{\dot{q}} ||J\dot{q} - \dot{e}^*||^2 \tag{11}$$

By applying the QP resolution scheme, both equalities and inequalities can be considered. Replacing b by $\dot{e}$, the reference part is then rewritten:

$$\underline{\dot{e}}^* \leq \dot{e} \leq \overline{\dot{e}}^* \tag{12}$$

[1]Both cases are similar in the sense that $\begin{bmatrix} J_1 \\ J_2 \end{bmatrix}$ is singular.

[2]Specific cases can be immediately implemented: $\underline{b}_k = \overline{b}_k$ in the case of equalities and $\underline{b}_k = -\infty$ or $\overline{b}_k = +\infty$ to handle single-bounded constraints.

[3]w is an implicit optimization variable whose explicit computation can be avoided when formulating the problem as a cascade. It does not appear in the vector of optimization variables u. See [26] for details.

For instance, in the case of two tasks with priority order $e_1 \prec e_2$, the expression of the QP is given by:

$$\begin{aligned} & \min_{\dot{q},w_2} ||w_2||^2 \\ s.t. \quad & \underline{\dot{e}}_1^* \leq J_1\dot{q} + w_1^* \leq \overline{\dot{e}}_1^* \\ & \underline{\dot{e}}_2^* \leq J_2\dot{q} + w_2 \leq \overline{\dot{e}}_2^* \end{aligned} \tag{13}$$

In robotics, when a constraint is expressed as an inequality, it is very likely to be put as the top priority: typically, joint limits and obstacle avoidance. Using this framework, it is also possible to handle inequalities at the second priority level (i.e. in the cost function). A typical case is to prevent visual occlusion when possible, or to keep a low velocity if possible, without disturbing the robot behavior when it is not necessary.

In the sequel, the HQP considering linear equalities and inequalities is extended from inverse kinematics to inverse dynamics.

III. Inverse Dynamics

In this section the case of a contact-free dynamical multi-body system without free-floating root is considered.

A. Task-space formulation

As previously stated, a task is defined by a task function e, a reference behavior and a differential mapping. At the dynamic level, the reference behavior is specified by the expected task acceleration $\ddot{e}^*$, while the control input is typically the joint torques τ. The operational-space inverse dynamics then refers to the problem of finding the torque control input τ that produces the task reference $\ddot{e}^*$, using any necessary joint acceleration $\ddot{q}$. The acceleration $\ddot{q}$ is then a side variable, that does not require to be explicitly computed during the resolution. Contrary to the case of kinematics, the mapping between the control input τ and the task space is obtained in two stages. First, the map between accelerations in the configuration space and in the task space is obtained by differentiating (5):

$$\ddot{e} = J\ddot{q} + \dot{J}\dot{q} \tag{14}$$

Then, the dynamic equation of the system expressed in the joint coordinates is deduced from the mechanical laws of motion [55].

$$A\ddot{q} + b = \tau \tag{15}$$

where $A = A(q)$ is the generalized inertia matrix of the system, $\ddot{q}$ is the joint acceleration, $b = b(q, \dot{q})$ includes all the nonlinear effects including Coriolis, centrifugal and gravity forces and τ are the joint torques. The generic form (1) is obtained by replacing $\ddot{q}$ in (15) with (14) [27]:

$$\ddot{e} - \dot{J}\dot{q} + JA^{-1}b = JA^{-1}\tau \tag{16}$$

This equation follows the template (1) with $Q = JA^{-1}$, $\mu = -\dot{J}\dot{q} + JA^{-1}b$ and $u = \tau$.

The torque τ^* that ensures $\ddot{e}^*$ is solved using the generic form (2). It is generally proposed to weight the inverse by the inertia matrix A. This weight ensures that the process is consistent with Gauss' principle [56], i.e. that the torques and accelerations corresponding to the redundancy of the task are the closest to the acceleration of the unconstrained multi-body system. This principle can be intuitively understood by considering the weight like a minimization of the acceleration pseudo energy $\ddot{q}^T A\ddot{q}$ [57], [31].

The redundancy can also be explicitly formulated during the inversion, using the form (3). A SOT can be iteratively built, with the lower-priority tasks being executed in the best possible way without disturbing the higher priority tasks [58], [59]:

$$\tau^* = \tau_1^* + P\tau_2 \tag{17}$$

where $P = I - J^T\ (JA^{-1}J^T)^+ JA^{-1}$ is the projector in the null space of JA^{-1} and $\tau_1^* = (JA^{-1})^{\#A}(\ddot{e}^* - \dot{J}\dot{q} + JA^{-1}b)$.

B. Projected inverse dynamics

As before, the differential map for the projected secondary task e_2 is obtained by replacing (17) into the robot dynamics equation in the task space $\ddot{e}_2 - \dot{J}_2\dot{q} + J_2A^{-1}b = J_2A^{-1}\tau$:

$$\ddot{e}_2 + \mu_2 = Q_2\tau_2 \tag{18}$$

with $\mu_2 = -\dot{J}_2\dot{q} + J_2A^{-1}b - J_2A^{-1}\tau_1^*$, and $Q_2 = J_2A^{-1}P$. The same weighted inverse is used to invert Q_2 [58], [59]. Accordingly, any number of tasks can be added iteratively until the projector becomes null.

The same singularities as in inverse kinematics may appear (the dynamics in itself does not bring any new singular case, since A is always full rank). To avoid any numerical problem, the damped weighted inverse is generally used. As for the kinematics, only tasks defined by equality constraints can be taken into account using this pseudoinverse-based resolution. To take into account inequalities, we propose to extend to the dynamics the HQP [25] that was previously introduced for the kinematics.

C. Application of the QP solver to the inverse dynamics

When resolving a given task e while taking into account the dynamics, both (14) and (15) must be fulfilled. There are two ways of formulating the QP. First, $\ddot{q}$ can be substituted from (14) into (15), to obtain the single reduced equation (16). In that case, the QP only requires to solve τ, the variable $\ddot{q}$ being not explicitly computed:

$$\min_{\tau} ||JA^{-1}\tau - \ddot{e}^* - \mu||^2 \tag{19}$$

Alternatively, (14) can be solved under the constraint (15). Using the hierarchy notation, the HQP is thus (15) $\prec$ (14), or using the standard QP notation:

$$\begin{aligned} & \min_{\tau,\ddot{q},w} ||w||^2 \\ s.t. \quad & A\ddot{q} + b = \tau \\ & \ddot{e}^* + w = J\ddot{q} + \dot{J}\dot{q} \end{aligned} \tag{20}$$

In that case, both τ and $\ddot{q}$ are explicitly computed. They constitute the vector of optimization variables $u = (\tau, \ddot{q})$.

QP (19) has a reduced form, but QP (20) allows any explicit formulation using the dynamics variables. In the following, such an exhaustive formulation is necessary to deal with the contact.

IV. Inverse dynamics under contact constraints

A. Insertion of the contact forces

In the previous section, the considered multi-body system was in free space (no contact forces) and fully actuated (no free-floating body, for example). The model of the humanoid robot includes both the contact forces and a zero-torque constraint on the six first DOF. First, the case of a single contact point denoted by x_c is considered:

$$A\ddot{q} + b + J_c^\top f = S^T\tau \tag{21}$$

where A and b are defined as before, $\ddot{q}$ is the vector of generalized joint accelerations[4], f is the 3D contact force applied at the contact point x_c, $J_c = \frac{\partial x_c}{\partial q}$ is the Jacobian matrix of x_c [5] and $S = [0\ I]$ is a matrix that allows to select the actuated joints.

The rigid-contact condition implies that there is no motion of the robot contact body x_c i.e. $\dot{x}_c = 0$, $\ddot{x}_c = 0$. For a given state, it implies the linear equality constraint:

$$J_c\ddot{q} = -\dot{J}_c\dot{q} \tag{22}$$

If multiplying (21) by J_cA^{-1} and substituting the expression of $J_c\ddot{q}$ given by (22), a constraint is obtained, that constrains the torque with respect to the contact force:

$$J_cA^{-1}J_c^\top f = J_cA^{-1}(S^T\tau - b) + \dot{J}_c\dot{q} \tag{23}$$

In this expression, the acceleration does not appear explicitly anymore. In the basic case, $J_cA^{-1}J_c^\top$ is invertible, and f can be deduced [35]:

$$f = (J_c^\top)^{A^{-1}\#}(S^T\tau - b) + (J_cA^{-1}J_c^\top)^{-1}\dot{J}_c\dot{q} \tag{24}$$

This expression of f can be re-injected in (21), to obtain a reformulated dynamic equation where the force variable does not appear explicitly anymore.

$$A\ddot{q} + b_c = P_cS^T\tau \tag{25}$$

where $P_c = (I - J_c{}^{\#A^{-1}}J_c)^T = (I - (J_cA^{-1})^{\#A}J_cA^{-1})$ is the projection operator of the contact[6], and $b_c = P_cb + J_c^\top(J_cA^{-1}J_c^\top)^{-1}\dot{J}_c\dot{q}$. As above, the differential map between the task and the torque input is expressed through the intermediate variable $\ddot{q}$ by inserting (25) in (14):

$$\ddot{e} + \mu = Q\tau \tag{26}$$

with $\mu = -\dot{J}\dot{q} + JA^{-1}b_c$ and $Q = JA^{-1}P_cS^T$. By inverting (26) and choosing a proper weighted inverse, the obtained formulation is equivalent to the operational-space inverse dynamics developed in [61] (see Appendix C). When inverting (26), it is possible to explicitly handle the redundancy using the inversion template (3). The scheme can be propagated to any levels of hierarchy. The general form of the inverse for the second level of the hierarchy is $J_2P_1A^{-1}P_cS^T$, where P_1 is the projector into the null space of the main task. In general, $rank(J_2P_1A^{-1}P_cS^T) \leq rank(J_2A^{-1}P_cS^T) \leq rank(J_2) \leq r_2$. If the first inequality is strict, this is the *algorithmic* singularity encountered in inverse kinematics. If the last inequality is strict, it is a *kinematic* singularity. If the intermediate inequality is strict, the singularity is due to the dynamic configuration of the multi-body system in contact, and could be called a *dynamic* singularity[7]. As above, a damped inverse is used in practice to avoid the numerical problems in the neighborhood of the singularity.

As before, (26) follows the template (2) and can be directly formulated as a QP. The QP can be expressed under a reduced form as proposed in [2]. Or more simply, the HQP (20) can be reformulated to consider the dynamics in contact. Using the HQP notation, the program for one task is (21) $\prec$ (22) $\prec$ (14). The variables f and $\ddot{q}$ are then explicitly computed: $u = (\tau, \ddot{q}, f)$. This HQP was proved to be equivalent to the reduced inversion in [1].

B. Rigid-point-contact condition

For a single point in rigid contact with a surface, there are two complementary possibilities: either the force along the normal to the contact surface is positive (the robot pushes against the surface and does not move), or the acceleration along the normal is positive (the robot contact point is taking off, and does not exert any force on the surface). Both possibilities are said to be complementary since one and only one of them is fulfilled. This is mathematically written:

$$\ddot{x} \geq 0 \tag{27}$$

$$f^\perp \geq 0 \tag{28}$$

$$\ddot{x}f^\perp = 0 \tag{29}$$

where $f^\perp$ is the component of f corresponding to the normal direction. The complementary condition is a direct expression of d'Alembert-Lagrange Virtual Work principle, in the simple case of rigid contact. By writing (21) and (22), it is implicitly considered that the robot is in the first case: no movement (22) and positive normal force. In consequence, the generated control must also fulfill the second condition (28).

Very often, only the zero-motion condition constraint (22) is considered [35]. As a consequence, an unfeasible dynamic motion can be generated since the second contact condition (28) is not explicitly verified. A first solution can be to saturate the part of the control that does not correspond to gravity compensation when the positivity condition is not satisfied [59]. However, such a solution is very restrictive compared to the motions that the robot can actually perform.

It is straightforward to take into account the two conditions above in a HQP. In that case, the contact forces have to be explicitly computed as one of the QP variables: $u = (\tau, \ddot{q}, f)$. The HQP is then (21) $\prec$ (22) $\prec$ (28) $\prec$ (14).

[4] To be exact, $\ddot{q}$ should be written $\begin{bmatrix}\dot{v}_f \\ \ddot{q}_A\end{bmatrix}$, where v_f is the 6D velocity of the robot root and q_A the position of the actuated joints. For the ease of notation q, $\dot{q}$ and $\ddot{q}$ are used in the article.

[5] The coordinates of x_c, f and J_c have to be expressed in the same frame, for example the one attached to the corresponding robot body

[6] The exact same form can be obtained if J_c is rank deficient [60].

[7] The three cases are similar in the sense that the matrix $\begin{bmatrix}J_1 & 0 & 0 \\ J_2 & 0 & 0 \\ A & J_c & -S^T\end{bmatrix}$ is singular.

The two first levels (21), (22) are always feasible. However, it may happen that (28) is not. This case is sometimes referred to as *strong contact instability* [62]: whatever the motions of the multi-body system are, the contact cannot be maintained. In practice, the solver finds an optimal u, but with nonzero slack variables corresponding to (28). The solution u is then meaningless, since it is dynamically inconsistent. To obtain a consistent control in that case, a change of behavior should be triggered, with the robot removing one of its contacts from (22) and trying to find another solution without this contact. However, the nonzero slack on (28) only appears in extreme cases, for example when the robot is already falling, and in general it is already too late to do anything to restore the balance.

The typical situation with a humanoid robot requires more than one contact point: for example, when one rectangular foot is in contact with the ground, at least four contacts points are needed, with as many force variables and contact constraints. It is then very costly to handle several bodies in contact. In the following, we focus on the case of planar rigid contact, and propose a reduced formulation such that the cost of the HQP does not increase linearly with the number of points in contact.

V. Reduced formulation of rigid planar contacts

Instead of considering one variable per contact force f, the contact forces are summarized by the generalized 6D (spatial) force exerted by the body contacting the environment.

$$A\ddot{q} + b + J_c^\top \phi = S^T \tau \tag{30}$$

where J_c is now the Jacobian of the contacting body expressed on any arbitrary fixed point c of the body, and ϕ is the 6D force (linear and angular components) expressed at c. The contact is supposed to occur between two rigid planar surfaces, one of them being a face of one robot body, the other one belonging to the environment. If the robot is in contact with two or more planar surfaces at the same time, several planar contacts are to be considered. The point c denotes the arbitrary origin of the reference frame attached to the robot body in contact (c can be on the contact surface as before or anywhere on the contact body, e.g. on the last joint). A rigid planar contact is defined by at least three unaligned points of the body p_i, $i = 1..l$ ($l \geq 3$), that define the boundaries of the contact polygon. For $i = 1..l$, f_i denotes the contact force applied to p_i. The vector f of the contact forces f_i is related to ϕ by:

$$\phi = \begin{bmatrix} \sum_i f_i \\ \sum_i p_i \times f_i \end{bmatrix} = X \begin{bmatrix} f_1 \\ \vdots \\ f_l \end{bmatrix} = Xf \tag{31}$$

with

$$X = \begin{bmatrix} I & I & ... & I \\ [p_1]_\times & [p_2]_\times & ... & [p_l]_\times \end{bmatrix}$$

where the first three components of ϕ are the linear part of the force vector, the second three components are the angular part and $[p_i]_\times$ is the cross-product matrix defined by $[p_i]_\times z = p_i \times z$ for any vector z. Using this notation, the necessary and sufficient condition to ensure the contact stability (in the sense that the contact remains in the same phase of the complementary condition, i.e. no take off) is that all the normal components $f_i^\perp$ of the contact forces f_i are positive, expressing the fact that the reaction forces of the surface are directed toward the robot:

$$f^\perp \geq 0 \tag{32}$$

with $f^\perp = S^n f = (f_1^\perp, f_2^\perp, \ldots, f_l^\perp)$ the vector of the normal components of the forces at the contact points and S^n the matrix selecting the normal components.

A. Including the contact forces within the QP Solver

Condition (32) must now be introduced in the HQP proposed at the end of Section IV-B

1) A first way of modeling the problem: The constraints should be written with respect to the optimization variables, while (32) depends on f. A first way of writing (32) with respect to the optimization variables is to use the linear map X between ϕ and f given by (31). In order to compute f, (31) should be inverted by using a particular generalized inverse $X^\#$:

$$f = X^\# \phi \tag{33}$$

The normal component $f^\perp$ is then given by:

$$f^\perp = S^n X^\# \phi = F\phi \tag{34}$$

The condition of positivity of $f^\perp$ is then written with respect to the optimization variables:

$$F\phi \geq 0 \tag{35}$$

The resulting HQP is (30) $\prec$ (22) $\prec$ (35) $\prec$ (14), with the vector of optimization variables being $u = (\ddot{q}, \tau, \phi)$.

However, it is possible to show that (35) is only a sufficient condition of (32), that is too restrictive. In fact, the map X is not invertible. Thus, by choosing a specific inversion $.^\#$, an unnecessary assumption is made, and it may happen that an admissible ϕ produces a negative $f^\perp = S^\perp X^\# \phi$. Fig. 2 displays the domain reached by the center of pressure: for a necessary and sufficient condition, the whole support polygon should be reached. Using the 2-norm, only the included diamond is reached, as presented in Fig. 2. Various included quadrilaterals are reached when using other norms for the inversion operator $^\#$.

2) Using contact forces as variables: The problem is that the forces f_i cannot be uniquely determined from ϕ, while it is possible to determine ϕ from f_i. To cope with this problem we propose to include the contact forces f in the optimization variables of the QP resolution. Condition (32) is then directly written with respect to the variables $u = (\tau, \ddot{q}, \phi, f)$, with the HQP: (30) $\prec$ (22) $\prec$ (31) $\prec$ (32) $\prec$ (14).

Compared to the HQP formulated at the end of Section IV-B, this new formulation considerably reduces the size of J_c, and thus the whole complexity of the resolution scheme. Adding ϕ inside the variables acts as a proxy on the bigger-dimension variable f. The contact forces only appear for the positivity condition (32) and in the relation with ϕ (31). The

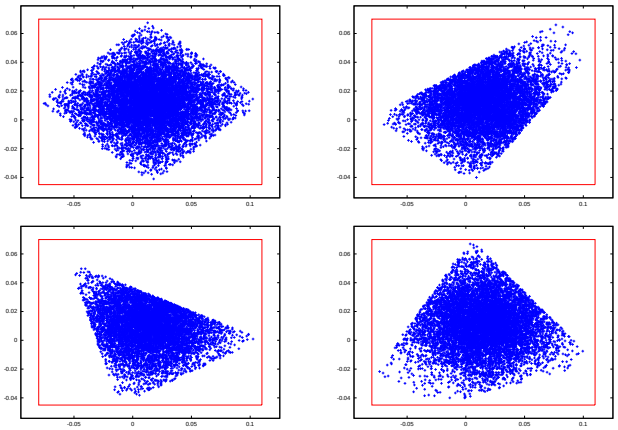


Fig. 2. Random sampling of the reached support region. The actual support polygon is the encompassing rectangle. The point clouds display the ZMP of random forces admissible in the sense of (35). Random forces ϕ are shot and the corresponding $f = X^{\#}\phi$ are computed. If ϕ respects (35), the corresponding center of pressure is drawn. Each sub-figure displays the admissible forces for a different weighted inversion (the Euclidean norm is used on the top left, and random norms for the three others). Only a sub-region of the support polygon can be reached, experimentally illustrating the fact that (35) is a too-restrictive sufficient condition.

HQP is now sparse on the column corresponding to f, which could be optimally exploited only if the solver is sparse. In the following, we rather propose to reduce the formulation while making the constraint matrix dense.

3) Reducing the size of the variable f: It is possible to decouple in (31) the relation between ϕ and the tangent components of f. ϕ was previously expressed at an arbitrary point c of the contact body ($\phi = {}^c\phi$). Consider the point o chosen at the interface of contact (e.g. o is the projection of c on the contact surface). ${}^o\phi$ denotes the 6D forces at o, expressed in terms of ${}^o\phi$ as follows:

$$ {}^o\phi = \begin{bmatrix} f_o \\ \tau_o \end{bmatrix} = \begin{bmatrix} I_3 & 0_3 \\ [{}^oc]_\times & I_3 \end{bmatrix} {}^c\phi = {}^oX_c \ {}^c\phi \tag{36} $$

with ${}^ox$ the coordinates of any quantity x in the frame $\mathcal{F}_o$ centered at o, having its z-axis normal to the contact surface. From (31) and (36), it comes:

$$ \left\{ \begin{array}{lll} {}^of^x = & \sum_i f_i^x & = {}^cf^x \\ {}^of^y = & \sum_i f_i^y & = {}^cf^y \\ {}^of^z = & \sum_i f_i^z & = {}^cf^z \\ {}^o\tau^x = & \sum_i -{}^op_i^z f_i^y + \sum_i {}^op_i^y f_i^z & = -c^z{}^cf^y + {}^c\tau^x \\ {}^o\tau^y = & \sum_i -{}^op_i^x f_i^z + \sum_i {}^op_i^z f_i^x & = c^z{}^cf^x + {}^c\tau^y \\ {}^o\tau^z = & \sum_i -{}^op_i^y f_i^x + \sum_i {}^op_i^x f_i^y & = {}^c\tau^z \end{array} \right. \tag{37} $$

Since o is coplanar with the p_i, the ${}^op_i^z$ are null. The previous expression reveals a decoupling in ${}^c\phi$: the forces ${}^of^{x,y}$ and the torque ${}^o\tau^z$ are expressed in terms of $f_i^{x,y}$. The force ${}^of^z$ and the torques ${}^o\tau^{x,y}$ are a function of f_i^z. In the QP, ${}^of^{x,y}$ and ${}^o\tau^z$ are unconstrained and can be removed along with the associated constraints (37.1), (37.2) and (37.6). The reduced rigid-contact constraint can be expressed as follows:

$$ \begin{array}{c} Q_c \begin{bmatrix} \phi \\ f^\perp \end{bmatrix} = 0 \\ f^\perp \geq 0 \end{array} \tag{38} $$

with

$$ Q_c = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & \dots & 1 \\ 0 & c^z & 0 & -1 & 0 & 0 & p_1^y & p_2^y & \dots & p_l^y \\ -c^z & 0 & 0 & 0 & -1 & 0 & -p_1^x & -p_2^x & \dots & -p_l^x \end{bmatrix} $$

The HQP is then (30) $\prec$ (22) $\prec$ (38) $\prec$ (14) with the optimization variables: $u = (\tau, \ddot{q}, \phi, f^\perp)$.

B. Generalization to multiple contacts

Eq. (30) considers one single body in contact. If several bodies are in contact or one body is in contact with several planes, a force ϕ_i is introduced for each couple plane-body in contact:

$$ A\ddot{q} + b + \sum_i J_i^\top \phi_i = S^T\tau \tag{39} $$

For each body in contact, the same reasoning can be applied separately. Support polygons and normal forces $f_i^\perp$ have to be introduced. For each contact, $f_i^\perp$ is constrained to be positive and can be mapped to ϕ_i using (36). The zero-motion constraint corresponding to contact i is denoted by (22.i) and the positivity constraint by (38.i), where i refers to the index.

C. Multiple tasks and final norm

Similarly for several tasks, (14.j) denotes the constraint for each task j (using the same notation where j refers to the index). After adding all the tasks, some DOF may remain unconstrained. In that case, it is desirable to comply with Gauss' principle. This is possible by imposing $\ddot{q} = a_0$ as the least priority, where a_0 is the acceleration of the unconstrained system[8]. This has strictly the same effect as weighting all the pseudoinverses by A^{-1}, as done in (17) [56]. However no damping mechanism acts in the corresponding DOF that would reduce the motion energy and stabilize the system. The task-function formalism requires the system to be fully constrained to ensure its stability in terms of automatic control (Lyapunov stability, [13]). On a physical robot, damping is always present. For perfect systems like simulations, where damping is absent or is perfectly compensated, it is better to introduce, at the very last level, a task to cope with the case of an insufficient number of tasks and constraints to fulfill the full-rank condition:

$$ \ddot{q} = -K\dot{q} \tag{40} $$

Various full-rank constraints could have been considered (minimum acceleration, distance to a reference posture, etc). The choice of using the minimum velocity constraint is arbitrary.

Finally, the complete HQP for n contacts and k tasks is written: (39) $\prec$ (22.1) $\prec$ (38.1) $\prec$ ... $\prec$ (22.n) $\prec$ (38.n) $\prec$ (14.1) $\prec$ ... $\prec$ (14.k) $\prec$ (40), with the optimization variables $u = (\tau, \ddot{q}, \phi_1, f_1^\perp, ..., \phi_n, f_n^\perp)$.

D. Opening to other classes of contacts

The model (22)-(38) is built on the rigid point contact. From the basic point model, many other variations can be built. In particular, it is straightforward to obtain edge contact. Elastic contact can be defined by modifying the equation

[8]Similarly, the constraint can be imposed on a least-square τ.

of motion (22) [40]. Linearized friction cones can also be considered, by replacing $J_c^\top f$ by $J_c^\top G\lambda$ and $f^\perp \geq 0$ by $\lambda \geq 0$, where G is a family of generators of the linearized cone, and λ are the multipliers of these generators [37], [38]. Motions with slips are made possible by removing the motion constraint (22) in the tangent directions, and setting a constraint on the tangent force to be outside the friction cone.

However, the limitation in the viewpoint of real-time control is the size of the obtained QP formulation: typically, a good cone approximation is obtained with twelve generators, which introduce twelve new variables per point of contact. The prospectives of this work for humanoid robot control are to find reduced formulations to handle these situations. In the remaining of the paper, the reduced rigid planar formulation is used, since it keeps a relatively low computational cost while covering many possible situations with the humanoid robot.

VI. Control law robustness

A. Comparison of (38) with the ZMP condition

A classical situation is to have one or two feet of the humanoid robot in contact with a flat horizontal floor. In this case, a classical condition to enforce the contact stability is to check that the ZMP stays inside the support polygon [63], [64]. In this section, this condition is proved to be equivalent to (32).

Proposition VI.1. *In the case of contact with a horizontal floor, the rigid contact condition* (32) *is equivalent to the well-known contact stability condition which requires that the ZMP belongs to the support polygon.*

1) Sufficient implication: As above, the robot is supposed to be in single support[9]. The contact surface is supposed to be horizontal. The ZMP (also called center of pressure − COP [65]) can be defined as the barycenter of the contact points p_i delimiting the contact surface of the foot with a horizontal floor, weighted by the normal component $f_i^\perp$ of the contact forces f_i at these points[10]:

$$z = \frac{1}{\Sigma_i f_i^\perp} \Sigma_i p_i f_i^\perp \tag{41}$$

In affine geometry, it is well known that the convex hull of a polygon can be written as the set of all positive-weight barycenters of the vertices [66]. The rigid contact condition defined by (32) ensures that each $f_i^\perp$ is positive. Consequently, (32) together with (41) ensures that the ZMP belongs to the convex hull of the contact points p_i which, by definition, is equal to the support polygon.

2) Necessary implication: On the other hand, if the ZMP belongs to the support polygon, there always exists a distribution of contact forces f_i at the points p_i, having positive components $f_i^\perp$, and such that the ZMP is the barycenter of the p_i weighted by the $f_i^\perp$. This is sometimes referred to as *weak contact* stability [62] for which the ZMP is known to be a reduced condition [67]. When the support polygon is defined by more than three contact points ($l > 3$), an infinite number of possible barycenter weights $f_i^\perp$ can be found to define the ZMP. For given weights, one of the $f_i^\perp$ can be negative (this is typically what happens in Fig. 2). However, since the ZMP is inside the convex hull, there is at least one combination of non-negative weights that reaches it. □

[9]The same reasoning holds with several bodies in contact with the same horizontal plane.

[10]The foot is usually a rectangle but any shape delimited by three or more contact points can be considered as well.

B. Brief stability analysis

The inverse dynamics schemes are known to be sensitive to modeling errors [68]. In particular, if the inertia parameters are not perfectly known, the application of the reference torques leads to different accelerations. The estimated value of X is denoted by $\widehat{X}$. The solution of the QP is equivalent to the solution given by the pseudoinverse if none of the positive-force constraints are active; otherwise, it has a similar form with an additional projection and can be written for one task:

$$\tau = (\widehat{J}\widehat{A}^{-1}\widehat{P}_f S^T)^+ (\ddot{e}^* + \widehat{\mu}_) \tag{42}$$

where P_f is the projection operator onto the contact zero-motion constraint (22) and onto the set of contact positive-force constraints (38) that are active. Using (26), the observed task acceleration when applying this control law, also denoted by $\hat{.}$, is:

$$\widehat{\ddot{e}} = JA^{-1}P_c S^T (\widehat{J}\widehat{A}^{-1}\widehat{P}_f S^T)^+ (\ddot{e}^* + \mu) - \mu \tag{43}$$

Since $P_c P_f = P_f$, $\widehat{\ddot{e}} = \ddot{e}^*$ if all the estimations are perfect. If the estimations are biased, applying the control (42) in closed-loop at the whole-body level is known to keep the stability properties of the control law $\ddot{e}^*$ in the task space *iff* $JA^{-1}P_c S^T (\widehat{J}\widehat{A}^{-1}\widehat{P}_f S^T)^+$ is definite positive [13]. When the estimation error is due to an inaccurate dynamic model, a classical solution to reduce the estimation error is to rely on a time-delay estimation, i.e reporting the biases observed at one iteration of the control on the next iteration [69]. However, this technique cannot perfectly cancel the errors of estimation, thus (43) still holds.

The reference $\ddot{e}^*$ is not perfectly tracked. It is also true for the contact forces computed by the solver. Indeed, the observed forces are:

$$\hat{\phi} = ({J_c}^T)^{\#A} \widehat{J}_c^T \phi^* + ({J_c}A^{-1}{J_c}^T)^{-1} J_c A^{-1} \widehat{A}\ddot{q}^* \tag{44}$$

where ϕ^* and $\ddot{q}^*$ are the reference force and acceleration computed by the HQP and $\dot{J}_c\dot{q}$ is neglected. The second term is close to 0 when $\widehat{A}$ is not too far from A. Similarly, the first term is nearly the identity matrix when the estimation is correct. The previous equation can be summarized by:

$$\hat{\phi} = (I + \epsilon_1)\phi^* + \epsilon_2 \ddot{q}^* \tag{45}$$

with ϵ_1 and ϵ_2 two matrices that tend to zero when the estimation tends to perfection. When the ϵ_i are not null, the observed force $\hat{\phi}$ is biased with respect to the solver predictions. If the bias is too great, there is no guarantee that the observed force $\hat{\phi}$ maintains the contact; then the property of stability can be lost.

In conclusion, applying the computed torques in closed-loop ensures the stability of the control as long as the observed forces respect the contact positivity-force constraint.

C. The contact condition as a qualitative robustness indicator

The previous stability analysis is not very instructive in practice, since it is barely possible to predict when the observed $\hat{\phi}$ keeps the contact stability. The robustness of the control scheme thus relies on the behavior of $\hat{\phi}$. It is interesting to provide an indicator of how easy it is for $\hat{\phi}$ to leave the acceptable domain. When considering one single point in contact as in Section IV, this indicator is straightforward to choose: consider the normal force value $f^{\perp *}$ computed by the solver. If $f^{\perp *}$ is large, then for small ϵ_1, ϵ_2, we can be very confident that $\hat{f}^{\perp}$ is positive and keeps the contact stable. Then, for one single contact point, the positivity of $f^{\perp *}$ is a good indicator of the robustness of the control.

For more than one single contact point, it is not possible to use a direct combination of the normal forces as an indicator. Indeed there is an infinite number of possible force values, all of them being equivalent in terms of the robot behavior. Once more, this is connected to the results displayed in Fig. 2: the computed solution may include one zero normal force, while another solution exists with strictly positive values. When considering a single planar contact, the ZMP is a good indicator of robustness: when the predicted ZMP z^* is far inside the support polygon, then we can be very confident that the observed ZMP $\hat{z}$ stays inside the support polygon, which means in return that all the $\hat{f}^{\perp}$ are positive.

If the contacts are not coplanar, the ZMP is not defined. In that case, the generalized zero-moment point (GZMP) [70] has been proposed. Contrary to the ZMP or to (38), the GZMP is not a constructive criterion, i.e. it has not been used to generate a motion or a control law. The idea of the GZMP is to find from the 3D contact points a plane that acts like the floor plane for the ZMP. On this plane, all the force boundaries are projected, defining a 2D polygon. The GZMP exists in this same plane. The contact-stability criterion says that the GZMP should remain inside the 2D polygon. The GZMP is easy to display. It is easy to visualize the distance to the boundaries and thus to have a qualitative evaluation of the motion robustness with respect to the contact stability. The GZMP needs some implementation work in order to be calculated, since the 2D projection plane is deduced from geometrical computations. Moreover, it is only an approximated criterion, since the friction forces are neglected. To cope with these limitations and obtain a generative criterion, the GZMP was augmented in [67]. However, this last criterion, like (38), cannot be easily plotted, and is thus not relevant to judge the robustness of the obtained motion.

Consider the six first rows of the dynamic equation. The dependency on τ disappears:

$$\bar{A}\ddot{q} + \bar{b} = \bar{J}_c f \tag{46}$$

where $\bar{A}$, $\bar{b}$ and $\bar{J}_c$ are the first six rows of respectively A, b and ${J_c}^T$. For a given $\ddot{q}^*$, the left term is constant, denoted by ψ^*. It corresponds to the actuation of the free-floating body that cannot be accomplished by the motors. The variable f can be partitioned in two parts $f = (f_\heartsuit, f_\spadesuit)$: $f_\heartsuit$ is unconstrained, while $f_\spadesuit$ is subject to the positivity constraint. $\bar{J}_c$ is similarly partitioned into $\bar{J}_\heartsuit$ and $\bar{J}_\spadesuit$. The set $K_\spadesuit := \{\psi = \bar{J}_\spadesuit f, f > 0\}$ is a 6D cone, that can be expressed by its facets. The motion is robust to the parameter error if the point $\psi^* - \bar{J}_\heartsuit f_\heartsuit = (I - \bar{J}_\heartsuit {\bar{J}_\heartsuit}^+)\psi^*$ is deep inside the cone. The distance from this point to the closest facet of $K_\spadesuit$ can be used as a measure of the robustness of the motion. The scaling between torques and forces is done using a characteristic length of the system (1m for a human-size robot). In the following, this criterion is referred to as robustness criterion VI-C.

VII. Experiments

Three sets of experiments are presented in this section. The first one presents a simple oscillatory motion that illustrates the saturation of the contact-stability constraints. The second one presents a complex sequence of tasks to make the robot sit in an armchair using several successive contacts. This motion is also executed by the real robot. The last experiment presents a dynamic transition of contacts. First, the setup is detailed.

A. Experimental setup

The inverse formulation of the dynamic equation of motion (30) is given to the HQP solver. However, since it computes explicitly both τ and $\ddot{q}$, it solves simultaneously the forward and inverse dynamics of the robot. Both values can then be used as control input. The acceleration $\ddot{q}$ can be integrated in simulation, or provided as control input to the robot servo control; or the torques can be given as the robot control, or provided to a dynamics simulator. On current humanoid robots, such as HRP-2, only the first solution is possible[11]. However, this solution has the drawback that the servo is on the position variables, while, as explained in the previous section, the robustness mainly relies on the accuracy of the force variables. In simulation, both solutions are possible. The second solution is more beneficial, since it makes it possible to double-check the dynamic computations.

In practice, we have used this last solution. The dynamic simulator AMELIF [73] was used to resolve the forward dynamics from the computed torques τ^*. The simulator checks the collision, computes the acceleration from the collision set and the torque input using a linear solver and numerically integrates $\ddot{q}$ using a classical Runge-Kutta of the fourth order. The current set of contacts is then provided to the control solver, along with the current position and velocity of the robot. The control is updated every $1ms$. It is computed using the control framework SOT [33] and the dedicated solver [26]. The result of this simulation is a joint trajectory of the robot, that complies to the multi-body dynamics. This trajectory is replayed on the real robot using a position-control mode.

The task set used in the three presented motions is the following. A first task function is used to control the position and orientation of one operational point of the robot (e.g. grippers, head, chest). The task error is the position p and angle-vector orientation $r\theta$ [74] of the operational point with respect to a reference p^*, $r\theta^*$ expressed in the world frame:

$$e_{op} = \begin{bmatrix} p - p^* \\ r\theta \ominus u\theta^* \end{bmatrix} \tag{47}$$

[11]The second solution will be possible with the next humanoid robot generation, e.g. Romeo [71] or DLR [72].

The reference acceleration is computed from this error as a proportional-derivative (PD) control law:

$$\ddot{e}^{\star}_{op} = -\lambda_p e_{op} - \lambda_d \dot{e}_{op} \tag{48}$$

where $\dot{e}_{op} = J_{op}\dot{q}$ is the velocity in the task space and the gains λ_p, λ_d are used to tune the convergence velocity (usually, $\lambda_d = 2\sqrt{\lambda_p}$). For tracking a moving target, a fixed high gain is used for λ_p. When reaching a fixed target, an adaptive gain is typically used:

$$\lambda_p : ||e|| \rightarrow (\lambda_0 - \lambda_\infty)e^{\beta||e||} + \lambda_\infty \tag{49}$$

where λ_0 is the gain when the error is null, λ_∞ is the gain far from the target, and β adjusts the switching behavior between the gains. A typical setting is $(\lambda_0, \lambda_\infty, \beta) = (450, 15, 100)$. A second task e_{gaze} is used to servo the projection s of one point of the environment on the right camera plane to a reference position s^* [14]:

$$e_{gaze} = s - s^* \tag{50}$$

The reference acceleration $\ddot{e}^*$ is also defined by (48). The torque magnitude is also bounded. Since the torques are included in the vector of optimization variables, it is trivial to express the torque limits by a simple bound on these variables:

$$\underline{\tau} \leq \tau \leq \overline{\tau} \tag{51}$$

with $\overline{\tau} = -\underline{\tau}$ the maximum torque value.

Similarly, bounds have to be set on the joint positions. Since the positions are not variables of the solver, the constraint is set on the joint accelerations:

$$\underline{q} \leq q + T_S\dot{q} + \frac{{T_S}^2}{2}\ddot{q} \leq \overline{q} \tag{52}$$

where $\underline{q}$ and $\overline{q}$ denote the lower and upper joint-limits respectively, and T_S is the length of the preview windows. In theory, the control sampling time $\Delta T = 1ms$ should be used for T_S. In practice, a smoother behavior can be obtained by adjusting this value $T_S := \frac{\Delta T}{\lambda_s}$ where λ_s can be tuned as the gain of the task. We used $\lambda_s = 0.1$ to generate the following motions.

B. Experiment A: Swing posture

1) Description: The objective of this experiment is to validate the contact stability constraint. It is inspired by a biomechanics experiment which aims at testing the human swinging posture behavior with respect to the same constraints [75]. A tracking task is imposed to the robot head to make it oscillate. Depending on the frequency and the amplitude of the oscillation, forces are obtained at the contact points, that may saturate the contact constraint. The task e_{head} given by (47) is imposed to the head operational point, where only the translation on the forward axis is selected. The reference position is given by a time-varying sinusoid, around a central point $x_c = 0.02$ and with amplitude of $5cm$ and frequency $0.3Hz$ (low frequency), $0.56Hz$ (medium frequency) or $0.9Hz$ (high frequency). The gain is set to $\lambda_p := 250$ to ensure good tracking. The complete SOT is: (39) $\prec$ (22) $\prec$ (38) $\prec$ (51) $\prec$ (52) $\prec$ e_{head} $\prec$ (40).

In theory, the contact points are defined from the 3D model of the robot. However, in practice, we never consider the real support polygon, but a smaller one. This simple trick ensures increased robustness of the motion when trying to replay it on the robot. For example, on the feet, the support polygon is often defined as a square of 4cm centered below the ankle axis [76], [77]. The obtained robustness can be evaluated afterward with respect to the real support polygon.

The motion is played four times. In the first two executions, both feet are flat on the ground and the reference is oscillating at low and medium frequencies respectively. For the next two executions, the right gripper contact is added and the motion is played at medium and high frequencies. In the following, the four motions are referred to as *2pt-low*, *2pt-medium*, *3pt-medium* and *3pt-high* respectively.

2) Results: The experiment is summed up by Figures 3 to 6. The motion is displayed in Fig. 3. The robot is oscillating forward and backward to follow the head reference. The two motions *2pt-low* and *2pt-medium* were already detailed in [1] where the plots of joint positions and torques can be found. When only the feet are contacting, the stability of the motion can be evaluated by displaying the ZMP, plotted in Fig. 4. At low frequency, the ZMP does not saturate because the demanded accelerations are small enough. At medium frequency, the accelerations are larger and the ZMP saturates. Since the real support polygon is about 20cm wide, there is a large offset that ensures a good robustness when executing this motion on the real robot.

The robustness can be evaluated using the criterion proposed in Section VI-C. The contact constraints of the solver are projected into the space of the spatial forces expressed at the waist point. Then the distance of the point ψ^* (46) to this constraint set is computed. The result is plotted in Fig. 5. First, the distance is computed to the constraint set of the solver (the $4cm$-wide support polygon). As expected, the distance is null when the ZMP saturates. More interestingly, the distance can be computed to the real constraints by taking into account the true polygon as well as the linearized friction cones at the contact points. The friction coefficient was set to $K = 0.5$. In that case, the robustness criterion is always strictly positive, showing that the motion is robust to small perturbations or model uncertainties.

Using only the feet as contacts, it is not possible to follow the reference at high velocity. A third contact point is added to increase the stability domain. The contact polygon is a square of $5cm$ centered at the gripper terminal point. Contrary to the ZMP, the robustness criterion VI-C is still valid with non-coplanar contacts. When the friction cones are not considered (slidingless contact), it is always possible to find a set of contact forces following a given CoM acceleration (the system is said to be in *force closure* [78]). In that case, the distance to the constraint set is always infinite. The robustness criterion is finite when the friction cones are considered. The friction coefficient at the gripper is set to $K = 0.1$. At medium frequency, the motion can be considered as very robust since the criterion is always very far from 0. If the frequency is increasing, the criterion remains smaller. It then jumps from one constraint edge to another, which explains the discontinuities. The computation time depends on the number of contacts, tasks and active constraints as shown in Fig. 6.

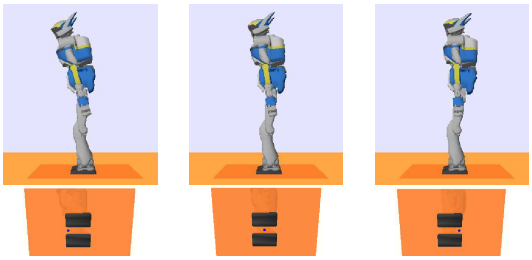


Fig. 3. Experiment A: Top: Snapshots of the oscillatory movement *2pt-medium*. Bottom: Feet and ZMP positions at the corresponding instants. The ZMP saturates on the front when the robot is reaching its top amplitude and decelerates to go backward. Similarly, the ZMP saturates on the back.

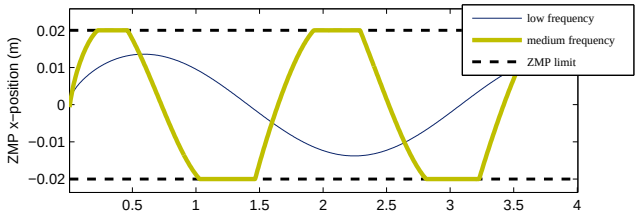


Fig. 4. Experiment A: ZMP position along the forward (x) axis for the two motions with only the feet contacts. The support polygon is a $4cm$-wide square centered on the ankle joint. The ZMP does not saturate when the motion oscillates at low frequency. It saturates at medium frequency.

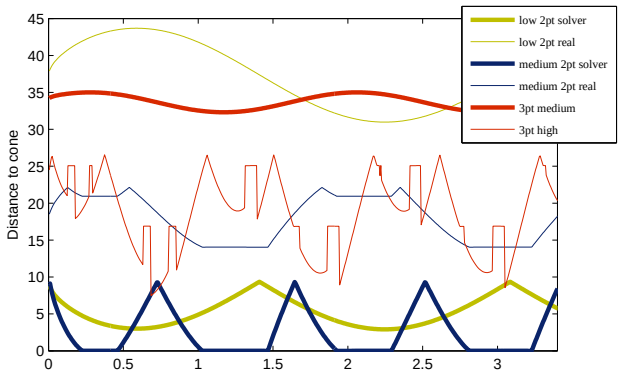


Fig. 5. Experiment A: robustness criterion VI-C. For the first two motions *2pt-low* and *2pt-medium*, the criterion is given with respect to the support polygon defined in the solver (small contact surface) in bold, and with respect to the real support polygon taking into account the friction cone (linearized by twelve facets) in nonbold. This criterion behaves similarly to the distance of the ZMP to the support polygon. The criterion is plotted for *3pt-medium* and *3pt-high*. If the solver support polygons are considered, the distance is infinite. It is only plotted for the distance to the friction cone.

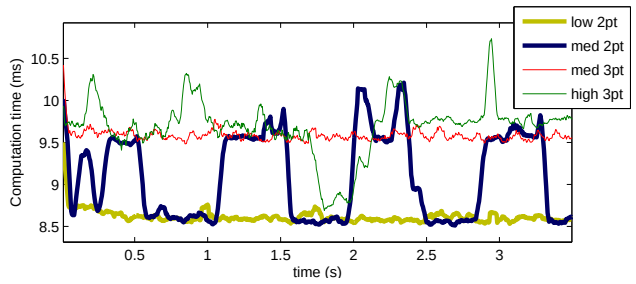


Fig. 6. Experiment A: Computation time. For the motion *2pt-medium*, the saturation of the force constraints clearly induces an increase of the computation cost, whereas for *2pt-low* the cost remains constant. For *3pt-medium*, the cost is constant (no saturation) but higher in average due to the additional contact. Finally, the cost of *3pt-high* is higher and varies when the constraints are saturated.

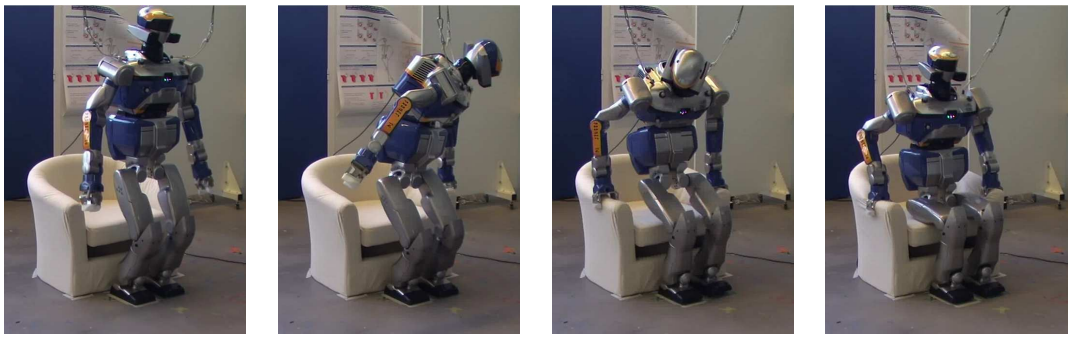
C. Experiment B: sitting in the armchair

1) Description: The second experiment illustrates the possibilities of multiple non-coplanar contacts during a more complex sequence of motion. The robot sits in an armchair (see Fig. 7). First, contacts of the left then right grippers are found with the armrests to increase the contact stability domain. Then, the pelvis is brought in contact with the seat.

At the highest priority of the stack, the limits (51) and (52) ensure that the joints and actuator limits are respected. Two tasks e_{rh} and e_{lh}, defined by (47), are set on each robot gripper to control the position and orientation toward the corresponding armrest. To prevent a collision when grasping, an intermediate point is first reached, above the grasping position. The contact of each gripper with the armrest is realized by the rear part of the opened gripper. The support polygon is then a $5cm$-wide square. To improve the naturalness of the motion, a task e_{gaze} defined by (50) is set to constrain the gaze toward the armrest to be grasped. After each grasp, the gaze is brought back in front of the robot. Finally, the waist is controlled by a task e_{waist} also defined by (47) where only the vertical position and sagittal rotation are active: the waist is constrained to remain vertical and to move down to the seat. The complete SOT is defined by: (39) $\prec$ (22) $\prec$ (38) $\prec$ (51) $\prec$ (52) $\prec$ e_{hand} $\prec$ e_{gaze} $\prec$ e_{waist} $\prec$ (40), with e_{hand} being the right or left hand task, when active. The temporal sequence of tasks is given in Fig. 8. Essentially, the robot looks left and bends to grasp the left handle; then it looks right and bends to grasp the right handle; finally, using both handle supports, it moves the pelvis down to sit.

2) Results: The experiment is summarized in Figures 7 to 13. The key frames of the motion executed by the robot are given in Fig. 7. The sequence of tasks is summarized in Fig. 8. On each of the following figures, the chronological sequence is recalled by vertical stems at the transition instants. During the motion, the joint range is extensively used. The most representative joint trajectories are plotted in Fig. 9. The neck joint reaches its limit while looking left. In reaction, all the other aligned joints move to overrun the neck limitation (chest joint of course, but also hip and ankle joints). The right hip then reaches its limit. In consequence, all the motions of both legs are stopped, due to a lack of DOF to compensate this limit. The chest joint absorbs all the subsequent overrun to fulfill the task. Again, the neck joint reaches its limit when looking right. This time, the velocity of the joint when it reaches its limit is higher, which leads to a strong acceleration of the chest, and consequently brings the neck out of its limit. This behavior could be damped if necessary by tuning λ_s in (52). The chest joint finally reaches its limit at the end of the right-grasp task, which produces a limited overrun on the other joints. All the joints are properly stopped at the limit, and can leave the neighborhood of the limit without being stuck as it may appear with some avoidance techniques.

The contact with the two armrests is very useful to control the descent of the waist. The vertical forces on each support are plotted in Fig. 10. In the beginning, the weight is fully supported by the two feet, as shown on Fig. 11. After $t = 8s$, the left arm is used to sustain the robot. However, the robot



t=0s t=7s t=15s t=19s

Fig. 7. Experiment B: Snapshots of the motion executed on the real HRP-2 robot. The robot is standing on both feet ($t = 0s$). It first looks left and grasps the left armrest $t = 7s$. It then looks right, grasps the right armrest ($t = 15s$) and finally sits ($t = 19s$).

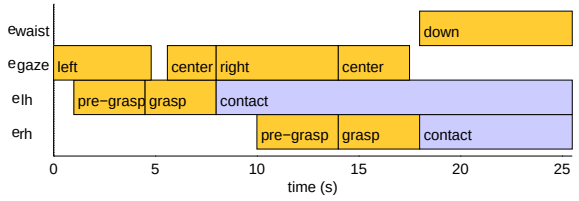


Fig. 8. Experiment B: Sequence of tasks and contacts. The gaze task focuses sequentially on the left and right armrests and on a virtual point in front of the robot. The pre-grasp tasks are set at the vertical $10cm$ above the grasp position.

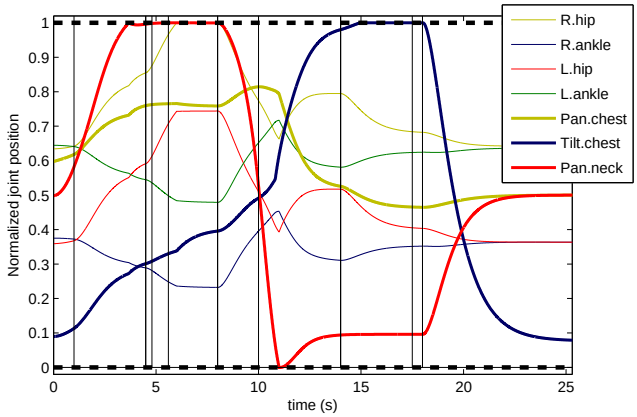


Fig. 9. Experiment B: normalized joint position (0 and 1 are resp. the lower and upper limits) of the right and left hip and ankle, chest and neck joints. The joint limits are properly avoided. When a limit is reached, one or several joints move in reaction to overcome the saturation.

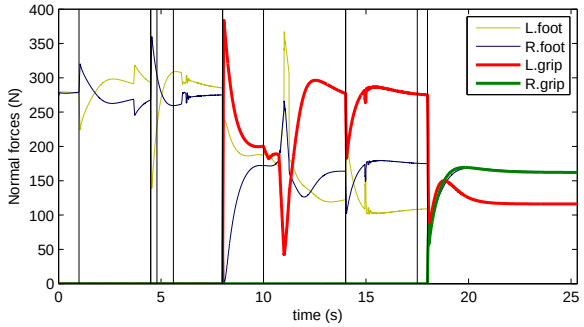


Fig. 10. Experiment B: Vertical forces distribution.

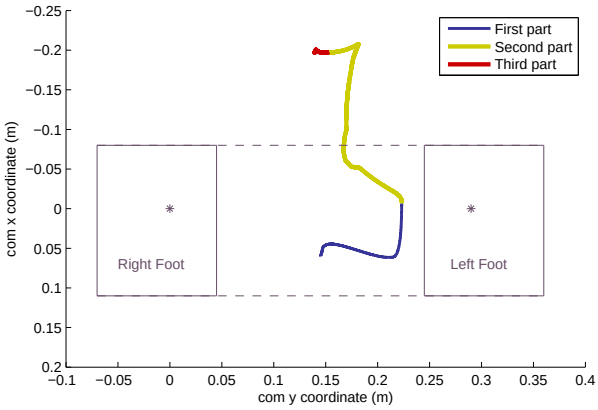


Fig. 11. Experiment B: Position of COM. The three phases correspond to changes in the number of contacts (first the two feet, then the left gripper and finally both feet and grippers). Firstly, the COM stays forward, but is finally moved backward to reach the second armrest and move the pelvis down to the seat.

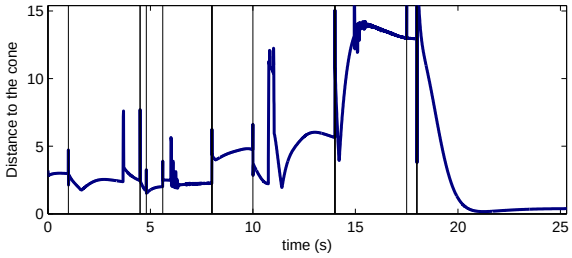


Fig. 12. Experiment B: Robustness criterion VI-C. The distance is computed with respect to the friction cones. The friction coefficient at the armrests is roughly estimated to be 5 times less than at the sole. The less robust part occurs during the final phase, where the waist moves down.

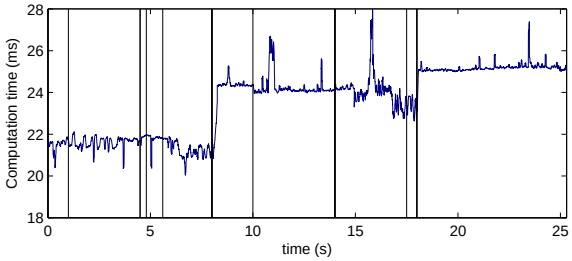


Fig. 13. Experiment B: Computation time.

upper body is still in front of the chair, and this contact is not fully used yet. In order to reach the second armrest, the robot has to move its weight back (see Fig. 11) and use the left-arm contact to ensure its balance: nearly half of the weight is then supported by the arm. Finally, the right armrest is grasped, and the robot distributes its weight on the four contacts equally.

Neither the center of mass (COM) nor the ZMP can give a proper estimation of the stability, since the motion is neither quasi-static nor supported by planar contacts. The robustness estimator presented in Section VI-C is plotted in Fig. 12 with respect to the linearized friction cones at both feet and grippers. The motion is very stable, except at the end of the motion, when the waist moves down. At that time, the robot is using the tangent forces of the grippers on the armrest, which nearly saturates the friction cone. In consequence, this part of the motion is the less robust when executed by the real robot. Indeed, since the armrests do not respect the hypothesis of rigid contact and due to this lack of robustness, it can be observed that the toes nearly leave the ground during this phase of the motion. This effect is very interesting, since it confirms the relevance of the robustness criterion. Of course, this undesirable effect could be avoided by setting a more accurate model of the environment or adding a safety limit to the positivity constraint in the solver.

Finally, the computation times are plotted in Fig. 13. The SOT is nearly full. In that case, the computation cost is around 20ms per iteration, i.e. five times the real-time if controlling the robot at $200Hz$. The computation cost depends on the number of tasks and even more on the number of contacts, as shown by the computation increase at $t = 8s$ and $t = 18s$.

D. Experiment C: Dynamic contact transition

1) Description: At the beginning of the motion, the robot is standing on both feet and its COM is artificially pushed forward using a task on its chest. The robot is then out of its domain of quasi-static stability: the only solution to restore the balance is to change the set of supports. The two grippers (first the left, then the right) are then sent forward to establish a contact with the wall, in order to increase the set of support contacts and to restore the balance. An overview of the motion is given in Fig. 14. Three tasks of type (47) are used: one task on the chest, that controls only the translation; another one on each gripper controls both the translation and the rotation. The COM is not explicitly controlled. The sequence of tasks and contacts is given in Fig. 15.

2) Results: The experiment is summarized in Fig. 14 to 18. If using only quasi-static movements (i.e. reaching while keeping the COM inside the feet support polygon), the maximal reaching distance of HRP-2 is around $85cm$. In this motion, the wall is positioned $1m$ in front of the robot, as shown in Fig. 14. The motions of the COM along with the forward direction are plotted in Fig. 16. The COM quickly leaves the support polygon in the beginning of the motion, due to the artificial motion of the chest. From $t = 0.7s$, the COM is out of the support polygon with a positive velocity: it is then impossible to bring it back to stability without changing the supports. The balance is restored after $t = 2.5s$, with the COM coming back to zero velocity. The stability is evaluated using the robustness criterion presented in Section VI-C. When only the feet are in contact, the ZMP is at the forward limit of the support polygon, which corresponds to a low robustness. The robustness increases when the first gripper enters into contact. However, at that time, the tangent forces of the gripper on the wall are high. The robot can then lose its balance by rotating on one of the gripper-foot edges, as already observed in [70]. The second gripper helps to improve the stability by decreasing the tangent forces at each contact point. The vertical forces are plotted in Fig. 18. On the grippers, the vertical direction corresponds to the tangent to the contact. Between $t = 1.9s$ and $t = 2.5s$, the tangent forces at the left gripper are high, at the limit of the friction cones, which corresponds to a weaker robustness of the motion (the gripper is close to slide).

VIII. Conclusion

This paper proposes a complete solution to perform task-space (operational-space) inverse dynamics while taking into account various tasks, unilateral constraints such as joint position or torque limits and preserving the contact stability. Complex motions can be composed from several tasks, constraints and contacts, by ensuring a strict hierarchy between conflicting references. Several models of unilateral contacts can be considered. The most usual one is the rigid point contact. We have also proposed a reduced formulation to express rigid planar contacts. The contact condition has been shown to be equivalent to the *ZMP-inside-the-support-polygon* constraint in the particular case of the humanoid robot standing on a flat floor. To quantify the quality of the generated motion in terms of distance to the contact-stability constraints, a generic criterion has been proposed, that can handle the rigid slidingless point contact, the rigid planar contact, but also friction cones.

The effectiveness of the approach has been demonstrated by generating different motions for the humanoid HRP-2. These motions have been generated off-line because the motion-generation algorithm is close to but still not real-time. They are fully consistent with the robot dynamics and can be replayed directly by the robot, as it was shown by making the real HRP-2 sit down in an armchair.

The future of this approach would be to apply the algorithm directly on the robot as a closed-loop control. This would require technical contributions to accelerate the solver computation cost, but also to consider an effective dynamic sensor-based control.

Appendix A
Generalized inverse

The notation $Q^{\#}$ denotes any reflexive generalized inverse of Q [46], i.e. that respects the two first conditions of Moore-Penrose:

$$QQ^{\#}Q = Q$$
$$Q^{\#}QQ^{\#} = Q^{\#}$$

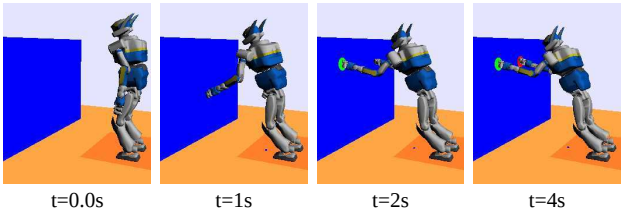


Fig. 14. Experiment C: Snapshots of the dynamic contact transition.

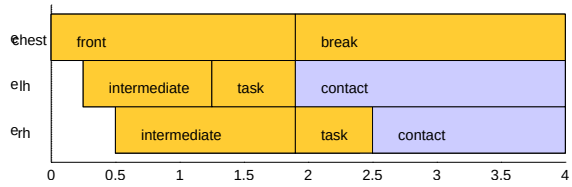


Fig. 15. Experiment C: Sequence of tasks and contacts. On each gripper, an intermediate point is used to ensure that the final contact motion is performed along the normal to the wall. The contact polygons of the feet and grippers are the same as previously.

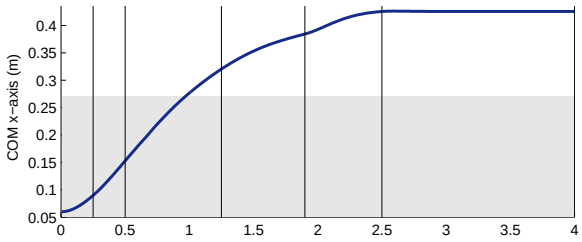


Fig. 16. Experiment C: Trajectory of the COM along the X-axis (forward direction). The grey rectangle marks the limit of the foot support. The COM starts inside the support polygon, quickly leaves it when the chest is thrown forward and finally converges to a fixed position when the grippers contact the wall and stabilize the motion.

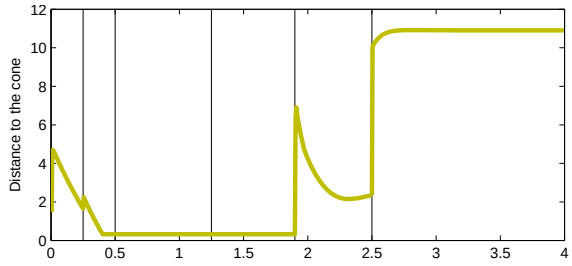


Fig. 17. Experiment C: Robustness criterion VI-C. The distance is computed with respect to the friction cones. The friction coefficient of the gripper with the wall is set to the same value than at the sole contact.

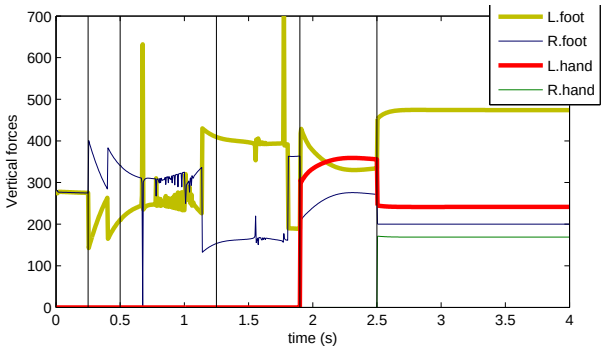


Fig. 18. Experiment C: Vertical forces (normal forces for the feet, tangent forces for the gripper).

In general, $Q^{\#}$ is chosen among the possible inverses as the one which minimizes the norm in both the task space and the control parameter (referred to as the pseudoinverse in the paper, and denoted by $.^{+}$); i.e. it verifies the two second conditions of Moore-Penrose:

$$QQ^{\#} \text{ is symmetrical.}$$

$$Q^{\#}Q \text{ is symmetrical.}$$

Alternatively, one of these two (or both) conditions can be relaxed to impose a different metric in the task space or on the control parameter. In particular, a weighted generalized inverse [45] can be chosen to impose a given minimum R-norm in the control space $||u||_R^2 = u^T R u$, where R is a given symmetric positive definite matrix; in that case, the inverse is given by $Q^{\#R} = \sqrt{R}(Q\sqrt{R})^{+} = RQ^T(QRQ^T)^{+}$, where $\sqrt{R}$ is any decomposition such that $\sqrt{R}^T\sqrt{R} = R$, for example the Choleski decomposition. A weighted generalized inverse can also impose a minimum L-norm in the task space $||\dot{e}^* - Qu||_L^2 = (\dot{e}^* - Qu)^T L(\dot{e}^* - Qu)$; in that case, it is $Q^{L\#} = (\sqrt{L}Q)^{+}\sqrt{L} = (Q^T L Q)^{+}Q^T L$. Of course, both R and L norms can be imposed by $Q^{L\#R} = \sqrt{R}(\sqrt{L}Q\sqrt{R})^{+}\sqrt{L}$.

APPENDIX B
HQP COMPLEXITY

Consider a HQP whose variable x is of dimension n and whose constraints have the following form: $Ax \leq b$. The choice of an active set $\mathcal{A}$ defines an equality-only HQP (eHQP), with fewer constraints whose form are $A_i x = b_i$, where A_i (resp. b_i) are the rows of A (resp. b) selected by $\mathcal{A}$. The eHQP solution can be computed by a set of pseudoinverses following (2). The active-search algorithm [52], [26] uses a heuristic to find the optimal active set, for which the eHQP computes the optimal x. The algorithm is presented in Alg. 1, see [26] for more details.

Basically, the eHQP routine costs $o(mn^2)$ where m is the number of rows of the problem, which is approximately $o(n^3)$ when the eHQP is nearly square. If p is the number of iterations in the loop Row #3, then the complexity[12] is roughly $o(pn^3)$.

In the case presented in this paper, the HQP is called at each iteration of the robot. Between two iterations, the HQP values vary slightly. Because the HQP is a continuous function of the constraints, the active set also varies slightly. If using the optimal active set of the previous robot time to initialize the current active search, then the number of HQP iterations p remains small (experimentally, p is null 99% of the time, and is never more than 10).

Consider now the robotics HQP presented in the paper. The number of DOF is denoted by n. In the inverse-kinematics HQP, the size of the variable is the number of DOF of the system n, and the robot is generally nearly completely constrained. The cost of the inverse kinematics by HQP is thus $o(pn^3)$, with experimentally $p < 10$.

[12]In fact, only the first eHQP call is in $o(n^3)$, the following ones can be updated for $o(n^2)$.

Algorithm 1 HQP active search

Input: $A, b, \mathcal{A}_0$
$\mathcal{A} := \mathcal{A}_0$
repeat
  $A_i := A_{\mathcal{A}}$, $b_i := b_{\mathcal{A}}$
  $x^*, w := eHQP(A_i, b_i)$
  if $\exists k \notin \mathcal{A}, w_k > 0$, **then**
    $\mathcal{A} += k$
    continue
  end if
  if $\exists k \in \mathcal{A}, w_k < 0$, **then**
    $\mathcal{A} -= k$
    continue
  end if
until never
return x^*

For inverse dynamics with q the size of the contact variable, the cost is $o(p(2n+q-6)^3)$. For the reduced planar model when only the feet are in contact, $q = 20$, which makes $N = 36 + 30 + 20$ for the HQP variable.

APPENDIX C
PROOF OF EQUIVALENCE

The equivalence is proved between the scheme proposed in Section IV and the control law proposed in [59].

a) Control scheme: The development of [59] are first recalled. The task Jacobian subject to a contact is defined by:

$$J_{t|c} = JP_c{}^T \tag{53}$$

where the subscript $t|c$ indicates that the task quantities are projected in a contact consistent space. Left-multiplying (25) by $(J_{t|c}{}^{\#A^{-1}})^T = \left(A^{-1}J_{t|c}{}^T(J_{t|c}A^{-1}J_{t|c}{}^T)^{-1}\right)^T$, the task-space dynamic evolution is obtained:

$$\Lambda_{t|c}\ddot{e} + \mu_{t|c} = Q_{t|c}S^T\tau \tag{54}$$

with $\Lambda_{t|c} = (J_{t|c}A^{-1}J_{t|c}{}^T)^{-1}$, $Q_{t|c} = (J_{t|c}{}^{\#A^{-1}})^T P_c$ and $\mu_{t|c} = Q_{t|c}b + (J_{t|c}{}^{\#A^{-1}{}^T}J_c{}^T(J_cA^{-1}J_c{}^T)^{-1}\dot{J}_c - \Lambda_{t|c}\dot{J})\dot{q}$. The reference torques are obtained by inverting (54):

$$\begin{aligned}\tau^* &= ((J_{t|c}{}^{\#A^{-1}})^T P_c S^T)^{\#} f^* \\ &= J^{\star T} f^*\end{aligned} \tag{55}$$

where $J^\star = J_{t|c}(SP_c{}^T)^{\#}$ and $F = \Lambda_{t|c}\ddot{e} + \mu_{t|c}$. This final equation corresponds to the standard map from the end-effector forces f^* to the joint torques by the transpose of the Jacobian of the robot.

b) Proof of equivalence: Control law (55) can be shown to be equivalent to the control law proposed in Section IV. On the one hand, since SP_c^T is full row rank, (55) can be rewritten:

$$\tau^* = (SP_c^TA^{-1}P_cS^T)^{-1}SP_c^TA^{-1}P_cJ^T(JP_c^TA^{-1}P_cJ^T)^{-1}\ddot{e}^* \tag{56}$$

On the other hand, the scheme proposed in Section IV can be written:

$$\tau = (JA^{-1}P_cS^T)^{\#W}\ddot{e}^* \tag{57}$$

with W a user-defined weight matrix. Developing the weighted inverse gives [45]:

$$\tau = WSP_c^TA^{-1}J^T(JA^{-1}P_cS^TWSP_c^TA^{-1}J^T)^{-1}\ddot{e}^*$$

The weight is chosen as $W = (SA^{-1}P_cS^T)^{-1} = (SP_c^TA^{-1}P_cS^T)^{-1}$ [57]. Since $A^{-1}P_c = P_c^TA^{-1} = P_c^TA^{-1}P_c$ [59], the equivalence between (56) and (57) is brought to prove that:

$$JA^{-1}P_cS^T(SA^{-1}P_cS^T)^{-1}SP_c^TA^{-1}J^T = (JP_c^TA^{-1}P_cJ^T)$$

We can recognize the term $(SP_c^T)^{\#A^{-1}} = A^{-1}P_cS^T(SA^{-1}P_cS^T)^{-1}$ in the previous equality. It thus reduces to:

$$J(SP_c^T)^{\#A^{-1}}SP_c^TA^{-1}J^T = (JP_c^TA^{-1}P_cJ^T) \tag{58}$$

In [59], it is proved that $(SP_c^T)^{\#}SP_c^T = P_c^T$, which concludes the proof. ■

REFERENCES

[1] L. Saab, N. Mansard, F. Keith, J.Y. Fourquet, and P. Soueres. Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints. In *IEEE Int. Conf. on Robotics and Automation (ICRA'11)*, Shangai, China, 2011.

[2] L. Saab, O. Ramos, N. Mansard, P. Soueres, and J.Y. Fourquet. Generic dynamic motion generation with multiple unilateral constraints. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'11)*, San Francisco, USA, September 2011.

[3] P. Várkonyi, D. Gontier, and J. Burdick. On the lyapunov stability of quasistatic planar biped robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA'12)*, Saint Paul, USA, May 2012.

[4] K. Bouyarmane, A. Escande, F. Lamiraux, and A. Kheddar. Potential field guide for humanoid multicontacts acyclic motion planning. In *IEEE Int. Conf. on Robotics and Automation (ICRA'09)*, Kobe,Japan, May 2009.

[5] T. Chang and R. Dubey. A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators. *IEEE Trans. on Robotics and Automation*, 11(2):286–292, April 1995.

[6] A. Escande, S. Miossec, and A. Kheddar. Continuous gradient proximity distances for humanoids free-collision optimized-postures generation. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoid'07)*, Pittsburgh, USA, November 2007.

[7] E. Marchand and G. Hager. Dynamic sensor planning in visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98)*, Leuven, Belgium, May 1998.

[8] L. Kavraki, M. Kolountzakis, and J.C. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Trans. on Robotics and Automation*, 14(1):166–171, February 1998.

[9] J. Kuffner and S. LaValle. Rrt-connect: An efcient approach to single-query path planning. In *IEEE Int. Conf. on Robotics and Automation (ICRA'00)*, San Francisco, USA, April 2000.

[10] D. Berenson, S. Srinivasa, and J. Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *Int. Journal of Robotics Research*, 30(12):1435–1460, October 2011.

[11] S. Dalibard, A. Nakhaei, F. Lamiraux, and J.P. Laumond. Whole-body task planning for a humanoid robot: a way to integrate collision avoidance. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoid'09)*, Paris, France, December 2009.

[12] H. Hanafusa, T. Yoshikawa, and Y. Nakamura. Analysis and control of articulated robot with redundancy. In *IFAC, 8th Triennal World Congress*, volume 4, pages 1927–1932, Kyoto, Japan, 1981.

[13] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.

[14] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[15] F. Chaumette and S. Hutchinson. Visual servo control, part ii: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118, March 2007.

[16] N. Mansard and F. Chaumette. Task sequencing for sensor-based control. *IEEE Trans. on Robotics*, 23(1):60–72, February 2007.

[17] J. Rosen. The gradient projection method for nonlinear programmimg, part i, linear constraints. *SIAM Journal of Applied Mathematics*, 8(1):181–217, March 1960.

[18] A. Liégeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans. on Systems, Man and Cybernetics*, 7(12):868–871, December 1977.

[19] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Longman Publishing, Boston, 1991.

[20] B. Siciliano and J-J. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *IEEE Int. Conf. on Advanced Robotics (ICAR'91)*, Pisa, Italy, June 1991.

[21] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90–98, March 1986.

[22] D. Raunhardt and R. Boulic. Progressive clamping. In *IEEE Int. Conf. on Robotics and Automation (ICRA'07)*, Roma, Italy, April 2007.

[23] C. Chevallereau and W. Khalil. A new method for the solution of the inverse kinematics of redundant robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA'88)*, pages 37–42, Philadelphia, USA, April 1988.

[24] A. Hofmann, M. Popovic, and H. Herr. Exploiting angular momentum to enhance bipedal center-of-mass control. In *IEEE Int. Conf. on Robotics and Automation (ICRA'09)*, Kobe, Japan, 2009.

[25] O. Kanoun, F. Lamiraux, F. Kanehiro, E. Yoshida, and Laumond J-P. Prioritizing linear equality and inequality systems: application to local motion planning for redundant robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA'09)*, Kobe, Japan, May 2009.

[26] A. Escande, N. Mansard, and P-B. Wieber. Fast resolution of hierarchized inverse kinematics with inequality constraints. In *IEEE Int. Conf. on Robotics and Automation (ICRA'10)*, Anchorage, USA, May 2010.

[27] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *International Journal of Robotics Research*, 3(1):43–53, February 1987.

[28] F.E. Udwadia and R.E. Kalaba. A new perspective on constrained motion. *Proceedings: Mathematical and Physical Sciences*, pages 407–410, 1992.

[29] A. Laulusa and O. Bauchau. Review of classical approaches for constraint enforcement in multibody systems. *Journal of Computational and Nonlinear Dynamics*, 3, 2008.

[30] O. Bauchau and A. Laulusa. Review of contemporary approaches for constraint enforcement in multibody systems. *Journal of Computational and Nonlinear Dynamics*, 3, 2008.

[31] J. Peters, M. Mistry, F. E. Udwadia, J. Nakanishi, and S. Schaal. A unifying framework for robot control with redundant DOFs. *Autonomous Robots*, 24(1):1–12, January 2008.

[32] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal. Operational space control: A theoretical and empirical comparison. *Int. Journal of Robotics Research*, 27(6):737–757, June 2008.

[33] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar. A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks. In *IEEE Int. Conf. on Advanced Robotics (ICAR'09)*, Munich, Germany, June 2009.

[34] J. Park and O. Khatib. Contact consistent control framework for humanoid robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA'06)*, Orlando, USA, May 2006.

[35] O. Khatib, L. Sentis, and J. Park. A unified framework for whole-body humanoid robot control with multiple constraints and contacts. In *European Robotics Symposium*, pages 303–312, Prague, Czech Republic, March 2008.

[36] N. Mansard, O. Khatib, and A. Kheddar. Integrating unilateral constraints inside the stack of tasks. *IEEE Trans. on Robotics and Automation*, 25(3), May 2009.

[37] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle. Dynamic balance control of humanoids for multiple grasps and noncoplanar frictional contacts. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoid'07)*, Pittsburgh, USA, December 2007.

[38] K. Bouyarmane and A. Kheddar. Multi-contact stances planning for multiple agents. In *IEEE Int. Conf. on Robotics and Automation (ICRA'11)*, Shangai, China, May 2011.

[39] S. Boyd and B. Wegbreit. Fast computation of optimal contact forces. *IEEE Trans. on Robotics*, 23(6):1117–1132, December 2007.

[40] J. Park and O. Khatib. A haptic teleoperation approach based on contact force control. *Int. Journal of Robotics Research*, 25(5):575–591, 2006.

[41] M. Vukobratović and D. Juricić. Contribution to the synthesis of biped gait. *IEEE Trans Biomed Eng.*, 16(1):1–6, Jan 1969.

[42] 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 *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, Taipei, Taiwan, September 2003.

[43] Y. Nakamura and H. Hanafusa. Inverse kinematics solutions with singularity robustness for robot manipulator control. *ASME Journ of Dyn. Sys., Measures and Control*, 108:163–171, September 1986.

[44] E. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26(9):394–395, 1920.

[45] K. Doty, C. Melchiorri, and C. Bonivento. A theory of generalized inverses applied to robotics. *Int. J. of Robotics Research*, 12(1):1–19, December 1993.

[46] A. Ben-Israel and T. Greville. *Generalized inverses: theory and applications*. CMS Books in Mathematics. Springer, 2nd edition, 2003.

[47] G. Golub and C. Van Loan. *Matrix computations*. John Hopkins University Press, 3rd edition, 1996.

[48] A. Deo and I. Walker. Robot subtask performance with singularity robustness using optimal damped least squares. In *IEEE Int. Conf. on Robotics and Automation (ICRA'92)*, pages 434–441, Nice, France, May 1992.

[49] T. Sugihara. Solvability-unconcerned inverse kinematics by the levenberg-marquardt method. *IEEE Trans. on Robotics*, 25(3):658–669, June 2011.

[50] P. Baerlocher and R. Boulic. An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 6(20):402–417, August 2004.

[51] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Trans. on Robotics and Automation*, 13(3):398–410, June 1997.

[52] R. Fletcher. A general quadratic programming algorithm. *IMA Journal of Applied Mathematics*, 7:76–91, February 1971.

[53] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[54] C. Maes. *A Regularized Active-Set Method for Sparse Convex Quadratic Programming*. PhD thesis, Institute for Computational and Mathematical Engineering, Stanford University, USA, November 2010.

[55] R. Featherstone. *Rigid body dynamics algorithms*. Springer, 2008.

[56] H. Bruyninckx and O. Khatib. Gauss' principle and the dynamics of redundant and constrained manipulators. In *IEEE Int. Conf. on Robotics and Automation (ICRA'00)*, San Fransisco, USA, April 2000.

[57] J. Park. *Control Strategies for Robots in Contact*. PhD thesis, Stanford University, Department of Aeronautics and Astronautics, USA, March 2006.

[58] O. Khatib, L. Sentis, J. Park, and J. Warren. Whole body dynamic behavior and control of human-like robots. *Int. Journal of Robotics Research*, 1(1):29–44, March 2004.

[59] L. Sentis. *Synthesis and Control of Whole-Body Behaviors in Humanoid Systems*. PhD thesis, Stanford University, Department of Electrical Engineering, USA, July 2007.

[60] N. Mansard. A dedicated solver for fast operational-space inverse dynamics. In *IEEE Int. Conf. on Robotics and Automation (ICRA'12)*, St Paul, USA, May 2012.

[61] L. Sentis, J. Park, and O. Khatib. Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Trans. on Robotics*, 26(3):483 – 501, April 2010.

[62] J. Pang and J. Trinkle. Stability characterizations of rigid body contact problems with coulomb friction. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 80(10):643–663, 2000.

[63] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi. *Introduction à la commande des robots humanoïdes*. Translated from the Japanese original version by S. Sakka. Springer, 2009.

[64] M. Vukobratović and B. Borovać. Zero moment point - thirty five years of its life. *Int. Journal of Humanoid Robotics*, 1(1):157–173, January 2004.

[65] P. Sardain and G. Bessonnet. Forces acting on a biped robot. center of pressure-zero moment point. *IEEE Transactions on Systems, Man and Cybernetics (Part A)*, 34(5):630–637, September 2004.

[66] M. Audin. *Geometry*, pages 29, Proposition 5.6. Translated from the 1998 French original version. Springer-Verlag, 2002.

[67] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa. A universal stability criterion of the foot contact of legged robots-adios zmp. In *IEEE Int. Conf. on Robotics and Automation (ICRA'06)*, 2006.

[68] K. Youcef-Toumi and O. Ito. A time delay controller for systems with unknown dynamics. In *IEEE American Control Conference, 1988*, pages 904–913, 1988.

[69] P.H. Chang and J.W. Jeong. Enhanced operational space formulation for multiple tasks by using time-delay estimation. *IEEE Trans. on Robotics*, 2012. To be published.
[70] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. Dynamics and balance of a humanoid robot during manipulation tasks. *IEEE Trans. on Robotics*, 22(3):568–575, June 2006.
[71] E. Guizzo. France developing advanced humanoid robot romeo. *IEEE Spectrum Automaton Blog*, December 13, 2010.
[72] C. Ott, C. Baumgartner, J. Mayr, M. Fuchs, R. Burger, D. Lee, O. Eiberger, A. Albu-Schaffer, M. Grebenstein, and G. Hirzinger. Development of a biped robot with torque controlled joints. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoid'10)*, Paris, France, December 2010.
[73] P. Evrard, F. Keith, J.-R. Chardonnet, and A. Kheddar. Framework for haptic interaction with virtual avatars. In *IEEE Int. Symp. on Robot and Human Interactive Communication*, Munich, Germany, August 2008.
[74] Y. Ma. *An invitation to 3-d vision: from images to geometric models*, volume 26, chapter 2: Representation of three-dimensionnal moving Scene. Springer Verlag, 2004.
[75] V. Bonnet, P. Fraisse, N. Ramdani, J. Lagarde, S. Ramdani, and B. Bardy. Modeling postural coordination dynamics using a closed-loop controller. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoid'08)*, Daejeon, Korea, December 2008.
[76] A. Herdt, H. Diedam, P.B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl. Online walking motion generation with automatic footstep placement. *Advanced Robotics*, 24(5-6):719–737, 2010.
[77] A. Escande, A. Kheddar, S. Miossec, and S. Garsault. Planning support contact-points for acyclic motions and experiments on hrp-2. In *International Symposium on Experimental Robotics (ISER'08)*, Athens, Greece, July 2008.
[78] C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE Int. Conf. on Robotics and Automation (ICRA'92)*, 1992.

Layale Saab received the M.Sc. and Ph.D. degrees in robotics from the University of Toulouse, Toulouse, France, under the supervision of P. Souères, J-Y. Fourquet, and N. Mansard.

She spent four years with the GEPETTO Team, Laboratory for Analysis and Architecture of Systems (LAAS), Centre National de la Recherche Scientifique (CNRS), University of Toulouse, Toulouse, France. In 2010, she was a Visiting Researcher with Tokyo University in the laboratory of Y. Nakamura. She is currently with EOS Innovation, Évry, France. Her research interests include humanoid robots, dynamics of mechanical systems, and generation of human-like movement for virtual prototyping.

Oscar E. Ramos Ponce received the Erasmus Mundus M.Sc. degree in computer vision and robotics from Heriot-Watt University, Edinburgh, U.K.; the University of Girona, Girona, Spain; and the University of Bourgogne, Dijon, France. He is currently working toward the Ph.D. degree in robotics in the GEPETTO group, LAAS-CNRS.

He was previously with the Automation and Robotics Group, Federal University of Rio Grande do Sul, Porto Alegre, Brazil, and with Underwater Robotics Lab, University of Girona. He is involved in the generation of dynamics motion for humanoid robots.

François Keith received the M.Eng. degree from the École Nationale Supérieure dInformatique dIndustrie et dEntreprise, Évry, France, in 2007, and the Ph.D. degree in computer science from the University of Montpellier II, Montpellier, France, in 2010.

He spent one year with the Institut National de Recherche en Informatique et en Automatique Grenoble. Since 2007, he has been with the Centre National de la Recherche ScientifiqueNational Institute of Advanced Industrial Science and Technologies Joint Robotics Laboratory , UMI3218/CRT, Tsukuba, Japan (JRL-Japan) and the Montpellier Laboratory of Informatics, Robotics, and Microelectronics, where he is currently a Research Engineer. His research interests include humanoid robots and task sequencing for planning and control.

Nicolas Mansard received the joint M.Eng. and M.Sc. degrees in robotics and image processing from the Nationale Supérieure dInformatique et de Mathématiques Appliquées de Grenoble, Grenoble, France, and the University Joseph Fourier, Grenoble, and the Ph.D. degree for his work with the Lagadic Group, Institut National de Recherche en Informatique et en Automatique, Rennes, France, in 2006.

He spent one year with Stanford University, Stanford, CA, with O. Khatib and one year with the JRL-Japan with A. Kheddar. He is currently with the GEPETTO Group, LAAS-CNRS. His research is concerned with sensor-based robot animation.

Philippe Souères received the M.Sc. degree in mathematics, the Ph.D. degree in robotics, and the Habilitation degree, all from the University of Toulouse, Toulouse, France, in 1990, 1993, and 2001, respectively.

From 1993 to 1994, he was with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, with Prof. S. Sastry. Since then, he has been involved in different facets of robot perception and control with the LAAS-CNRS. Since 2003, he has been cooperating with neuroscientists on multisensory and sensorimotor integration. From 2006 to 2007, he was with the Brain and Cognition Research Center (UMR5549), Toulouse. Since 2008, he has been with the GEPETTO Group, LAAS-CNRS, where, since 2010, he has been leading the group. His research interests include control, robot vision and audio, wheeled and flying robots, humanoid systems, and neuroscience.

Jean-Yves Fourquet received the Ph.D. degree in 1446 robotics from the University of Toulouse, Toulouse, France, in 1990.

He is currently a Professor with the National School of Engineers in Tarbes (ENIT), University of Toulouse, where, since 2011, he has been the Head of Laboratoire Génie de Production (LGP). From 1991 to 1992, he was an Associate Professor with the Computing and Systems Engineering Program, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil. From 1992 to 2000, he was an Associate Professor with the University of Toulouse and a Researcher with the LAAS-CNRS. In 2000, he joined the LGP-ENIT, where he led the Intelligent Manufacturing Group from 2007 to 2009 and the Dynamic Decision and Interactions for Systems team from 2009 to 2011. He has participated or led various projects at the industrial, regional, national, or European levels. His research interests mainly concern mobile manipulation, human motion simulation, and interactive assembly scenarios.