



HAL
open science

Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints

Layale Saab, Nicolas Mansard, François Keith, Jean-Yves Fourquet, Philippe Souères

► **To cite this version:**

Layale Saab, Nicolas Mansard, François Keith, Jean-Yves Fourquet, Philippe Souères. Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints. ICRA 2011 - IEEE International Conference on Robotics and Automation, May 2011, Shanghai, China. pp.1091-1096. hal-00963612

HAL Id: hal-00963612

<https://hal.science/hal-00963612>

Submitted on 21 Mar 2014

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.



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://oatao.univ-toulouse.fr/>
Eprints ID: 8393

To link to this article: DOI:10.1016/j.corsci.2005.05.031
URL: <http://dx.doi.org/10.1016/j.corsci.2005.05.031>

To cite this version: Saab, Layale and Mansard, Nicolas and Keith, François and Fourquet, Jean-Yves and Soueres, Philippe *Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints*. (2011) In: IEEE International Conference on Robotics and Automation, 9-13 may 2011, Shanghai, China.

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Generation of Dynamic Motion for Anthropomorphic Systems under Prioritized Equality and Inequality Constraints

L. Saab, N. Mansard, F. Keith, J-Y. Fourquet, P. Soueres

Abstract—In this paper, we propose a solution to compute full-dynamic motions for a humanoid robot, accounting for various kinds of constraints such as dynamic balance or joint limits. As a first step, we propose a unification of task-based control schemes, in inverse kinematics or inverse dynamics. Based on this unification, we generalize the cascade of quadratic programs that were developed for inverse kinematics only. Then, we apply the solution to generate, in simulation, whole-body motions for a humanoid robot in unilateral contact with the ground, while ensuring the dynamic balance on a non horizontal surface.

I. INTRODUCTION

This paper focuses on whole-body motion generation in humanoid robotics. We are interested in resolving tasks that involve simultaneous motion of upper and lower parts of the body. This gives rise to trajectory equality constraints. Thus kinematic redundancy and interaction between tasks are central issues. These problems have already been tackled in the robotics literature. The first attempts were concerned by multiple equality constraints at kinematic level in Resolved Motion Rate Control schemes [15]. Prioritization has been adopted following the pioneering work of Nakamura [20] with fixed or adjustable priority [1]. However, in these works, inequality constraints were not exactly taken into account. To this end, several authors applied a potential field approach projected in the null space of equality constraints [12], [19]; whereas others used clamping for joint limit avoidance [3], [24]. In fact, the kinematic formulation can be written as a quadratic problem, as proposed in [11], where it is possible to find an approximate solution based on successive prioritization, even when inequality constraints forbid an exact solution. Whole-body motion is naturally concerned with dynamics and contact forces. Therefore, a dynamical formulation is necessary and tasks dynamics are written as linear equalities whose unknowns are the generalized torques. Several approaches consider prioritization techniques within a dynamic formulation given at the operational level. This is particularly the case for the works by Khatib and colleagues

L. Saab is with CNRS, LAAS, 7, avenue du Colonel Roche, F-31077 Toulouse, France, Université de Toulouse(UPS) lsaab@laas.fr

N. Mansard is with CNRS, LAAS, 7, avenue du Colonel Roche, F-31077 Toulouse, France, Université de Toulouse(UPS) nmansard@laas.fr

F. Keith is with the CNRS-AIST JRL, UMI3218/CRT, Tsukuba, Japan and the CNRS-UM2 LIRMM, Montpellier, France francois.keith@aist.go.jp

J-Y.Fourquet is with LGP-ENIT, 47, avenue d'Azereix, 65016, Tarbes, France, fourquet@enit.fr

P.Soueres is with CNRS, LAAS, 7, avenue du Colonel Roche,F-31077 Toulouse, France, Université de Toulouse(UPS) soueres@laas.fr

[26], [14]. The survey paper [22] gives a good comparison between operational dynamics approaches. In this family of methods, prioritization is fixed and inequalities are treated by projected potential techniques. Recently, different techniques have been developed: an original scheme to compute a generic control law from a hierarchic set of both unilateral constraints and bilateral tasks was proposed in [17] and solvers of dynamic and static quadratic problems for multi-contact were designed in [4].

This paper exploits the quadratic nature of the task space dynamic formulation including unilateral contact constraints in order to take into account both equality and inequality constraints in a way similar to the one proposed at kinematic level in [11]. Thus, our problem formulation enables the resolution of different types of dynamic constraints defined in any order of priority. First, we recall the classical schemes for inverse kinematics and dynamics in Section II. A generic formulation is drawn from these schemes, that can be used to build a unified resolution. In Section III, we prove the equivalence of the unified methodology with the control scheme of the literature, in the case of the free-floating humanoid robot. Then, in Section IV, we generalize the hierarchy of quadratic programs, first used for inverse kinematics only, to the dynamic case. A set of specific tasks is then explicated and applied to the humanoid robot model in simulation in Section V. The experiments consist in performing a whole body task while preserving the robot balance on a sloping surface.

II. NUMERICAL INVERSE RESOLUTION

In this section, the inverse algorithms for both kinematics-based and dynamics-based control are developed. The similarities in term of algorithmic resolution are then drawn and used to generate common solvers for motion generation.

A. Task function approach

The task-function approach [25] (or operational space approach [13], [21]) consists in designing the motion to be performed as a control law in a subspace of small dimension, and then back-projecting this control law to the state space of the robot. A task is defined by the triple (e, \dot{e}^*, G) , where e belongs to the task space, \dot{e}^* is the reference behavior in the tangent space to the task space at e , and G is the differential mapping between the task space and the control space of the robot:

$$\dot{e} + \mu = Gu \quad (1)$$

where u is the control and μ is the drift of the task. The interest of defining the robot motion inside a task space rather than directly at the joint level is double: first, the task space is chosen such that the control law can be easily designed (typically, in visual servoing the task space is the space of measurable visual features), making the link between sensor feedback and control direct [6]; second, the interference between two task spaces can be easily prevented and then concurrent simultaneous objectives can be decoupled, using a projection operator. To compute a specific robot control u^* that performs the reference \dot{e}^* , any numerical inverse of G can be used. The generic control law is then

$$u^* = G^\#(\dot{e}^* + \mu) \quad (2)$$

where $\#$ is a generalized inverse operator.

B. Inverse kinematics

In inverse kinematics, the control input u is simply the robot joint velocity $u = \dot{q}$. The differential link between the task and the control is the task Jacobian $G = J = \frac{\partial e}{\partial q}$. In that case, the drift μ is null. For inverse kinematics, the generalized inverse is most of the time the pseudo inverse denoted by \cdot^+ which gives the least Euclidean norm [2], [8]. The control law is then:

$$\dot{q}^* = \dot{q}_1^* + P\dot{q}_2 \quad (3)$$

where $\dot{q}_1^* = J^+ \dot{e}^*$ and P is the projection operator into the null space of J , that allows to consider any secondary control objectives \dot{q}_2 without disturbing the realization of the main objective \dot{e}^* [15]. A typical reference behavior is an exponential decay of e to zero: $\dot{e}^* = -\lambda e, \lambda > 0$.

C. Projected inverse kinematics

The projector P represents the redundancy of the robot with respect to the task. If a secondary task (e_2, \dot{e}_2^*, J_2) has to be performed, then \dot{q}_2 can be used as the new control input. The template (2) is obtained by replacing (3) in $\dot{e}_2 = J_2 \dot{q}$:

$$\dot{e}_2 - J_2 \dot{q}_1^* = J_2 P \dot{q}_2 \quad (4)$$

In that case, the differential link is the projected Jacobian $G = J_2 P$, and the drift is $\mu = -J_2 \dot{q}_1^*$. The control input \dot{q}_2^* is obtained once more by numerical inversion [28], [1]:

$$\dot{q}_2^* = (J_2 P)^+(\dot{e}_2 - J_2 \dot{q}_1^*) + P_2 \dot{q}_3 \quad (5)$$

where P_2 is the projector into the null space of $J_2 P$. The same scheme can be reproduced iteratively to account for any number of tasks until P_i is null (no more redundancy).

D. Inverse dynamics

In dynamics, the input of the system is the robot motor torques $u = \tau$. The state of the robot is the pair (q, \dot{q}) which is linked to the pair (e, \dot{e}) in the task space, and the reference behaviour is homogeneous to an acceleration denoted as \ddot{e} . Contrarily to the kinematic case, the mapping to the control input is obtained in two stages. First, we set the dynamic equation of the system, typically:

$$A\ddot{q} + b = \tau \quad (6)$$

where A is the generalized inertia matrix of the system, \ddot{q} is the robot configuration second derivative, and b is the dynamic drift (typically, gravity torques and Coriolis accelerations). Second, from the derivation of the kinematic mapping $\dot{e} = J\dot{q}$, we get:

$$J\ddot{q} + \dot{J}\dot{q} = \ddot{e} \quad (7)$$

Multiplying (1) by JA^{-1} , the differential link between τ and \ddot{e} is obtained:

$$\ddot{e} - \dot{J}\dot{q} + JA^{-1}b = JA^{-1}\tau \quad (8)$$

This last equation corresponds to the template (2) with $G = JA^{-1}$ and $\mu = -\dot{J}\dot{q} + JA^{-1}b$. The inverse dynamics control law is then directly obtained by inverting G . To fit with the dynamics of the system, the use of the generalized inverse operator $\cdot^{\#A}$ weighted by A was proposed in [13].¹ This generalized inverse gives the least norm $\|\tau\|_A = \sqrt{\tau^\top A^{-1}\tau}$, which corresponds to a minimization of the acceleration pseudo-energy $\dot{q}^\top A \ddot{q}$ [23]. Introducing τ_2 as the torque devoted to a secondary objective we get:

$$\tau^* = \tau_1^* + P\tau_2 \quad (9)$$

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

E. Projected inverse dynamics

Similarly as before, the differential link is obtained by replacing τ^* in the robot dynamics equation.

$$\ddot{e}_2 + \mu_2 = G_2 \tau_2 \quad (10)$$

with $\mu_2 = -\dot{J}_2 \dot{q} + J_2 A^{-1}b - J_2 A^{-1}\tau_1^*$ and $G_2 = J_2 A^{-1}P$. The same weighted inverse is used to inverse G_2 [26]. As before, any number of tasks can be added iteratively until the projector becomes null.

F. Inverse dynamics with rigid contacts

When the robot is in contact with the environment, the dynamic equation of the system becomes:

$$A\ddot{q} + b + J_c^\top f_c = \tau \quad (11)$$

where $J_c = \frac{\partial x_c}{\partial q}$ is the Jacobian matrix of the contact points x_c and f_c are the respective contact forces. The rigid contact implies that there is no motion of the robot contact points:

$$\dot{x}_c = 0, \quad \ddot{x}_c = 0 \Rightarrow J_c \ddot{q} = -\dot{J}_c \dot{q} \quad (12)$$

The torques are then linked to the forces with no dependency on the acceleration by multiplying (11) by $J_c A^{-1}$:

$$J_c A^{-1} J_c^\top f_c = J_c A^{-1}(\tau - b) + \dot{J}_c \dot{q} \quad (13)$$

In the basic case, $J_c A^{-1} J_c^\top$ is invertible, and f_c is deduced [14]:

$$f_c = (J_c^\top)^{\#A^{-1}}(\tau - b) + (J_c A^{-1} J_c^\top)^{-1} \dot{J}_c \dot{q} \quad (14)$$

¹The weighted generalized inverse of a matrix X by a weight W takes one of the following forms:

- if X is full row rank: $X^{\#W} = W X^\top (X W X^\top)^{-1}$
- if X is full column rank: $X^{\#W} = (X^\top W X)^{-1} X^\top W$

since J_c^\top is a full column rank matrix. The obtained force can be reinjected in (11), to obtain the *force-free* dynamic equation:

$$A\ddot{q} + b_p = P_c\tau \quad (15)$$

where $P_c = (I - J_c \#^{A^{-1}} J_c)^\top = (I - (J_c A^{-1}) \#^A J_c A^{-1})$ is the projection operator of the contact, and $b_p = P_c b + J_c^\top (J_c A^{-1} J_c^\top)^{-1} \dot{J}_c \dot{q}$. A more complete solution for dealing with redundant contact can be found in [27]. As previously, the differential link between the task and the torque input is expressed through the intermediate variable \dot{q} :

$$\ddot{e} - \dot{J}\dot{q} + JA^{-1}b_p = JA^{-1}P_c\tau \quad (16)$$

In that case, an interesting task is to control the force f_c to a reference value f_c^* . Using $e = x_c$, and setting $\ddot{e}^* = \Lambda_c f_c^*$, with $\Lambda_c = (J_c A^{-1} J_c^\top)^{-1}$ the apparent mass matrix at the contact point, it can be shown that $f_c = f_c^*$.

G. Generic actuators

In the generic case, the system actuation is given by:

$$A\ddot{q} + b + J_c^\top f_c = J_a^\top \tau \quad (17)$$

where J_a gives the link between the motors and the motion. The case of the robot is a simplification of this more generic case, since the motors control the joints. Two cases are of particular interest for us. First, the case of the humanoid robot or avatar, where some of the state parameters are not actuated and J_a is a selection matrix. Second, the case of the cable-driven actuation (typically the human body), where J_a gives the map of the forces distribution. The task differential link is obtained as before with $G = JA^{-1}P_c J_a^\top$. When the rank of J_a is smaller than the configuration of the robot, the system is under-actuated. However, it may have no direct impact on the task, since G may stay full rank independently of J_a . But, when the task space is equal to the configuration space, $G = A^{-1}P_c J_a^\top$ has at most the rank of J_a and is thus rank deficient.

III. CASE OF THE FREE-FLOATING DYNAMICS

In this section, we show the equivalence of the generalization proposed above, with the work on humanoid robot control using inverse dynamics. The humanoid robot dynamic model can be written as:

$$A \begin{bmatrix} \ddot{v}_b \\ \ddot{q} \end{bmatrix} + b + J_c^\top f_c = S^\top \tau \quad (18)$$

where $S = [0 \ I]$ is the matrix selecting the actuated joints and v_b is the velocity of the free floating robot base.

A. Control scheme from [26]

We first recall the control law for such a system proposed in [26]. In this work, an equivalent Jacobian J^* is derived from the supporting-contact constraint, that acts as a *classical Jacobian*, but respects naturally the contact constraint. As mentioned in (12), the velocity constraint on the supporting contact implies that the robot velocities (base and joint together) have to belong to the null space of the support

Jacobian. Given any velocity $(v_b, \dot{q})^\top$, the closest acceptable velocity $(v_b^*, \dot{q}^*)^\top$, is given by:

$$\begin{bmatrix} v_b^* \\ \dot{q}^* \end{bmatrix} = P_c^\top \begin{bmatrix} v_b \\ \dot{q} \end{bmatrix} \quad (19)$$

Referring to [26], the task velocities can be expressed in terms of articulated joint velocities as:

$$\dot{e} = J \begin{bmatrix} v_b^* \\ \dot{q}^* \end{bmatrix} = J(SP_c^\top)^\# \dot{q}^* \quad (20)$$

In this last expression, we can recognize a *classical* Jacobian, relying only on the actuated parameters. This equivalent Jacobian is denoted by:

$$J^* = J(SP_c^\top)^\#$$

The chosen generalized inverse for SP_c^\top is the A^{-1} -weighted inverse. It is contact consistent, which means that it can now be used directly, without any care to the contact or to the under-actuation. Indeed, the torques that perform the reference task can then simply be written as a transpose of the consistent Jacobian [26]:

$$\tau = J^{*\top} F \quad (21)$$

where

$$F = \Lambda_{t|c} \ddot{e}^* + \mu_{t|c}$$

and

$$\begin{cases} \Lambda_{t|c} = (J_{t|c} A^{-1} J_{t|c}^\top)^{-1} \\ J_{t|c} = J P_c^\top \\ \mu_{t|c} = J_{t|c}^\# b - (\Lambda_{t|c} \dot{J}_{t|c} - J_{t|c}^\# J_c^\top \Lambda_c \dot{J}_c) \begin{bmatrix} v_b \\ \dot{q} \end{bmatrix} \end{cases}$$

where the subscript $t|c$ indicates that the task space quantities are projected in the space consistent with the contact constraints.

B. Proof of the equivalence with the proposed generic scheme

As SP_c^\top is full row rank, by using the above expressions of J^* and F and considering only the task corresponding torque part, the torque of (21) can be rewritten:

$$\tau_{ref} = (SP_c^\top A^{-1} P_c S^\top)^{-1} SP_c^\top A^{-1} P_c J^\top (J P_c^\top A^{-1} P_c J^\top)^{-1} \ddot{e}^*$$

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

$$\tau = (JA^{-1}P_c S^\top)^\# W \ddot{e}^* \quad (22)$$

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

$$\tau = W SP_c^\top A^{-1} J^\top (JA^{-1}P_c S^\top W SP_c^\top A^{-1} J^\top)^{-1} \ddot{e}^* \quad (23)$$

Choosing $W = (SA^{-1}P_c S^\top)^{-1} = (SP_c^\top A^{-1} P_c S^\top)^{-1}$; since $A^{-1}P_c = P_c^\top A^{-1}$, we only need to demonstrate that:

$$JA^{-1}P_c S^\top (SA^{-1}P_c S^\top)^{-1} SP_c^\top A^{-1} J^\top = (J P_c^\top A^{-1} P_c J^\top)$$

Applying the A^{-1} weighted inverse to SP_c^\top , the previous equation reduces to:

$$J(SP_c^\top)^\# A^{-1} SP_c^\top A^{-1} J^\top = (J P_c^\top A^{-1} P_c J^\top) \quad (24)$$

It has also been demonstrated [26] that $(SP_c^\top)^\# SP_c^\top = P_c^\top$. Finally:

$$JP_c^\top A^{-1} J^\top = (JP_c^\top A^{-1} P_c J^\top) \quad (25)$$

because $P_c^\top A^{-1} = P_c^\top A^{-1} P_c$ [26]. \square

IV. INEQUALITIES IN THE LOOP

So far, we only considered tasks defined by equality constraints $\dot{e} = \dot{e}^*$. However, many objective functions describing a motion have to be defined by inequalities. Typically, they are joint limits, obstacles, balance constraint, visibility of landmark in the field of view or behind occlusion, actuator limits, etc. A very well-known solution to handle such constraints is to define a potential function [12], whose gradient acts as a virtual force that drives the robot away from the obstacles [19], or that is used to weight the Jacobian inverse to penalize the motion toward the obstacle [3]. These solutions take advantage of the null space of the main tasks. However, there are two main limitations. First, the potential functions can only be defined at the configuration level and cannot be used to cope with constraints such as actuator limits. Second, the potential functions can only be taken into account when the robot is redundant with respect to the main task. When there is not enough redundancy to account for the avoidance field, solutions have been proposed to remove some parts of the main task [16] or to add properly-chosen equality constraints at the top-priority level to prevent any further motion [26], [24]. However, such solutions are very costly in the neighborhood of several boundaries [17].

A. Quadratic programming

It has been proposed in [11] to replace the pseudo inverses used in inverse kinematics by a quadratic solver. Since quadratic solvers are able to handle indifferently equalities and inequalities, it is then possible to have both inequalities and equalities in the task definition. The reference part is then rewritten:

$$\underline{\dot{e}}^* \leq \dot{e} \leq \bar{e}^* \quad (26)$$

with $\underline{\dot{e}}^* = \bar{e}^*$ in the case of equalities, and $\underline{\dot{e}}^* = -\infty$ or $\bar{e}^* = +\infty$ to handle single-bounded constraints. Most of the time, unilateral constraints have priority over any other constraints: typically, joint limits and avoidance would be put above a grasping task. However to handle less-common cases (like insuring visibility when performing a visually-guided grasping), the method proposed in [11] was generalized to cope with a hierarchy of tasks, including possibly inequalities at any level. In [5], it was shown that this approach was applicable at low computation cost on full-size systems such as humanoid robots.

1) *One task, equalities only:* When considering a single task, the inversion (2) corresponds to the optimal solution of the problem:

$$\min_u \|Gu - \dot{e}^* - \mu\|^2 \quad (27)$$

2) *One task, inequalities and equalities together:* It is straightforward to introduce inequality constraints into a quadratic program. However, this would introduce also a de-facto hierarchy between the inequality part and the equalities. It was then proposed [11], [9] to rely on slack variables. The quadratic program for both equalities and inequalities is then written:

$$\begin{aligned} & \min_{u,w} \|w\|^2 \\ \text{s.t. } & \underline{\dot{e}}^* \leq Gu - \mu + w \leq \bar{e}^* \end{aligned} \quad (28)$$

with $\underline{\dot{e}}^* = \bar{e}^*$ for the equality parts of the task. The effect of the slack variable is to relax the parts of the task that are not feasible, and therefore to insure that the task is fulfilled at the best (in the sense of the norm of the rest).

3) *Two tasks with priority:* When the first task is solved, it was proposed [11] to use the optimal slack denoted by w^* to formalize the hierarchy with a secondary task. After the resolution of the first quadratic program, a secondary task is solved by

$$\begin{aligned} & \min_{u,w_2} \|w_2\|^2 \\ \text{s.t. } & \underline{\dot{e}}^* \leq Gu - \mu + w^* \leq \bar{e}^* \\ & \underline{\dot{e}}_2^* \leq G_2 u - \mu_2 + w_2 \leq \bar{e}_2^* \end{aligned} \quad (29)$$

In this secondary program, w^* is no more a variable. Indeed, the first task is now priority, and should be solved at least as accurately as done by the first program. If the two tasks are not compatible, the second task will be relaxed, and then less accurately executed, thanks to its slack variable. Similarly, slack variables can be introduced iteratively for any number of tasks. In [5], this cascade of quadratic program was performed by means of a dedicated optimization solver. It was possible to resolve a set of 4 tasks, for a total of 45 constraints on a 36-degrees of freedom humanoid robot in 3ms. In the sequel, we will use the generalization introduced in Section II to apply the same dedicated solver for a hierarchy of tasks while taking into account the full dynamics of the robot.

B. Weighted inverse

The previous quadratic program gives a least square solution which corresponds to a least norm for both the parameter $\|u\|$ and the rest of the optimization $\|\dot{e}^* + \mu - Gu\|$. However, as shown earlier, a norm for a specific weight is preferred for the dynamic inverse. It is easy to show that a weighting of the inverse can be equivalently obtained by adding the following constraints at the lowest priority:

$$\begin{aligned} & G_w = \dot{e}_w^* + \mu_w \\ \text{with } & G_w = \sqrt{W}, \quad \dot{e}_w^* = 0, \quad \mu_w = 0 \end{aligned} \quad (30)$$

with \sqrt{W} any square root of the weight W (for example, a Cholesky decomposition of W). Indeed, the quadratic rest of such a task is $u^\top W u = \|u\|_W^2$. The weighting can then be obtained by adding this quadratic program at the last stage:

$$\begin{aligned} & \min_u u^\top W u \\ \text{s.t. } & \forall i, \underline{\dot{e}}_i^* \leq G_i u - \mu_i + w_i^* \leq \bar{e}_i^* \end{aligned} \quad (31)$$

V. EXPERIMENTS

Four different tasks were used during the experiments. The first one, denoted by e_q , is a regulation of the robot posture. The task space is the actuated-joint space, and the desired acceleration in this space is a proportional derivative (PD) to a given reference position at zero velocity:

$$\ddot{q}^* = -\lambda_P(q - q^*) - \lambda_D\dot{q} \quad (32)$$

In that case, the Jacobian is simply the selection matrix S . The second task e_{rh} is a regulation in position and orientation of one body of the robot (for example the right hand, or the head). The reference acceleration is also a PD on the robot:

$$\ddot{x}^* = -\lambda_P \begin{bmatrix} p \\ r\theta \end{bmatrix} - \lambda_D\dot{x} \quad (33)$$

where \dot{x} is the velocity of the controlled point in its own frame, and $\begin{bmatrix} p \\ r\theta \end{bmatrix}$ is the gap between the current configuration of the controlled body and the desired value. The third task e_{jl} is the constraint of joint limits in the actuated-joint space. The task is defined by a second order Euler integration:

$$q \leq q + \Delta T\dot{q} + \frac{\Delta T^2}{2}\ddot{q} \leq \bar{q} \quad (34)$$

where q and \dot{q} are known before the resolution. Finally, the last task e_{bal} ensures an immediate balance control, by preventing the contact points from leaving the ground. The task space is the space of forces normal to the ground at the contact, and the task is to prevent them from vanishing:

$$f_{\perp} = S_{\perp} J_c \#^{\top} (S^{\top} \tau - b) \leq \epsilon \quad (35)$$

where f_{\perp} are the normal components of the contact forces f_c , S_{\perp} is the matrix that allows to select the corresponding lines of the contact Jacobian J_c , and the link with τ is defined by (14). The parameter ϵ is used to ensure a security margin to cope with perturbations. We set $\epsilon = 10N$ in the experiments. This last constraint is equivalent to the well-known ZMP constraint [29], [10] when all the contact points are planar and horizontal. Indeed, the ZMP is defined as the barycenter of the contact points weighted by their normal forces:

$$z = \frac{\sum_{i \in c} f_{\perp}^i p_c^i}{\sum_{i \in c} f_{\perp}^i} \quad (36)$$

The ZMP constraint states that *the point z must stay inside the convex hull of the contact points*. In the case of a barycenter, this is equivalent to say that all the weights must be positive. Since we do not own a torque-feedback controlled humanoid robot, the experiments were performed in simulation. We used the dynamic simulator AMELIF [7] that computes the direct dynamics from the current configuration and motor torques, resolves the collision and finally integrates the result. The control law was integrated in the control framework SoT [18], using the dedicated inequalities solver developed for inverse kinematics [5]. We reproduce a well known experiment of physiology: the subject is asked to follow an oscillatory reference with the legs. When the oscillations frequency or amplitude increases, the required

acceleration increases, until the natural contact constraint is saturated. An opposite oscillation then naturally appears on the chest to counteract the oscillation of the legs, and preserve the constraint. When put on a force sensor, the subject's ZMP was shown to present a saturation at the maximum of the amplitude. The robot is standing on one leg. An oscillatory acceleration is given as reference, that requires the whole body to remain static, except for one joint of the support leg. The amplitude of the acceleration is then increased until the support constraint saturates. To prove the generality of the proposed balance constraint (with opposition for example to the ZMP constraint), the robot was placed on a sloping ground, rotated by an angle of $10deg$ to the skyline around the pitch axis of the robot. The robot configuration at the maximum of the oscillation is shown in Fig. 1. The robot is bending on its left, with the hip roll axis moving. The balance-constraint saturation comes both from the bending (center of mass outside the support polygon), and from the acceleration in the opposite direction required to come back to the rest position. Fig. 2 shows the normal forces at the four corners of the foot during the motion. The minimal acceptable force is set to $10N$. Around iteration 1000, the force corresponding to the front right of the foot saturates. This corresponds to the time of maximal acceleration. Fig. 3 shows the acceleration of the hip joint (roll) and chest joint (yaw). The chest joint is required to remain fixed. However, when the contact constraint saturates, this part of the task becomes infeasible. Therefore, the chest is used to compensate for the motion of the hip, and prevent the foot from leaving the ground. When the acceleration of the hip decreases, the contact constraint leaves the saturation area, and the chest comes back to a zero acceleration.

VI. CONCLUSION

Based on a normalization of both the inverse-kinematics and inverse-dynamics control schemes, a solution was proposed to use the hierarchy of quadratic problems to generate dynamic movements of a humanoid robot. We proved that the given solution was generic enough to use existing inverse-dynamics schemes. The proposed schemes were then applied to control the robot motion while keeping its dynamic balance on a sloping ground. Many types of constraints and tasks can be taken into account within this framework. Future works will focus on the integration of the most classical ones: obstacles and occlusion avoidance, and their integration to generate more complex motions.

REFERENCES

- [1] 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.
- [2] A. Ben-Israel and T. Greville. *Generalized inverses: theory and applications*. CMS Books in Mathematics. Springer, 2nd edition, 2003.
- [3] 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.
- [4] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle. Dynamic balance control of humanoids for multiple grasps and noncoplanar frictional contacts. In *IEEE Int. Conf. on Humanoid Robots*, Pittsburgh, USA, 2007.

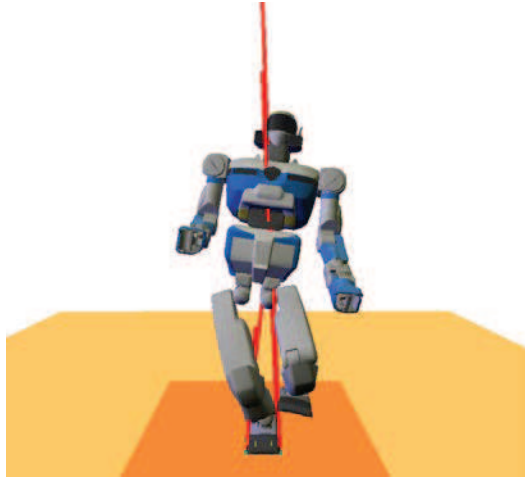


Fig. 1. Position of the robot at iteration 1000, when the maximal acceleration is reached. The red strikes correspond to the forces applied by the ground on the robot. The front forces are smaller, with the front-right force nearly null.

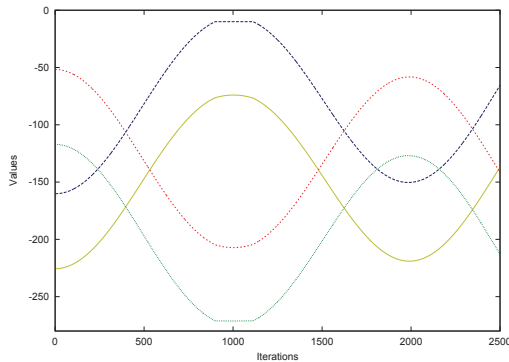


Fig. 2. Normal forces to the ground during the motion. The saturation threshold is fixed to 10N. Around iteration 1000, when the acceleration reached its highest value, the force of the front-right corner (blue) of the foot reached the saturation. A side effect of this saturation is the saturation of the right-back corner force (green), that is not due to a constraint.

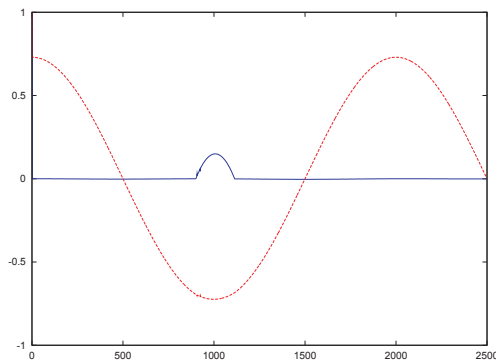


Fig. 3. Acceleration of the hip roll joint (red) and chest yaw joint (blue) during the motion. The hip is required to oscillate with a period of 2000 iterations (2s), and an amplitude of 0.72. The chest is required to stay static. Around iteration 1000, when the contact forces saturate, the chest starts to move to compensate the motion of the hip, and preserve the robot balance.

- [5] 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*, Anchorage, USA, May 2010.
- [6] 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.
- [7] 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*, 2008.
- [8] G. Golub and C. Van Loan. *Matrix computations*. John Hopkins University Press, 1996.
- [9] 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*, Kobe, Japan, 2009.
- [10] 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*, pages 1620–1626, Taipei, Taiwan, September 2003.
- [11] O. Kanoun, F. Lamiroux, 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*, Kobe, Japan, may 2009.
- [12] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90–98, Spring 1986.
- [13] 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.
- [14] 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, 2008.
- [15] 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.
- [16] N. Mansard and F. Chaumette. Task sequencing for sensor-based control. *IEEE Trans. on Robotics*, 23(1):60–72, February 2007.
- [17] N. Mansard, O. Khatib, and A. Kheddar. Integrating unilateral constraints inside the stack of tasks. *IEEE Trans. on Robotics and Automation*, 2009.
- [18] 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 *Int. Conf. on Advanced Robotics*, number 119, 2009.
- [19] E. Marchand and G. Hager. Dynamic sensor planning in visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1988–1993, Leuven, Belgium, May 1998.
- [20] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Longman Publishing, Boston, 1991.
- [21] 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.
- [22] 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.
- [23] J. Peters, M. Mistry, F. E. Udwardia, J. Nakanishi, and S. Schaal. A unifying framework for robot control with redundant dofs. *Autonomous Robots*, pages 1–12, 2008.
- [24] D. Raunhardt and R. Boulic. Progressive clamping. In *IEEE Int. Conf. on Robotics and Automation*, Roma, Italy, April 2007.
- [25] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.
- [26] L. Sentis. *Synthesis and Control of Whole-Body Behaviors in Humanoid Systems*. PhD thesis, Stanford University, USA, July 2007.
- [27] 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), June 2010.
- [28] B. Siciliano and J.-J. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *IEEE Int. Conf. on Advanced Robot*, Pisa, Italy, June 1991.
- [29] M. Vukobratović and D. Jurčić. Contribution to the synthesis of biped gait. *IEEE Trans Biomed Eng.*, 16(1):1–6, January 1969.