Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks*

Oussama Kanoun, Florent Lamiraux and Pierre-Brice Wieber

Abstract—The redundant mechanical systems like humanoid robots are designed to fulfill multiple tasks at a time. A task, in velocity-resolved inverse kinematics, is a desired value for a function of the robot configuration that can be regulated with an Ordinary Differential Equation. When facing simultaneous tasks, the corresponding equations can be grouped in a single system, or better, sorted in priority and solved each in the solutions set of higher priority tasks. This elegant framework for hierarchical task regulation has been implemented as a sequence of Least Squares problems. Its limitation lies in the handling of inequality constraints, which are usually transformed into more restrictive equality constraints through potential fields. In this paper, we propose a new prioritized task regulation framework based on a sequence of quadratic programs (QP) that removes the limitation. At the basis of the proposed algorithm is a study of the optimal sets resulting from the sequence of QPs. The algorithm is implemented and illustrated in simulation on the humanoid robot HRP-2.

Index Terms—Inverse kinematics, inequality constraints, task priority, hierarchy, redundancy, control, humanoid robot

I. INTRODUCTION

A *task*, in the context of robot motion control, can be a kinematic or a dynamic goal. For a robotic arm, a kinematic goal is for example a position for its end effector and a dynamic goal is a force it should apply on an object. Whether a robot is able or not to complete a goal depends on its own physical limitations (shape, power of actuators, etc) and on additional difficulties imposed by the environment (terrain, obstacles, etc). We take interest in generic, optimization-based control frameworks that account for such constraints.

In early frameworks, it was proposed to place artificial repulsive force fields around obstacles and place attraction fields over goals. The control was computed along the gradient of the resulting potential field [1]. This method proved especially efficient for mobile robots with lateral range sensors to navigate without colliding.

However, imposing a repulsive force in the vicinity of an obstacle can be inadequate and have adverse effects for robots

who are expected to enter this vicinity, such as oscillations or impossibility to cross narrow passages [2]. For this reason, it has been proposed in [3] to introduce instead a nonlinear damping which limits the velocity of the robot in the direction of obstacles, what appears to be much more adequate. A quadratic cost function is formed according to the goal and optimized under the velocity constraints using a numerical Quadratic Program (QP) solver. This QP formulation is less restrictive on the motion of the robot and expresses an exact hierarchy between constraints and tasks, what is not the case with the potential field approach. But this formulation considers only a single cost function to account for all the potentially desired goals.

The need to specify several goals has appeared with redundant manipulators and humanoid robots. Most of the time, the primary task of these systems is a manipulation task but their highly-articulated structure allows them to fulfill other goals simultaneously, such as keeping a reference posture, orienting a vision system, etc. Tasks that are simultaneously feasible in a given configuration may become conflicting with motion. In this case, considering a single cost function to represent all tasks would invariably lead to trade-off configurations that do not satisfy any of the tasks.

The *task priority* framework addresses this problem. Much like the QP formulation separates the tasks and the constraints in two distinct levels, this framework affects the tasks themselves with a strict priority order. The case involving two priority levels was formulated in [4] and generalized to any number of priority levels in [5], in what appears to be a sequence of equality-constrained Least Squares problems [6], defining a sequence of linear systems that need to be solved. This framework has been successfully implemented on many robotic platforms, for instance in kinematic-based control [7] and torque-based control [8]. But it only considers equality tasks, and is not designed to take into account inequality constraints such as the velocity damping introduced in [3] for obstacle avoidance and joint limits.

To observe joint limits in this framework, it was proposed [9] to compute the solution to the unbounded problem then shrink the contributions level by level back within boundaries. This is a restrictive method that can produce sub-optimal solutions, which could be avoided with a classical active set algorithm [10].

Other implementations recurred to adding repulsive fields to the highest-priority task level [8]. This solution has the aforementioned drawbacks, linked to the systematic conver-

Pierre-Brice Wieber is with JRL and INRIA Grenoble, 38334 Saint Ismier Cedex, France.pierre-brice.wieber@inria.fr

Florent Lamiraux is with JRL and LAAS-CNRS, University of Toulouse, 7 avenue du Colonel Roche 31077 Toulouse, France.florent@laas.fr

Oussama Kanoun is with Nakamura Laboratory, Department of Mechano-Informatics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku, Tokyo 113-8656, Japan okanoun@ynl.t.u-tokyo.ac.jp

sion of an inequality constraint into an equality task. A similar approach, proposed by [11] in the context of avatar animation combines the task priority framework with repulsive fields to avoid obstacles for reaching motions. Their solution appears at first to be equivalent to placing a finite repulsive field in lowest priority level [12] as long as the corresponding inequality constraint is satisfied. When it is no longer the case, the inequality constraint is converted into a regular equality constraint, moved to highest priority and the whole problem is solved again until a posture update satisfying all constraints is reached. The problem that we see in this approach resides in the priority inversion that is likely to produce a discontinuity, what should be avoided for robotic systems.

The approach by [13] solves this issue. Given k prioritized tasks and m inequality constraints, they first solve the 2^m task priority problems corresponding to every combination of inequality constraints taken as highest equality constraints. From these 2^m controls they produce a weighted solution. The weights are chosen proportionally to the distances left before saturation of the inequality constraints. The output of this method has nice regularity properties but the involved cost is unfortunately exponential in the number of inequalities.

What we propose is a new task regulation framework based on a hierarchy of Quadratic Programs that generalizes the constrained QP approach of [3] to any number of priority levels. Within this framework it becomes possible to forward the constraint-task separation across priority levels, eliminating the need for converting inequality constraints into equalities. Moreover, an inequality constraint is generalized to the notion of *inequality task* and becomes an element that can be given a priority rank as well.

To reach this result, we start in Section II by recalling the sequential Least Squares formulation of the classical task priority framework. Section III introduces the inequality tasks that we propose as a new prioritized element. We spend Section IV studying the solutions sets of the Quadratic Programs that we associate to equality and inequality tasks. The properties that we establish are the basis of the algorithm described in Section V. The purpose of the last section is to show examples of implementation with this framework, applied in simulation on the humanoid robot HRP-2.

II. CLASSICAL PRIORITIZED INVERSE KINEMATICS

A. Definitions

Let us consider a kinematic structure with n degrees of freedom, a configuration vector $q \in \mathbb{R}^n$, and a sequence for $k \in \{1, \ldots p\}$ of vector functions $f_k(q)$ specifying kinematic properties that need to be controlled to some desired values, which can be defined without loss of generality as

$$f_k(q) = 0 \tag{1}$$

Calling these equations *constraints* or *tasks* is just a question of context. Now, these vector functions are often nonlinear and without trivial inverses, so we have to rely on numerical methods to solve them.

B. Solving one task

Let us consider the following Ordinary Differential Equation (ODE),

$$\frac{\partial f_1(q)}{\partial q}\,\dot{q} = -\lambda f_1(q) \tag{2}$$

with a positive real constant λ. When following this ODE, the configuration vector converges exponentially to a limit q^* verifying $f_1(q^*) = 0$. In the case of kinematic structures with a high number of degrees of freedom, this ODE often appears to be under-determined. Of particular interest then is the solution with minimum norm,

$$\dot{q}_1 = \arg\min_x \ \frac{1}{2}\|x\|^2 \tag{3}$$
$$\text{s.t. } A_1 x = b_1 \tag{4}$$

with $A_1 = \partial f_1(q)/\partial q$, $b_1 = -\lambda f_1(q)$ (the ODE is only reformulated here in the constraint (4) without any modification). Sometimes, equation (4) is over-constrained or rank-deficient, so the solution is more generally formulated as

$$\dot{q}_1 = \arg\min_{x \in \mathcal{S}_1} \ \frac{1}{2}\|x\|^2, \tag{5}$$

where

$$\mathcal{S}_1 = \{\arg\min_x \ \frac{1}{2}\|A_1 x - b_1\|^2\} \tag{6}$$

This corresponds in fact to the result of the action of the pseudo-inverse of the matrix A_1:

$$\dot{q}_1 = A_1^+ b_1. \tag{7}$$

The set $\mathcal{S}_1$ is an affine subspace with the following closed-form expression,

$$\mathcal{S}_1 = \{\dot{q}_1 + P_1 z_1, \ z_1 \in \mathbb{R}^n\} \tag{8}$$

where z_1 is an arbitrary vector projected orthogonally on the null-space of the matrix A_1 by the operator

$$P_1 = I - A_1^+ A_1. \tag{9}$$

A fundamental observation then is that this vector z_1 gives some freedom to the control of the robot [12], which can be used to consider secondary objectives within the set $\mathcal{S}_1$ of solutions which already satisfy the constraint (2) (in the Least Squares sense). This observation is at the heart of the algorithm described in the next section, which considers a sequence of kinematic tasks of decreasing priority.

C. Solving a hierarchy of tasks

Let us consider now a second ODE,

$$A_2 \dot{q} = \frac{\partial f_2(q)}{\partial q}\,\dot{q} = -\lambda f_2(q) = b_2. \tag{10}$$

Proceeding as in the previous section, we can consider the set

$$\mathcal{S}_2 = \{\arg\min_{x \in \mathcal{S}_1} \ \frac{1}{2}\|A_2 x - b_2\|^2\} \tag{11}$$

of solutions to this ODE in the Least Squares sense, but *within* the set $\mathcal{S}_1$ where the first ODE was satisfied first of all. This gives a priority to the first ODE, which appears to be satisfied without taking into account the second ODE, whereas potential

solutions to this second ODE are considered only once the first ODE has been satisfied. Note that this second set is a subset of the first one, by definition.

Once again, we are interested in the solution with minimum norm,

$$\dot{q}_2 = \arg\min_{x \in \mathcal{S}_2} \ \frac{1}{2}\|x\|^2, \tag{12}$$

which can be obtained very easily as before with the help of the pseudo-inverse of the matrix A_2, but projected beforehand by the operator P_1 on the null-space of the matrix A_1 [4]:

$$\dot{q}_2 = \dot{q}_1 + (A_2 P_1)^+ (b_2 - A_2 \dot{q}_1). \tag{13}$$

Going from the solution (7) to the solution (13), a recursive formulation becomes apparent, which can be iterated to take into account as many kinematic tasks as desired with a priority decreasing at each iteration [5].

This classical algorithm, based on a sequence of pseudo-inverses of projected constraints, appears therefore to compute the solution to a sequence of Quadratic Programs (6), (11)... which define a shrinking sequence of subsets $\mathcal{S}_1$, $\mathcal{S}_2$... within which the solution with minimum norm is selected. This is the basis of our generalization to the case of inequality tasks.

D. Cases of ill-conditioning

When the linear equations (2) and (10) are ill conditioned the pseudo-inverse solutions (7) and (13) grow unbounded. This situation occurs mostly when some task becomes unfeasible with respect to higher priority ones. The regularization of the Least-Squares problems

$$\dot{q}_1 = \arg\min_{x} \ \frac{1}{2}\|A_1 x - b_1\|^2 + \frac{1}{2}\rho_1^2\|x\|^2, \ \rho_1 \in \mathbb{R}^* \tag{14}$$

and

$$\dot{q}_2 = \arg\min_{x \in \mathcal{S}_1} \ \frac{1}{2}\|A_2 x - b_2\|^2 + \frac{1}{2}\rho_2^2\|x\|^2, \ \rho_2 \in \mathbb{R}^* \tag{15}$$

induces a controlled error in the regulation of the task but is important for the numerical stability of the process[14]. The closed form (8) of the set $\mathcal{S}_1$ is kept unchanged. These adjusted solutions are usually obtained seamlessly with simple modifications of the algorithm computing the pseudo-inverses of the matrices A_1 and $A_2 P_1$.

III. INTRODUCING INEQUALITY TASKS

A. Attractive aspects of inequality tasks

Suppose that because of an obstacle, a humanoid robot must keep its hands at a height less than 1 meter above the ground. The algorithm presented in the previous section allows controlling this height to any given value. But here, no precise value is required, this height just needs to be below a certain value. Fixing a precise value would constrain the motions of the robot more than necessary, which could interfere with other goals given to the robot in ways that could and should be avoided. There lies a need for considering not only tasks introduced through equalities as in (1), but also tasks introduced through inequalities of the same form:
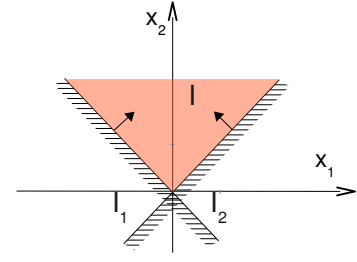
$$g_k(q) \leq 0. \tag{16}$$

Fig. 1. The linear inequalities $y \geq x$ and $y \geq -x$ determine the filled convex polytope.

Instead of the ODE (2), we introduce here an Ordinary Differential *Inequality* (ODI)

$$\frac{\partial g_k(q)}{\partial q} \dot{q} \leq -\lambda g_k(q). \tag{17}$$

Gronwall's lemma gives us

$$g_k(q) \leq g_k(q_0) e^{-\lambda(t-t_0)} \tag{18}$$

where $t > t_0$ and $q_0 = q(t_0)$. We have at least an exponential convergence to the desired inequality (16). The alternative to potential fields proposed by [3] is based on this ODI formulation. Defining d the distance between a robot and an obstacle, they wrote a collision avoidance constraint as

$$d \geq d_{min} \tag{19}$$

and derived the ODI

$$-\dot{d} \leq -\lambda(d_{min} - d) \tag{20}$$

which did not impose a value on $\dot{d}$, but only a lower bound depending on the distance to the obstacle and the convergence rate factor λ.

Apart from being less restrictive on the controls, the ODI is straightforwardly derived from the expression of the inequality task which is an attractive aspect for implementations. Equalities $f(q) = 0$ can even be seen as special cases of such ranges, as $0 \leq f(q) \leq 0$. For an equality task $f(q) = 0$, one can monitor the convergence to the solution by evaluating the norm $\|f(q)\|$. For an inequality task $g(q) \leq 0$, we may use instead the convex function $\|\max\{0, g(q)\}\|$, which is one example of an exterior penalty function, as it appears in non-linear constrained optimization [15].

B. Priority and inequality tasks

We have seen in Section II that the classical algorithm for prioritized inverse kinematics computes a shrinking sequence of affine sub-spaces $\mathcal{S}_k$ which are solutions in the Least Squares sense to a sequence of systems of linear equalities. We can try follow the same approach with inequalities, but solutions to systems of linear inequalities are not affine sub-spaces of $\mathbb{R}^n$ as in the case of equalities, but convex polytopes, volumes of $\mathbb{R}^n$ which may be finite or infinite (Figure 1). The algorithm needs to be modified accordingly.

For example, the sequence of affine sub-spaces $\mathcal{S}_k$ is computed in Section II with a recursive formula involving pseudo-inverses and projection matrices. In the general case, problems
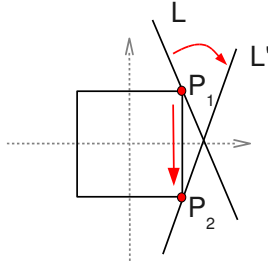
Fig. 2. The square represents a level set of the L_∞ norm in $\mathbb{R}^2$. It touches the lines L and L' at the points P_1 and P_2 with minimal L_∞ norm. A jump from P_1 to P_2 occurs even if the line L rotates in a continuous and differentiable way towards L'.

with inequalities cannot be solved efficiently with pseudo-inverses: they usually require full-fledged QP solvers. We need therefore to devise a new iterative process for computing a shrinking sequence of convex polytopes, directly involving a QP solver at each iteration.

IV. PROPERTIES OF LEAST SQUARES SOLUTIONS TO SYSTEMS OF LINEAR EQUALITIES AND INEQUALITIES

We propose here to follow closely the approach of the classical algorithm, and consider solutions to the systems of linear inequalities in a Least Squares sense.

Let A and C be matrices in $\mathbb{R}^{m \times n}$ and b and d vectors in $\mathbb{R}^m$ with $(m, n) \in \mathbb{N}^2$. We will consider in the following either a system of linear equalities

$$Ax = b \tag{21}$$

or a system of linear inequalities

$$Cx \leq d \tag{22}$$

or both. When $m = 1$, (21) is reduced to one linear equation and (22) to one linear inequality.

A. Choice of the norm L_2

We take no particular hypothesis on the linear systems (21) and (22), which might be rank-deficient or even without solutions. In the pure equality case, we saw that the problem is solved in a generic manner by the Least Squares formulation in the sense of norm L_2. This particular norm has an advantage for the problem of control. In [16], it is suggested that norms L_1 or L_∞ could be used as alternatives for the optimal resolution of inequality constraints in a control problem. Here, we give an intuition as to why these norms could be less adapted than L_2 for a control problem.

Consider in Figure 2 the square representing a level set of the norm L_∞ and the lines representing the solution sets of an equality system ($Ax = b$) at different instants. The point of junction between the square and a line is the point realizing the minimal L_∞ norm in the set $\{x \in \mathbb{R}^2 : Ax = b\}$. Even with a differentiable motion of this set, the point of minimal norm is bound to jump from a corner of the square to an adjacent one, which could cause an unwanted irregularity in the control. The same geometrical reasoning can be applied for the L_1 norm whose level sets are diamonds in $\mathbb{R}^2$. The

norm L_2 which we inherit from the previous framework has the advantage of defining strictly convex level sets (circles) that prevent this problem from occurring.

In what follows, we show that the solutions to systems of linear equalities or inequalities in the Least Squares sense are polytopes of $\mathbb{R}^n$ and can be fully determined given a single solution point.

B. System of linear inequalities

When trying to satisfy a system (22) of linear inequalities while constrained to a non-empty convex set $\Omega \subset \mathbb{R}^n$, let us consider the set

$$\mathcal{S}_i = \arg \min_{x \in \Omega} \frac{1}{2} \|w\|^2 \tag{23}$$

with

$$w \geq Cx - d, \quad w \in \mathbb{R}_+^m \tag{24}$$

where w plays now the role of a vector in $\mathbb{R}_+^m$ of *slack variables*. Since the minimized function is coercive, this set is non-empty.

Proposition 4.1: Given a point $x^* \in \mathcal{S}_i$ and considering each inequality $c^j x \leq d^j$ of the system (22) separately, we have:

$$\mathcal{S}_i = \Omega \cap \begin{cases} c^j x \leq d^j & \text{if } c^j x^* \leq d^j \\ c^j x = c^j x^* & \text{if } c^j x^* > d^j \end{cases} \tag{25}$$

In other words, all optimal solutions satisfy a same set of inequalities and violate the others by a same amount.

Proof: Let us consider an optimal solution x^*, w^* to the minimization problem (23)-(24). If x^* is on the boundary of Ω, the Karush-Kuhn-Tucker optimality conditions give that for every vector v not pointing outside Ω from x^*,

$$w^{*T} C v \geq 0 \tag{26}$$

and

$$w^* = \max \{0, Cx^* - d\}. \tag{27}$$

If x^* is in the interior of Ω, v can be any vector in $\mathbb{R}^n$ and the above conditions hold. The last condition indicates that if an inequality in the system (22) is satisfied, the corresponding element of w^* is zero, and when an inequality is violated, the corresponding element of w^* is equal to the value of the violation.

First we establish that the optimal slack variables w^* are unique. Suppose that we have two optimal solutions, x_1^*, w_1^* and x_2^*, w_2^*. Since the set Ω is convex, the direction $x_2^* - x_1^*$ points towards its inside from x_1^*, so we have

$$w_1^{*T} C(x_2^* - x_1^*) \geq 0 \tag{28}$$

which is equivalent to

$$w_1^{*T}(Cx_2^* - d) - w_1^{*T}(Cx_1^* - d) \geq 0. \tag{29}$$

The optimality condition (27) gives

$$w_1^{*T} w_2^* \geq w_1^{*T}(Cx_2^* - d) \tag{30}$$

and

$$w_1^{*T} w_1^* = w_1^{*T}(Cx_1^* - d), \tag{31}$$

so we obtain

$$w_1^{*T} w_2^* - \|w_1^*\|^2 \geq 0. \tag{32}$$

The same can be written from x_2^*,

$$w_2^{*T} w_1^* - \|w_2^*\|^2 \geq 0, \tag{33}$$

so that we obtain

$$\|w_2^* - w_1^*\|^2 = \|w_2^*\|^2 + \|w_1^*\|^2 - 2 w_2^{*T} w_1^* \leq 0, \tag{34}$$

but this squared norm cannot be negative, so it must be zero and w^* is unique.

Let I denote the subset of indices j verifying $w_j^* \leq 0$ and let J be the complementing set. From the condition (27) we deduce that $\forall x \in \mathcal{S}_i$, if $j \in I$ then $c^j x \leq d^j$ and if $j \in J$ then $c^j x = d^j + w_j^* = c^j x^*$. This establishes the inclusion of $\mathcal{S}_i$ in the set of interest (25). Now let $x \in \Omega$ such that $\forall j \in I$, $c^j x \leq d^j$ and $\forall j \in J$, $c^j x = c^j x^*$. Relaxing the second equality into an inequality, we see that $\forall j$, $c^j x - d^j \leq w_j^*$ which establishes the opposite inclusion and concludes the proof. ∎

C. System of linear equalities

We are interested in the solution set of the problem (6) while further constraining the solutions in a non-empty convex set Ω:

$$\mathcal{S}_e = \arg\min_{x \in \Omega} \frac{1}{2} \|Ax - b\|^2. \tag{35}$$

Proposition 4.2: The set (35) is non-empty, and given $x^* \in \mathcal{S}_e$ we have:

$$\mathcal{S}_e = \{x \in \Omega : Ax = Ax^*\}. \tag{36}$$

The proof is very similar to the case of the inequality systems seen above.

D. Mixed system of linear equalities and inequalities

We can observe that the optimization problems (35) and (23)-(24) have similar lay-outs and similar properties. The generalization of these results to mixed systems of linear equalities and inequalities is straightforward through the following minimization problem:

$$\min_{x \in \Omega} \frac{1}{2} \|Ax - b\|^2 + \frac{1}{2} \|w\|^2 \tag{37}$$

with

$$Cx - w \leq d, \quad w \in \mathbb{R}_+^m \tag{38}$$

The optimal set is obtained by direct application of propositions (4.2) and (4.1).

V. PRIORITIZING LINEAR SYSTEMS OF EQUALITIES AND INEQUALITIES

A. Formulation

Let us consider now the problem of trying to satisfy a set of systems of linear equalities and inequalities with a strict order of priority between these systems. At each level of priority $k \in \{1, \dots p\}$, both a system of linear equalities (21) and a system of linear inequalities (22) are considered, with matrices
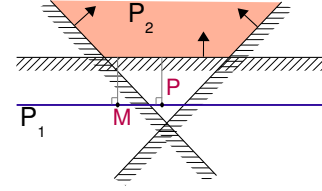

Fig. 3. The primary linear equality P_1 and the secondary system of three linear inequalities P_2 are without common solutions. M and P are solutions of P_1 minimizing the euclidean distance to P_2's set, however, P should be preferred since it satisfies two inequalities out of three while M satisfies only one. This is readily obtained by the objective function in (40).

and vectors A_k, b_k, C_k, d_k indexed by their priority level k. At each level of priority, we try to satisfy these systems while strictly enforcing the solutions found for the levels of higher priority. We propose to do so by solving at each level of priority a minimization problem such as (37)-(38). With levels of priority decreasing with k, that gives:

$$\mathcal{S}_0 = \mathbb{R}^n, \tag{39}$$

$$\mathcal{S}_{k+1} = \arg\min_{x \in \mathcal{S}_k} \frac{1}{2} \|A_k x - b_k\|^2 + \frac{1}{2} \|w\|^2 \tag{40}$$

$$\text{with } C_k x - w \leq d_k, \quad w \in \mathbb{R}_+^m. \tag{41}$$

B. Properties

A first direct implication of propositions (4.2) and (4.1) is that throughout the process (39)-(41),

$$\mathcal{S}_{k+1} \subseteq \mathcal{S}_k. \tag{42}$$

This means that the set of solutions found at a level of priority k is always strictly enforced at lower levels of priority, which is the main objective of all this prioritization scheme.

A second direct implication of these propositions is that if $\mathcal{S}_k$ is a non-empty convex polytope, $\mathcal{S}_{k+1}$ is also a non-empty convex polytope. Furthermore, the polytopes can be described using systems of equality and inequality constraints

$$\forall k, \ \exists \bar{A}_k, \bar{b}_k, \bar{C}_k, \bar{d}_k \text{ such that } x \in \mathcal{S}_k \Leftrightarrow \left\{ \begin{array}{l} \bar{A}_k x = \bar{b}_k \\ \bar{C}_k x \leq \bar{d}_k \end{array} \right. \tag{43}$$

With this representation, the step (40)-(41) in the prioritization process appears to be a simple Quadratic Program with linear constraints that can be solved efficiently. When only systems of linear equalities are considered, with the additional final requirement of choosing x^* with a minimal norm, the prioritization process (39)-(41) boils down to the classical algorithm described in Section II.

An important property on the solutions can be seen in Figure 3: the points M and P lie on the constraint P_1, and are at an equal Euclidean distance to the set P_2, but the point P satisfies two of the inequalities defining P_2 when M satisfies only one of them. Obviously, the point P should be preferred, and this is exactly what the minimization in (40) expresses: looking for the minimal amount of violation of constraints, the point P will be favored over the point M. Figure 4 further illustrates the optimal set for different priority orderings.

(a) The linear systems have common solutions so the priority does not matter.

(b) Equality has priority over inequality. The error is minimized with respect to inequalities that could not be satisfied.

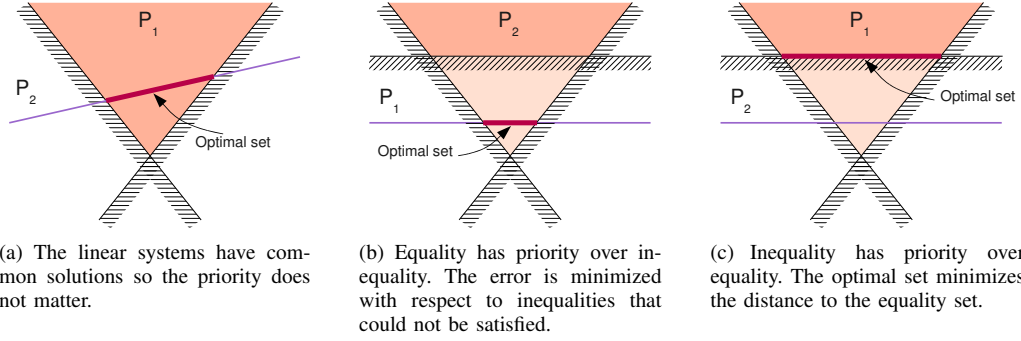(c) Inequality has priority over equality. The optimal set minimizes the distance to the equality set.

Fig. 4. Illustration of the optimal sets for prioritization problems involving both linear equality and inequality systems.

C. Algorithm

We showed how the optimal prioritization of linear systems of equalities and inequalities can be formulated as a sequence of linearly-constrained Least Squares problems. As we go down the priority levels, the admissible set S_k in which the k-th Least Squares problem is solved keeps the optimality of upper stages. The propositions (4.2) and (4.1) showed that the admissible set S_{k+1} differs from the set S_k by linear constraints that can be determined from the optimal point x_k^*. We have all we need to build an algorithm that solves a stack of prioritized linear systems.

Knowing that the algorithm repeats the same steps for every stage, we illustrate a single stage depending on the type of task at hand. Let the initial admissible set be defined by the linear constraints

$$S_1 = \left\{ \begin{array}{l} \bar{A}_0 x = \bar{b}_0 \\ \bar{C}_0 x \le \bar{d}_0 \end{array} \right. \tag{44}$$

Take the case A) where the first system to solve is a system of equalities $A_1 x = b_1$. We solve the QP

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|A_1 x - b_1\|^2 \tag{45}$$

$$\text{s.t} \quad \begin{array}{l} \bar{A}_0 x = \bar{b}_0 \\ \bar{C}_0 x \le \bar{d}_0 \end{array} \tag{46}$$

and obtain an optimal point x_1^*. Suppose that we have another linear system with lower priority, $A_2 x = b_2$. Then we need to determine the new admissible set S_2 where $\|A_1 x - b_1\|$ remains minimal. For this, proposition (4.2) indicates that the equality constraint $\bar{A}_0 x = \bar{b}_0$ is to be augmented with $A_1 x = A_1 x^*$.

If we are in case B) where the first system to solve is a system of m inequalities $C_1 x \le d_1$ then, based on the formulation (23)-(24), we solve the QP

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|w\|^2 \tag{47}$$

$$\text{s.t} \quad \begin{array}{l} \bar{A}_0 x = \bar{b}_0 \\ \bar{C}_0 x \le \bar{d}_0 \\ C_1 x - w \le d_1, \quad w \in \mathbb{R}_+^m \end{array} \tag{48}$$

for the point x_1^*. Now to find the linear constraints defining the next admissible set S_2, we use proposition (4.1): in the set of m task inequalities $\{c_1^j x \le d_1^j\}$, we identify the subset I_1 of inequalities enforced at point x_1^* and the complement subset J_1 of inequalities that were not satisfied. S_2 is nothing more than

S_1 further constrained with the inequalities $\{c_1^j x \le d_1^j\}_{j \in I_1}$ and the equalities $\{c_1^j x = c_1^j x_1^*\}_{j \in J_1}$.

As for the general case C) where the target linear system is a mixture of both types, the solution x_1^* and the next admissible set S_2 are obtained by a straightforward combination of cases A) and B). The steps are summarized in Algorithm 1.

The output of the proposed algorithm is the last stage's optimal set and a point in it. One might be interested in a more particular control realizing a minimal norm or maximizing the distance to the boundaries of the optimal set. This can easily be expressed as an additional optimization over the last optimal set.

The cost of every stage is polynomial in the number of inequality constraints, which makes this framework viable for real time implementations.

Algorithm 1 Solve prioritized linear systems
___
1: Set the system of equality constraints: $\bar{A}_0$, $\bar{b}_0$.
2: Set the system of inequality constraints: $\bar{C}_0$, $\bar{d}_0$.
3:
4: **for** $k = 0$ to $p - 1$ **do**
5:
6:     Solve the Quadratic Program (40)-(41) to obtain $\mathcal{S}_{k+1}$.
7:
8:     $\bar{A}_{k+1} \leftarrow \begin{bmatrix} \bar{A}_k \\ A_k \end{bmatrix}, \bar{b}_{k+1} \leftarrow \begin{bmatrix} \bar{b}_k \\ A_k x_k^* \end{bmatrix}.$
9:
10:     $\bar{C}_{k+1} \leftarrow \bar{C}_k, \bar{d}_{k+1} \leftarrow \bar{d}_k.$
11:
12:     **for all** c_k^j in C_k **do**
13:       **if** $c_k^j x_k^* \le d_k^j$ **then**
14:
15:         $\bar{C}_{k+1} \leftarrow \begin{bmatrix} \bar{C}_{k+1} \\ c_k^j \end{bmatrix}, \bar{d}_{k+1} \leftarrow \begin{bmatrix} \bar{d}_{k+1} \\ d_k^j \end{bmatrix}.$
16:
17:     **else**
18:
19:         $\bar{A}_{k+1} \leftarrow \begin{bmatrix} \bar{A}_{k+1} \\ c_k^j \end{bmatrix}, \bar{b}_{k+1} \leftarrow \begin{bmatrix} \bar{b}_{k+1} \\ c_k^j x_k^* \end{bmatrix}.$
20:
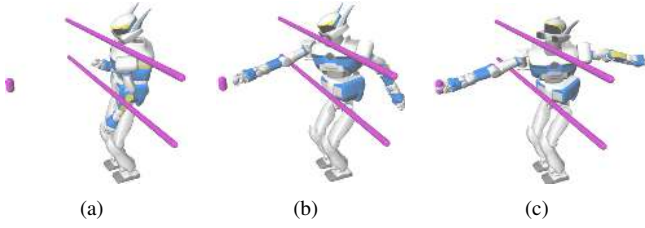21:   **end if**
22:     **end for**
23: **end for**

Fig. 5.   Scenario 1: reaching an object behind obstacles

D. Cases of ill-conditioning

This algorithm shares with the classical algorithm described in Section II the same problems with respect to ill-conditioning, that have been discussed in Section II-D. The same solution can be adopted here, balancing the Least Squares problem (40)-(41) with the norm of the resulting solution, solving instead the QP

$$S_{k+1} = \arg\min_{x \in S_k} \frac{1}{2}\|A_k x - b_k\|^2 + \frac{1}{2}\|w\|^2 + \frac{1}{2}\rho^2\|x\|^2 \quad (49)$$

$$\text{with } C_k x - w \le d_k \quad w \in \mathbb{R}^m_+. \quad (50)$$

VI. SIMULATIONS

A. Setting

The purpose of this section is to show how the values of equality and inequality tasks evolve using the proposed algorithm. We propose two scenarios in velocity-based control for the system HRP-2. In each scenario, the parameters that we solve for are all the joint velocities $\dot{q}$. They are at the number of 28: 6 in each limb, 2 in the trunk and 2 in the neck.

The algorithm uses a Quadratic Program to solve the required optimization at every priority stage. This optimization can be done using any off-the-shelf numerical QP solver. We are currently developing a solver specialized for the case where all inequalities are hard constraints, and we use it here to solve scenario 1. For the general case where inequality tasks are placed anywhere in the priority stack, we temporarily recur to all-purpose QP solvers like [17].

B. Scenario 1

In this scenario, the robot must reach for an object while standing behind horizontal bars.

The robot is subject to permanent equality and inequality constraints. The constraints are for enforcing the joint limits, preventing collision and maintaining the pose of the feet and the position of the projection of the center of mass on the ground. The shape of the robot is simplified to a set of cylinders so that only a small number of pairs of segments are constrained against collision. The simplification is specific to HRP-2 and out of the scope of this work. We implement the inequality constraints as shown in section III and choose all the gains λ in the differential inequalities equal to 0.5s^{-1}.

In a first priority stage S_1, we place an equality task to position the right hand at a fixed target point behind the bars. We call this task *reaching* and write it $\overrightarrow{p_h}(q) - \overrightarrow{p_t} = 0$ where $\overrightarrow{p_h}$ is the position of the center of the right hand and $\overrightarrow{p_t}$ the target position. The corresponding ODE has the form $J_h\dot{q} =$
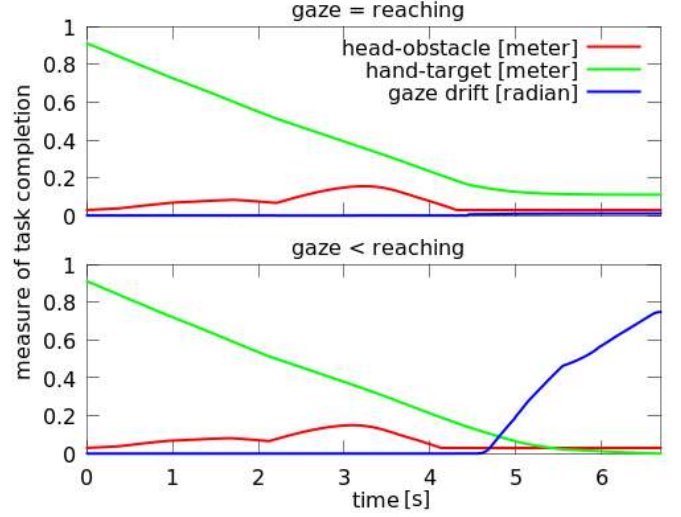

Fig. 6.   Effect of prioritization in Scenario 1. Upper window: a single priority stage is used to solve the *reaching* and *gaze* under constraints: neither of the tasks is completed. Lower window: the gaze task placed in priority 2 diverges to the benefit of reaching for the target point (priority 1).

$-\lambda_1(\overrightarrow{p_h} - \overrightarrow{p_t})$ where J_h is the jacobian of $\overrightarrow{p_h}(q)$ with respect to q. The gain λ_1 is taken such as $\|\lambda_1(\overrightarrow{p_h} - \overrightarrow{p_t})\|$ is bounded above by the maximum velocity of 0.2m.s^{-1}.

In a second priority stage S_2, an equality task keeps the vision target focused on the target. We call this task *gaze* and define it as:

$$\overrightarrow{ob} \times \overrightarrow{op_t} = \vec{0} \quad \text{for } \overrightarrow{ob}.\overrightarrow{op_t} \ge 0 \quad (51)$$

where o is a point on the optical axis, $\overrightarrow{ob}$ is a vector lying on the optical axis ahead of o. The ODE is

$$[\overrightarrow{op_t} \times (\overrightarrow{ob} \times J_w) - \overrightarrow{ob} \times J_o]\dot{q} = -\lambda_2(\overrightarrow{ob} \times \overrightarrow{op_t}) \quad (52)$$

where J_w and J_o respectively stand for the head orientation jacobian and position jacobian at point o. The scaling factor λ_2 is chosen to bound $\|\lambda_2(\overrightarrow{ob} \times \overrightarrow{op_t})\|$ above by $4.10^{-4}\text{m}^2.\text{s}^{-1}$. For each of the tasks, we had to set the regularization parameter ρ^2 to the order of magnitude of 0.01 in order to prevent instabilities when the tasks become conflicting.

Figure 6 is divided in two windows: the upper window shows how this scenario is poorly solved if the gaze and reaching tasks were to share the same priority level. The obstacle stops the head too soon for the hand to reach its target and because both tasks have the same priority, the trade-off posture in Figure 5(b) is the final result. The lower window shows that lowering the priority of the gaze task makes it possible for the reaching task to be solved. We see that the moment the head cannot move any closer to the upper bar, the direction of the gaze starts deviating from its acquired target so that the hand may continue to move, ending in the posture of Figure 5(c). The computation time using the implementation prototype is reported in Figure 7, it should be improved in the future.

C. Scenario 2

This scenario is also a reaching scenario, the difference consists in adding a last stage inequality task in order to

Fig. 8. Scenario 2: reach for an object and avoid the vision field if possible
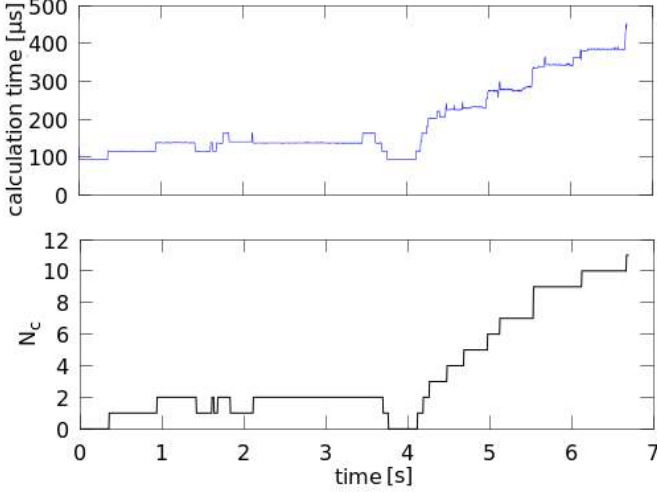
Fig. 7. Scenario 1. Upper window: cost of preparing and solving the sequence of prioritized linear systems per time step. Lower window: number N_c of saturated inequality constraints per time step. Every iteration counted 33 inequality constraints for collision avoidance and 56 for lower and upper bounds. The program was run in a single thread using the notebook processor Intel Core i5-540M at 2.52 GHz.

prevent the moving hand from entering the vision field as long as possible.

The permanent constraints are those of the first scenario, if we exclude the external collision avoidance constraints.

A first stage S_1 has an attractor field designed to move a point p_h in the right hand of the robot to a target point p_t,

$$P_a(q) = \frac{1}{2}K\|\overrightarrow{p_t p_h}\|^2 = 0 \tag{53}$$

with the positive gain $K = 2.10^{-2}$. For this task we solve the ODE

$$\langle\overrightarrow{p_t p_h}|J_h\rangle\dot{q} = -\lambda_1 P_a(q) \tag{54}$$

λ_1 is chosen such as $\lambda_1 P_a(q)$ is bounded above and below respectively by $10^{-3}\mathrm{m}^2.\mathrm{s}^{-1}$ and $2.5\times10^{-5}\mathrm{m}^2.\mathrm{s}^{-1}$. The lower bound forces a minimum attraction towards the target point, which is useful near convergence. The above equation allows the free translation of the point p_h on the plane orthogonal to the direction $\overrightarrow{p_h p_t}$.

A second stage S_2 is occupied by the same gaze task seen in scenario 1.

A third priority stage S_3 holds an inequality task that forbids a point p_h on the moving hand from entering the vision field. The vision field is modeled as a finite cylinder (see Figure 8). This inequality $d > r$, where d is the distance of the point to the core of the vision field and r the radius of the field ($14cm$), leads to the ODI [3]

$$-\langle J_h|\vec{n}\rangle\dot{q} \le \lambda_3(d-r) \tag{55}$$

with $\vec{n}$ being the unitary radial vector of the cylinder pointing out to p_h. When the inequality is not satisfied, the effect of this task is to pull the hand out of the vision field. Like any other task in a priority stack, the amplitude of the feedback $\lambda_3(d-r)$ should be bounded to avoid numerical instabilities.
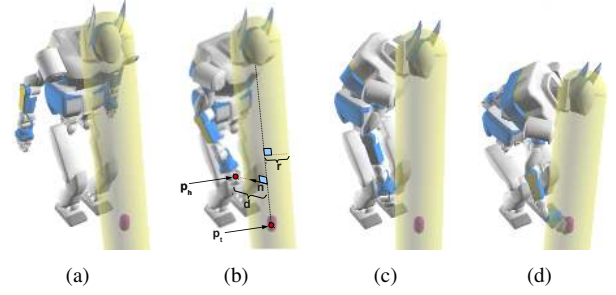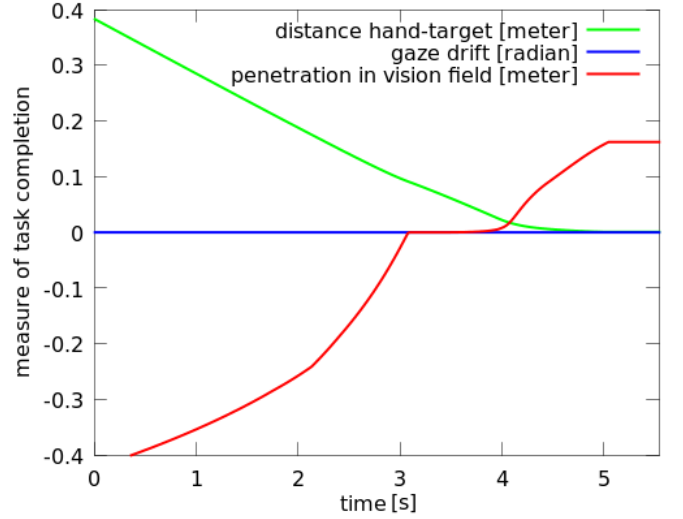

Fig. 9. Effect of prioritization in Scenario 2. The inequality task responsible for keeping the center of the hand (point p_h) out of the vision field holds until becoming conflicting with the higher-priority reaching task.

We replace the above ODI by the following,

$$-\langle J_h|\vec{n}\rangle\dot{q} \le max(-\varepsilon, \lambda_3(d-r)) \tag{56}$$

where $\varepsilon = 5 \times 10^{-5}\mathrm{m.s}^{-1}$ and $\lambda_3 = 0.5\mathrm{s}^{-1}$.

Finally, the regularization parameters ρ^2 were chosen with an order of magnitude of 0.01 for the three task stages. Figure 9 shows the evolution of task residual errors along the iterations. The vision field is protected until the attractor field and the inequality task become conflicting. From there, the point p_h is drawn inside the field.

It must be mentioned that the choices of gains at every priority stage were intricate: ε had to be large enough not to make the last inequality task fail too early, and small enough to prevent stability issues when conflicting with the upper reaching task given the regularization parameters. We think that an automated optimization of these parameters would be a good addition to the framework.

VII. CONCLUSION

In the context of kinematic control, we proposed a new theoretical framework based on a hierarchy of Quadratic Programs that generalizes the constrained QP approach of [3] to any number of priority levels. Within this framework it becomes possible to forward the constraint-task separation

across priority levels, eliminating the need for converting inequality constraints into equalities through potential fields. The proposed framework is general enough to permit the prioritization of the inequality constraints themselves, which become defined as prioritized inequality tasks.

The proposed framework was illustrated for regulating configuration-dependant functions on a redundant manipulator by controlling the joint velocities, but it could have been illustrated for regulating torque-dependant or acceleration-dependant functions. Similarly to a basic, non prioritized regulation control, it is not well adapted to time-dependant functions, for instance when trying to track a reference trajectory. The proposed algorithm for prioritizing inequality-constrained Least Squares problems remains nonetheless a general tool that may serve in contexts other than kinematic control.

This generalized framework comes at the cost of replacing an equality-constrained Least Squares optimization at every priority stage with an inequality-constrained one. The complexity of the algorithm remains polynomial but the performance will be subject to how well the inequality constraints are handled by the Quadratic Program solver. A control problem is a time-continuous one, therefore keeping track of the saturated inequalities after a control iteration is a valuable information for predicting the saturated set of the following iteration and saving time. This is one of the principles of the algorithm proposed by [18]. We plan to investigate in this direction to further lower the computational costs.

Acknowledgment

References

[1] O. Khatib, "The potential field approach and operational space formulation in robot control," *Adaptive and Learning Systems: Theory and Applications*, pp. 367–377, 1986.

[2] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *IEEE International Conference on Robotics and Automation*, 1991, pp. 1398–1404.

[3] B. Faverjon and P. Tournassoud, "A local based approach for path planning of manipulators with a high number of degrees of freedom," in *IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 1152–1159.

[4] Y. Nakamura, *Advanced robotics: redundancy and optimization*. Addison-Wesley Longman Publishing, 1990.

[5] B. Siciliano and J. Slotine, "A general framework for managing multiple tasks in highly redundantrobotic systems," in *International Conference on Advanced Robotics*, 1991, pp. 1211–1216.

[6] Å. Björck, *Numerical Methods for Least Squares Problems*. Philadelphia: SIAM, 1996.

[7] N. Mansard and F. Chaumette, "Task Sequencing for High-Level Sensor-Based Control," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 60–72, 2007.

[8] L. Sentis and O. Khatib, "Control of Free-Floating Humanoid Robots Through Task Prioritization," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 1718–1723.

[9] G. Antonelli, G. Indiveri, and S. Chiaverini, "Prioritized closed-loop inverse kinematic algorithms for redundant robotic systems with velocity saturations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 5892–5897.

[10] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: John Wiley & Sons, 1987.

[11] M. Peinado, R. Boulic, B. Le Callennec, and D. Meziat, "Progressive cartesian inequality constraints for the inverse kinematic control of articulated chains," *Eurographics*, pp. 93–96, 2005.

[12] A. Liégeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[13] N. Mansard, O. Khatib, and A. Kheddar, "Integrating unilateral constraints inside the Stack Of Tasks," *IEEE Transactions on Robotics*, 2008.

[14] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of dynamic systems, measurement, and control*, vol. 108, no. 3, pp. 163–171, 1986.

[15] A. Fiacco and G. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques*, ser. Classics in Applied Mathematics. Society for Industrial Mathematics, 1987.

[16] W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter, "Extending iTaSC to support inequality constraints and non-instantaneous task specification," in *IEEE International Conference on Robotics and Automation*, 2009.

[17] K. Schittkowski, "QLD: A FORTRAN Code for Quadratic Programming, Users Guide," Mathematisches Institut, Universitt Bayreuth, Germany, Tech. Rep., 1986.

[18] A. Escande, N. Mansard, and P.-B. Wieber, "A Dedicated Quadradic Program for Fast Hierarchical-Inverse-Kinematic Resolution," in *IEEE International Conference on Robotics and Automation*, 2010.

Oussama Kanoun graduated from École des Mines de Paris, France, in 2005. He obtained a Master degree from École Normale Superieure de Cachan in Mathematics for Machine Learning and Vision in 2006. He received his Ph.D dregree in Robotics from The University of Toulouse in 2009. He is currently a JSPS Postdoctoral Fellow at Nakamura Laboratory - The University of Tokyo, working on motion control and planning for human and humanoid models.

Florent Lamiraux graduated from the École Polytechnique Paris in 1993. He received the Ph-D degree in Computer Science from the Institut National Polytechnique de Toulouse in 1997 for his research on Mobile Robots. He worked two years at Rice University as a Research Associate. He is currently Directeur de Recherche at LAAS-CNRS, working in humanoid robots.

Pierre-Brice Wieber graduated from the École Polytechnique, Paris, France, in 1996 and received his PhD degree in Robotics from the École des Mines de Paris, France, in 2000. Since 2001, he has been with the INRIA Grenoble Rhne-Alpes in the Bipop team but is presently with the CNRS/AIST JRL in Tsukuba, Japan, as an EU Marie Curie visiting fellow. His research interests include the modeling and control of humanoid robots, model predictive control, and optimization.