Noname manuscript No.
(will be inserted by the editor)

Whole-Body Hierarchical Motion and Force Control for Humanoid Robots

Mingxing Liu, Ryan Lober, and Vincent Padois

Received: date / Accepted: date

Abstract Robots acting in human environments usually need to perform multiple motion and force tasks while respecting a set of constraints. When a physical contact with the environment is established, the newly activated force task or contact constraint may interfere with other tasks. The objective of this paper is to provide a control framework that can achieve real-time control of humanoid robots performing both strict and non strict prioritized motion and force tasks. It is a torque-based quasi-static control framework, which handles a dynamically changing task hierarchy with simultaneous priority transitions as well as activation or deactivation of tasks. A quadratic programming problem is solved to maintain desired task hierarchies, subject to constraints. A generalized projector is used to quantitatively regulate how much a task can influence or be influenced by other tasks through the modulation of a priority matrix. By the smooth variations of the priority matrix, sudden hierarchy rearrangements can be avoided to reduce the risk of instability. The effectiveness of this approach is demonstrated on both a simulated and a real humanoid robot.

Keywords Whole-body control · Physical contact · Torque-based control · Humanoid robots

M. Liu · R. Lober · V. Padois
Sorbonne Universités, UPMC Univ Paris 06, UMR 7222, Institut des Systèmes Intelligents et de Robotique (ISIR), F-75005, Paris, France

M. Liu · R. Lober · V. Padois
CNRS, UMR 7222, ISIR, F-75005, Paris, France
E-mail: {liu, lober, padois}@isir.upmc.fr

1 INTRODUCTION

Humanoids are expected to perform complex tasks, including physical interactions with environments (see Figure 1) through the control of their whole-body motion. When both motion and force tasks are involved, three problems should be handled. First, when the degrees of freedom (DoF) of a motion and a force task are not orthogonal to each other, i.e. when both motion and force tasks defined at the end-effector frame require the same DoF, then the priorities between these two tasks should be handled, since both of them may not be satisfied all the time. Second, as motion and contact forces applied at different body frames can interfere with each other through robot dynamics, the controller must ensure that task hierarchies are respected to achieve an appropriate whole-body performance. Third, if constraints need to be satisfied, for example when foot contact forces need to be maintained within friction cones to avoid foot slippage, then the hierarchy of tasks should be consistent with such constraints. This paper focuses on the whole-body control of humanoid robots performing prioritized motion and force tasks subject to a set of constraints.

The motion and force control problem was first studied to control robotic manipulators. An approach to handle a pair of end-effector motion and force tasks is proposed in [11]. This approach uses task specification matrices to restrict operational space positional freedom in the subspace orthogonal to the directions of force that is to be applied by the end-effector. With the development of humanoid robots, several whole-body motion and force control approaches have been proposed. A dynamic balance force controller [25] is developed for the control of center of mass (CoM) motion and contact forces of humanoid robots, where an addi-

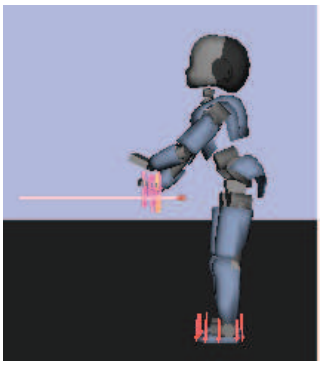

Fig. 1 Example of a humanoid robot in physical interaction with its environment.

tional task force is computed based on a CoM dynamics model and external forces to ensure balance. In these approaches, the control of an arbitrary number of prioritized tasks is not dealt with.

The problem of prioritized multi-task control of redundant robots is addressed by hierarchical control algorithms. Some of these control algorithms focus on the handling of *strict task priority* hierarchies, such as analytical methods based on null-space projectors [6, 7, 18, 24], which ensure that lower priority tasks are performed only in the null-space of higher priority tasks. However, these methods handle task priorities by relying on the use of pseudo-inverses and null space projectors, resulting in a formulation that is mathematically not compatible with inequality constraints. Therefore, constraints, such as those restricting contact forces inside friction cones to avoid foot slippage of humanoids, can not be properly implemented. The handling of prioritized tasks as well as equality and inequality constraints is addressed by hierarchical quadratic programming (HQP) algorithms [5, 9, 20, 21]. The idea of HQP is to first solve a quadratic program (QP) to obtain a solution for a higher priority task; and then to solve another QP for a lower priority task, without increasing the obtained minimum of the previous task objective. This prioritization process corresponds to solving lower-priority tasks in the null-space of higher-priority tasks while respecting constraints.

Another type of redundant robot control framework handles *non-strict task hierarchies* by using weighting strategies [1, 2, 3, 13, 22], the solution of which is a trade-off among task objectives with different weights. These weighting strategies are based on the use of optimization techniques. All the constraints and task objectives are solved in one quadratic program. One limitation of weighting strategies is that strict priorities cannot be achieved, and the performance of higher-priority tasks cannot be guaranteed by simply adjusting the weights of task objectives. Although this problem is addressed by a prioritized control framework [14], which ensures the performance of a higher-priority task with a user defined tolerance margin, this approach handles priorities of only two levels.

An important difference between strict and non-strict hierarchies is how efficiently they achieve hierarchy rearrangements. For robots acting in dynamically changing contexts, task priorities may have to be switched, and certain tasks may have to be activated or deactivated to cope with changing situations, for example, during frequent establishment and break of contacts. In this case, a sudden rearrangement of task hierarchies may lead to a large discontinuity in control laws and an increased risk of system instability. Such a sudden rearrangement may occur when the hierarchies are handled by strict hierarchical control algorithms, which organize tasks by using discrete priority levels. Therefore, to achieve smooth hierarchy rearrangements within strict hierarchies, some specific methods have been developed. The method presented in [10, 19] achieves smooth priority rearrangement between only two levels of tasks. An approach to hierarchical control with continuous null-space projections is presented in [4], but the use of a specific activator makes this approach difficult to implement for separatly handling different task directions. Priority transition between multiple tasks is achieved by the use of a specific inverse operator [16] or by using intermediate desired values in the task space [12], but the computation time of these two methods increases with the number of simultaneous priority transitions. On the other hand, priority transitions can be easily achieved within a non-strict hierarchical control framework by the continuous variation of task weights [22]. This method is used in HQP approaches to swap task priorities [8] smoothly. But this implementation may require a set of swaps before bringing a task to the desired priority level, since a swap is needed each time the task is moved from its actual priority level to a consecutive one.

The above mentioned works handle task hierarchies organized in a lexicographic way [21], which is not flexible since a lexicographic hierarchy does not allow one to handle the priority between each pair of tasks separately. For example, in the case of a weighting strategy, the priority between each pair of tasks is determined by the ratio of their task weights. For a hierarchy containing three tasks: task 1, task 2, and task 3, once the value of ω_{t_1} is fixed, the desired priorities of task 2 and task 3 over task 1 can be achieved by tuning ω_{t_2} and ω_{t_3} respectively. But in this way, the priority of task 3 over task 2 is fixed by the previously tuned ω_{t_2} and ω_{t_3}, and it is thus not possible to tune the priority relation between task 2 and task 3 any more.

Recently, a generalized projector has been developed and used in [15] for hierarchical control. The novelty of hierarchical control algorithms based on the use of this generalized projector is that they can handle not only a single standard lexicographic hierarchy as the HQP and weighting strategies do, but also a complex priority network of hierarchies with both strict and non-strict priorities. The priority between each pair of tasks can be handled separately. Only one swapping phase is needed to move an arbitrary number of tasks to their desired priority levels concurrently. Moreover, this generalized projector improves the smoothness during hierarchy rearrangements, because it can regulate to what extent a lower-priority task is projected into the null-space of higher-priority tasks (e.g. completely, partially, or not at all). In [15], the generalized projector is implemented in an optimization based dynamic control framework, which is applied to control a simulated KUKA LWR robot. However, the application of this control framework in real-time control of humanoid robots is limited, because its computational cost is sensitive to the number of tasks and the number of DoF of the robot.

The contribution of this paper is the implementation of the aforementioned generalized projector in a quasi-static torque control framework and on humanoid robots. Compared with the control framework used in [15], the computational cost of this quasi-static framework is much less sensitive to task numbers or robot complexity. This makes it possible to handle a complex network of task priorities by using the generalized projector, with a computation cost that can be suitable for real-time control of humanoid robots.

This paper is organized as follows. The robot model, as well as task definitions and task priority parametrization used in this paper are presented in Section 2. The control framework is developed in Section 3. Some experimental results are presented in Section 4 to demonstrate the framework capabilities. Finally, the conclusion and future works are presented in Section 5.

2 Modeling

Consider a robot as an articulated mechanism with n DoF including n_a actuated DoF. The dynamics of the robot in terms of its generalized coordinates $\boldsymbol{q} \in \mathbb{R}^n$ are written as follows

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \mathbf{S}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \boldsymbol{\tau} + \mathbf{J}_c(\boldsymbol{q})^T \boldsymbol{w}_c, \quad (1)$$

where $\mathbf{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the generalized inertia matrix; $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ and $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ are the vector of velocity and the vector of acceleration in generalized coordinates, respectively; $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^n$ is the vector of Coriolis and centrifugal induced joint torques; $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the vector of gravity induced joint torques; $\mathbf{S}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \in \mathbb{R}^{n \times n_a}$ is a selection matrix for the actuated DoF; $\boldsymbol{\tau} \in \mathbb{R}^{n_a}$ is the vector of the actuation torques; $\mathbf{J}_c(\boldsymbol{q})^T = \begin{bmatrix} \mathbf{J}_{c,1}(\boldsymbol{q})^T & \dots & \mathbf{J}_{c,n_c}(\boldsymbol{q})^T \end{bmatrix}$ is the transpose of a Jacobian matrix, with $\mathbf{J}_{c,\beta}(\boldsymbol{q})$, the Jacobian matrix associated with a contact point β; $\boldsymbol{w}_c = \begin{bmatrix} \boldsymbol{w}_{c,1}^T & \dots & \boldsymbol{w}_{c,n_c}^T \end{bmatrix}^T$ are the external contact wrenches applied to the robot, with n_c the number of contact points.

2.1 Motion and force tasks

Consider a robot performing motion and force tasks. Each task i is associated with its task wrench $\boldsymbol{w}_{t,i}$. For a goal directed Cartesian motion task i, the task wrench $\boldsymbol{w}_{t,i}$ should drive the task frame to perform the desired motion. The desired task wrench can be, for example, the output of a proportional-derivative (PD) controller

$$\boldsymbol{w}_{t,i}^d = \mathbf{K}_{P,i}\boldsymbol{e}_i + \mathbf{K}_{D,i}\dot{\boldsymbol{e}}_i, \quad (2)$$

where $\boldsymbol{e}_i$ and $\dot{\boldsymbol{e}}_i$ are task position and velocity errors, respectively; and $\mathbf{K}_{P,i}$ and $\mathbf{K}_{D,i}$ are symmetric, positive definite gain matrices. For a posture task, the task wrench $\boldsymbol{w}_{t,i}$ is in fact a torque in joint space.

For a goal directed wrench task, the desired task wrench can be the output of, for example, a proportional-integral controller with a feed-forward term

$$\boldsymbol{w}_{t,i}^d = \boldsymbol{w}_{t,i}^\star + \mathbf{K}_{P,i}\boldsymbol{e}_w + \mathbf{K}_{I,i}\int \boldsymbol{e}_w dt, \quad (3)$$

where $\boldsymbol{w}_{t,i}^\star$ is the desired task wrench applied by the robot on the environment, $\boldsymbol{e}_w$ is the error of task wrench, and $\mathbf{K}_{P,i}$ and $\mathbf{K}_{I,i}$ are symmetric, positive definite gain matrices. The integral component here is used to reduce steady state force tracking errors.

For a non goal directed task, such as the foot contact task for supporting the whole-body balance, or a whole-body posture task, which ensures the uniqueness of robot control input solution, the desired wrench is unknown *a priori*. The appropriate values of these task wrenches are computed by the controller.

In this paper, $\mathcal{I}$ denotes the set of all the tasks, including both goal directed and non goal directed tasks. $\mathcal{N}$ is a subset in $\mathcal{I}$, which contains non goal directed tasks only. $\boldsymbol{w}_t$ denotes the vector of all the task wrenches.

Basically, each task wrench $\boldsymbol{w}_{t,i}$ acts on the robot dynamics (1) through its equivalent joint torques ($\boldsymbol{\tau}_{t,i} = \mathbf{J}_{t,i}(\boldsymbol{q})^T \boldsymbol{w}_{t,i}$ with $\mathbf{J}_{t,i}$ being the task Jacobian [1]). These

[1] The dependence of Jacobian matrices on $\boldsymbol{q}$ is omitted for clarity reasons.

equivalent joint torques are accounted for in the computation of $\boldsymbol{\tau}$, which is used to drive the robot.

2.2 Priority parametrization

The priority parametrization used in [15] is applied here. The relative importance levels of each task i with respect to a set of n_t tasks, including task i, is characterized by a priority matrix $\mathbf{A}_i$

$$\mathbf{A}_i = diag\left(\alpha_{i1}\mathbf{I}_{m_1}, \ldots, \alpha_{ij}\mathbf{I}_{m_j}, \ldots, \alpha_{in_t}\mathbf{I}_{m_{n_t}}\right) \tag{4}$$

where m_j is the dimension of task j, $\mathbf{A}_i$ is a diagonal matrix, the main diagonal blocks of which are square matrices: $\alpha_{ij}\mathbf{I}_{m_j}$. $\mathbf{I}_{m_j}$ is the $m_j \times m_j$ identity matrix, and $\alpha_{ij} \in [0, 1]$. In this paper, the notation $i \triangleright j$ indicates that task i has a strict higher priority over task j. By convention, the coefficient α_{ij} indicates the priority of task j with respect to task i.

- $\alpha_{ij} = 1$ corresponds to the case where task j has a strict higher priority with respect to task i ($j \triangleright i$).
- $0 < \alpha_{ij} < 1$ corresponds to a soft (non-strict) priority between the two tasks: the larger the value of α_{ij}, the higher the importance level of task j with respect to task i.
- $\alpha_{ij} = 0$ corresponds to the case where task i is not at all restricted by task j.

2.2.1 Task insertion and deletion

There is a particular case induced by the proposed parametrization, which corresponds to the influence of task i on itself. This self-influence can be interpreted in terms of task existence, modulated by α_{ii}.

- If $\alpha_{ii} = 1$, then task i has a strict higher priority over itself, or in other words the task is projected into its own null space, which means the task is deactivated.
- If $\alpha_{ii} = 0$, then task i is not restricted by itself, which means the task is fully activated.
- If $0 < \alpha_{ii} < 1$, then the task is partially activated. Decreasing α_{ii} from 1 to 0 implies that the task is introduced in the set of activated tasks gradually. Increasing α_{ii} from 0 to 1 implies that the task is removed from the set of activated tasks gradually.

When being added or suppressed, the influence of task i with respect to other tasks also has to be defined and this can be done by the regulation of α_{ij}.

3 Control problem formulation

The hierarchical control framework proposed in this paper extends the quasi-static torque control framework introduced in [14], which is summarized in section 3.1. This paper relies on a quasi-static control because it is fast enough to achieve real-time control of robots with a high number of DoF. Section 3.2 summarizes the generalized projector developed in [15], which is implemented in the control framework in section 3.3 to achieve a quasi-static hierarchical control.

3.1 Quasi-static control with weighting strategy

The quasi-static control framework in [14] handles multiple prioritized tasks subject to equality and inequality constraints. This multi-objective control problem is formulated as a QP problem, where all the task objectives and constraints are solved simultaneously in one QP. More specifically, this approach first solves the QP for optimal task wrenches, and then it applies the Jacobian-transpose method to compute joint torques that are equivalent to the optimized task wrenches.

The QP problem is formulated as

$$\underset{\boldsymbol{w}_{t,i}}{\arg\min} \sum_{i \in \mathcal{I}/\mathcal{N}} \left\| \boldsymbol{w}^d_{t,i} - \boldsymbol{w}_{t,i} \right\|^2_{\mathbf{Q}_{t_i}} + \sum_{i \in \mathcal{N}} \left\| \boldsymbol{w}_{t,i} \right\|^2_{\mathbf{Q}_{r_i}} \tag{5a}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}} \mathbf{J}^{rt^T}_{t_i} \boldsymbol{w}_{t_i} + \boldsymbol{g}^{rt} = \mathbf{0} \tag{5b}$$

$$\mathbf{G}\boldsymbol{w}_t \leq \mathbf{h}, \tag{5c}$$

where the matrices $\mathbf{Q}_{t_i}$ and $\mathbf{Q}_{r_i}$ are diagonal weighting matrices with $\mathbf{Q}_{t_i} = \omega_{t_i}\mathbf{I}_{m_i}$, $\mathbf{Q}_{r_i} = \omega_{r_i}\mathbf{I}_{m_i}$. Here ω is the scalar parameter of a task weight, $\mathbf{I}_a$ is the $a \times a$ identity matrix, and m_i is the dimension of task i. The norms of the wrench errors of goal directed tasks are minimized to achieve a compromise among all these tasks weighted by $\mathbf{Q}_{t_i}$. If a task i is more important than another task j, then $\omega_{t_i} > \omega_{t_j}$.

$\mathbf{Q}_{r_i}$ is the weighting matrix of the regulation term, which minimizes the norm of wrench variables of non goal directed tasks. For a standing humanoid robot, the non goal directed tasks may include the foot contact tasks and the whole-body posture task. As the redundancy of the humanoid robot may allow multiple control input solutions satisfying the same task objectives, this regulation term is useful for ensuring the uniqueness of the solution. As the regulation term may increase the error of goal directed tasks, ω_{r_i} is usually set to a very small value compared to ω_{t_i}.

The equality constraint (5b) is the static equilibrium of the root body under $\boldsymbol{w}_{t,i}$ and $\boldsymbol{g}$. The superscript rt stands for the root (free-floating base) DoF and (5b) corresponds to the six unactuated lines in (1).

The matrix $\mathbf{G}$ and the vector $\boldsymbol{h}$ in (5c) express some other equality or inequality constraints, such as non-sliding contact constraints and bounds on wrench

variables or on joint torques. For example, joint torque bound constraints can be formulated as

$$\underline{\boldsymbol{\tau}} \leq \sum_i \mathbf{J}_{t_i}^{acT} \boldsymbol{w}_{t_i} + \boldsymbol{g}^{ac} \leq \overline{\boldsymbol{\tau}}, \tag{6}$$

where $\underline{\boldsymbol{\tau}}$ and $\overline{\boldsymbol{\tau}}$ are the lower and upper bounds of $\boldsymbol{\tau}$. The superscript ac denotes the actuated DoF, which correspond to the actuated lines in (1).

Let $\boldsymbol{w}_{t_i}^*$ denotes the solution of (5). Joint torques are computed as follows

$$\boldsymbol{\tau} = \sum_i \mathbf{J}_{t_i}^{acT} \boldsymbol{w}_{t_i}^* + \boldsymbol{g}^{ac}. \tag{7}$$

In (5), a weighting strategy is used to handle a lexicographic hierarchy of multiple prioritized tasks, and strict priority cannot be achieved. This control framework is extended in section 3.3 to allow one to control the priority between each pair of tasks separately and to change the priority gradually from a non strict case to a strict case. This is achieved by using the generalized projector explained in section 3.2.

3.2 Generalized projector

The generalized projector $\mathbf{P}_i(\mathbf{A}_i) \in \mathbb{R}^{n\times n}$ introduced in [15] can be used here to modify task torques $\boldsymbol{\tau}_i$ by an appropriate projection $(\mathbf{P}_i(\mathbf{A}_i)\boldsymbol{\tau}_i)$ to account for the hierarchy information contained in $\mathbf{A}_i$. This generalized projector can completely or partially project a task in the null-space of other tasks. It can handle both strict and non-strict priorities, since it allows the precise regulation of how much a task is affected by other tasks. This section provides a short outline of the computation of $\mathbf{P}_i(\mathbf{A}_i) \in \mathbb{R}^{n\times n}$ as needed in this paper. More details of this computation can be found in [15].

In order to compute the generalized projector $\mathbf{P}_i(\mathbf{A}_i)$, a preliminary processing of $\mathbf{A}_i$ and of the augmented Jacobian $\mathbf{J}$, which concatenates the Jacobian matrices of all the n_t tasks in a hierarchy ($\mathbf{J} = \left[\mathbf{J}_1^T \dots \mathbf{J}_j^T \dots \mathbf{J}_{n_t}^T\right]^T$), is carried out according to the priorities of all the tasks with respect to task i. As each row of $\mathbf{J}$ is associated to the same row in $\mathbf{A}_i$, the rows of $\mathbf{J}$ can be sorted in descending order with respect to the values of the diagonal elements in $\mathbf{A}_i$. The resulting matrix $\mathbf{J}_{s_i}$ is thus constructed so that tasks which should be the least influenced by task i appear in its first rows, while tasks which can be the most influenced by task i appear in its last rows. The values in $\mathbf{A}_i$ are sorted accordingly, leading to $\mathbf{A}_i^s$, the diagonal elements of which are organized in descending order starting from the first row.

Based on $\mathbf{J}_{s_i}$, a projector into the null space of $\mathbf{J}$ can be computed. This can be done by first computing a matrix $\mathbf{B}_i(\mathbf{J}_{s_i}) \in \mathbb{R}^{r\times n}$, where $r = rank(\mathbf{J}_{s_i})$ is the rank of $\mathbf{J}_{s_i}$. The rows of $\mathbf{B}_i(\mathbf{J}_{s_i})$ form an orthonormal basis of the joint space obtained by using elementary row transformations on $\mathbf{J}_{s_i}$. Then this projector can be computed as $\mathbf{P}_i^{'} = \mathbf{I}_n - \mathbf{B}_i^T\mathbf{B}_i$. When performing task i by using the projected joint torques $\mathbf{P}_i^{'}\boldsymbol{\tau}_i = (\mathbf{J}_i\mathbf{P}_i^{'})^T\boldsymbol{w}_i$, the projector $\mathbf{P}_i^{'}$ basically cancels any joint torque that impacts all the n_t tasks, including task i itself.

The computation of the projector $\mathbf{P}_i^{'}$ can be modified such that tasks having strict priority over task i are perfectly accounted for; tasks over which task i has a strict priority are not considered; and all other tasks with soft priorities are accounted for, according to the value of their respective priority parameters in $\mathbf{A}_i$. The generalized projector accounting for all these requirements is given by

$$\mathbf{P}_i(\mathbf{A}_i) = \mathbf{I}_n - \mathbf{B}_i(\mathbf{J}_{s_i})^T \mathbf{A}_{i,r}^s(\mathbf{A}_i, \boldsymbol{origin})\mathbf{B}_i(\mathbf{J}_{s_i}), \tag{8}$$

where $\mathbf{A}_{i,r}^s$ is a diagonal matrix of degree r. The vector $\boldsymbol{origin} \in \mathbb{R}^r$ is a vector of the row indexes of $\mathbf{J}_{s_i}$ selected during the construction of the orthonormal basis $\mathbf{B}_i$. Each of these r rows in $\mathbf{J}_{s_i}$ is linearly independent to all the previously selected ones. The diagonal elements of $\mathbf{A}_{i,r}^s$ are restricted to the r diagonal elements of $\mathbf{A}_i^s$, which correspond to the r rows of $\mathbf{J}_{s_i}$, the row indexes of which belong to $\boldsymbol{origin}$.

Note that by varying the value of each α_{ij} in $\mathbf{A}_i$, one can regulate the priority of each task j in the n_t tasks with respect to task i separately. Moreover, if $\alpha_{ii} = 1$, then task i is projected into its own null-space, *i.e.* it is essentially canceled out. Decreasing α_{ii} continuously from 1 to 0 activates task i gradually. Conversely, increasing α_{ii} continuously from 0 to 1 provides one with a proper task deletion procedure.

3.3 Generalized Quasi-Static Hierarchical Control

The control framework presented in Section 3.1 is extended here to account for both strict and non strict task priorities. Moreover, an advantage of this approach is that a priority rearrangement can be performed between any two tasks.

The QP problem to be solved here is

$$\arg\min_{\boldsymbol{w}_{t_i}} \sum_{i\in\mathcal{I}/\mathcal{N}} \left\|\boldsymbol{w}_{t_i}^d - \boldsymbol{w}_{t_i}\right\|_{\mathbf{I}}^2 + \sum_{i\in\mathcal{N}} \left\|\boldsymbol{w}_{t_i}\right\|_{\mathbf{Q}_{r_i}}^2 \tag{9a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}} \mathbf{P}_{t_i}(\mathbf{A}_i)\mathbf{J}_{t_i}^{rtT}\boldsymbol{w}_{t_i} + \boldsymbol{g}^{rt} = \mathbf{0} \tag{9b}$$

$$\mathbf{G}(\{\mathbf{P}_{t_i}(\mathbf{A}_i)\})\boldsymbol{w}_t \leq \mathbf{h}. \tag{9c}$$

where $\{\mathbf{P}_{t_i}(\mathbf{A}_i)\}$ is the set of generalized projectors of all the tasks.

The control input $\boldsymbol{\tau}$ is computed by using modulated task wrenches ($\mathbf{P}_{t_i}\mathbf{J}_{t_i}^{acT}\boldsymbol{w}_{t_i}$) to account for desired task hierarchies

$$\boldsymbol{\tau} = \sum_{i\in\mathcal{I}} \mathbf{P}_{t_i}(\mathbf{A}_i)\mathbf{J}_{t_i}^{acT}\boldsymbol{w}_{t_i}^{*} + \boldsymbol{g}^{ac}. \tag{10}$$

The major difference between the formulation of the proposed hierarchical control framework and that of the control framework reviewed in Section 3.1 is that each task Jacobian $\mathbf{J}_{t_i}$ is modulated by the generalized projector here to account for the desired hierarchies. As the task hierarchy in (9) is handled by generalized projectors instead of task weights, the weighting matrix $\mathbf{Q}_{t_i}$ in (5) is set to the identity matrix for goal directed task objectives in (9). The weight ω_{r_i} of the regulation term is set to a value which is very small compared to 1.

In this framework, foot contact tasks are considered as non goal directed tasks. These foot contact tasks are crucial for maintaining the balance of the robot. Their task wrenches are constrained not only by the static equilibrium (9b), but also by the linearized friction cone constraints included in (9c) to avoid foot slippage. It is important to ensure that no foot slippage is generated due to other goal directed tasks. This is achieved by setting the projectors of force contact tasks to the identity matrix ($\mathbf{P}_{t_i}(\mathbf{A}_i) = \mathbf{I}_n$ for foot contact tasks); so that in both the constraints and the computation of joint torques, these foot contact tasks are not projected into the null space of any other task.

Bounds of joint torques (11) are implemented as inequality constraints within this framework using modulated task Jacobians

$$\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} = \sum_{i\in\mathcal{I}} \mathbf{P}_{t_i}(\mathbf{A}_i)\mathbf{J}_{t_i}^{acT}\boldsymbol{w}_{t_i} + \boldsymbol{g}^{ac} \leq \overline{\boldsymbol{\tau}}. \tag{11}$$

Indeed, all the equality and inequality constraints have a higher priority over goal directed task hierarchies in this framework. This is because the constraints are expressed in terms of the modulated task wrenches accounting for desired task hierarchies; and these modulated task wrenches, which are ensured to satisfy these constraints by solving (9), are used to compute the equivalent control signal of joint torques.

4 RESULTS

The proposed control approach has been implemented on a free-floating humanoid robot iCub in simulation and a fixed-based real iCub robot. The iCub robot has 38 DoFs, including 6 DoFs of its root body, and 32 DoFs of its joints. The simulations are carried out on the simulator XDE [17], which is a software environment that manages physics simulation in real time. The QP problem (9) is solved by using the QLD solver [23]. In the experiments, the control period is $10ms$.

4.1 Task priority rearrangements for table pounding

In this experiment, the simulated iCub robot is required to stand on the ground and switch its hands to apply a contact force of $30N$ on a table periodically (see Figure 2). The table surface is connected with the ground through a spring with a stiffness of 2000 N/m and a damping of 89 Ns/m. The displacement of the spring is used to measure the hand contact force.

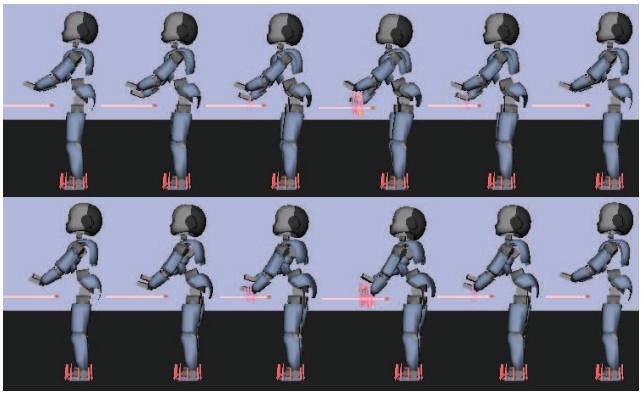

Fig. 2 Snapshots of the robot switching its hands to apply a contact force on a table periodically by using the control framework proposed in this paper.

Eighteen tasks are considered, namely the 2-D center of mass (CoM) task, the 3-D right hand (rh) and left hand (lh) position tasks, the 3-D right hand and left hand orientation tasks, the 1-D right hand force (rhf) and left hand force (lhf) tasks, the 1-D head orientation task, the 32-D whole-body posture task, the 5-D back posture task, and four 3-D contact force tasks on each feet. The static equilibrium constraint (9b) is applied to the free-floating base. Non-sliding contact constraints are applied to contact points on the feet.

During the experiment, the CoM task has the strict higher priority over all the other tasks (by setting all the $\alpha_{i,CoM} = 1$ and all the $\alpha_{CoM,i} = 0$) to ensure the balance of the robot. The posture task, which is used for redundancy resolution, is always assigned with the lowest priority. The hand orientation tasks, back posture task, and head orientation task are of lower priorities than the hand position tasks and hand force tasks. The priority relations between pairs of tasks, including the left and right hand position tasks, the left and right hand orientation tasks, the left and right hand force tasks, and the head and hand orientation tasks are left free (by setting relevant $\alpha_{ij} = \alpha_{ji} = 0$).

A finite state machine (FSM) is used to describe the switching sequence of tasks. The states are: *idle*, *rh-reaching*, *rh-contact*, *rh-release*, *lh-reaching*, *lh-contact*, *lh-release*. As the table is connected with the ground by a spring, the table surface will move downward when the hand pushes it strongly. Hand task targets during contact states are fixed on the surface of the initial table position; while the actual hand position during this state should be lower than this target position to be able to increase the contact force to $30N$. This means that during the periodic behavior of contact establishment and break between the hands and the table, priorities between hand force tasks and hand position tasks should be modified. Task priorities with respect to different states are illustrated in Figure 3.

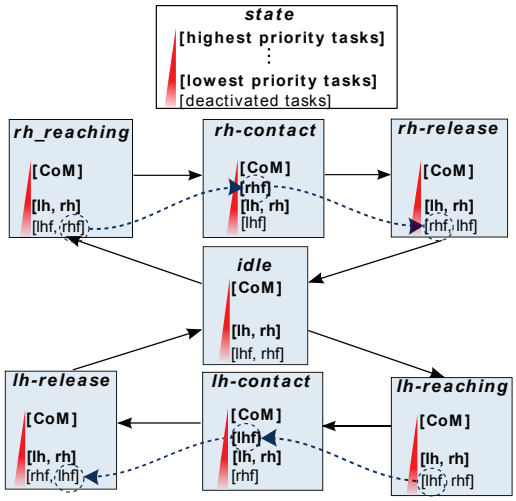

Fig. 3 Task priorities with respect to different states of the finite state machine. Priorities of the CoM task, hand position tasks, and hand force tasks are shown, and those of the other tasks are omitted for clarity.

- At the beginning, the robot is in *idle* state. During this state, its hands are not in contact with the table. The hand force tasks are deactivated, and they have a strict lower priority than hand position tasks by default.
- In *rh/lh-reaching* state, the hand moves toward the table.
- When a contact is established with the table, the FSM enters *rh/lh-contact* state. When entering this state, the hand force task is gradually activated and its priority increases gradually over hand position task to enhance the control of hand contact force.
- When *rh/lh-release* state starts, the hand should move away from the table to a target position above it. When entering this state, the hand force task is gradually deactivated and its priority with respect to hand position task decreases to enhance hand position control.

The following functions are used for the smooth variation of an α_{ij} (conversely α_{ji}) from 0 to 1 during the transition time period ($[t_1, t_2]$)

$$\begin{aligned} \alpha_{ij}(t) &= 0.5 - 0.5\cos\left(\frac{t-t_1}{t_2-t_1}\pi\right), \quad t \in [t_1, t_2], \\ \alpha_{ji}(t) &= 1 - \alpha_{ij}(t). \end{aligned} \tag{12}$$

The experiment is first conducted with the hierarchy rearrangement period ($t_2 - t_1$) being set to $0.6s$. The result of α, hand contact forces, as well as the errors of the CoM and the hand position tasks are shown in Figure 4. At the beginning, $\alpha_{lhf,lhf} = 1$ and $\alpha_{rhf,rhf} = 1$, which means that the force tasks are deactivated since they are projected in their own null-spaces. When the hand touches the table, $\alpha_{lhf,lhf}$ (or $\alpha_{rhf,rhf}$) decreases to zero smoothly to activate the force task gradually. During the contact phase, $\alpha_{rhf,rh}$ (or $\alpha_{lhf,lh}$) decreases to zero and $\alpha_{rh,rhf}$ (or $\alpha_{lh,lhf}$) increases smoothly so that the priority of hand force task increases gradually over hand position task. After this hierarchy rearrangement, as can be observed in Figure 4, the hand task error increases while the force task tracks its reference well.

Moreover, during the experiment, the equilibrium of the robot is maintained and no foot slippage is observed, which illustrates the fact that this approach can handle a task hierarchy subject to both equality constraint (static equilibrium) and inequality constraint (non-sliding contacts).

An advantage of this approach is that the rearrangement of task hierarchy can be carried out gradually and more smoothly to avoid abrupt hierarchy changes and thus reduce system instability. To demonstrate this, the same experiment is carried out with a sudden change of relevant αs (during $0.015s$ which is much faster than in the previous experiment). The resulting hand contact forces are shown in Figure 5. Hand force task errors with both gradual and sudden hierarchy rearrangements are shown in Figure 6. The energy consumption measured by the sum of squares of the joint torques ($\boldsymbol{\tau}^T\boldsymbol{\tau}$) is shown in Figure 7.

Figure 5, 6, and 7 show that larger force task errors with larger peaks and more energy consumptions (larger squared sum of joint torques) can be observed when hierarchies are rearranged suddenly, compared with the previous case where hierarchies are changed more slowly by smoother variations of α_{ij}. The desired transition duration should be defined by user requirements. It can be related to criteria such as less energy consumption, less joint jerks, etc. The point here is

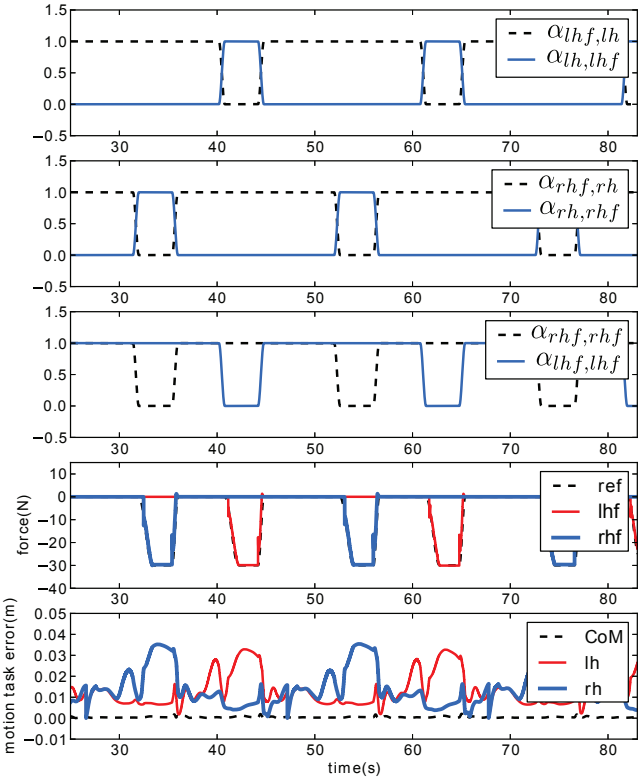

Fig. 4 Change of α (top), desired and real hand contact forces (middle), and the errors of the CoM and the hand position tasks (bottom). Hierarchy rearrangement period lasts $0.6s$.

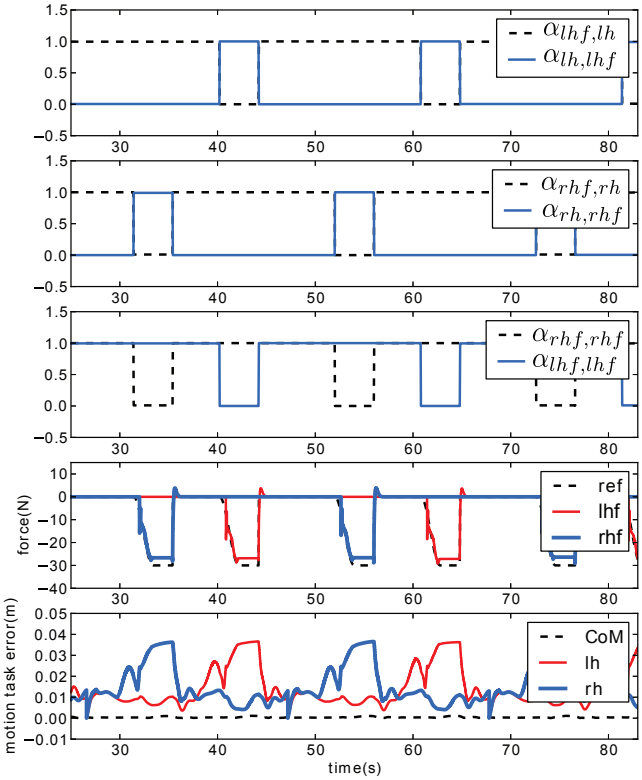

Fig. 5 Change of α (top), desired and real hand contact forces (middle), and the errors of the CoM and the hand position tasks (bottom). Hierarchy rearrangement period lasts $0.015s$.

to show that the proposed approach provides a way to change hierarchy rearrangement speed, and to show that a slower transition actually reduces energy consumption.

4.1.1 Computation time

The computation time for the proposed control algorithm for the table pounding experiment presented in section 4.1 is shown in Figure 8. It can be seen in Figure 8 that for the iCub robot with $n = 38$ DoF performing $k = 18$ tasks, and with the total task dimension of $m = \sum_{i \in \mathcal{I}} m_i = 78$, the computation time for the control algorithm is within $10ms$ (cpu-time $= 1.52 \pm 0.36$ ms, max $= 6.14$ ms). Compared to the control framework presented in [15], which needs a computation time with the order of magnitude of $80ms$ for the iCub robot performing 5 tasks, the computation time of the framework proposed in this paper is largely reduced. This is mainly because, for the approach in [15], the dimension of the optimization variable for the shown simulation is $kn = 684$ (task number multiplied with robot DoF), whereas for the approach proposed in this paper, this dimension is reduced to $m = 78$ (total task dimension). Note that the computation time provided here is the result without any optimization of program efficiency, leaving some space to further reduce the computation time of the control algorithm.

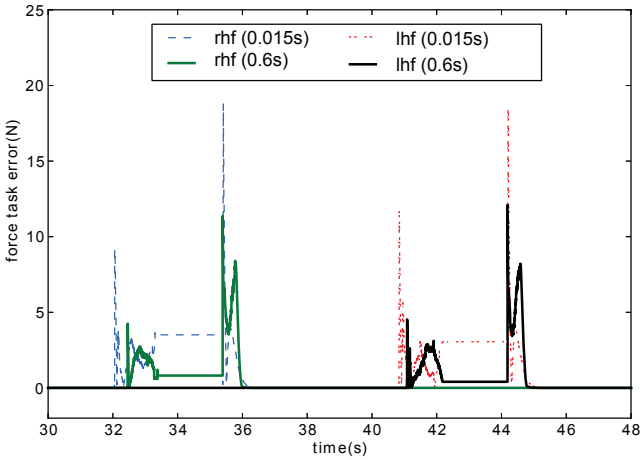

Fig. 6 Hand contact force errors with a slower hierarchy rearrangement of $0.6s$ (solid lines) and the faster rearrangement of $0.015s$ (dotted lines).

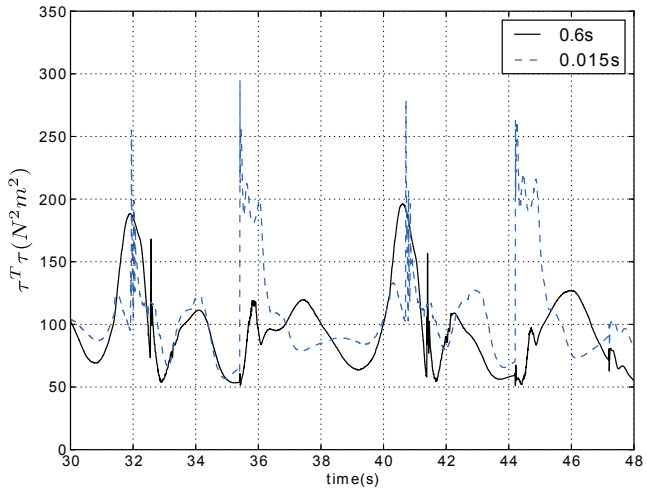

Fig. 7 Squared sum of joint torques. Faster hierarchy rearrangement (dotted line) consumes more energy compared to slower hierarchy rearrangement (solid line).

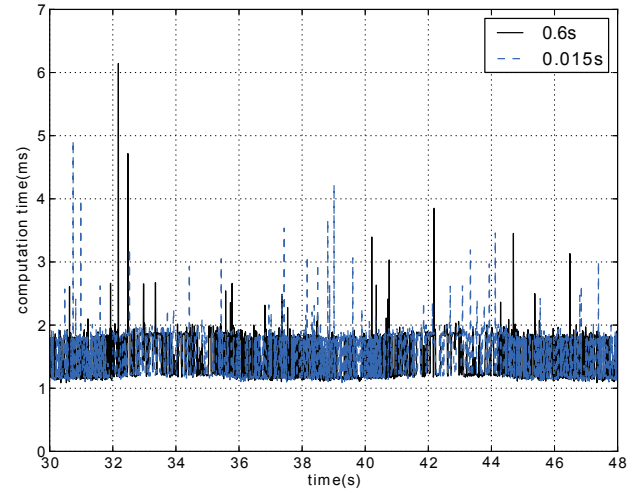

Fig. 8 Computation time for the control algorithm for the table pounding experiment.

4.2 Experiments on a real humanoid robot

In this section, the proposed approach is applied to control a fixed based iCub robot to perform multiple prioritized motion tasks (see Figure 9). The experiments show task performances during priority switching. Moreover, the influence of priority parameters on steady state task errors are studied.

4.2.1 Experimental setup

In the experiments, four tasks are controlled simultaneously:

- a 3-D head position task: the head task target is a static position to keep the head up;

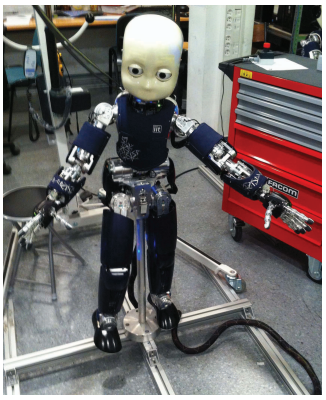

Fig. 9 The iCub robot being controlled by the proposed approach to perform a head position task, a left hand position task, a posture task, and a joint limit avoidance task.

- a 3-D left hand (lh) position task: the lh task target is a static position in front on the right of the robot, and this task can be in conflict with the head task, because it requires the robot to lean its upper body, including the head, forward in order to reach the left hand target;
- a 32-D whole-body posture task: the posture task handles posture redundancy;
- a joint limit (jl) avoidance task: a task for the avoidance of joint limits, and the target position for each considered joint is its neutral position.

The priority between the left hand task and the head task is varied by different values of $\alpha_{lh,head}$ and $\alpha_{head,lh}$. The posture task has the lowest priority, with $\alpha_{posture,i} = 1$ and $\alpha_{i,posture} = 0$ with $i \in \{lh, head, jl\}$. The handling of joint limit is achieved by changing the priority of the joint limit avoidance task. By default, this task has a low priority. But whenever a joint is close to its limit, the priority of the task corresponding to this joint is increased with respect to all the other tasks to draw the joint position away from its limit.

Note that the output of the control framework proposed in this paper is joint torques. There are two ways to illustrate the performance of a torque control framework on robots, as mentioned in [21]. The first way is to apply the control signal $\boldsymbol{\tau}$ to the simulated robot, and integrate $\ddot{\boldsymbol{q}}$ resulting from the simulation to control the real robot; the second way is to directly use $\boldsymbol{\tau}$ to control the real robot (this is clearly the most appropriate way). Since the iCub used in this experiment is a position controlled robot, the demonstration of the control algorithm on this robot is achieved using the first method.

4.2.2 Priority switching on real robot

In this experiment, the priorities between the left hand task and the head task are switched. The switching is achieved by the continuous variation of $\alpha_{lh,head}$ from 0 to 1 and $\alpha_{head,lh}$ from 1 to 0 using (12). Figure 10 shows the errors of the two tasks. It can be seen that

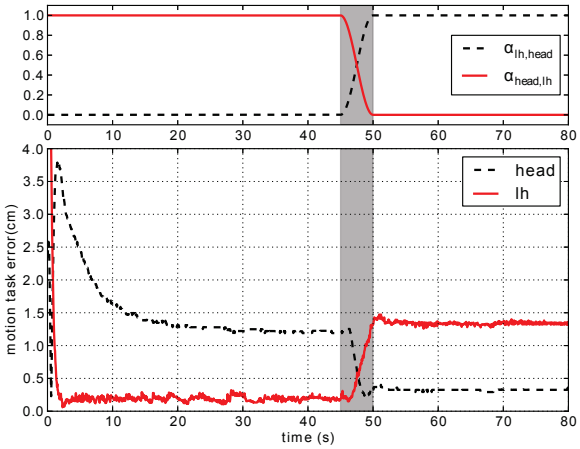

Fig. 10 Variation of priority parameters (top) and the errors of the head task and the left hand task (bottom) with priority switching between the two tasks.

the switch of task priorities are successfully achieved.

4.2.3 Influence of priority parameters on steady state task errors

In the previously mentioned experiments, the priority parameters of each task pair satisfy $\alpha_{ij} + \alpha_{ji} = 1$ at all the time. However, $\alpha_{ij} + \alpha_{ji} = 1$ is not a necessary condition for task priority assignments. Indeed, the error of task i is related not only to how task i is restricted by other tasks through α_{ij}, but also to how task i is allowed to influence other tasks through α_{ji}. To explain the influence of α_{ij} on task errors in detail, the steady state task errors with respect to different choices of priority parameter values are studied here.

Figure 11 presents the errors of the head task and the left hand task with respect to different values of $\alpha_{lh,head}$ and $\alpha_{head,lh}$. It can be seen in Figure 11 that when $\alpha_{lh,head} = 0$ and $\alpha_{head,lh} = 1$, the steady state left hand task error is very small. Generally, there are different ways to reduce the head task error. The first way is to increase $\alpha_{lh,head}$, which leads to the left hand task being restricted more by the head task, and the head task being less affected by the left hand task. The second way is to decrease $\alpha_{head,lh}$, which leads to the head task being less restricted by the left hand task, and the left hand task being more affected by the head

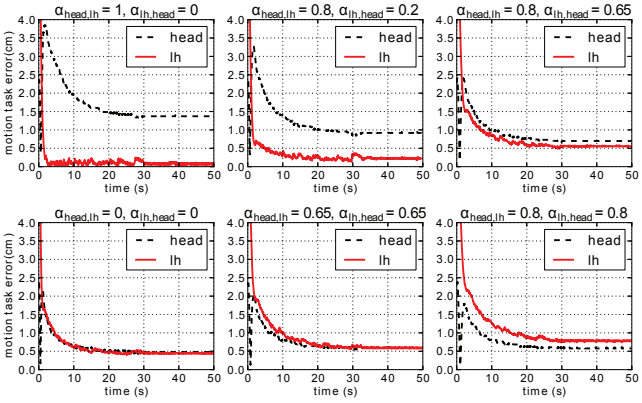

Fig. 11 Errors of the head task and the left hand task with respect to different priority parameter values.

task. For example, when $\alpha_{lh,head}$ increases from 0.2 to 0.65 with $\alpha_{head,lh}$ fixed at 0.8, the steady state error of the left hand task is increased, and the steady state error of the head task is decreased.

The results in Figure 11 also show that task performances are not simply influenced by the ratio between α_{ij} and α_{ji}. For example, when $\alpha_{lh,head} = \alpha_{head,lh} = a \leq 0.65$, the errors of the two tasks are close. But when a is inceased to a high value of 0.8, both tasks are restricted a lot by each other, and such restriction affects especially the left hand task in this example.

Note that the proposed approach parametrizes task priorities in a continuous way and can encode priorities between each pair of tasks, therefore, it is richer and more informative compared with a discrete parametrization used by strict priorities.

5 Conclusions and Future Works

This paper proposes a hierarchical control approach to handle multiple motion and force tasks for a humanoid robot. The generalized projector is used to precisely regulate how much a task can influence or be influenced by other tasks through the modulation of a priority matrix. The same mechanism is used to activate and deactivate tasks.

Experiments demonstrate that both motion and contact force tasks of different priorities can be handled by this approach. Task priorities can be maintained and switched, and the switching duration can be adjusted to achieve smoother hierarchy rearrangement. The computation cost is largely reduced compared to the previous work [15].

The proposed approach provides the possibility to automatically regulate a complex priority network, since this approach uses a continuous parametrization of the priority between each pair of tasks. A potential appli-

cation of the proposed framework could be to combine with robot learning techniques to incrementally learn and improve the tuning of priority parameters for different scenarios of interactions with the environment.

Acknowledgements This work was partially supported by the European Commission, within the CoDyCo project (FP7-ICT-2011-9, No. 600716) and by the RTE company through the RTE/UPMC chair "Robotics Systems for field intervention in constrained environment" held by Vincent Padois.

References

1. Abe Y, da Silva M, Popović J (2007) Multiobjective control with frictional contacts. In: Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation, pp 249–258
2. Bouyarmane K, Kheddar A (2011) Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp 4414–4419, DOI 10.1109/IROS.2011.6094483
3. Collette C, Micaelli A, Andriot C, Lemerle P (2007) Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts. In: 7th IEEE-RAS International Conference on Humanoid Robots, pp 81–88
4. Dietrich A, Albu-Schaffer A, Hirzinger G (2012) On continuous null space projections for torque-based, hierarchical, multi-objective manipulation. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp 2978–2985, DOI 10.1109/ICRA.2012.6224571
5. Escande A, Mansard N, Wieber PB (2014) Hierarchical quadratic programming: Fast online humanoid-robot motion generation. The International Journal of Robotics Research p 0278364914521306
6. Flacco F, De Luca A, Khatib O (2012) Prioritized multi-task motion control of redundant robots under hard joint constraints. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pp 3970–3977, DOI 10.1109/IROS.2012.6385619
7. Hsu P, Mauser J, Sastry S (1989) Dynamic control of redundant manipulators. Journal of Robotic Systems 6(2):133–148, DOI 10.1002/rob.4620060203, URL http://dx.doi.org/10.1002/rob.4620060203
8. Jarquin G, Escande A, Arechavaleta G, Moulard T, Yoshida E, Parra-Vega V (2013) Real-time smooth task transitions for hierarchical inverse kinematics. In: 13th IEEE-RAS International Conference on Humanoid Robots, pp 528–533, DOI 10.1109/HUMANOIDS.2013.7030024
9. Kanoun O, Lamiraux F, Wieber PB, Kanehiro F, Yoshida E, Laumond JP (2009) Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots. In: IEEE International Conference on Robotics and Automation (2009), pp 2939–2944
10. Keith F, Wieber N P-Band Mansard, Kheddar A (2011) Analysis of the discontinuities in prioritized tasks-space control under discreet task scheduling operations. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3887–3892, DOI 10.1109/IROS.2011.6094706
11. Khatib O (1987) A unified approach for motion and force control of robot manipulators: The operational space formulation. Robotics and Automation, IEEE Journal of 3(1):43–53
12. Lee J, Mansard N, Park J (2012) Intermediate desired value approach for task transition of robots in kinematic control. Robotics, IEEE Transactions on 28(6):1260–1277, DOI 10.1109/TRO.2012.2210293
13. Liu M, Micaelli A, Evrard P, Escande A, Andriot C (2011) Interactive dynamics and balance of a virtual character during manipulation tasks. In: IEEE International Conference on Robotics and Automation (ICRA), pp 1676–1682
14. Liu M, Micaelli A, Evrard P, Escande A, Andriot C (2012) Interactive virtual humans: A two-level prioritized control framework with wrench bounds. IEEE Transactions on Robotics 28(6):1309–1322, DOI 10.1109/TRO.2012.2208829
15. Liu M, Tan Y, Padois V (2015) Generalized hierarchical control. Autonomous Robots pp 1–15, DOI 10.1007/s10514-015-9436-1, URL http://dx.doi.org/10.1007/s10514-015-9436-1
16. Mansard N, Remazeilles A, Chaumette F (2009) Continuity of varying-feature-set control laws. Automatic Control, IEEE Transactions on 54(11):2493–2505, DOI 10.1109/TAC.2009.2031202
17. Merlhiot X, Le Garrec J, Saupin G, G A (2012) The xde mechanical kernel: Efficient and robust simulation of multibody dynamics with intermittent nonsmooth contacts. In: The Second Joint International Conference on Multibody System Dynamics
18. Mistry M, Nakanishi J, Schaal S (2007) Task space control with prioritization for balance and locomotion. In: Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, pp 331–338, DOI 10.1109/IROS.2007.4399595
19. Petrič T, Žlajpah L (2013) Smooth continuous transition between tasks on a kine-

matic control level: Obstacle avoidance as a control problem. Robotics and Autonomous Systems 61(9):948–959, DOI http://dx.doi.org/10.1016/j.robot.2013.04.019
20. Saab L, Mansard N, Keith F, Fourquet JY, Soueres P (2011) Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints. In: IEEE International Conference on Robotics and Automation (ICRA), pp 1091–1096
21. Saab L, Ramos O, Keith F, Mansard N, Soueres P, Fourquet JY (2013) Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. Robotics, IEEE Transactions on 29(2):346–362, DOI 10.1109/TRO.2012.2234351
22. Salini J, Padois V, Bidaud P (2011) Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp 1283 –1290, DOI 10.1109/ICRA.2011.5980202
23. Schittkowski (1986) QLD: A FORTRAN Code for Quadratic Programming, users guide. Mathematisches Institut, Universität Bayreuth, Germany, Tech. Rep.
24. Sentis L, Khatib O (2004) Prioritized multi-objective dynamics and control of robots in human environments. In: 4th IEEE/RAS International Conference on Humanoid Robots, vol 2, pp 764–780 Vol. 2, DOI 10.1109/ICHR.2004.1442684
25. Stephens B, Atkeson C (2010) Dynamic balance force control for compliant humanoid robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 1248–1255

Mingxing Liu is a post-doctoral fellow at Institut des Systèmes Intelligents et de Robotique (ISIR, UMR CNRS 7222) at Université Pierre et Marie Curie (UPMC), Paris, France. In 2009, she received both her engineering degree from Ecole Centrale Paris, France and her master's degree in Automatic Control from Shanghai Jiao Tong University, China. From 2009 to 2012, she is a PhD student at the Systems and Technologies Integration Laboratory of the French Atomic Energy Commission (CEA-LIST). She received the Ph.D. degree in Robotics in 2013 from UPMC. Her research interests include automatic control, whole-body control for redundant robots such as industrial manipulators and humanoid robots, as well as human-robot interaction.

Ryan Lober is a doctoral fellow at the Institut des Systèmes Intelligents et de Robotique (ISIR, UMR CNRS 7222) at the Université Pierre et Marie Curie (UPMC) in Paris, France. In 2011, he received his Bachelor of Science in Mechanical Engineering from The Georgia Institute of Technology in Atlanta, Georgia USA. In 2013, he received his Master of Science in Robotics and Systems from the École Nationale Supérieure d'Arts et Métiers in Paris, France. His research focuses on how whole-body control schemes and machine learning can be integrated in order solve problems that would be otherwise intractable using one or the other. One example involves the management and optimization of multiple tasks for humanoid robots.

Vincent Padois is an associated professor of Robotics and Computer Science and a member of the Institut des Systèmes Intelligents et de Robotique (ISIR, UMR CNRS 7222) at Université Pierre et Marie Curie in Paris, France. In 2001, he receives both an engineering degree from the Ecole Nationale d'Ingénieurs de Tarbes (ENIT), France and his master's degree in Automatic Control from the Institut National Polytechnique de Toulouse (INPT), France. From 2001 to 2005, he is a PhD student in Robotics of the ENIT/INPT Laboratoire Genie de Production. In 2006 and 2007, he is a post-doctoral fellow in the Stanford Artificial Intelligence Laboratory and more specifically in the group of Professor O. Khatib. Since 2007, his research activities at ISIR are mainly focused on the automatic design, the modeling and the control of redundant and complex systems such as wheeled mobile manipulators, humanoid robots as well as standard manipulators evolving under constraints in complex environments. He is also involved in research activities that aim at bridging the gap between adaptation and decision making techniques provided by Artificial Intelligence and low-level, reactive control. Since 2011, he holds the "Intervention Robotics" RTE/UPMC chair position.