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

Whole-Body Hierarchical Motion and Force Control for Humanoid Robots

Mingxing Liu, 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 establishes, the newly activated force task or contact constraint may interfere with other tasks. This paper aims at handling a dynamically changing hierarchy of motion and force tasks of different priority levels. A torque-based control framework is proposed, which solves a quadratic programming problem to maintain desired task hierarchies, subject to constraints. This approach can achieve simultaneous priority transitions as well as activation or deactivation of tasks. A novel generalized projector is used to regulate quantitatively 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 a free-floating humanoid robot in simulation.

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

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 freedoms of a motion and a force task are not orthogonal to each other, for example when both motion and force tasks defined at the end-effector frame require the same degrees of freedom (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, a control of each task with respect to their importance level is desired to achieve an appropriate whole-body performance. Third, if non-sliding contact 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 develops a whole-body control framework for handling prioritized motion and force tasks during physical interactions. It focuses on the generalized framework formulation that allows for the regulation of priorities between motion and force tasks, as well as the activation or deactivation of tasks.

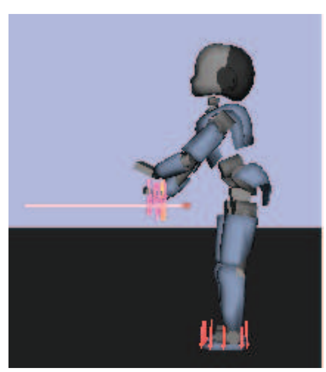

Fig. 1: Example of a humanoid robot in physical interactions with its environment.

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 [1].

The authors are with Institut des Systèmes Intelligents et de Robotique, CNRS UMR 7222 & Université Pierre et Marie Curie, 4 place Jussieu, 75252 Paris cedex 05, France. e-mail: {liu, padois}@isir.upmc.fr

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 [2] is developed for the control of center of mass (CoM) motion and contact forces of humanoid robots, where an additional 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.

Recently, some hierarchical control frameworks have been proposed for the control of redundant robots. These control frameworks can be divided into three categories. The first category refers to analytical methods based on null-space projectors [3–6]. These approaches ensure *strict task hierarchies*, which means that lower-priority tasks are performed only in the null-space of higher priority tasks. Equality constraints can be implemented by the projection into the null-space of the Jacobian of constraint equations. However, it is difficult to impose inequality constraints with these approaches since they use pseudo-inverses and projection matrices. Therefore, non-sliding constraints that restrict contact forces inside friction cones can not be properly implemented.

The second category of approaches that can handle inequality constraints is hierarchical quadratic programming (HQP) [7–9], which is applied to whole-body motion control under unilateral constraints [10]. The idea of HQP is to first solve a QP to obtain a solution for a higher priority task objective; 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. Therefore, HQP also deals with *strict task hierarchies*.

In many contexts, organizing tasks by assigning them with strict priorities is not generic, *i.e.* can have some limitations. A task may not always have a strict priority over another one, and strict priorities can sometimes be too conservative so that they may completely block lower-priority tasks. Unlike the first two categories, the third one handles *non-strict task hierarchies* using weighting strategies [11–15]. These control frameworks solve all the constraints and task objectives in one QP and provide a trade-off among task objectives with different weights. As the performances of higher priority tasks cannot be guaranteed by simply adjusting the weights of task objectives, a prioritized control framework is proposed in [16] to ensure the performance of a higher-priority task within a user defined tolerance margin. However, this approach handles priorities of only two levels.

Another 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, frequent establishment and break of contacts. In this case, a sudden rearrangement of task hierarchy may lead to a great discontinuity in control laws and the increase of system instability. Recently, different methods based on strict hierarchies have been developed to handle priority transition problems. An approach to smooth priority rearrangement between two tasks is proposed in [17,18]. Approaches for continuous and simultaneous transitions of multiple tasks are developed in [19,20]. In [19], a specific inverse operator ensures continuous inverse in the analytical computation of control laws. The approach presented in [20] is based on intermediate desired values in the task space. When the number of task transitions increases, this approach suggests to apply an approximation to reduce the computational cost. An approach of hierarchical control with continuous null-space projections is presented in [21]. In this approach, an activator associated to directions in the right singular vectors of a task Jacobian matrix is regulated to activate or deactivate these directions. However, the design of such an activator makes this approach difficult to be implemented for the separate handling of different task directions. On the other hand, priority transitions can be easily achieved within a non-strict hierarchy by the continuous variation of task weights [14].

This paper addresses such smooth hierarchy rearrangement problems. The contribution of this paper is a whole-body hierarchical control framework, which can handle both strict and non-strict hierarchies, and can perform task hierarchy rearrangements gradually so that system instability due to such rearrangements can be reduced. This control framework formulates and solves all tasks and constraints in one QP, where linear inequality constraints, such as non-sliding contacts, can be implemented. Hierarchy rearrangements within this framework are based on the use of a novel generalized projection matrix, which has been developed in our previous work [22]. This projection matrix regulates to what extent a lower-priority task is projected into the null-space of a higher-priority task. In other words, it allows a task to be completely, partially, or not at all projected into the null-space of some other tasks. The priority levels can be changed by the modulation of the generalized projector. In [22] this projector is used in a dynamic control framework on a robotic manipulator. In this paper, it is the first time that such a projector is implemented in the control of a free-floating robot.

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 on a humanoid robot are presented in Section 4 to demonstrate the framework capabilities. Future research directions regarding the potential application of the proposed approach are presented in Section 5.

2 Modeling

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

$$M(q)\ddot{q} + n(q,\dot{q}) + g(q) = S(q,\dot{q})^T \tau + J_c(q)^T w_c, \tag{1}$$

where $M(q) \in \mathbb{R}^{n\times n}$ is the generalized inertia matrix; $\dot{q} \in \mathbb{R}^n$ and $\ddot{q} \in \mathbb{R}^n$ are the vector of velocity and the vector of acceleration in generalized coordinates, respectively; $n(q,\dot{q}) \in \mathbb{R}^n$ is the vector of Coriolis and centrifugal induced joint torques; $g(q) \in \mathbb{R}^n$ is the vector of gravity induced joint torques; $S(q,\dot{q})^T \in \mathbb{R}^{n\times n_a}$ is a selection matrix for the actuated DoF; $\tau \in \mathbb{R}^{n_a}$ is the vector of the actuation torques; $J_c(q)^T = \left[J_{c,1}(q)^T \ \dots \ J_{c,n_c}(q)^T\right]$ is the transpose of a Jacobian matrix, with $J_{c,n_\beta}(q)$, the Jacobian matrix associated to a contact point β; $w_c = \left[w_{c,1}^T \ \dots \ w_{c,n_c}^T\right]^T$ is the external contact wrenches applied to the robot, with n_c the number of contact points.

2.1 Motion and force task

Consider a robot performing motion and force tasks. Each task is associated with its task wrench. For a motion task i, the desired task wrench is the output of a task space proportional-derivative (PD) controller

$$w_{t,i}^d = K_{P,i} e_i + K_{D,i} \dot{e}_i, \tag{2}$$

where e_i and $\dot{e}_i$ are task position and velocity errors, respectively; and $K_{P,i}$ and $K_{D,i}$ are symmetric, positive definite gain matrices.

For a force task, the desired task wrench is the output of a proportional-integral controller with a feed-forward term

$$w_{t,i}^d = w_{t,i}^\star + K_{P,i} e_w + K_{I,i} \int e_w dt, \tag{3}$$

where $w_{t,i}^\star$ is the desired task wrench, e_w is the error of task wrench, and $K_{P,i}$ and $K_{I,i}$ are symmetric, positive definite gain matrices.

2.2 Priority parametrization

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 α_i

$$\alpha_i = diag\left(\alpha_{i1} I_{m_1}, \dots, \alpha_{ij} I_{m_j}, \dots, \alpha_{in_t} I_{m_{n_t}}\right) \tag{4}$$

where m_j is the dimension of task j, α_i is a diagonal matrix, the main diagonal blocks of which are square matrices: $\alpha_{ij} I_{m_j}$. 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.

- $\alpha_{ij} = 0$ corresponds to the case where task j has strict lower priority with respect to task i ($i \triangleright j$).
- $0 < \alpha_{ij} < 1$ corresponds to a soft (non-strict) priority between the two tasks: the greater the value of α_{ij}, the higher the importance level 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 ($j \triangleright i$).

3 Control problem formulation

In this work, multiple tasks with different priority levels subject to equality and inequality constraints have to be handled. This kind of multi-objective control problem can be formulated as a Linear Quadratic Programming problem (LQP), where all the task objectives and constraints are solved simultaneously in one LQP. Moreover, Jacobian-transpose method is adopted here to compute joint torques that are equivalent to task wrenches w_i applied at task frames.

This Section first briefly illustrate the control framework based on LQP and Jacobian-transpose method in 3.1, then develops a generalized projector in 3.2, which is implemented in this LQP-based control framework in 3.3 for handling a user-defined task hierarchy.

3.1 Control framework based on Linear Quadratic Programming and Jacobian-transpose method

The hierarchical control framework proposed by this paper extends the multi-objective control framework introduced in [16], which implements a weighting strategy to handle the importance levels of multiple elementary tasks. This Jacobian-transpose based quasi-static control framework is adopted in this paper because it is fast enough to achieve real-time control of robots with a high number of degrees of freedom.

This control framework proposes to solve a constrained multi-objective control problem in two steps. The first step solves the LQP problem (3.1) to obtain optimal task forces and contact forces.

$$\underset{w_{t,i},w_{c,j}}{\arg\min} \sum_i \left\|w^d_{t,i} - w_{t,i}\right\|^2_{\mathbf{Q}_{t_i}} + \sum_j \left\|w_{c,j}\right\|^2_{\mathbf{Q}_{c_j}} \tag{5a}$$

$$\text{s.t. } \sum_i \mathbf{J}^{rt\,T}_{t_i} w_{t_i} + \sum_j \mathbf{J}^{rt\,T}_{c_j} w_{c_j} + g^{rt} = 0 \tag{5b}$$

$$\mathbf{G}\begin{bmatrix} w_{t_i} \\ w_{c_j} \end{bmatrix} \leq \mathbf{h}. \tag{5c}$$

where the matrices $\mathbf{Q}_{t_i}, \mathbf{Q}_{c_j}$ are diagonal weighting matrices with $\mathbf{Q}_{t_i} = \omega_{t_i}\mathbf{I}_{m_i}$ and $\mathbf{Q}_{c_j} = \omega_{c_j}\mathbf{I}_3$, with ω, the scalar parameter of a task weight, $\mathbf{I}_a$, the $a \times a$ identity matrix, and m_i, the dimension of task i. If a task i is more important than another task j, then $\omega_{t_i} > \omega_{t_j}$. The norms of the task wrench errors are minimized to achieve a compromise among all the weighted tasks. The equality constraint (5b) is the static equilibrium of the root body under $w_{t,i}$, w_{c_j}, and g, where the superscript rt stands for the root (free-floating base) DoFs. The inequality constraints (5c) may include non-sliding contact constraints and bounds on wrench variables or on joint torques

$$\underline{\tau} \leq \sum_i \mathbf{J}^{ac\,T}_{t_i} w_{t_i} + \sum_j \mathbf{J}^{ac\,T}_{c_j} w_{c_j} + g^{ac} \leq \overline{\tau}, \tag{6}$$

where $\underline{\tau}$ and $\overline{\tau}$ are the lower and upper bounds of τ.

The second step is the computation of joint torques by using the solution of the first step. Let $w^*_{t_i}$ and $w^*_{c_j}$ denote the solution of (5). Joint torques are computed as follows

$$\tau = \sum_i \mathbf{J}^{ac\,T}_{t_i} w^*_{t_i} + \sum_j \mathbf{J}^{ac\,T}_{c_j} w^*_{c_j} + g^{ac} \tag{7}$$

where the superscript ac denotes the actuated DoFs.

3.2 Projectors for hierarchical control

Strict priorities can be handled by analytical methods using a null-space projector $N_j = I - J^\dagger_j J_j$ as defined in [23], where $J^\dagger_j$ is the Moore-Penrose pseudo-inverse of the Jacobian J_j[1]. The projection of a task i in the null-space of another task j can ensure that task i is performed without producing any motion for a task j. The idea of the use of null space projections to handle strict task priorities can be generalized to handle either strict or non-strict task priorities by either a complete or a partial projection of a task in the null-space of other tasks. This generalization leads to the development of the generalized projector $P_i(\alpha_i) \in \mathbb{R}^{n\times n}$ [22], which can handle both strict and non-strict priorities in a generalized way by the precise regulation of how much a task is affected by other tasks. For a torque controlled robot, the projector $P_i(\alpha_i)$ should be able to modify task torques τ_i by an appropriate projection ($P_i(\alpha_i)\tau_i$) to account for the hierarchy information contained in α_i. This section provides a short outline of the development of the generalized projector as needed in this paper. For more details refer to [22].

In order to compute the generalized projector $P_i(\alpha_i)$, a preliminary processing of α_i and the augmented Jacobian J, which concatenates the Jacobian matrices of all the n_t tasks in a hierarchy ($J = \left[J_1^T \dots J_j^T \dots J_{n_t}^T\right]^T$), is carried out according to the priorities of all the tasks with respect to task i. As each row of J is associated to the same row in α_i, the rows of J can be sorted in descending order with respect to the values of the diagonal elements in α_i. The resulting matrix 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 α_i are sorted accordingly, leading to α^s_i, the diagonal elements of which are organized in descending order starting from the first row.

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

The computation of the projector $P^{'}_i$ 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 α_i. The generalized projector taking account of all these requirements is given by

$$P_i(\alpha_i) = I_n - B_i(J_{s_i})^T \alpha^s_{i,r}(\alpha_i, origin) B_i(J_{s_i}), \tag{8}$$

where $\alpha^s_{i,r}$ is a diagonal matrix of degree r. The vector $origin \in \mathbb{R}^r$ is a vector of the row indexes of J_{s_i} selected during the construction of the orthonormal basis B_i. Each of these r rows in J_{s_i} is linearly independent to all the previously selected ones. The diagonal elements of $\alpha^s_{i,r}$ are restricted to the r diagonal elements of α^s_i, which correspond to the r rows of J_{s_i}, the row indexes of which belong to $origin$.

Algorithm (1) and (2) summarize the construction of the generalized projector $P_i(\alpha_i)$. As any numerical scheme, tolerances are used for numerical comparison, such as ε, which is defined as the smallest value greater than zero in line #11 of Algorithm (2).

Note that by varying the value of each α_{ij} in α_i, one can regulate the priority of each task j in the n_t tasks with respect to task i separately.

[1] The dependence to q is omitted for clarity reasons.

Algorithm 1: Generalized projector computation - task i

Data: α_i, J
Result: P_i
begin
  $n \longleftarrow GetNbCol(J)$
  $index \longleftarrow GetRowsIndexDescOrder(\alpha_i)$
  $\alpha_i^s \longleftarrow SortRows(\alpha_i, index)$
  $J_{s_i} \longleftarrow SortRows(J, index)$
  $B_i, origin, r \longleftarrow GetOrthBasis(J_{s_i})$ ▷`Alg. (2)`
  $\alpha_{i,r}^s \longleftarrow GetSubDiagMatrix(\alpha_i^s, origin)$
  $P_i \longleftarrow I_n - B_i^T \alpha_{i,r}^s B_i$
  return P_i
end

Algorithm 2: Orthonormal basis computation - $GetOrthBasis(A)$

Data: A, ε
Result: B, $origin$, r
begin
  $n \longleftarrow GetNbCol(A)$
  $m \longleftarrow GetNbRow(A)$
  $i \longleftarrow 0$
  for $k \leftarrow 0$ **to** $m-1$ **do**
    if $i \geq n$ **then**
      break
      $B[i,:] \longleftarrow A[k,:]$
    for $j \leftarrow 0$ **to** $i-1$ **do**
      $B[i,:] \longleftarrow B[i,:] - \left(B[i,:]B[j,:]^T\right)B[j,:]$
    if $norm(B[i,:]) > \varepsilon$ **then**
      $B[i,:] \longleftarrow B[i,:]/norm(B[i,:])$
      $origin[i] \longleftarrow k$
      $i \longleftarrow i+1$
end
$r \longleftarrow i$
return B, $origin$, r

3.2.1 *Task insertion and deletion*

There is a particular case induced by the proposed formulation and corresponding to the influence of task i on itself. Even though not intuitive, this self-influence has to be interpreted in terms of task existence, modulated by α_{ii}. If $\alpha_{ii} = 1$ then task i is projected into its own null-space, *i.e.* it is basically canceled out. Decreasing α_{ii} continuously to 0 is a simple and elegant way to introduce the task in the set of tasks. Conversely, increasing α_{ii} continuously from 0 to 1 provides with a proper task deletion procedure. When being added or suppressed, the influence of task i with respect to other tasks also has to be defined and here again this can be done by the regulation of α_{ij}.

3.3 Hierarchical Motion and Force Control

The control framework presented in Section 3.1 can qualitatively regulate the relative importance levels of tasks through task objective weights, but it cannot precisely ensure strict priorities among tasks. This control framework is extended in this paper 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 major difference between our hierarchical control framework and the control framework reviewed in Section 3.1 is that each task Jacobian $\mathbf{J}_{t_i}$ is modulated by the generalized projector to account for the desired priority levels. Therefore, the control framework presented in (3.1) is modified here by the application of modulated Jacobians. The LQP problem to be solved is

$$\underset{w_{t_i}, w_{c_j}}{\arg\min} \quad \textstyle\sum_i \left\| w_{t_i}^d - w_{t_i} \right\|_{\mathbf{I}}^2 + \sum_j \left\| w_{c_j} \right\|_{\mathbf{Q}_{c_j}}^2 \tag{9a}$$

$$\text{s.t. } \textstyle\sum_i \mathbf{P}_{t_i} \mathbf{J}_{t_i}^{rt\,T} w_{t_i} + \sum_j \mathbf{J}_{c_j}^{rt\,T} w_{c_j} + g^{rt} = 0 \tag{9b}$$

$$\mathbf{G}(\mathbf{P}_{t_i}) \begin{bmatrix} w_{t_i} \\ w_{c_j} \end{bmatrix} \leq \mathbf{h}(\mathbf{P}_{t_i}). \tag{9c}$$

The control input τ is computed from modulated task wrenches (7).

$$\tau = \sum_i \mathbf{P}_{t_i} \mathbf{J}_{t_i}^{ac\,T} w_{t_i}^* + \sum_j \mathbf{J}_{c_j}^{ac\,T} w_{c_j}^* + g^{ac} \tag{10}$$

Here, the task objective weighting matrix Q_i is set to the identity matrix, and the task hierarchy is handled by generalized projectors. The weighting matrix $Q_r = \omega_r I_{n+n_a+3n_c}$ of the contact force objective is set to a diagonal matrix with the weight value ω_r being very small compared to 1. The contact force objective here is not a target tracking objective. It is used to ensure the uniquess of the optimization solution. The norm of the contact wrenches w_{c_j} is minimized, as their desired values are unknown *a priori*. Appropriate value of w_{c_j}, which satisfies the static equilibrium and non-sliding contact constraints, is computed by solving the LQP.

Non-sliding contact constraints are implemented as inequality constraints, where contact forces are constrained inside linearized Coulomb friction cones. Bounds of joint torques (11) can be implemented as inequality constraints within this framework using modulated task Jacobians.

$$\underline{\tau} \leq \sum_i \mathbf{P}_{t_i} \mathbf{J}_{t_i}^{ac\,T} w_{t_i} + \sum_j \mathbf{J}_{c_j}^{ac\,T} w_{c_j} + g^{ac} \leq \overline{\tau}. \tag{11}$$

It should be noticed that, in this framework, the real task wrenches applying on the system are not w_{t_i}, but those modulated by generalized projectors. As generalized projectors are always directly applied to task Jacobians instead of task wrenches, it is difficult to apply bounds or non-sliding contact constraints directly on modulated task wrenches. Therefore, crucial contacts, such as foot contacts that should not slip, are set as constraints instead of tasks.

4 RESULTS

The proposed control approach has been implemented on a free-floating humanoid robot iCub and validated by experiments. The experiments are carried out on the simulator XDE [24], which is a software environment that manages physics simulation in real time. The robot has 38 DoFs, including 6 DoFs of its root body, and 32 DoFs of its joints. It 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.

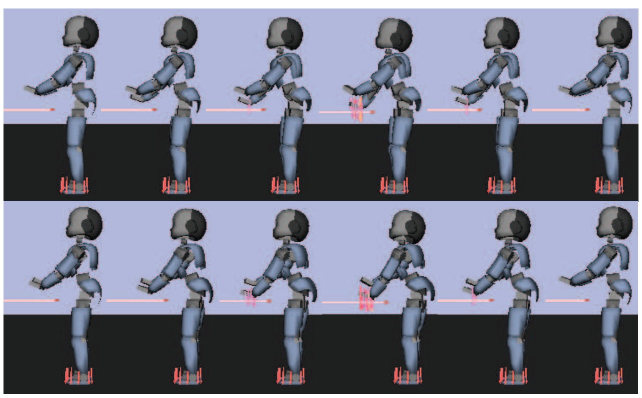

Fig. 2: Snapshots of the robot switch its hands to apply a contact force on a table periodically by using the control framework proposed in this paper.

Four 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 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 highest priority over all the other tasks to ensure the balance of the robot. The posture task, which is used for redundancy resolution, is always assigned with the lowest priority.

A finite state machine (FSM) is applied. 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 periodical 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 FSM states are illustrated in Figure 3.

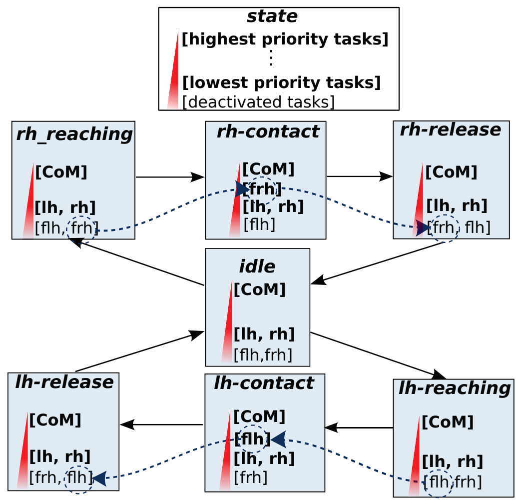

Fig. 3: Task priorities with respect to different states of the finite state machine. Priorities of the CoM task, hand position tasks, and hand force tasks are shown, and those of the other tasks are omitted for clarity.

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

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

The experiment is first conducted with the hierarchy rearrangement period ($t_2 - t_1$) being set to 0.08s. 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. During this hierarchy rearrangement, as can be observed in Figure 4, the hand task error increases while the force task tracks better its reference.

Moreover, during the experiment, the equilibrium of the robot is maintained and no foot slippage is observed, which demonstrates 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 0.03s which is much faster than in the previous experiment). The resulting hand contact forces are shown in Figure 5, and hand force task errors with both gradual and sudden hierarchy rearrangements are shown in Figure 6.

Figure 5 and 6 show that greater force task errors with larger peaks can be observed when hierarchies are rearranged suddenly, compared with the previous experiment where hierarchies are changed more slowly (by the smoother variations of α). In fact, this hierarchical force and motion control approach allows for the precious adjustment of hierarchy rearrangement speed. As this approach parametrizes task priorities in a continuous way and can encode priorities between each pair of tasks, it is richer and more informative compared with a discrete parametrization used in approaches that handle strict-priorities only (such as analytical approaches based on null-space projection and HQP). A potential application of this framework could be a combination with robot learning techniques to incrementally learn and improve the tuning of priority parameters α for different scenarios of interactions with the environment.

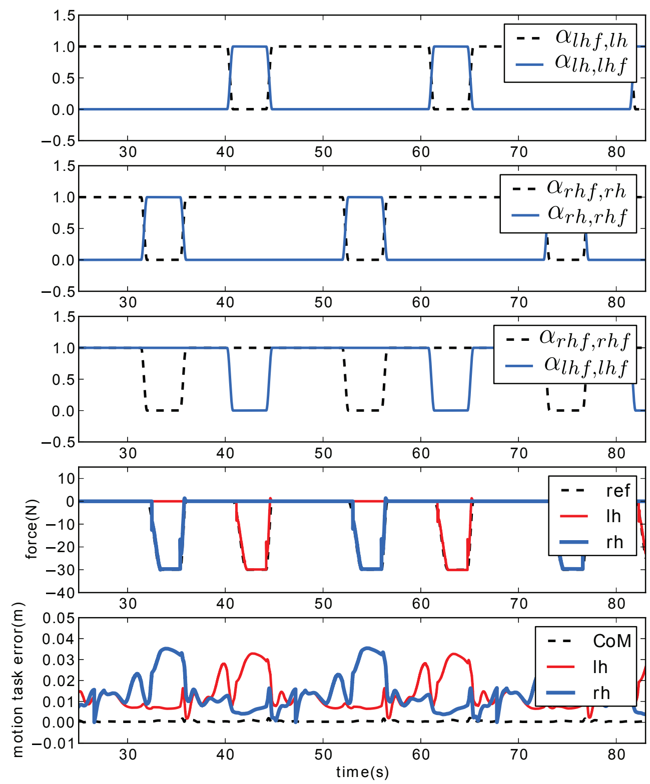

Fig. 4: Change of α (top), desired and real hand contact forces (middle), and the errors of the CoM and the hand position tasks (bottom). Hierarchy rearrangement period lasts 0.08s.

5 Conclusions and Future Works

This paper proposes a novel hierarchical control approach to handling multiple motion and force tasks for a free-floating humanoid robot. A novel 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. By using the same mechanism, tasks can be easily activated or deactivated. 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 while respecting both equality and inequality constraints.

The control framework presented here is quasi-static; however, the hierarchical control based on generalized projectors is not restricted in such a quasi-static case. In fact, it can also be used in other types of controllers, such as a dynamic controller. The basic idea is to associate each task with a task variable in joint space ($\dot{q}'_i$, $\ddot{q}'_i$, or τ'_i), then to apply generalized projectors to modulate these task variables, and finally the global joint space variable is the sum of each projected task variables ($P_i(\alpha_i)\dot{q}'_i$, $P_i(\alpha_i)\ddot{q}'_i$, or $P_i(\alpha_i)\tau'_i$).

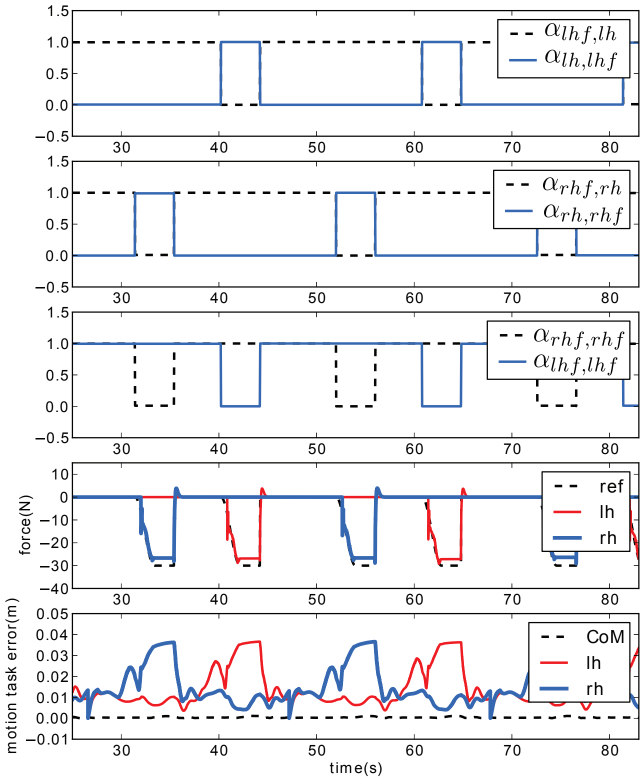

Fig. 5: Change of α (top), desired and real hand contact forces (middle), and the errors of the CoM and the hand position tasks (bottom). Hierarchy rearrangement period lasts 0.03*s*.

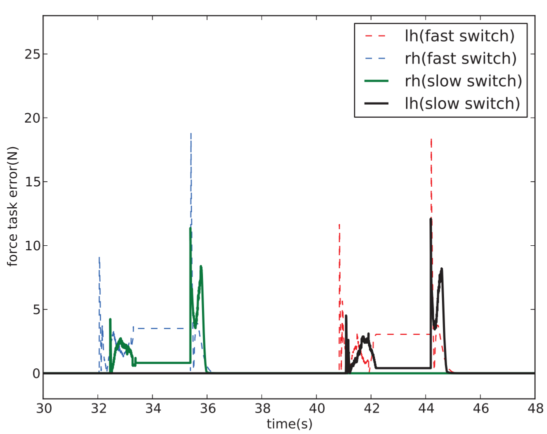

Fig. 6: Hand contact force errors with a slower hierarchy rearrangement of 0.08*s* (solid lines) and the faster rearrangement of 0.03*s* (dotted lines).

A future research direction is to extend this approach for the dynamic control of humanoid robots and to analyze its computational cost. As in the dynamic case, the relation between joint accelerations, joint torques, and contact forces has to be considered, so the optimization problem should handle more variables. Finally, as this approach uses a continuous parametrization of task priorities, the use of robot learning techniques to incrementally learn and improve the tuning of these priority parameters is also of great interest.

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. O. Khatib, Robotics and Automation, IEEE Journal of **3**(1), 43 (1987)
2. B. Stephens, C. Atkeson, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2010), pp. 1248–1255
3. L. Sentis, O. Khatib, in *4th IEEE/RAS International Conference on Humanoid Robots*, vol. 2 (2004), vol. 2, pp. 764–780 Vol. 2. DOI 10.1109/ICHR.2004.1442684
4. M. Mistry, J. Nakanishi, S. Schaal, in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (2007), pp. 331–338. DOI 10.1109/IROS.2007.4399595
5. P. Hsu, J. Mauser, S. Sastry, Journal of Robotic Systems **6**(2), 133 (1989). DOI 10.1002/rob.4620060203. URL http://dx.doi.org/10.1002/rob.4620060203
6. F. Flacco, A. De Luca, O. Khatib, in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (2012), pp. 3970–3977. DOI 10.1109/IROS.2012.6385619
7. O. Kanoun, F. Lamiraux, P.B. Wieber, F. Kanehiro, E. Yoshida, J.P. Laumond, in *IEEE International Conference on Robotics and Automation (2009)* (2009), pp. 2939–2944
8. L. Saab, N. Mansard, F. Keith, J.Y. Fourquet, P. Soueres, in *IEEE International Conference on Robotics and Automation (ICRA)* (2011), pp. 1091–1096
9. A. Escande, N. Mansard, P.B. Wieber, The International Journal of Robotics Research p. 0278364914521306 (2014)
10. L. Saab, O. Ramos, F. Keith, N. Mansard, P. Soueres, J.Y. Fourquet, Robotics, IEEE Transactions on **29**(2), 346 (2013). DOI 10.1109/TRO.2012.2234351
11. Y. Abe, M. da Silva, J. Popović, in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 249–258
12. C. Collette, A. Micaelli, C. Andriot, P. Lemerle, in *7th IEEE-RAS International Conference on Humanoid Robots* (2007), pp. 81–88
13. M. Liu, A. Micaelli, P. Evrard, A. Escande, C. Andriot, in *IEEE International Conference on Robotics and Automation (ICRA)* (2011), pp. 1676–1682
14. J. Salini, V. Padois, P. Bidaud, in *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (2011), pp. 1283 –1290. DOI 10.1109/ICRA.2011.5980202
15. K. Bouyarmane, A. Kheddar, in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (2011), pp. 4414–4419. DOI 10.1109/IROS.2011.6094483
16. M. Liu, A. Micaelli, P. Evrard, A. Escande, C. Andriot, IEEE Transactions on Robotics **28**(6), 1309 (2012). DOI 10.1109/TRO.2012.2208829

17. F. Keith, N. Wieber, P.-B.and Mansard, A. Kheddar, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2011), pp. 3887–3892. DOI 10.1109/IROS.2011.6094706
18. T. Petrič, L. Žlajpah, Robotics and Autonomous Systems **61**(9), 948 (2013). DOI http://dx.doi.org/10.1016/j.robot.2013.04.019
19. N. Mansard, A. Remazeilles, F. Chaumette, Automatic Control, IEEE Transactions on **54**(11), 2493 (2009). DOI 10.1109/TAC.2009.2031202
20. J. Lee, N. Mansard, J. Park, Robotics, IEEE Transactions on **28**(6), 1260 (2012). DOI 10.1109/TRO.2012.2210293
21. A. Dietrich, A. Albu-Schaffer, G. Hirzinger, in *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (2012), pp. 2978–2985. DOI 10.1109/ICRA.2012.6224571
22. M. Liu, Y. Tan, V. Padois, Generalized Hierarchical Control (2014). URL http://hal.archives-ouvertes.fr/hal-01068404
23. A. Liégeois, IEEE Transactions on Systems, Man and Cybernetics **7**(12), 868 (1977)
24. CEA-List. XDE eXtended Dynamic Engine. http://www.kalisteo.fr/lsi/en/aucune/a-propos-de-xde

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.

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.