



Generalized Projector for Task Priority Transitions During Hierarchical Control

Mingxing Liu, Sovannara Hak, Vincent Padois

► To cite this version:

Mingxing Liu, Sovannara Hak, Vincent Padois. Generalized Projector for Task Priority Transitions During Hierarchical Control. IEEE International Conference on Robotics and Automation, May 2015, Seattle, United States. hal-01116410

HAL Id: hal-01116410

<https://hal.science/hal-01116410>

Submitted on 13 Feb 2015

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

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

Generalized Projector for Task Priority Transitions During Hierarchical Control

Mingxing Liu, Sovannara Hak, and Vincent Padois

Abstract—Redundant robots performing multiple tasks of different priority levels are often handled by hierarchical control frameworks. Existing hierarchical controllers can handle either strict task priorities by using null space projections or a sequence of quadratic programs, or non strict task priorities by using a weighting strategy. This paper proposes a general approach to handle both strict and non-strict task priorities, and to achieve multiple priority rearrangements simultaneously. Priority modulations are achieved by the regulation of a novel generalized projector. This projector allows a task to be completely projected in the null-space of a set of tasks, while partially projected in those of some other tasks. Such a projector is implemented in an analytical control framework here. It can also be implemented in many optimization-based multi-task control frameworks. The effectiveness of this approach is demonstrated on a KUKA LWR robot performing task priority rearrangements as well as task insertion and deletion.

I. INTRODUCTION

Motion control systems for redundant robots are nowadays expected to handle multiple tasks simultaneously. As all the task objectives cannot be satisfied simultaneously, tasks are usually assigned with different priority levels, and a hierarchical control framework is usually needed for the management of task objectives of different priorities. Two main types of hierarchical control approaches are presented in the robotics literature. The first one deals with *strict task hierarchies*, in which critical tasks are fulfilled with higher priorities and lower-priority tasks are performed only in the null-space of higher priority tasks. This type of approaches includes analytical approaches based on the use of null-space projectors [1]–[5] and hierarchical quadratic programming (HQP) approaches [6]–[8]. The idea of the latter 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 while trying to satisfy lower-priority tasks at best. The second one handles *non-strict task hierarchies*, which is usually formulated as a quadratic program and the solution is a compromise among task objectives with different weights. Approaches using weighting strategies [9]–[12] are of this type.

Generally, for an approach based on strict hierarchy, the relative importance of one task with respect to another one of different priority level is parametrized in a lexicographic

way [13]. However, in many contexts, organizing tasks by assigning them with strict priorities is not generic, *i.e.* can have some limitations. First, a strict priority is just an extreme case of the relations of task importance levels. In fact, a task may not always have a strict priority over another one and it is usually difficult to define a strict hierarchy among a set of tasks. Second, strict priorities can sometimes be too conservative so that they may completely block lower-priority tasks. On the other hand, for approaches using weighting strategies, task priorities are expressed by continuous task weights. Compared with a discrete parametrization, a continuously one is richer and more informative. However, this kind of priority handling is still not generic, because it cannot be extended to the extreme case of strict priorities. Even though the work in [14] on soft constraints in model predictive control could probably be adapted to provide a way to reach strict priorities, the existing robotic applications of these frameworks do not extend to this extreme case.

In a more general context, robots may need to deal with both strict and non-strict hierarchies. Moreover, for robots acting in dynamically changing contexts, non-strict priorities may become strict ones and task priorities may have to be changed to cope with changing situations. Recently, different methods have been developed to handle priority transition problems. An approach to smooth priority rearrangement between only two levels of tasks is proposed in [15,16], which can hardly be extended to the case of simultaneous transitions among multiple priority levels. An approach for continuous and simultaneous transitions of multiple tasks is developed in [17], however the computation of the specific inverse operator required by this approach is time consuming. The approach presented in [18] 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 [19]. 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, task transitions can be easily achieved by using a weighting strategy through the continuous variation of task weights [20]. This method is used in HQP approaches to swap priorities [21], where tasks to swap are merged in the same priority and their weights are tuned to comply with the priorities before and after the transition phase. However this task transition strategy

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. {liu, hak, padois}@isir.upmc.fr

requires a set of swaps to be performed before bringing a task to the desired priority level, since each swap handles only two tasks coming from two consecutive priority levels.

The approach presented in this paper can handle both strict and non-strict hierarchies simultaneously, to change priorities from a non-strict case to a strict case, to swap priorities among multiple tasks simultaneously, as well as to insert and delete tasks in an elegant way. This approach is based on a novel generalized projector developed in our previous work [22]. This generalized projector can regulate quantitatively to what extent a lower-priority task is projected into the null-space of a higher-priority task. In other words, this generalized projector allows a task to be completely, partially, or not at all projected into the null-space of some other tasks by using a continuous priority parametrization. Moreover, this projector allows to handle multiple priority transitions among an arbitrary number of tasks easily. Indeed, priorities are defined by pairs of tasks and are encoded by a priority matrix. One advantage of such a priority representation is that a priority rearrangement can be performed between any two tasks. This choice of priority representation can handle not only a single standard lexicographic hierarchy as HQP does, but also a complex priority network. For example, it can represent two lexicographic hierarchies $1 \triangleright 2 \triangleright 3^1$ and $4 \triangleright 5 \triangleright 6$, with an additional relationship $2 \triangleright 5$, leaving the relationships among all the other pairs of tasks free.

The main contribution of this paper is to demonstrate the effectiveness of this generalized projector by its implementation in the control of a real robot. In [22], this projector is used in an optimization-based multi-objective control framework on simulated robots. In this paper, it is the first time that such a projector is implemented in an analytical control framework based on the Jacobian-transpose method, and applied on a torque controlled robot.

II. MODELING

Consider a fixed-based 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 $\mathbf{q} \in \mathbb{R}^n$ is written as follows

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

where $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the generalized inertia matrix; $\dot{\mathbf{q}} \in \mathbb{R}^n$ and $\ddot{\mathbf{q}} \in \mathbb{R}^n$ are the vector of velocity and the vector of acceleration in generalized coordinates, respectively; $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ is the vector of Coriolis, centrifugal and gravity induced joint torques; $\boldsymbol{\tau} \in \mathbb{R}^{n_a}$ is the vector of the actuation torques; $S \in \mathbb{R}^{n_a \times n}$ is a selection matrix for the actuated DoF.

A. Motion task control

Consider a robot performing motion tasks. Each task i is associated with its task torque τ_i

$$\tau_i = J_i^T (K_{p,i} \mathbf{e}_i + K_{d,i} \dot{\mathbf{e}}_i) = J_i^T \mathbf{w}_i, \quad (2)$$

¹The notation $i \triangleright j$ indicates that task i has a strict higher priority over task j .

which is the output of a task space proportional-derivative (PD) controller projected in joint space using the Jacobian-transpose method. Here \mathbf{e}_i and $\dot{\mathbf{e}}_i$ are task position and velocity errors, respectively; and $K_{p,i}$ and $K_{d,i}$ are symmetric, positive definite gain matrices. For a whole-body joint space posture task, the task Jacobian J_i is the identity matrix. For a Cartesian task, J_i represents the differential kinematics mapping from joint space to Cartesian task space. In this case, the PD control part is called task wrench \mathbf{w}_i here, and τ_i is equivalent to the wrench \mathbf{w}_i applied at the task frame.

B. 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 $\alpha_i = \text{diag}(\alpha_{i1} I_{m_1}, \dots, \alpha_{ij} I_{m_j}, \dots, \alpha_{in_t} I_{m_{n_t}})$, 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 $\alpha_{ij} \in [0, 1]$. 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 non-strict priority: 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 ($j \triangleright i$).

III. GENERALIZED PROJECTOR

Strict priorities can be handled by analytical methods using a null-space projector $N_j = I - J_j^\dagger J_j$ [23], where J_j^\dagger is the Moore-Penrose pseudo-inverse of the Jacobian J_j ². 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 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 please refer to [22].

First, consider a one-dimensional task j . The following matrix can be applied to achieve activation or deactivation of task j direction by the variation of the scalar α_{jj} .

$$N_j' = I - \alpha_{jj} \frac{J_j^T}{\|J_j\|} \frac{J_j}{\|J_j\|}, \quad (3)$$

²The dependence to \mathbf{q} is omitted for clarity reasons.

The form of the generalized projector is similar to that of the above matrix N_j , and it is extended here to handle an arbitrary number of tasks [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 ($J = [J_1^T \dots J_j^T \dots J_{n_t}^T]^T$), 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 $\alpha_{i,r}^s$, 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 $r = \text{rank}(J_{s_i})$ 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 $P'_i = I_n - B_i^T B_i$, 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 non-strict 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

$$P_i(\alpha_i) = I_n - B_i(J_{s_i})^T \alpha_{i,r}^s(\alpha_i, \text{origin}) B_i(J_{s_i}), \quad (4)$$

where $\alpha_{i,r}^s$ is a diagonal matrix of degree r . The vector $\text{origin} \in \mathbb{R}^r$ 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 $\alpha_{i,r}^s$ are restricted to the r diagonal elements of α_i^s , 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) *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

Algorithm 1: Generalized projector computation - task i

Data: α_i, J
Result: P_i

```

1 begin
2    $n \leftarrow \text{GetNbCol}(J)$ 
3    $\text{index} \leftarrow \text{GetRowsIndexDescOrder}(\alpha_i)$ 
4    $\alpha_i^s \leftarrow \text{SortRows}(\alpha_i, \text{index})$ 
5    $J_{s_i} \leftarrow \text{SortRows}(J, \text{index})$ 
6    $B_i, \text{origin}, r \leftarrow \text{GetOrthBasis}(J_{s_i}) \triangleright \text{Alg. (2)}$ 
7    $\alpha_{i,r}^s \leftarrow \text{GetSubDiagMatrix}(\alpha_i^s, \text{origin})$ 
8    $P_i \leftarrow I_n - B_i^T \alpha_{i,r}^s B_i$ 
9 return  $P_i$ 
10 end
```

Algorithm 2: Orthonormal basis computation - $\text{GetOrthBasis}(A)$

Data: A, ϵ
Result: B, origin, r

```

1 begin
2    $n \leftarrow \text{GetNbCol}(A)$ 
3    $m \leftarrow \text{GetNbRow}(A)$ 
4    $i \leftarrow 0$ 
5   for  $k \leftarrow 0$  to  $m - 1$  do
6     if  $i \geq n$  then
7       break
8      $B[i, :] \leftarrow A[k, :]$ 
9     for  $j \leftarrow 0$  to  $i - 1$  do
10       $B[i, :] \leftarrow B[i, :] - (B[i, :] B[j, :]^T) B[j, :]$ 
11      if  $\text{norm}(B[i, :]) > \epsilon$  then
12         $B[i, :] \leftarrow B[i, :] / \text{norm}(B[i, :])$ 
13         $\text{origin}[i] \leftarrow k$ 
14         $i \leftarrow i + 1$ 
15 end
16  $r \leftarrow i$ 
17 return  $B, \text{origin}, r$ 
```

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} .

IV. IMPLEMENTATION IN AN ANALYTICAL CONTROL FRAMEWORK

The control problem that solves one task i , while taking into account the influence of a set of other tasks over it, can be written as $\tau = P_i(\alpha_i) \tau_i$, where the task torque (2) is multiplied by the generalized projector defined by (4). Now consider the control problem for solving n_t tasks. In analytical control frameworks, such as the one presented in [5], a prioritized hierarchy is established through a linear combination of task torques, each of which being projected in the null-space of all the other higher priority tasks ($\tau = \sum_{i=1}^{n_t} N_i \tau_i$). Similarly, here the overall joint torques accounting for the sets of relative importance parameters $(\alpha_1, \dots, \alpha_{n_t})$ is given by the sum of the each task torque

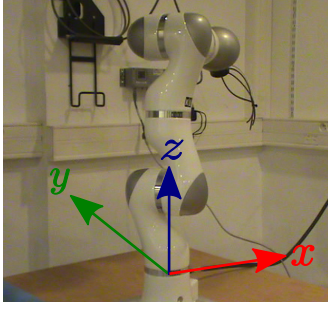


Fig. 1. The KUKA LWR robot used in the experiments. Its base frame is defined by the X,Y and Z axes shown in the figure.

multiplied by a generalized projector

$$\tau = \sum_{i=1}^{n_t} P_i(\alpha_i) \tau_i = \sum_{i=1}^{n_t} P_i(\alpha_i) J_i^T \mathbf{w}_i. \quad (5)$$

Note that when $\alpha_i = 0$ or 1, (5) becomes a controller that handles strict hierarchies.

This control approach is robust to both kinematic and algorithmic singularities. It does not require the inversion of task Jacobian matrices, therefore, it does not have problems of numerical singularities due to kinematic singularities. Moreover, unlike approaches using the pseudo-inverse of projected Jacobians ($J_i N_j$), which requires special treatment for handling algorithmic singularities when the projected Jacobian drops rank [24], this approach does not need the inversion of projected Jacobians ($J_i P_j$).

V. RESULTS

Some experiments are carried out to demonstrate that generalized projectors allow handling desired task hierarchies, task priority rearrangements, as well as the insertion and deletion of tasks. The approach presented in this paper is applied to the control of a 7-DoF KUKA Lightweight robot (LWR) shown in Fig. 1. The robot controller runs at 1 ms. This control is performed in joint torques using a real-time OS (Linux + Xenomai). The controller is implemented as a C++ OROCOS [25] component, which communicates with the control unit of the robot using the KUKA FRI. More specifically, the joint specific impedance control mode is applied, with the control law [26]

$$\tau_{\text{cmd}} = k_j(\mathbf{q}_{\text{FRI}} - \mathbf{q}_{\text{msr}}) + D(d_j) + \tau_{\text{FRI}} + \mathbf{f}_{\text{gravity}}(\mathbf{q}), \quad (6)$$

where $k_j(\mathbf{q}_{\text{FRI}} - \mathbf{q}_{\text{msr}})$ is a joint space virtual spring, $D(d_j)$ is a damping term, $\mathbf{f}_{\text{gravity}}(\mathbf{q})$ is for gravity compensation, and τ_{FRI} is a superposed control torque modifiable by the user. In our experiments, τ_{FRI} is the joint torques computed in (5).

In the experiments, three tasks as well as their targets are defined as follows.

- Task 1: a 7D posture task. The target of this task is the static posture shown in Fig. 1.
- Task 2: a 3D Cartesian position task of the end-effector. The target of this task is either a static point or a lemniscate shaped trajectory.

TABLE I
STEADY STATE TASK ERRORS

priority	1 ▷ 2 ▷ 3		
task	1	2	3
	1.94e-6 rad	0.10 m	0.19 m
priority	2 ▷ 3 ▷ 1		
task	1	2	3
	0.39 rad	9.4e-3 m	0.20 m
priority	3 ▷ 2 ▷ 1		
task	1	2	3
	0.52 rad	0.20 m	7.0e-3 m
priority	2 ▷ 1		
task	1	2	3
	0.13 rad	6.6e-3 m	0.24 m
priority	3 ▷ 2 ▷ 1		
task	1	2	3
	0.53 rad	0.20 m	5.5e-3 m

- Task 3: a 3D Cartesian position task of the elbow. The target of this task is a static point.

A local PD controller is applied for the computation of each task torque or task wrench (2). For the posture task, the PD gains are set to $k_{p,1} = 5$ N and $k_{d,1} = 0.01$ N·s; for the end-effector task and the elbow task, these gains are set to $k_{p,1} = 300$ N/m and $k_{d,1} = 80$ N·s/m. The priority matrices associated with the three tasks are: $\alpha_i = \text{diag}(\alpha_{i1} I_3, \alpha_{i2} I_3, \alpha_{i3} I_7)$, with $i=1,2,3$. For example, if the end-effector task has a strict higher priority over the elbow task, then $\alpha_{23} = 0$ and $\alpha_{32} = 1$, according to the priority parametrization convention described in subsection II-B. 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 $[t_1, t_2]$

$$\alpha_{ij}(t) = 0.5 - 0.5 \cos\left(\frac{t - t_1}{t_2 - t_1} \pi\right), \text{ with } t \in [t_1, t_2], \quad (7)$$

$$\alpha_{ji}(t) = 1 - \alpha_{ij}(t).$$

During the experiments, the robot initial posture is the same as the desired posture. The evolution of task hierarchies are defined as $1 \triangleright 2 \triangleright 3 \Rightarrow 2 \triangleright 3 \triangleright 1 \Rightarrow 3 \triangleright 2 \triangleright 1 \Rightarrow 2 \triangleright 1 \Rightarrow 3 \triangleright 2 \triangleright 1$, where the notation \Rightarrow stands for a transition of the hierarchy setting. At the beginning, the tasks, in a priority level decreasing order, are the posture task, the end-effector task, and the elbow task. Then the end-effector task priority increases over both the elbow and the posture task, becoming the task with the highest priority, and the elbow task priority increases over the posture task. Afterward, the priorities of the end-effector task and the elbow task are switched. Then the elbow task is removed from the actual task set. Finally, the elbow task is inserted as the highest priority task. The process of the insertion and the deletion of the elbow task is performed by the continuous change of α_{33} .

The evolution of α s and task errors when all the task targets are static are presented in Fig. 2. Steady state task errors for each task hierarchy configuration are shown in Table I. In these results, task errors are represented by their Euclidean norm.

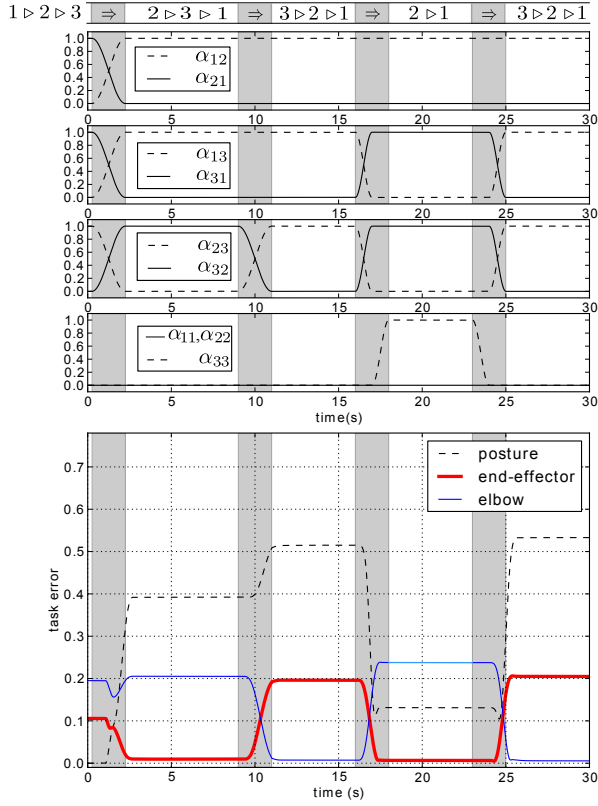


Fig. 2. Evolution of α s (top) and task errors (bottom), where task targets are static. Priority transitions as well as the insertion and deletion of the elbow task are performed.

When a lemniscate-shaped end-effector trajectory is used, the end-effector task is to move along this lemniscate orbit periodically, with an orbital period of $2\pi s$. Task errors are presented in Fig. 3. Figure 4 shows the desired and the resulting end-effector trajectory, when the end-effector task has the highest priority. A video of this experiment is attached to this paper.

These experimental results demonstrate that this approach can successfully perform priority transitions. As mentioned before, the modulation of priority matrices in generalized projectors can regulate quantitatively to what extent lower-priority tasks are projected in the null-space of higher-priority tasks. As a result, priorities can be easily rearranged by setting relevant priority parameters. The priority between two tasks i and j becomes non-strict when the value of α_{ij} is between 0 and 1. Generally, when the priority of task j increases over another task i by the augmentation of α_{ij} , its task error decreases. Such an evolution can be observed during priority transition periods in Fig. 2 and Fig. 3.

Strict priorities are achieved when α_{ij} achieves its limits (0 or 1). The results of task errors in Table I show that strict priorities are well respected. The steady state errors of highest priority tasks are very small after each hierarchy rearrangement, with the order of magnitude within 10^{-3} m for Cartesian tasks and 10^{-6} rad for posture tasks. In the experiment when the end-effector moves along a lemniscate shaped orbit, its task error is reduced when its priority is

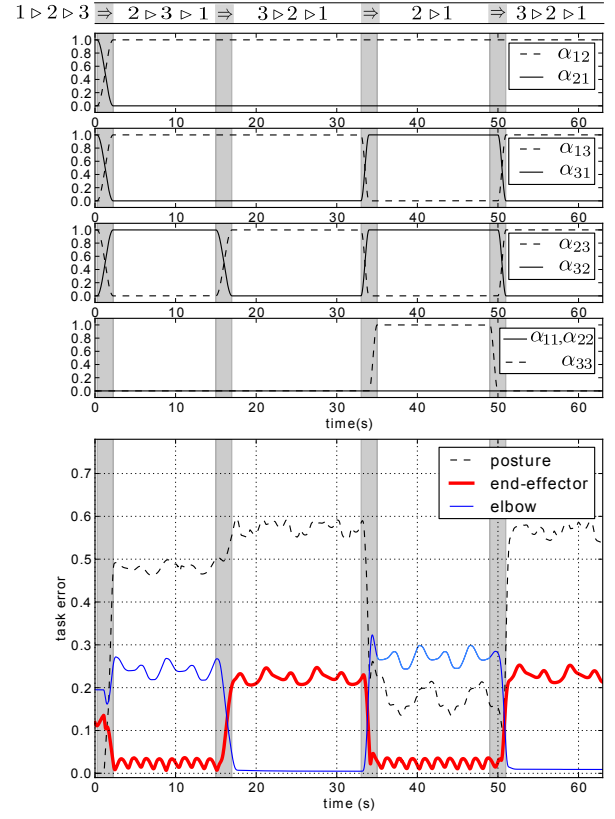


Fig. 3. Evolution of α s (top) and task errors (bottom), with the end-effector task target as a lemniscate-shaped trajectory. Priority transitions as well as the insertion and deletion of the elbow task are performed.

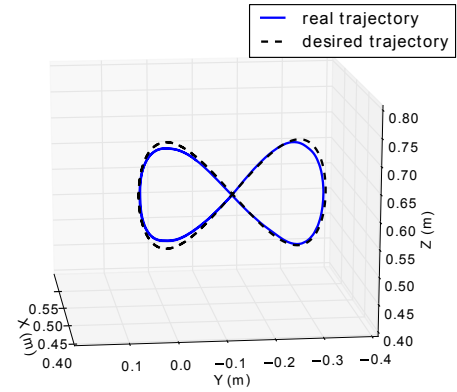


Fig. 4. The desired and the resulting end-effector trajectory, when the end-effector task has the highest priority. The end-effector moves along the lemniscate-shaped trajectory with an orbital period of $2\pi s$. Positions are represented with respect to the base frame of the robot shown in Fig. 1

increasing. When the end-effector task priority is the highest, the small and periodical variations of its errors are due to the lack of the dynamic model compensation in the control (6) in our current experimental conditions. These small errors may be reduced once the dynamic model can be estimated.

The insertion and deletion of a task can be successfully achieved. When the elbow task is removed by the increase of α_{33} to 1, its task error increases, as the task is no longer

controlled; while when this task is inserted in the same hierarchical level as before, task errors change reasonably.

VI. CONCLUSIONS AND FUTURE WORKS

This paper proposes a novel control approach for handling multiple tasks with strict and non-strict priorities. A generalized projector is developed. It can precisely regulate how much a task can influence or be influenced by other tasks through the modulation of a priority matrix: a task can be completely, partially, or not at all projected in the null-space of other tasks. By using this generalized projector, the priority relation between any pair of tasks can be adjusted, multiple simultaneous changes of task priorities can be achieved, and tasks can be easily inserted or deleted. Experiments on a KUKA LWR robot are conducted to demonstrate that this approach allows proper handling of task hierarchies and achieve 1 ms control.

In this work, the approach is illustrated based on analytical control approaches and the Jacobian-transpose method; however, the generalized projector introduced here is not restricted to this case. Immediate future work includes the application of such generalized projectors in the control of more complex robotic systems, such as (free-floating) humanoid robots. To achieve this, an efficient control framework that can take into account generalized projectors, as well as constraints associated with a robot body and its environment should be developed. This can possibly be achieved by the implementation of generalized projectors in control frameworks based on quadratic programming techniques [20,27].

ACKNOWLEDGMENT

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 environments held by Vincent Padois.

REFERENCES

- [1] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 43–53, 1987.
- [2] M. Mistry, J. Nakanishi, and S. Schaal, "Task space control with prioritization for balance and locomotion," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007, pp. 331–338.
- [3] P. Hsu, J. Mauser, and S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, vol. 6, no. 2, pp. 133–148, 1989. [Online]. Available: <http://dx.doi.org/10.1002/rob.4620060203>
- [4] F. Flacco, A. De Luca, and O. Khatib, "Prioritized multi-task motion control of redundant robots under hard joint constraints," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3970–3977.
- [5] L. Sentis and O. Khatib, "Prioritized multi-objective dynamics and control of robots in human environments," in *4th IEEE/RAS International Conference on Humanoid Robots*, vol. 2, 2004, pp. 764–780 Vol. 2.
- [6] O. Kanoun, F. Lamiroux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots," in *IEEE International Conference on Robotics and Automation (2009)*, may 2009, pp. 2939–2944.
- [7] L. Saab, N. Mansard, F. Keith, J.-Y. Fourquet, and P. Soueres, "Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2011, pp. 1091–1096.
- [8] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, p. 0278364914521306, 2014.
- [9] Y. Abe, M. da Silva, and J. Popović, "Multiobjective control with frictional contacts," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 249–258.
- [10] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle, "Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts," in *7th IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 81–88.
- [11] K. Bouyarmane and A. Kheddar, "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*, 2011, pp. 4414–4419.
- [12] M. Liu, A. Micaelli, P. Evrard, A. Escande, and C. Andriot, "Interactive virtual humans: A two-level prioritized control framework with wrench bounds," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1309–1322, 2012.
- [13] L. Saab, O. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *Robotics, IEEE Transactions on*, vol. 29, no. 2, pp. 346–362, 2013.
- [14] K. E. C. and M. J. M., "Soft constraints and exact penalty functions in model predictive control," in *Proceedings of the UKACC International Conference*, Cambridge, UK, September 2000.
- [15] F. Keith, N. Wieber, P.-B. and Mansard, and A. Kheddar, "Analysis of the discontinuities in prioritized tasks-space control under discrete task scheduling operations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 3887–3892.
- [16] T. Petrič and L. Žlajpah, "Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem," *Robotics and Autonomous Systems*, vol. 61, no. 9, pp. 948–959, 2013.
- [17] N. Mansard, A. Remazeilles, and F. Chaumette, "Continuity of varying-feature-set control laws," *Automatic Control, IEEE Transactions on*, vol. 54, no. 11, pp. 2493–2505, 2009.
- [18] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *Robotics, IEEE Transactions on*, vol. 28, no. 6, pp. 1260–1277, 2012.
- [19] A. Dietrich, A. Albu-Schaffer, and G. Hirzinger, "On continuous null space projections for torque-based, hierarchical, multi-objective manipulation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 2978–2985.
- [20] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 1283–1290.
- [21] G. Jarquin, A. Escande, G. Arechavaleta, T. Moulard, E. Yoshida, and V. Parra-Vega, "Real-time smooth task transitions for hierarchical inverse kinematics," in *13th IEEE-RAS International Conference on Humanoid Robots*, Oct 2013, pp. 528–533.
- [22] M. Liu, Y. Tan, and V. Padois, "Generalized Hierarchical Control," Jun 2014. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-01068404>
- [23] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 12, pp. 868–871, dec. 1977.
- [24] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Dynamic multi-priority control in redundant robotic systems," *Robotica*, vol. 31, pp. 1155–1167, 10 2013.
- [25] P. Soetens, "RTT: Real-Time Toolkit," <http://www.orocos.org/rtt>.
- [26] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the kuka lightweight robot," in *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA)*, 2010.
- [27] M. Liu, "Personnage virtuel : contrôleur hybride couplant commande dynamique multi-objectifs et mouvements capturés," Ph.D. thesis, Université Pierre et Marie Curie - Paris VI, Sept. 2012. [Online]. Available: <http://tel.archives-ouvertes.fr/tel-00825375>