Robotica (2013) volume 31, pp. 1155–1167. © Cambridge University Press 2013
doi:10.1017/S0263574713000416

Dynamic multi-priority control in redundant robotic systems[1]

Hamid Sadeghian†,‡*, Luigi Villani‡, Mehdi Keshmiri† and Bruno Siciliano‡

†*Department of Mechanical Engineering, Isfahan University of Technology (IUT), Isfahan 84156-83111, Iran*
‡*PRISMA Lab, Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università di Napoli Federico II, Italy*

(Accepted April 15, 2013. First published online: May 22, 2013)

SUMMARY
This paper presents a dynamic-level control algorithm to meet simultaneously multiple desired tasks based on allocated priorities for redundant robotic systems. It is shown that this algorithm can be treated as a general framework to achieve control over the whole body of the robot. The control law is an extension of the well-known acceleration-based control to the redundant robots, and considers also possible interactions with the environment occurring at any point of the robot body. The stability of this algorithm is shown and some of the previously developed results are formulated using this approach. To handle the interaction on robot body, null space impedance control is developed within the multi-priority framework. The effectiveness of the proposed approaches is evaluated by means of computer simulation.

KEYWORDS: Redundant robots; Multi-priority control; Null space impedance control.

1. Introduction

Robots are termed kinematically redundant when they possess more degrees of freedom (DOF) than those necessary to achieve a desired task. Redundant degrees of freedom can be conveniently used to perform some additional tasks besides the main task. These additional tasks can be a performance objective or, for example, a given Cartesian position of a point on the robot body. There are plenty of papers that deal with how to use redundancy effectively to optimize some performance objective besides the main task control. This optimization is usually performed in the null space of the main task to ensure its perfect tracking. In order to solve the conflict between tasks in the case where several objective functions are going to be satisfied simultaneously, the so-called task priority strategy was developed.[2,3] The formulation was later extended in a general framework for managing multiple tasks by Siciliano and Slotine.[4]

This formulation uses the first-order differential kinematics equation and solves redundancy in the least-squares (LS) sense, based on the assigned priority by resorting to pseudo-inverse solution. Because of using the pseudo-inverse of the projected Jacobians – the Jacobians of the lower priority tasks that are projected into the null space of the higher priority tasks – the formulation may suffer from high norms during transition into and out of algorithmic singularities. Usually singularity-robust pseudo-inverse that allows limiting joint velocities at the expense of small tracking error in lower priority tasks is the first remedy to treat this problem. Efficient damping techniques have been suggested by Nakamura and Hanafusa,[5] Wampler,[6] and also by Nenchev and Sotirov[7] for the case of multiple priorities.

Chiaverini[8] proposed the singularity-robust task-priority resolution without using the projected Jacobian. This formulation always involves tracking errors in the additional tasks, but singularities do not occur as long as the Jacobian of each additional task is full rank. The stability of this formulation has been also shown.[9]

De Santis *et al.*[10] apply the concept of multi-point control and virtual end-effectors (VEEs) for human–robot interaction (HRI). The VEEs are parts of the manipulator structure whose positions are to be controlled in addition to the control of the end-effector of the robot manipulator. They proposed a nested closed-loop inverse kinematics algorithm with a priority management strategy in order to control robot in a cluttered environment.

To address dynamic uncertainties of the system, an adaptive multi-priority control has been proposed by the authors.[11] By this method, asymptotic stability and convergence of tracking error are achieved for the main task as well as the subtasks, according to the allocated priority.

Instead of velocity-based control, acceleration-based control computes the desired joint accelerations for given task references.[12] Synthesis of joint acceleration in a redundant robot usually requires a more involved analysis, but for systems such as robots, this formulation is most natural and offers improved tracking ability due to the explicit incorporation of acceleration information.

The problem of internal instability in the acceleration-level redundancy resolution was first noted by Hsu *et al.*[13] The nature of this instability was further analyzed[14] and it was shown that the divergence of joint velocity norm in finite time in linear acceleration-based redundancy resolution is possible because of rapid increase in the null space component of joint velocities. A complete theoretical and empirical evaluation of different dynamic methods has been investigated by Nakanishi *et al.*[15]

* Corresponding author. E-mail: h.sadeghian@me.iut.ac.ir
[1] A preliminary version of this paper appeared in the proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

The problem of redundancy resolution and multi-priority control has recently received more attention, especially within the context of highly redundant complex system such as humanoid robots. Khatib *et al.*[16] and Sentis and Khatib[17] proposed the extension of the operational space formulation[18] to control behavioral primitives in a humanoid robot at torque level.

A framework based on optimal control for robot with redundant degrees of freedom has been proposed.[19] It is shown how a variety of control laws can be derived using this framework.

Recently, Platt *et al.*[20] proposed a multi-priority Cartesian impedance control method by resorting to acceleration resolution. They also used multi-priority control algorithm to control the impedance of the Robonaut 2[21] arms in both operational space and redundant space.[22]

The motivation behind multi-priority control is the fact that a highly redundant robot is generally required to execute multiple tasks simultaneously. This paper, which is an extension of our previous work[1], contributes to multi-priority control at the acceleration level for redundant robotic systems and establishes a general framework to achieve dynamic control over the whole body of robot. It is shown that by the proper choice of additional tasks, besides preserving the stability of internal motion, it is possible to derive some of the previous results in the literature within this framework. Specifically, Hsu's controller,[13] which ensures the stability of null space motion, is reformulated in a more intuitive and simple expression under multi-priority control framework. Moreover, a comparison with hierarchical torque-level control algorithm[16,17] is performed. To cope with algorithmic singularities, a possible solution is also presented. Null space impedance control, as a result of task prioritization, is introduced to handle the robot–environment interaction.

The rest of the paper is organized as follows. In Section 2 some preliminaries related to the generalized inverse are reviewed. The multi-priority inverse kinematics at velocity level is briefly described in Section 3. The main results, including acceleration-level multi-priority control and null space impedance, are presented in Sections 4 and 5. A comparison between torque level and acceleration level is given in Section 6. Some of the main results are verified by simulation in Section 7. The conclusions are given in the final section.

2. Background on Generalized Inverse

In this section some of the fundamental properties of the pseudo-inverse are surveyed in brief.[3,23]

For a linear equation $\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{q}}$, where $\boldsymbol{J} \in R^{m\times n}$, $\dot{\boldsymbol{x}} \in R^m$ and $\dot{\boldsymbol{q}} \in R^n$, the general form of the least-squares solution that minimizes the error norm $||\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}}||$ is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{x}} + (\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J})\boldsymbol{\zeta}, \tag{1}$$

where $\boldsymbol{\zeta} \in R^n$ is an arbitrary vector and $\boldsymbol{I}$ is an identity matrix. $\boldsymbol{J}^{\dagger} \in R^{n\times m}$ is the unique matrix, called the pseudo-inverse, which satisfies the following properties:

$$\begin{aligned} &\boldsymbol{J}\boldsymbol{J}^{\dagger}\boldsymbol{J} = \boldsymbol{J}, \qquad &&\boldsymbol{J}^{\dagger}\boldsymbol{J}\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\dagger},\\ &(\boldsymbol{J}\boldsymbol{J}^{\dagger})^T = \boldsymbol{J}\boldsymbol{J}^{\dagger}, &&(\boldsymbol{J}^{\dagger}\boldsymbol{J})^T = \boldsymbol{J}^{\dagger}\boldsymbol{J}. \end{aligned} \tag{2}$$

Moreover, the following equalities hold

$$\begin{aligned} &(\boldsymbol{J}^{\dagger})^{\dagger} = \boldsymbol{J},\\ &(\boldsymbol{J}^{\dagger})^T = (\boldsymbol{J}^T)^{\dagger},\\ &\boldsymbol{J}^{\dagger} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{\dagger}. \end{aligned} \tag{3}$$

For the case where $m < n$ and rank $(\boldsymbol{J}) = m$, the pseudo-inverse is given by $\boldsymbol{J}^{\dagger} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}$, and thus $\boldsymbol{J}\boldsymbol{J}^{\dagger} = \boldsymbol{I}$.

The matrix $\boldsymbol{N} = (\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J})$ is an orthogonal projector that projects every vector into the null space of $\boldsymbol{J}$ and thus is idempotent, i.e., $\boldsymbol{N}^2 = \boldsymbol{N}$. This matrix is also symmetric, and it can be easily shown that

$$\begin{aligned} &\boldsymbol{N}(\boldsymbol{B}\boldsymbol{N})^{\dagger} = (\boldsymbol{B}\boldsymbol{N})^{\dagger}, \text{ for any matrix } \boldsymbol{B} \in R^{m\times n},\\ &\boldsymbol{N}^{\dagger} = \boldsymbol{N},\\ &\boldsymbol{J}\boldsymbol{N} = \boldsymbol{N}\boldsymbol{J}^{\dagger} = \boldsymbol{0}. \end{aligned} \tag{4}$$

Equation (1) is the general form of least-squares solution based on Euclidean norm. The first term in (1) is the solution that minimizes $||\dot{\boldsymbol{q}}||$. In many robotic applications, it is desired to find the solution based on an appropriate weighting (usually the inertia matrix $\boldsymbol{M}$) of the components. In these situations the general form of $\boldsymbol{M}-$weighted norm solution is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}\dot{\boldsymbol{x}} + (\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J})\boldsymbol{\eta}, \tag{5}$$

where $\boldsymbol{M} \in R^{n\times n}$ is a symmetric positive definite weighting matrix and $\boldsymbol{J}^{\#}$ is the weighted generalized inverse, which is given by $\boldsymbol{J}^{\#} = \boldsymbol{M}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)^{-1}$ for a full rank matrix $\boldsymbol{J}$. Equation (5) represents a weighted orthogonal decomposition, where the first term minimizes $||\dot{\boldsymbol{q}}||_M$. Null space matrix $\boldsymbol{N}_{\#} = (\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J})$ is a projection matrix and is idempotent, thus $\boldsymbol{J}\boldsymbol{N}_{\#} = \boldsymbol{0}$. The following properties hold for full rank matrix $\boldsymbol{J}$:

$$\begin{aligned} &\boldsymbol{J}\boldsymbol{J}^{\#} = \boldsymbol{I},\\ &\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{N}_{\#}^T = \boldsymbol{0},\\ &\boldsymbol{M}\boldsymbol{N}_{\#} = \boldsymbol{N}_{\#}^T\boldsymbol{M},\\ &\boldsymbol{N}_{\#}(\boldsymbol{B}\boldsymbol{N}_{\#})^{\#} = (\boldsymbol{B}\boldsymbol{N}_{\#})^{\#}, \text{ for any matrix } \boldsymbol{B} \in R^{m\times n}. \end{aligned} \tag{6}$$

3. Multi-Priority Inverse Kinematics

Multi-priority inverse kinematics is a well-established framework to manage the tasks in a kinematically redundant robotic system. Assume that the task is composed of two prioritized tasks. The first priority task (main task) is specified using the first manipulation variable, $\boldsymbol{x}_1 \in R^{m_1}$, and the second priority task (subtask) is specified by the second manipulation variable, $\boldsymbol{x}_2 \in R^{m_2}$. The kinematic relationships between the joint vector $\dot{\boldsymbol{q}} \in R^n$ and the vectors of task variables are expressed by

$$\dot{\boldsymbol{x}}_k = \boldsymbol{J}_k(\boldsymbol{q})\dot{\boldsymbol{q}} \ (k = 1, 2), \tag{7}$$

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

where $\boldsymbol{J}_k(\boldsymbol{q}) \in R^{m_k \times n}$ is the Jacobian matrix of the kth task.

The inverse kinematics solution considering the priority of the main task over the subtask can be found in literature[3] and for the sake of completeness it is briefly recalled now. First, the set of all solutions that satisfy main task is obtained as $\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\dot{\boldsymbol{x}}_1 + (\boldsymbol{I} - \boldsymbol{J}_1^{\dagger}\boldsymbol{J})\boldsymbol{\zeta}$, where $\boldsymbol{\zeta}$ is an arbitrary vector. Then this solution is substituted in $\dot{\boldsymbol{x}}_2 = \boldsymbol{J}_2\dot{\boldsymbol{q}}$, and $\boldsymbol{\zeta}$ that minimizes $||\dot{\boldsymbol{x}}_2 - \boldsymbol{J}_2\dot{\boldsymbol{q}}||$, is computed. After substitution of this value of $\boldsymbol{\zeta}$, the final solution is obtained as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{x}}_1 + \bar{\boldsymbol{J}}_2^{\dagger}(\dot{\boldsymbol{x}}_2 - \boldsymbol{J}_2\boldsymbol{J}_1^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}}_1) + \boldsymbol{N}_2\boldsymbol{\eta}, \tag{8}$$

where $\bar{\boldsymbol{J}}_2 = \boldsymbol{J}_2\boldsymbol{N}_1$ is the projected Jacobian, which gives the available range for the subtask to be executed without affecting the main task, $\boldsymbol{N}_1 = (\boldsymbol{I} - \boldsymbol{J}_1^{\dagger}\boldsymbol{J}_1)$, $\boldsymbol{N}_2 = \boldsymbol{N}_1(\boldsymbol{I} - \bar{\boldsymbol{J}}_2^{\dagger}\bar{\boldsymbol{J}}_2)$, and $\boldsymbol{\eta}$ is an arbitrary vector. A recursive extension of (8) has also been proposed.[4]

At this point, let us remark that in multi-priority algorithm if $\boldsymbol{J}_k$ is singular (task singularity), then the kth task cannot be satisfied regardless of all other tasks. If $\bar{\boldsymbol{J}}_k$ is singular, without $\boldsymbol{J}_k$ being singular (algorithmic singularity), the kth task cannot be satisfied given the previous $k-1$ tasks. This will happen when tasks are dependent. Namely, two generic tasks are dependent when

$$\rho(\boldsymbol{J}_1) + \rho(\boldsymbol{J}_2) > \rho\left(\begin{bmatrix}\boldsymbol{J}_1\\ \boldsymbol{J}_2\end{bmatrix}\right), \tag{9}$$

where $\rho(.)$ denotes the rank of a matrix. More details about the task dependency and the algorithmic singularity can be found in related references.[8,9,24]

Thus, it is obvious that singularities may occur from the lower priority tasks. In the case of a free task priority assignment, dynamic task priority allocation[7] is crucial to the overall performance of the system. For a singularity-robust task-priority handling, Chiaverini[8] proposed the following formulation for a case with two tasks:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\dot{\boldsymbol{x}}_1 + \boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\dot{\boldsymbol{x}}_2. \tag{10}$$

Comparing with (8), algorithmic singularities are absent, but there is typically a greater tracking error for the subtask.[9,25,26]

4. Multi-Priority Control at Acceleration Level

4.1. General formulation

The goal of dynamic multi-priority control is to derive a control torque which will cause the system to track the desired main task exactly, while, at the same time, system redundancy is exploited to realize a number of subtasks according to some desired priorities.

The dynamic model of a robot manipulator can be written in compact form as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_{\text{ext}} = \boldsymbol{\tau} \tag{11}$$

with known notation. In this formulation, $\boldsymbol{\tau}_{\text{ext}}$ is the external torque resulting from any interaction on the manipulator. The kinematic relationships between the joint variable $\boldsymbol{q} \in R^n$ and the task variable $\boldsymbol{x}_k$ at acceleration level are expressed by

$$\ddot{\boldsymbol{x}}_k = \boldsymbol{J}_k\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}}. \tag{12}$$

Following a derivation similar to that used for extracting (8), the corresponding solution for the joint space command acceleration $\ddot{\boldsymbol{q}}_c$, for given task space command accelerations $\ddot{\boldsymbol{x}}_{1c}$, $\ddot{\boldsymbol{x}}_{2c}$ is given by

$$\begin{aligned}\ddot{\boldsymbol{q}}_c = {} & \boldsymbol{J}_1^{\dagger}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}) \\ & + \bar{\boldsymbol{J}}_2^{\dagger}[\ddot{\boldsymbol{x}}_{2c} - \dot{\boldsymbol{J}}_2\dot{\boldsymbol{q}} - \boldsymbol{J}_2\boldsymbol{J}_1^{\dagger}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}})] + \boldsymbol{N}_2\boldsymbol{\eta}.\end{aligned} \tag{13}$$

The basic issue in this formulation is the differential order at which resolution takes place. Actually the first term on the right-hand side ensures minimization of $||\ddot{\boldsymbol{x}}_1 - \ddot{\boldsymbol{x}}_{1c}||$, and by the second term the minimization of $||\ddot{\boldsymbol{x}}_2 - \ddot{\boldsymbol{x}}_{2c}||$ in the null space of the main task is obtained. Further investigation enables us to propose the general solution for L tasks as follows:

$$\begin{aligned}\ddot{\boldsymbol{q}}_c &= \sum_{k=1}^{L}\ddot{\boldsymbol{q}}_{kc}, \\ \ddot{\boldsymbol{q}}_{kc} &= \bar{\boldsymbol{J}}_k^{\dagger}(\ddot{\boldsymbol{x}}_{kc} - \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} - \boldsymbol{a}_k), \\ &\begin{cases}\boldsymbol{a}_1 = \boldsymbol{0}, & k = 1\\ \boldsymbol{a}_k = \boldsymbol{J}_k\sum_{i=1}^{k-1}\ddot{\boldsymbol{q}}_{ic}, & k = 2, \ldots, L\end{cases},\end{aligned} \tag{14}$$

where

$$\begin{aligned}&\bar{\boldsymbol{J}}_k = \boldsymbol{J}_k\boldsymbol{N}_{k-1}, \\ &\boldsymbol{N}_k = \textstyle\prod_{j=1}^{k}(\boldsymbol{I} - \bar{\boldsymbol{J}}_j^{\dagger}\bar{\boldsymbol{J}}_j), \\ &\boldsymbol{N}_0 = \boldsymbol{I},\end{aligned} \tag{15}$$

and the equality $\boldsymbol{N}_{k-1}\bar{\boldsymbol{J}}_k^{\dagger} = \bar{\boldsymbol{J}}_k^{\dagger}$ has been used.

Note that the null space matrix $\boldsymbol{N}_k$ can be written as

$$\boldsymbol{N}_k = \left(\boldsymbol{I} - \sum_{j=1}^{k}\bar{\boldsymbol{J}}_j^{\dagger}\bar{\boldsymbol{J}}_j\right), \tag{16}$$

or, equivalently, in the form

$$\begin{aligned}\boldsymbol{N}_k &= (\boldsymbol{I} - \boldsymbol{J}_{Ak}^{\dagger}\boldsymbol{J}_{Ak}), \\ \boldsymbol{J}_{Ak} &= \begin{bmatrix}\boldsymbol{J}_1\\ \cdots\\ \boldsymbol{J}_k\end{bmatrix},\end{aligned} \tag{17}$$

where $\boldsymbol{J}_{Ak}$ is the augmented Jacobian.[4]

Once the command acceleration $\ddot{\boldsymbol{q}}_c$ is obtained, well-known concept of inverse dynamics can be used to find the

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

driving torques

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_c + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_{\text{msr}}. \tag{18}$$

In this equation, $\boldsymbol{\tau}_{\text{msr}}$ is the vector of measured or estimated external torques acting on the robot. Under the assumption of perfect torque measurement, we have $\boldsymbol{\tau}_{\text{msr}} \approx \boldsymbol{\tau}_{\text{ext}}$.

Equations (14) and (18) can be treated as a general framework to control the whole behavior of the redundant robotic system by multiple priorities and even multi-point control. The acceleration-level prioritization, in contrast to velocity-level prioritization, gives full trajectory planning with complete time information. A nice property of (14) is that, if desired, it can be easily used for any kind of force motion control by a proper choice of the operational command acceleration. The method also enables us to give priority to one task over another, even in a non-redundant robotic system. This kind of formulation can explain the previously proposed acceleration-level resolution techniques in the framework of task prioritization by a proper choice of subtasks.

Regarding (14), the following crucial points must be taken into account:

(1) The first $k-1$ tasks influence the performance of task k. Hence, the order of priority allocation will be crucial to the performance.[7]
(2) Internal instability of joint velocity in the residual null space of the overall augmented Jacobian may arise.[13,14]
(3) The general scheme suffers from high-norm solutions in the neighborhood of singularities because of using pseudo-inverse.

The last two issues will be addressed in the next two sections.

4.2. Stability analysis

By the compensation of Coriolis/centrifugal, gravity and external torque in (18), the closed-loop behavior of the system is obtained as

$$\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_c, \tag{19}$$

where $\ddot{\boldsymbol{q}}_c$ is given by (14) for general L tasks. Multiplying both sides by $\boldsymbol{J}_1\boldsymbol{J}_1^{\dagger}\boldsymbol{J}_1$ and using the idempotency of $\boldsymbol{J}_1\boldsymbol{J}_1^{\dagger}$ give

$$\boldsymbol{J}_1\boldsymbol{J}_1^{\dagger}[\ddot{\boldsymbol{x}}_{1c} - \ddot{\boldsymbol{x}}_1] = \boldsymbol{0}, \tag{20}$$

which under the full rank assumption for $\boldsymbol{J}_1$ gives $\ddot{\boldsymbol{x}}_1 = \ddot{\boldsymbol{x}}_{1c}$. This implies that, for instance, for command acceleration

$$\ddot{\boldsymbol{x}}_{1c} = \ddot{\boldsymbol{x}}_{1d} + \boldsymbol{K}_{D1}(\dot{\boldsymbol{x}}_{1d} - \dot{\boldsymbol{x}}_1) + \boldsymbol{K}_{P1}(\boldsymbol{x}_{1d} - \boldsymbol{x}_1), \tag{21}$$

with desired trajectory $\boldsymbol{x}_{1d}$ and suitable positive Proportional-Derivative (PD) gains $\boldsymbol{K}_{P1}$ and $\boldsymbol{K}_{D1}$, asymptotic tracking for main task is achieved.

Multiplying both sides of (19) by $\bar{\boldsymbol{J}}_i\bar{\boldsymbol{J}}_i^{\dagger}\boldsymbol{J}_i$, in order to find the closed-loop equation for ith subtask, and considering the idempotency of $\bar{\boldsymbol{J}}_i\bar{\boldsymbol{J}}_i^{\dagger}$ and properties (4), yield

$$\bar{\boldsymbol{J}}_i\bar{\boldsymbol{J}}_i^{\dagger}[\ddot{\boldsymbol{x}}_{ic} - \ddot{\boldsymbol{x}}_i] = \boldsymbol{0}, \tag{22}$$

which ensures the minimization of $||\ddot{\boldsymbol{x}}_i - \ddot{\boldsymbol{x}}_{ic}||$ subject to all higher priority tasks. Thus, if ith task is independent from all higher priority tasks and thus $\bar{\boldsymbol{J}}_i$ is full rank, this subtask is correctly executed. In other case, when $\bar{\boldsymbol{J}}_i$ is not full rank, the ith task is not performed completely and just the above norm minimization is performed with respect to all higher priority tasks.

Note that if the rank of the overall augmented Jacobian is lower than the number of joints coordinate n, then the residual null space velocity must be controlled to avoid unstable behavior.[13,14]

Without loss of generality, let us assume that only one task (the main task) is assigned, and the null space of $\boldsymbol{J}_1$ is non-empty. To ensure the stability of the null space velocity, let us choose the null space velocity as a subtask, namely, $\dot{\boldsymbol{x}}_2 = \boldsymbol{N}_1\dot{\boldsymbol{q}}$, with the desired velocity, $\dot{\boldsymbol{x}}_{2d} = \boldsymbol{N}_1\boldsymbol{\xi}$, and command acceleration, $\ddot{\boldsymbol{x}}_{2c} = \ddot{\boldsymbol{x}}_{2d} + \boldsymbol{K}(\dot{\boldsymbol{x}}_{2d} - \dot{\boldsymbol{x}}_2)$, with positive definite matrix $\boldsymbol{K}$. From (13) the corresponding command acceleration in joint space is obtained as

$$\ddot{\boldsymbol{q}}_c = \boldsymbol{J}_1^{\dagger}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}) + \boldsymbol{N}_1[\ddot{\boldsymbol{x}}_{2c} - \dot{\boldsymbol{N}}_1\dot{\boldsymbol{q}}], \tag{23}$$

where $\boldsymbol{N}_1^{\dagger} = \boldsymbol{N}_1$ and $\boldsymbol{N}_1\boldsymbol{J}_1^{\dagger} = \boldsymbol{0}$ have been used.

By the above command acceleration, the closed-loop equation for the subtask from (22) is given by

$$\boldsymbol{N}_1[\dot{\boldsymbol{e}}_N + \boldsymbol{K}\boldsymbol{e}_N] = \boldsymbol{0}, \tag{24}$$

where $\boldsymbol{e}_N = \boldsymbol{N}_1(\boldsymbol{\xi} - \dot{\boldsymbol{q}})$ is the null space velocity error. The above equation does not guarantee that $\boldsymbol{e}_N$ goes to zero, since $\boldsymbol{N}_1$ is not full rank. Computing the time derivative of the identity $\boldsymbol{e}_N = \boldsymbol{N}_1\boldsymbol{e}_N$ and using (24) give the error dynamics

$$\dot{\boldsymbol{e}}_N = \boldsymbol{N}_1\dot{\boldsymbol{e}}_N + \dot{\boldsymbol{N}}_1\boldsymbol{e}_N = -(\boldsymbol{N}_1\boldsymbol{K} - \dot{\boldsymbol{N}}_1)\boldsymbol{e}_N. \tag{25}$$

To prove the stability of the null space velocity error dynamics, consider the positive definite Lyapunov function candidate

$$V = \frac{1}{2}\boldsymbol{e}_N^T\boldsymbol{e}_N \tag{26}$$

whose time derivative along the trajectories of the system (25) is

$$\dot{V} = -\boldsymbol{e}_N^T(\boldsymbol{N}_1\boldsymbol{K} - \dot{\boldsymbol{N}}_1)\boldsymbol{e}_N = -\boldsymbol{e}_N^T\boldsymbol{K}\boldsymbol{e}_N + \boldsymbol{e}_N^T\dot{\boldsymbol{N}}_1\boldsymbol{e}_N. \tag{27}$$

The last term of (27) is null, since matrix $\boldsymbol{N}_1\dot{\boldsymbol{N}}_1\boldsymbol{N}_1$ is null, as can be easily shown using (4). Therefore, $\dot{V} = -\boldsymbol{e}_N^T\boldsymbol{K}\boldsymbol{e}_N$ is negative definite for all positive definite matrix $\boldsymbol{K}$, and the equilibrium $\boldsymbol{e}_N = \boldsymbol{0}$ is asymptotically stable.

In conclusion, the internal stability is guaranteed in the same multi-priority framework by a simple choice for the lowest priority task. In the above equations, one can choose $\boldsymbol{\xi} = \boldsymbol{0}$, and thus as long as manipulator avoids singularity, null space velocity will go to zero. However, if it is appealed to minimize some performance index $m(\boldsymbol{q})$, $\boldsymbol{\xi}$ can be chosen as the gradient of performance index, i.e., $\boldsymbol{\xi} = \alpha\nabla m$.

It is worth observing that the null space command acceleration $\boldsymbol{\phi}_N = \boldsymbol{N}_1[\ddot{\boldsymbol{x}}_{2c} - \dot{\boldsymbol{N}}_1\dot{\boldsymbol{q}}]$ in (23), which guarantees

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

internal stability, coincides with the null space stabilizing control introduced by Hsu *et al*.[13] In their paper, the following control acceleration was proposed:

$$\ddot{\boldsymbol{q}}_c = \boldsymbol{J}_1^{\dagger}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}) + \boldsymbol{N}_1[\dot{\boldsymbol{\xi}} + \boldsymbol{K}\boldsymbol{N}_1(\boldsymbol{\xi} - \dot{\boldsymbol{q}})] - (\boldsymbol{J}_1^{\dagger}\dot{\boldsymbol{J}}_1\boldsymbol{J}_1^{\dagger} + \dot{\boldsymbol{J}}_1^{\dagger})\boldsymbol{J}_1(\boldsymbol{\xi} - \dot{\boldsymbol{q}}). \tag{28}$$

To show this coincidence, from (23) the null space vector can be written as

$$\begin{aligned}\boldsymbol{\phi}_N &= \boldsymbol{N}_1[\ddot{\boldsymbol{x}}_{2c} - \dot{\boldsymbol{N}}_1\dot{\boldsymbol{q}}] \\ &= \boldsymbol{N}_1[\dot{\boldsymbol{N}}_1(\boldsymbol{\xi}) + \boldsymbol{N}_1\dot{\boldsymbol{\xi}} + \boldsymbol{K}\boldsymbol{N}_1(\boldsymbol{\xi} - \dot{\boldsymbol{q}}) - \dot{\boldsymbol{N}}_1\dot{\boldsymbol{q}}]. \\ &= \boldsymbol{N}_1\dot{\boldsymbol{N}}_1(\boldsymbol{\xi} - \dot{\boldsymbol{q}}) + \boldsymbol{N}_1[\dot{\boldsymbol{\xi}} + \boldsymbol{K}\boldsymbol{N}_1(\boldsymbol{\xi} - \dot{\boldsymbol{q}})]\end{aligned} \tag{29}$$

Substituting $\boldsymbol{N}_1\dot{\boldsymbol{N}}_1 = -\boldsymbol{N}_1\dot{\boldsymbol{J}}_1^{\dagger}\boldsymbol{J}_1$ and using equality $\dot{\boldsymbol{J}}_1\boldsymbol{J}_1^{\dagger} = -\boldsymbol{J}_1\dot{\boldsymbol{J}}_1^{\dagger}$ give null space vector in (28).

Recently, Nakanishi *et al*.[15] tried to find an intuition behind this, rather complex, null space vector. They analyzed these terms to find a relation with analytical differentiation of velocity-based redundancy resolution approach. Here by (23) and (24) we give a reasonable intuition behind (28) with a more appealing and simple form.

4.3. Singularity treatment in multi-priority control

Since the acceleration-level formulation is based on least-squares, the first remedy to treat singularity is the Damped Least-Squares (DLS) method. A complete analysis of using DLS in the second-order kinematic control for non-redundant robots has been performed.[27] It is observed that using the DLS method at the acceleration level in multi-priority algorithm causes oscillatory motion near singularities. This oscillation is illustrated in the simulation section of the case study. The oscillation is different from the instability of internal motion, which occurs for residual null space. By using DLS in acceleration level, the sum of the norm of joint space acceleration (multiplied by the damping coefficient) and the norm of the task error are minimized. This minimization does not guarantee proper behavior of joint velocities.

Without loss of generality, assume that we have a main task $\boldsymbol{x}_1$, and only one subtask $\boldsymbol{x}_2$, producing an algorithmic singularity. To cope with this problem, the damped pseudo-inverse is adopted for the subtask, and an additional subtask $\boldsymbol{x}_3 = \boldsymbol{q}$ with command $\ddot{\boldsymbol{x}}_{3c} = -\boldsymbol{B}\dot{\boldsymbol{q}} + \boldsymbol{K}(\boldsymbol{q}_{ds} - \boldsymbol{q})$ is introduced, with the same priority level of subtask $\boldsymbol{x}_2$, aimed at damping the oscillation in the neighborhood of singularity. The joint vector $\boldsymbol{q}_{ds}(t)$ is a suitable configuration that can be set to push the system far from the singularity. Hence, the command acceleration is formulated as

$$\begin{aligned}\ddot{\boldsymbol{q}}_c = {} & \boldsymbol{J}_1^{\dagger}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}) \\ & + \boldsymbol{N}_1\bar{\boldsymbol{J}}_2^{\dagger\lambda_1}[\ddot{\boldsymbol{x}}_{2c} - \dot{\boldsymbol{J}}_2\dot{\boldsymbol{q}} - \boldsymbol{J}_2\boldsymbol{J}_1^{\dagger}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}})] \\ & + k\lambda_1\boldsymbol{N}_1[-\boldsymbol{B}\dot{\boldsymbol{q}} + \boldsymbol{K}(\boldsymbol{q}_{ds} - \boldsymbol{q})],\end{aligned} \tag{30}$$

where $\boldsymbol{B}$ and $\boldsymbol{K}$ are suitable positive matrices and k is a positive constant. In this formulation λ_1 is the damping factor, which can be given, for example, as[27]

$$\lambda_1^2 = \begin{cases} 0 & \sigma \geq \varepsilon \\ \left(1 - (\frac{\sigma}{\varepsilon})^2\right)\lambda_M^2 & \sigma < \varepsilon \end{cases}, \tag{31}$$

where σ is the singular value of the related projected Jacobian, ε defines the size of singular region and λ_M properly shapes the solution near the singularity. In this way, inside the region of singularity, the factor λ_1 increases in proportion to the closeness to the singularity to ensure continuity and good shaping of the solution.

Note that setting $\boldsymbol{K} = \boldsymbol{0}$ in (30) actually damps out the oscillatory motion without any pre-assigned configuration. In the simulation section these cases will be illustrated with more details.

It is worth to mention that by the last term in (30) and near to the singularity, in fact, the so-called null space impedance is brought into the second priority by factor $k\lambda_1$. In this way the singularity problem is solved within the same multi-priority framework. Null space impedance will be introduced in the next section.

5. Null Space Impedance Control

Assume that the end-effector of the redundant robot is going to follow the desired task space trajectory $\boldsymbol{x}_d(t)$ or maybe to realize an impedance behavior as the main task. One interesting choice for subtask $\boldsymbol{x}_2$ is the whole joint space configuration, i.e., $\boldsymbol{x}_2 = \boldsymbol{q}$. By this choice, we aim to have control over the behavior of the robot in the joint space at the second priority. Note that by the above choice, two tasks are always in conflict, but we do not have problems with singularities. Usually singularity may give troubles during transition from a nonsingular to a singular configuration because of the use of a pseudo-inverse in the solution.

Let us give the command impedance acceleration as $\ddot{\boldsymbol{x}}_{2c} = \ddot{\boldsymbol{q}}_d + \boldsymbol{M}_d^{-1}(\boldsymbol{B}_d\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_d\tilde{\boldsymbol{q}} - \boldsymbol{\tau}_{\mathrm{msr}})$, where $\boldsymbol{M}_d$, $\boldsymbol{B}_d$ and $\boldsymbol{K}_d$ are the impedance matrices, $\tilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q}$ and $\boldsymbol{q}_d(t)$ is the desired joint space configuration. The command joint acceleration and the closed-loop behaviors are obtained from (13) and (22) as follows:

$$\ddot{\boldsymbol{q}}_c = \boldsymbol{J}_1^{\dagger}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}) + \boldsymbol{N}_1[\ddot{\boldsymbol{q}}_d + \boldsymbol{M}_d^{-1}(\boldsymbol{B}_d\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_d\tilde{\boldsymbol{q}} - \boldsymbol{\tau}_{\mathrm{msr}})], \tag{32}$$

$$\ddot{\boldsymbol{x}}_1 = \ddot{\boldsymbol{x}}_{1c}, \tag{33}$$

$$\boldsymbol{N}_1[\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{M}_d^{-1}(\boldsymbol{B}_d\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_d\tilde{\boldsymbol{q}}) - \boldsymbol{M}_d^{-1}\boldsymbol{\tau}_{\mathrm{msr}}] = \boldsymbol{0}. \tag{34}$$

Actually this command lets us to realize joint space impedance despite the main task objective in the null space of the main task. By a proper choice of null space impedance matrices, it is possible to achieve a compliant behavior for the robot body. This compliant behavior is useful in the case where the robot works in a cluttered environment and interaction may occur. The desired trajectory $\boldsymbol{q}_d(t)$ in the above equation can be chosen as a rest point or as a trajectory to meet a given objective function. Note that, generally it is not necessary for $\boldsymbol{q}_d(t)$ to be consistent with the task space

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

trajectory $\boldsymbol{x}_d(t)$, that is, $\boldsymbol{x}_d = \boldsymbol{f}(\boldsymbol{q}_d)$. Actually, in the absence of interaction if $\boldsymbol{q}_d(t)$ is not consistent, i.e., $\boldsymbol{x}_d \neq \boldsymbol{f}(\boldsymbol{q}_d)$, a local minimization can be achieved in the null space which ensures the configuration to reach $\boldsymbol{q}_d(t)$ in the sense of least square. However, when the desired null space trajectory is consistent with the task space, both the desired task space objective and the null space configuration are expected to achieve simultaneously.

The closed-loop equations (33) and (34) were obtained under the assumption that the external forces applied to the manipulator have been entirely compensated. When external torque information is not available to compensate $\boldsymbol{\tau}_{\text{ext}}$, the controller is given by

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_c + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}), \tag{35}$$

$$\ddot{\boldsymbol{q}}_c = \boldsymbol{J}_1^{\dagger}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}) + \boldsymbol{N}_1[\ddot{\boldsymbol{q}}_d + \boldsymbol{M}_d^{-1}(\boldsymbol{B}_d\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_d\tilde{\boldsymbol{q}})], \tag{36}$$

and the closed-loop behavior is obtained as

$$\ddot{\boldsymbol{x}}_{1c} - \ddot{\boldsymbol{x}}_1 = \boldsymbol{J}_1\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\text{ext}}, \tag{37}$$

$$\boldsymbol{N}_1[\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{M}_d^{-1}(\boldsymbol{B}_d\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_d\tilde{\boldsymbol{q}}) - \boldsymbol{M}^{-1}\boldsymbol{\tau}_{\text{ext}}] = \boldsymbol{0}. \tag{38}$$

Note that by using dynamically consistent[18] generalized inverse, $\boldsymbol{J}_1^{\#} = \boldsymbol{M}^{-1}\boldsymbol{J}_1^T(\boldsymbol{J}_1\boldsymbol{M}^{-1}\boldsymbol{J}_1^T)^{-1}$in (36) and choosing $\boldsymbol{M}_d = \boldsymbol{M}$ from (38), the following equation is obtained:

$$\boldsymbol{N}_{\#1}^T[\boldsymbol{M}\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{B}_d\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_d\tilde{\boldsymbol{q}} - \boldsymbol{\tau}_{\text{ext}}] = \boldsymbol{0}, \tag{39}$$

where equality $\boldsymbol{M}\boldsymbol{N}_{\#1} = \boldsymbol{N}_{\#1}^T\boldsymbol{M}$ with $\boldsymbol{N}_{\#1} = (\boldsymbol{I} - \boldsymbol{J}_1^{\#}\boldsymbol{J}_1)$ has been used. It must be mentioned that for both the cases, with and without the external torque information, by using the above choices for the generalized inverse and $\boldsymbol{M}_d$, the closed-loop equations in the null space are almost the same. The only difference is that in (34) the impedance is realized with respect to the measured/estimated external torque whereas in (39) it is realized with respect to the physical external torque.

Note that in view of (37) the external interaction affects the operational task. To alter this error on the main task, a disturbance observer has been proposed by the author of this paper.[28] The null space impedance control has been also applied for a dual arm cooperating manipulator during Cartesian impedance control.[29]

For the case where the external interaction just occurs at the end-effector and dynamically consistent generalized inverse is used, we have

$$\boldsymbol{N}_{\#1}^T\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{N}_{\#1}^T\boldsymbol{J}_1^T\boldsymbol{F}_{\text{ext}} = \boldsymbol{0}. \tag{40}$$

It means that the force on the end-effector does not affect the null space behavior (39) as it is expected by using dynamically consistent generalized inverse.

From the closed-loop behavior (39) and, more generally, from (22), it is clear that the desired impedance can never be realized completely within the null space. However, it is realized as much as possible in the sense of least-squares in the null space of the main task (the norm $||\ddot{\boldsymbol{x}}_2 - \ddot{\boldsymbol{x}}_{2c}||$ is minimized subject to the main task). Actually, the name of null space impedance control implies this fact and it can be regarded as an extension of the null space stiffness or damping that is, as known from the literature, aimed at stabilizing null space motion.[15,30] By this formulation, the redundancy of the robot can be exploited for safe interaction between the robot body and the unknown environment.

Even though the above choice of whole configuration as the second task with the joint impedance command is quite intuitive, the stability analysis of (39) even in the absence of external interaction seems rather difficult.[15] This problem can be overcome by using the minimal representation of null space.[31,32] In detail, a full rank null space base matrix $\boldsymbol{Z}(\boldsymbol{q}) \in R^{n\times(n-m)}$, such that $\boldsymbol{J}_1\boldsymbol{Z} = \boldsymbol{0}$, is chosen and a velocity vector $\boldsymbol{v}$, such that $\boldsymbol{N}_1\dot{\boldsymbol{q}} = \boldsymbol{Z}\boldsymbol{v}$, is defined. It can be shown that a convenient choice of the minimal velocity vector $\boldsymbol{v}$ is given by the left inertia-weighted generalized inverse, $\boldsymbol{v} = \boldsymbol{Z}^{\#}\dot{\boldsymbol{q}} = (\boldsymbol{Z}^T\boldsymbol{M}\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{M}\dot{\boldsymbol{q}}$.

The minimal representation of null space impedance can be formulated within the multi-priority framework by the choice of minimal velocities as the second priority task, i.e., $\dot{\boldsymbol{x}}_2 = \boldsymbol{v}$. In this way, $\boldsymbol{Z}^{\#}$ can be seen as the Jacobian of the second priority task. Using (13) with the dynamically consistent generalized inverse, the command acceleration in this case is

$$\ddot{\boldsymbol{q}}_c = \boldsymbol{J}_1^{\#}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}) + \boldsymbol{Z}(\ddot{\boldsymbol{x}}_{2c} - \dot{\boldsymbol{Z}}^{\#}\dot{\boldsymbol{q}}), \tag{41}$$

where the equality $\boldsymbol{Z}^{\#}\boldsymbol{N}_{\#1} = \boldsymbol{Z}^{\#}$ has been used. Following the same procedure as in the non-minimal case and by the choice of $\ddot{\boldsymbol{x}}_{2c} = \dot{\boldsymbol{v}}_d + \boldsymbol{\Lambda}_d^{-1}(\boldsymbol{B}_d\tilde{\boldsymbol{v}} + \boldsymbol{Z}^T\boldsymbol{K}_d\tilde{\boldsymbol{q}})$ as the command acceleration, with $\boldsymbol{\Lambda}_d = \boldsymbol{Z}^T\boldsymbol{M}\boldsymbol{Z}$, the closed-loop dynamics in the null space is obtained as

$$(\boldsymbol{Z}^T\boldsymbol{M}\boldsymbol{Z})\dot{\tilde{\boldsymbol{v}}} + \boldsymbol{B}_d\tilde{\boldsymbol{v}} + \boldsymbol{Z}^T\boldsymbol{K}_d\tilde{\boldsymbol{q}} = \boldsymbol{Z}^T\boldsymbol{\tau}_{\text{ext}}, \tag{42}$$

being $\tilde{\boldsymbol{v}} = \boldsymbol{v}_d - \boldsymbol{v}$ and $\boldsymbol{B}_d$ and $\boldsymbol{K}_d$ as proper impedance matrices.

By this formulation a minimal representation of the null space impedance with the projection of external torques on the minimal space $\boldsymbol{Z}^T\boldsymbol{\tau}_{\text{ext}}$ is realized. The stability of the system can be shown following the solution proposed by Ott *et al.*[32]

For the sake of completeness, it is worth to remind that the nature of a specified task is important for the command $\ddot{\boldsymbol{x}}_c$ if it is related to the system position or orientation. In general, position and orientation controls need to be considered separately, since the position control is rather straightforward, while the orientation control is more complex. A critical comparison between various type of orientation error (Euler angles feedback, quaternion feedback and angle/axis feedback) has been performed and evaluated experimentally.[33] Therefore, in order to use the presented methods in a task space control, which contains orientation tasks, task space command must be given based on proper orientation error definition.

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

6. Comparison with Torque-Level Multi-Priority Control

Khatib *et al.*[14,17] developed a whole-body control framework for prioritized multiple task control in humanoid robots. Hand location, mass center control and obstacle and joint limit avoidance are the common choices for tasks in a humanoid robot. They labeled all the behaviors not affecting the main task as the posture space. This formulation allows for posture objectives to be controlled without dynamically interfering with the main (operational) task.

Let us use the subscripts 1 and 2 for the task space and the posture space respectively. In the absence of external torque and in order to obtain a decoupled behavior for the operational task $\boldsymbol{x}_1$ and the posture $\boldsymbol{x}_2$, the originally proposed control law[16] can be written as

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{\tau}_{\text{task}} + \boldsymbol{\tau}_{\text{posture}},\\ \boldsymbol{\tau}_{\text{task}} &= \boldsymbol{J}_1^T[\boldsymbol{\Lambda}_1(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}) + \boldsymbol{J}_1^{\#T}(\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g})],\\ \boldsymbol{\tau}_{\text{posture}} &= \bar{\boldsymbol{J}}_2^T[\bar{\boldsymbol{\Lambda}}_2(\ddot{\boldsymbol{x}}_{2c} - \dot{\bar{\boldsymbol{J}}}_2\dot{\boldsymbol{q}} - \ddot{\boldsymbol{x}}_{2-\text{bias}}) + \bar{\boldsymbol{J}}_2^{\#T}(\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g})],\end{aligned} \tag{43}$$

where $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ are the Jacobians associated to the operational and posture tasks, $\bar{\boldsymbol{J}}_2 = \boldsymbol{J}_2\boldsymbol{N}_{\#1}$ with $\boldsymbol{N}_{\#1} = (\boldsymbol{I} - \boldsymbol{J}_1^{\#}\boldsymbol{J}_1)$ being the task consistent null space, $\boldsymbol{J}_1^{\#}$ is the dynamically consistent generalized inverse of Jacobian, $\boldsymbol{\Lambda}_1 = (\boldsymbol{J}_1\boldsymbol{M}^{-1}\boldsymbol{J}_1^T)^{-1}$ is the task inertia matrix and $\bar{\boldsymbol{\Lambda}}_2 = (\bar{\boldsymbol{J}}_2\boldsymbol{M}^{-1}\bar{\boldsymbol{J}}_2^T)^{-1}$ is the posture-related inertia matrix. Furthermore, $\ddot{\boldsymbol{x}}_{1c}$ and $\ddot{\boldsymbol{x}}_{2c}$are command accelerations for the task and posture spaces, respectively, and $\ddot{\boldsymbol{x}}_{2-\text{bias}}$ is the acceleration induced by $\boldsymbol{\tau}_{\text{task}}$ in the posture space. The extension of the above formulation for a multi-level control has been also proposed.[17]

For the sake of comparison between acceleration- and torque-level formulations for general L tasks with allocated priorities, the control torque proposed by Sentis and Khatib[17] is rewritten in a suitable form with consistent notation as follows:

$$\begin{aligned}\boldsymbol{\tau} &= \sum_{k=1}^{L}\bar{\boldsymbol{\tau}}_k,\\ \bar{\boldsymbol{\tau}}_k &= \bar{\boldsymbol{J}}_k^T[\bar{\boldsymbol{\Lambda}}_k(\ddot{\boldsymbol{x}}_{kc} - \dot{\bar{\boldsymbol{J}}}_k\dot{\boldsymbol{q}} - \ddot{\boldsymbol{x}}_{k-\text{bias}}) + \bar{\boldsymbol{J}}_k^{\#T}(\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g})],\end{aligned} \tag{44}$$

where

$$\begin{aligned}\bar{\boldsymbol{\Lambda}}_k &= (\bar{\boldsymbol{J}}_k\boldsymbol{M}^{-1}\bar{\boldsymbol{J}}_k)^{-1},\\ \bar{\boldsymbol{J}}_k^{\#} &= \boldsymbol{M}^{-1}\bar{\boldsymbol{J}}_k^T\bar{\boldsymbol{\Lambda}}_k.\ k = 1,\ \ldots,\ L\end{aligned} \tag{45}$$

and $\bar{\boldsymbol{J}}_k$ is given by (15), using inertia-weighted generalized inverse $\bar{\boldsymbol{J}}_j^{\#}$ in lieu of $\bar{\boldsymbol{J}}_j^{\dagger}$.

Note that in this general formulation, $k = 1$ gives the control torque for the main task, i.e.,$\bar{\boldsymbol{\tau}}_1 = \boldsymbol{\tau}_{\text{task}}$. In (44) $\ddot{\boldsymbol{x}}_{k-\text{bias}}$ is a bias acceleration induced by the coupling of preceding higher priority objective. It is given by

$$\begin{cases}\ddot{\boldsymbol{x}}_{k-\text{bias}} = \boldsymbol{0}, & k = 1\\ \ddot{\boldsymbol{x}}_{k-\text{bias}} = \boldsymbol{J}_k\boldsymbol{M}^{-1}\sum_{i=1}^{k-1}\bar{\boldsymbol{\tau}}_i + \bar{\boldsymbol{\Lambda}}_k^{-1}\bar{\boldsymbol{J}}_k^{\#T}(\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g})\\ \quad -\boldsymbol{\Lambda}_k^{-1}\boldsymbol{J}_k^{\#T}(\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g}) - \dot{\bar{\boldsymbol{J}}}_k\dot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}}. & k = 2, \ldots, L\end{cases} \tag{46}$$

Writing the torque-level controller in this form reveals a close relationship of the above controller with the acceleration-resolved controller (14) and (18). In fact, in (43) and generally in (44), Coriolis/centrifugal and gravity terms are compensated in the operational and posture space. It is often more suitable to perform full Coriolis/centrifugal and gravity compensation in the joint space.[15] Having done so, we can show that by using a dynamically consistent generalized inverse in the acceleration-based controller, we will end up with exactly the same result as the torque-level controller. This can be shown as follows. Compensating the Coriolis/centrifugal and gravity terms in the joint space and substituting $\ddot{\boldsymbol{x}}_{k-\text{bias}}$ in the controller (44), we obtain

$$\begin{aligned}\boldsymbol{\tau} &= \sum_{k=1}^{L}\bar{\boldsymbol{\tau}}_k + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g},\\ \bar{\boldsymbol{\tau}}_1 &= \boldsymbol{J}_1^T\boldsymbol{\Lambda}_1(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}),\\ \bar{\boldsymbol{\tau}}_k &= \bar{\boldsymbol{J}}_k^T\bar{\boldsymbol{\Lambda}}_k(\ddot{\boldsymbol{x}}_{kc} - \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} - \boldsymbol{J}_k\boldsymbol{M}^{-1}\sum_{i=1}^{k-1}\bar{\boldsymbol{\tau}}_i),\ \ k = 2, \ldots, L.\end{aligned} \tag{47}$$

By using the inertia-weighted generalized inverse, it can be shown that the controller (47) is equal to the controller given by (14) and (18) in the absence of external torques. Substituting from (14) in (18), we have

$$\boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}}_c + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g}\ = \sum_{k=1}^{L}\boldsymbol{M}\ddot{\boldsymbol{q}}_{ck} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g}. \tag{48}$$

So we only need to show that $\bar{\boldsymbol{\tau}}_{\boldsymbol{k}} = \boldsymbol{M}\ddot{\boldsymbol{q}}_{ck}$. For $k = 1$, we have

$$\boldsymbol{M}\ddot{\boldsymbol{q}}_{c1} = \boldsymbol{M}\boldsymbol{J}_{1^{\#}}(\ddot{\boldsymbol{x}}_{1c} - \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}}) = \bar{\boldsymbol{\tau}}_1. \tag{49}$$

For $k > 1$, $\boldsymbol{M}\ddot{\boldsymbol{q}}_{ck}$ is given by

$$\begin{aligned}\boldsymbol{M}\ddot{\boldsymbol{q}}_{ck} &= \boldsymbol{M}\boldsymbol{N}_{\#k-1}\bar{\boldsymbol{J}}_k^{\#}(\ddot{\boldsymbol{x}}_{kc} - \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} - \boldsymbol{J}_k\sum_{i=1}^{k-1}\ddot{\boldsymbol{q}}_{ci})\\ &= \boldsymbol{N}_{\#k-1}^T\boldsymbol{M}\bar{\boldsymbol{J}}_k^{\#}(\ddot{\boldsymbol{x}}_{kc} - \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} - \boldsymbol{J}_k\boldsymbol{M}^{-1}\sum_{i=1}^{k-1}\boldsymbol{M}\ddot{\boldsymbol{q}}_{ci})\\ &= \bar{\boldsymbol{J}}_k^T\bar{\boldsymbol{\Lambda}}_k(\ddot{\boldsymbol{x}}_{kc} - \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} - \boldsymbol{J}_k\boldsymbol{M}^{-1}\sum_{i=1}^{k-1}\boldsymbol{M}\ddot{\boldsymbol{q}}_{ci}).\end{aligned} \tag{50}$$

Since $\bar{\boldsymbol{\tau}}_1 = \boldsymbol{M}\ddot{\boldsymbol{q}}_{c1}$, (47) and (50) imply $\bar{\boldsymbol{\tau}}_1 = \boldsymbol{M}\ddot{\boldsymbol{q}}_{c2}$ and by induction,$\bar{\boldsymbol{\tau}}_k = \boldsymbol{M}\ddot{\boldsymbol{q}}_{ck}$.

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

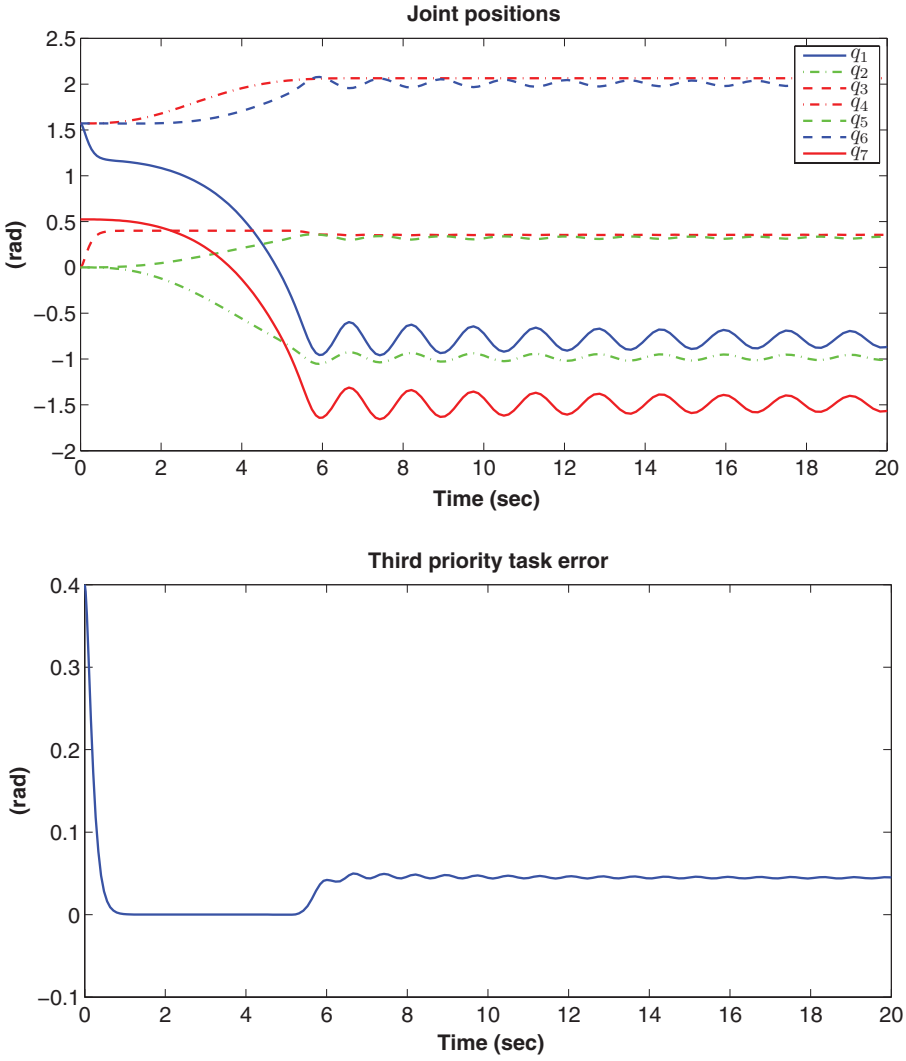


Fig. 1. (Colour online) Joint positions and the third priority task error using DLS method.

In the above derivation the following properties have been used:

$$\boldsymbol{N}_{\#k-1}^{T}\bar{\boldsymbol{J}}_{k}^{T}=\bar{\boldsymbol{J}}_{k}^{T}, \tag{51}$$

$$\boldsymbol{M}\boldsymbol{N}_{\#k-1}=\boldsymbol{N}_{\#k-1}^{T}\boldsymbol{M}. \tag{52}$$

The first one is straightforward by using idempotency property of $\boldsymbol{N}_{\#k-1}$. The second equation can be easily obtained for a general case as follows

$$\begin{aligned}\boldsymbol{M}\boldsymbol{N}_{\#k}&=\boldsymbol{M}\left(\boldsymbol{I}-\sum_{j=1}^{k}\bar{\boldsymbol{J}}_{j}^{\#}\bar{\boldsymbol{J}}_{j}\right)=\boldsymbol{M}-\sum_{j=1}^{k}\boldsymbol{M}\bar{\boldsymbol{J}}_{j}^{\#}\bar{\boldsymbol{J}}_{j}\\&=\boldsymbol{M}-\sum_{j=1}^{k}\boldsymbol{M}(\boldsymbol{M}^{-1}\bar{\boldsymbol{J}}_{j}^{T}\bar{\boldsymbol{\Lambda}}_{j})\,\bar{\boldsymbol{J}}_{j}\boldsymbol{M}^{-1}\boldsymbol{M}\\&=\boldsymbol{M}-\sum_{j=1}^{k}\bar{\boldsymbol{J}}_{j}^{T}\bar{\boldsymbol{J}}_{j}^{\#T}\boldsymbol{M}=\boldsymbol{N}_{\#k}^{T}\boldsymbol{M}.\end{aligned} \tag{53}$$

It must be mentioned that the controller (44) assumes full rank projected Jacobians $\bar{\boldsymbol{J}}_k$ and thus this assumption is also being held in the above procedure.

It can be easily verified that regardless of the choice of generalized inverse used in (14) the lower priority tasks do not affect on higher priority tasks' acceleration. The use of inertia-weighted generalized inverse may be preferable because by this choice no null space acceleration is induced by the external forces applied in the task spaces $\boldsymbol{x}_i$.[34] Despite the possibility of choosing any weighting matrix in acceleration-level control, having the joint space command acceleration given by this formulation, a variety of control algorithms can be applied because of explicit provision of trajectory in joint space. This can be regarded as another advantage of working in acceleration level. On the other hand, the above analysis reveals that the stability and singularity problems are also important issues in torque-level hierarchical control and care must be taken using this method.

It is worth to mention that Featherstone and Khatib[35] have also shown that the inertia-weighted generalized inverse is independent of load and end-effector inertia. They have

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

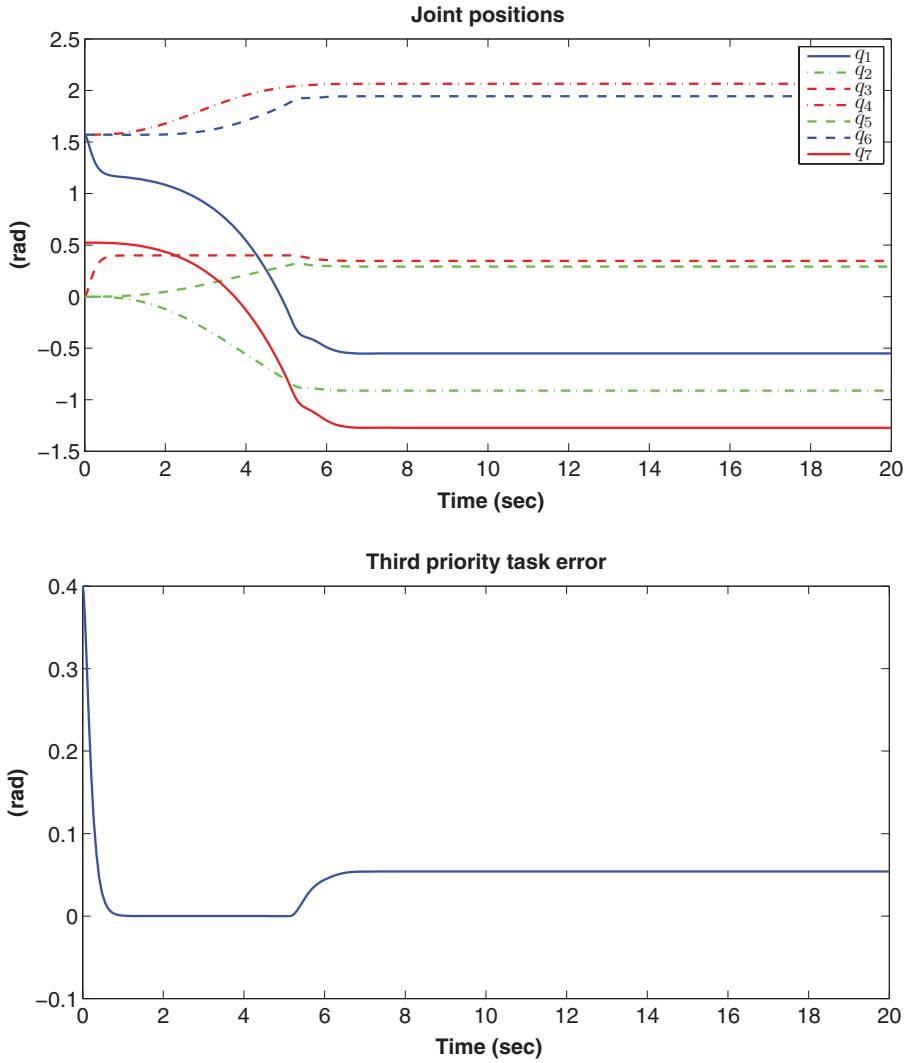


Fig. 2. (Colour online) Joint positions and the third priority task error using proposed method during singularity ($\boldsymbol{K} = 2\ \boldsymbol{I}$).

also had a comparison between acceleration and torque redundancy resolution in that paper.

7. Computer Simulation

In order to have a better understanding about the above findings, a 7-DOF KUKA lightweight arm is considered for the simulation study, neglecting the presence of joint elasticity and dissipative effects. The simulations are performed for two different case studies. In both the cases the position trajectory of the end-effector is assumed as the main task. By this choice the robot has 4 degrees of redundancy. The first case study illustrates the singularity condition in multi-priority control and shows the performance of the proposed solution to cope with singularities. In the second case study the performance of null space impedance control is shown.

7.1. Case study I

The end-effector main task is selected as a linear position trajectory from $\boldsymbol{x}_i$ to $\boldsymbol{x}_f$ with constant rotation matrix $\boldsymbol{R}_i$ as the second priority task,

$$\begin{aligned} \boldsymbol{x}_i &= [0, -0.39, 0.61]^T, \\ \boldsymbol{x}_f &= [0.085, -0.085, 0.597]^T, \\ \boldsymbol{R}_i &= \begin{pmatrix} -0.866 & -0.5 & 0 \\ 0.5 & -0.866 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} \tag{54}$$

The third joint position is controlled as the third priority task with the desired value of $x_3(t) = 0.4$. The controller (30) without the last term is used. The proportional and derivative gains for the first and second tasks are chosen as $\boldsymbol{K}_{Pi} = 100\ \boldsymbol{I}$, $\boldsymbol{K}_{Di} = 20\ \boldsymbol{I}$ and for the third task as $\boldsymbol{K}_{P3} = 81\ \boldsymbol{I}$, $\boldsymbol{K}_{D3} = 18\ \boldsymbol{I}$. The damping parameters are $\boldsymbol{\lambda}_M = 0.2$ and $\varepsilon = 0.01$.

The joint positions and the subtask errors are depicted in Fig. 1. The main task error and the second priority task error exponentially go to zero and are not shown here for brevity. Before around $t = 5.5$ sec all the tasks are performed successfully. After that, singularity

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

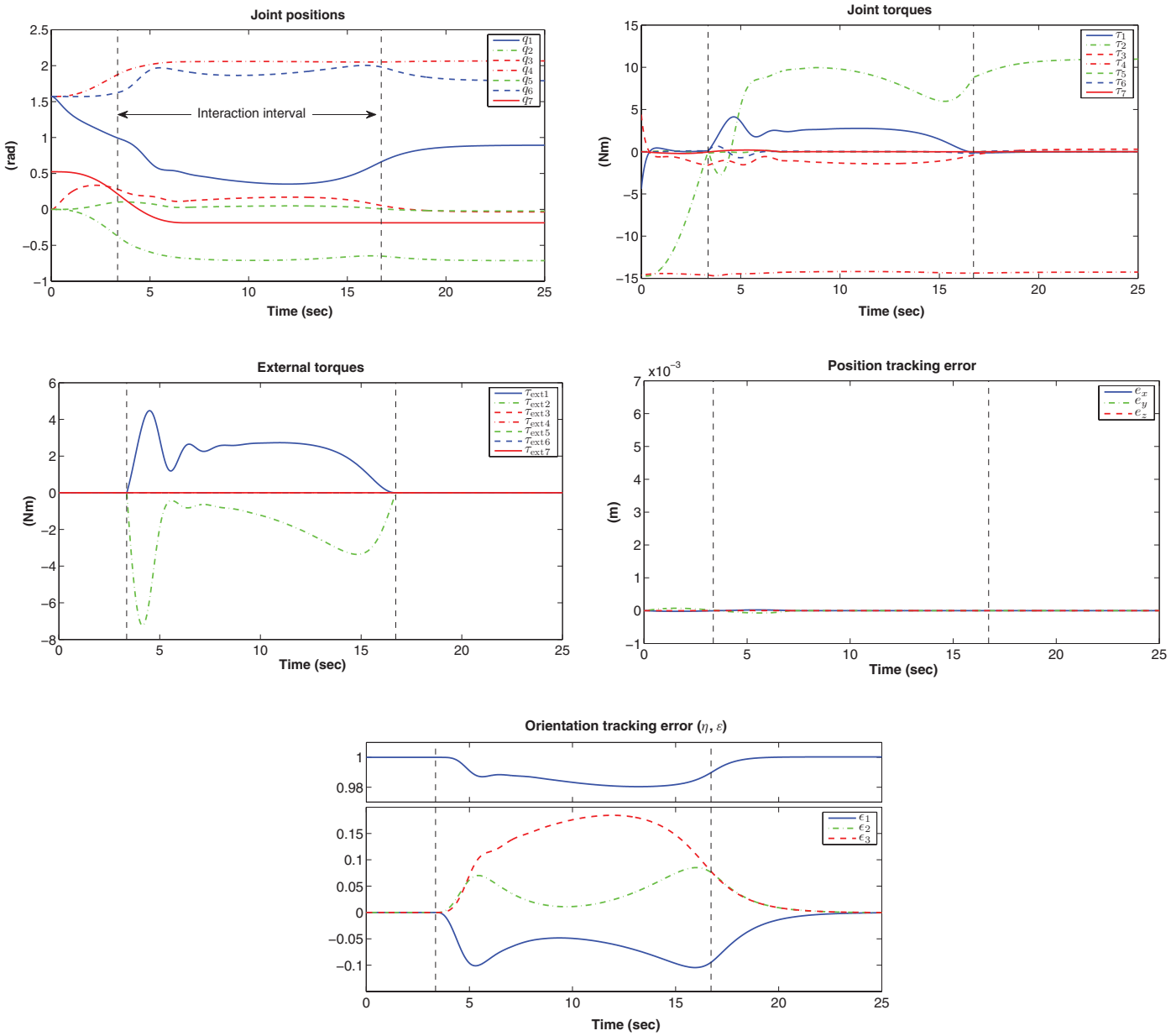


Fig. 3. (Colour online) Performance of the system under null space impedance control during contact with obstacle using external torque information.

occurs because $\bar{\boldsymbol{J}}_3$ drops rank, and hence the third priority task is sacrificed. It can be seen that using DLS in acceleration level produces an oscillatory motion near singularity.

The same task has been executed by using (30) and (31), as the command acceleration, with the same control gains as before to control the system during singularity. The other parameters in command acceleration (30) are chosen as $\boldsymbol{B} = 2\,\boldsymbol{I}$, $k = 10$. The results are reported in Fig. 2. It can be seen that the oscillatory motion during singularity is removed. Actually, with $\boldsymbol{K} = \boldsymbol{0}$ the oscillation is simply damped out, whereas with $\boldsymbol{K} = 2\,\boldsymbol{I}$ and, for instance, the choice $\boldsymbol{q}_{ds} = [\pi/4, -\pi/4, 0, \pi/2, 0, \pi/2, 0]^T$, the configuration of the robot during singularity is controlled at the same priority level of the singular subtask, and consequently a higher tracking error for the subtask is achieved.

7.2. Case study II

In order to test the performance of the null space impedance control, assume the scenario in which the body of a manipulator hits an obstacle during its maneuver. The interaction is modeled based on elastic contact between a moving obstacle (human) with the stiffness of 1000 N/m and the fourth joint of the robot arm. The simulations are performed for two cases. In both the cases the position trajectory of the end-effector given by (54) is assumed as the main task. The end-effector orientation and a constant desired joint configuration are considered as the desired task for null space. To this end, based on the second-order inverse

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

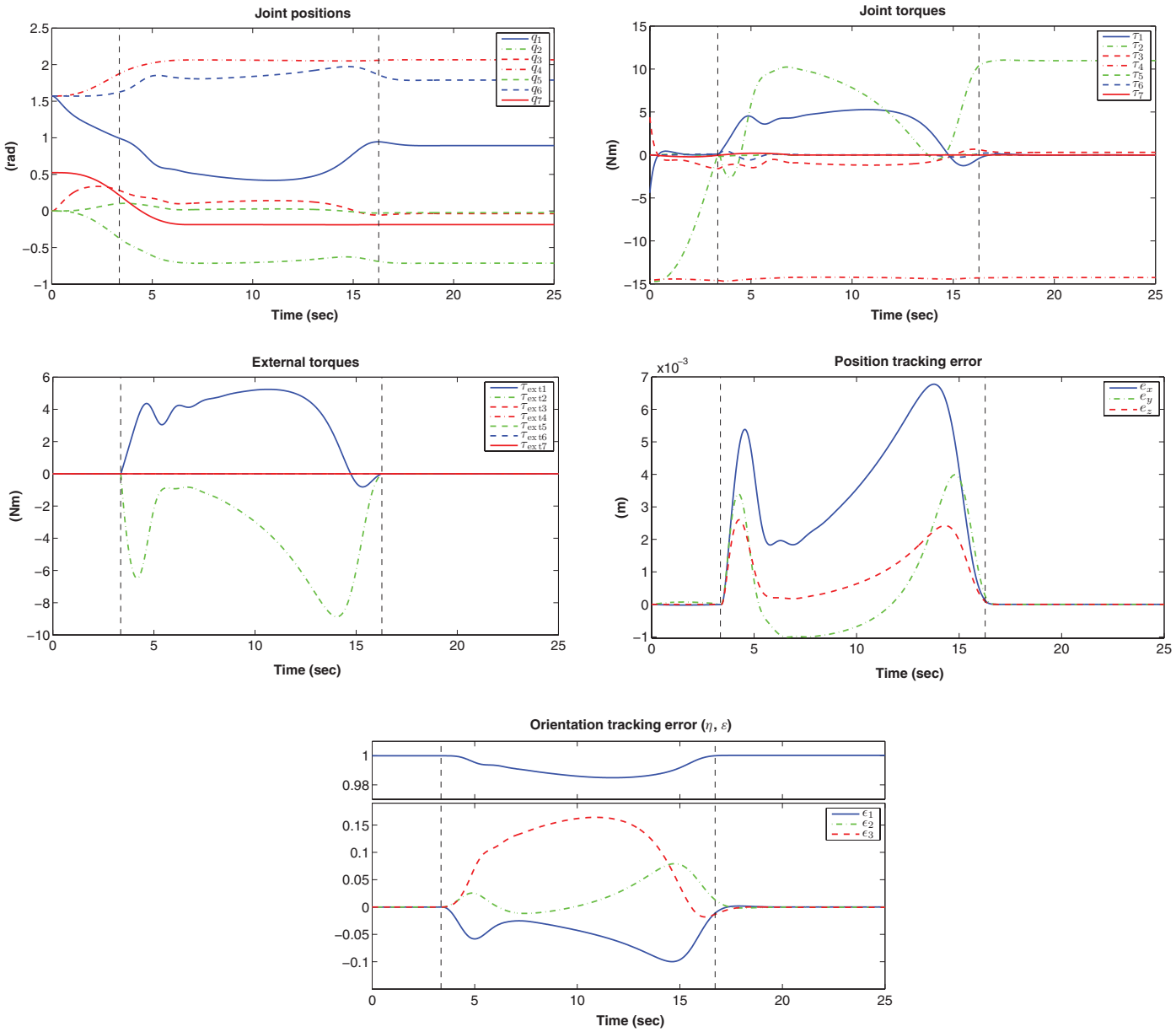


Fig. 4. (Colour online) Performance of the system under null space impedance control during contact with obstacle without using external torque information.

kinematics algorithm, the joint space-desired trajectory is computed and fed as the desired trajectory for the null space impedance.

In the first simulation we assume that an estimation of external torque resulted from interaction, $\boldsymbol{\tau}_{\text{ext}}$, is available by torque sensors. The command acceleration given by (32) is used. The proportional and derivative gains for the main task are chosen as $\boldsymbol{K}_{P1} = 100\ \boldsymbol{I}$ and $\boldsymbol{K}_{D1} = 20\ \boldsymbol{I}$, and the null space impedance matrices are chosen as $\boldsymbol{M}_d = 3\ \boldsymbol{I}$, $\boldsymbol{B}_d = 8\ \boldsymbol{I}$ and $\boldsymbol{K}_d = 4\ \boldsymbol{I}$. The results are illustrated in Fig. 3.

Here the first priority task error is zero, thanks to the use of the external torque estimation in the controller. Also during interaction, the orientation of the manipulator experiences errors since it is performed in the lower priority level as the desired trajectory for the null space impedance. Note that by this algorithm the robot does not leave the collision area, but it has a physical interaction with obstacle. As soon as the obstacle leaves the robot working area, the orientation errors go to zero and the arm comes back to its desired configuration.

In the next simulation illustrated in Fig. 4, the above scenario is performed without using external torque information in the controller. The command acceleration (36) with dynamically consistent generalized inverse and impedance parameters as $\boldsymbol{M}_d = \boldsymbol{M}$, $\boldsymbol{B}_d = 7\boldsymbol{I}$, $\boldsymbol{K}_d = 9\ \boldsymbol{I}$ is used. Other parameters chosen are the same as in the first simulation. Equations (37) and (39) show that in the absence of external torque information, by proper choice of impedance matrices and desired rest configuration, a satisfactory compliance behavior in the null space can

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

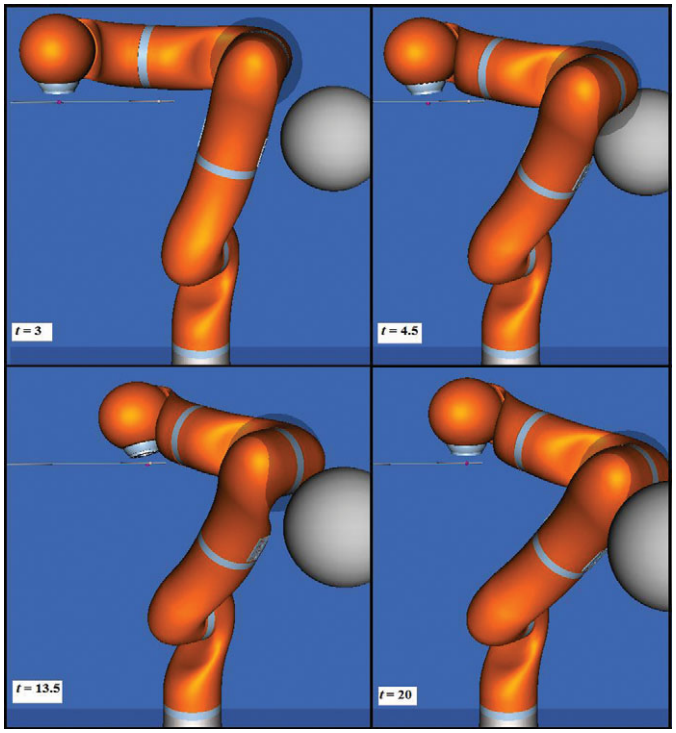


Fig. 5. (Colour online) Snapshots of the system during interaction.

be achieved. Indeed, there is an error on the main task during interaction, which can be tolerated by using high gain for the main task control or a disturbance observer[28] while giving high compliance to the null space through impedance matrices. The snapshot of the system is depicted in Fig. 5.

8. Conclusions

Acceleration-level multi-priority control algorithm has been presented in this paper. It has been shown how this formulation can be treated as a general framework to achieve control over the whole body of robot. The major contribution of the paper is the generality of the algorithm, which has been shown by bringing several control algorithms within the same framework. Specifically, the internal stability during multi-priority control has been demonstrated. A complete comparison between torque- and acceleration-level multi-priority control was performed. Null space impedance control with possible solution to cope with singularities has been proposed as a result of task prioritization, and the ability of this impedance to control the interaction of robot body has been shown by simulation.

Acknowledgements

This work was partially funded by the Italian Ministry of University and Research within the PRIN 2009 20094WTJ29_003 ROCOCO' project and by the European Community, within the FP7 ICT-287513 SAPHARI project.

References

1. H. Sadeghian, L. Villani, M. Keshmiri and B. Siciliano, "Multi-Priority Control in Redundant Robotic Systems," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Francisco, CA, 2011) pp. 3752–3757.
2. Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.* **6**(2), 3–15 (1987).
3. Y. Nakamura, *Advanced Robotics Redundancy and Optimization* (Addison-Wesley, Boston MA, 1991).
4. B. Siciliano and J. J. Slotine, "A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems," **In**: *Proceedings of the 5th International Conference on Advanced Robotics* (Pisa, Italy, 1991) pp. 1211–1216.
5. Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME J. Dyn. Syst. Meas. Control* **108**, 163–171 (1986).
6. C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. Syst. Man Cybern.* **16**, 93–101 (1986).
7. D. Nenchev and Z. M. Sotirov, "Dynamic Task-Priority Allocation for Kinematically Redundant Robotic Mechanisms," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Munich, 1994) pp. 518–524.
8. S. Chiaverini, "Singularity-robust task priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.* **13**, 398–410 (1997).
9. G. Antonelli, "Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation* (Pasadena, CA, USA, 2008) pp. 1993–1998.
10. A. De Santis, P. Pierro and B. Siciliano, "The Virtual End-Effectors Approach for Human–Robot Interaction," **In**: *Advances in Robot Kinematics* (J. Lenarcic and B. Roth, eds.) (Kluwer, Dordrecht, Netherlands, 2006) pp. 133–144.
11. H. Sadeghian, M. Keshmiri, L. Villani and B. Siciliano, "Priority Oriented Adaptive Control of Kinematically Redundant Manipulators," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation* (Saint Paul, Minnesota, USA, 2012) pp. 293–298.
12. A. De Luca, G. Oriolo and B. Siciliano, "Robot redundancy resolution at the acceleration level," *Lab. Robot. Autom.* **4**, 97–106 (1992).
13. P. Hsu, J. Hauser and S. Sastry, "Dynamic control of redundant manipulators," *J. Robot. Syst.* **6**, 133–148 (1989).
14. K. A. O'Neil, "Divergence of linear acceleration-based redundancy resolution schemes," *IEEE Trans. Robot. Autom.* **18**(4), 625–631 (2002).
15. J. Nakanishi, R Cory, M. J. Peters and S. Schaal, "Operational space control: A theoretical and empirical comparison," *Int. J. Robot. Res.* **27**, 737–757 (2008).
16. O. Khatib, L. Sentis, J. H. Park and J. Warren, "Whole-body dynamic behavior and control of human-like robots," *Int. J. Humanoid Robot.* **1**, 29–43 (2004).
17. L. Sentis and O. Khatib, "Prioritized Multi-Objective Dynamics and Control of Robots in Human Environments," **In**: *Proceedings of the IEEE/RAS International Conference on Humanoid Robots* (Santa Monica, CA, USA, 2004) pp. 764–780.
18. O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.* **3**, 1115–1120 (1987).
19. J. Peters, M. Mistry, F. Udwadia, J. Nakanishi and S. Schaal, "A unifying framework for robot control with redundant DOFs," *Auton. Robots* **24**, 1–12 (2008).
20. R. Platt, M. Abdallah and C. Wampler, "Multi-priority Cartesian Impedance Control," *Robotics Science and Systems VI* (online proceedings) (Zaragoza, Spain, 2010).
21. M. Diftler, J. Mehling, M. Abdallah, N. Radford, L. Bridgwater, A. Sanders, S. Askew, M. Linn, J. Yamokoski, F. Permenter, B. Hargrave, R. Platt, R. Savely and R. Ambrose, "Robonaut 2 the First Humanoid Robot in Space,"

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press

In: *Proceedings of the IEEE International Conference on Robotics and Automation* (Shanghai, China, 2011) pp. 2178–2183.
22. R. Platt, Jr., M. Abdallah and C. Wampler, "Multiple Priority Impedance Control," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation* (Shanghai, China, 2011) pp. 6033–6038.
23. A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed. (Springer, New York, 2002).
24. J. Park, Y. Choi, W. K. Chung and Y. Youm, "Multiple Tasks Kinematics Using Weighted Pseudo-Inverse for Kinematically Redundant Manipulators," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation* (Seoul, Korea, 2001) pp. 4041–4047.
25. G. Antonelli, F. Arrichiello and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems," *Intell. Serv. Robot.* **1**, 27–39 (2008).
26. P. Baerloclier and R. Boulic, "Task-Priority Formulations for the Kinematic Control of Highly Redundant Articulated Structures," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Victoria, BC, CANADA, 1998) pp. 323–329.
27. F. Caccavale, S. Chiaverini and B. Siciliano, "Second-order kinematic control of robot manipulators with Jacobian damped least-squares inverse: Theory and experiments," *IEEE/ASME Trans. Mechatronics* **2**(3), 188–194 (1997).
28. H. Sadeghian, M. Keshmiri, L. Villani and B. Siciliano, "Null-Space Impedance Control with Disturbance Observer," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura, Algarve, Portugal, 2012) pp. 2795–2800.
29. H. Sadeghian, F. Ficuciello, L. Villani and M. Keshmiri, "Global Impedance Control of Dual-Arm Manipulation for Safe Interaction," **In**: *Proceedings of the 10th IFAC Symposium on Robot Control* (Dubrovnik, Croatia, 2012) pp. 767–772.
30. A. Albu-Schaffer, C. Ott, U. Frese and G. Hirzinger, "Cartesian Impedance Control of Redundant Robots; Recent Results with the DLR-Light-Weight Arms," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation* (Taipei, Taiwan, 2003) pp. 3704–3709.
31. Y. Oh, W. Chung and Y. Youm, "Extended impedance control of redundant manipulators based on weighted decomposition of joint space," *J. Robot. Syst.* **15**(5), 231–258 (1998).
32. C. Ott, A. Kugi and Y. Nakamura, "Resolving the Problem of Non-Integrability of Null-Space Velocities for Compliance Control of Redundant Manipulators by Using Semi-Definite Lyapunov Functions," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation* (Pasadena, CA, USA, 2008) pp. 1999–2004.
33. F. Caccavale, C. Natale, B. Siciliano and L. Villani, "Resolved acceleration control of robot manipulators: A critical review with experiments," *Robotica* **16**, 565–573 (1998).
34. B. Nemec and L. Zlajpah, "Null-space velocity control with dynamically consistent pseudo-inverse," *Robotica* **18**, 513–518 (2000).
35. R. Featherstone and O. Khatib, "Load independence of the dynamically consistent inverse of the Jacobian matrix," *Int. J. Robot. Res.* **16**, 168–170 (1997).

https://doi.org/10.1017/S0263574713000416 Published online by Cambridge University Press