Redundancy Resolution Scheme for Manipulators Subject to Inequality Constraints

International Journal of Control, Automation and Systems VV(X) (YYYY) 1-15
http://dx.doi.org/10.1007/s12555-xxx-xxxx-x
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Redundancy Resolution Scheme for Manipulators Subject to Inequality Constraints

Daniele Proietti Pagnotta*, Andrea Monteriù, Alessandro Freddi, Sauro Longhi, and Anthony Maciejewski

Abstract: The aim of this paper is the development of a redundancy resolution scheme for manipulators able to cope with kinematic constraints. In detail, the structure of the controller is of Weighted Least Norm (WLN) type. The constraints are modeled as unilateral inequalities and can be general scalar functions (linear or nonlinear) of both the joint position and the joint velocity variables. In this work, a general procedure is proposed in order to include constraints of different types, namely functions of joint position or velocity only, functions of both joint position and velocity with a time dependent or time independent threshold. Simulations are performed in Matlab-Simulink environment and two tests are performed: the first employs a single 7-DOF arm, while in the second a dual-arm system composed of two 7-DOF manipulators is used. Results show that the proposed redundancy resolution scheme is capable of satisfying complex inequality constraints where other known methods fail.

Keywords: Constrained Redundancy Resolution, Kinematic Control, Manipulators Motion Control, Redundant Robots

1. INTRODUCTION

In the last few decades, dealing with constraints and additional subtasks in the control loop has been one of the key challenges in the manipulator kinematic control field. Typical constraints that are commonly required are joint position [1, 2], velocity [3, 4], and/or torque limit avoidance [5, 6], obstacle avoidance [7, 8], fault tolerance [9–11], or other constraints imposed by the environment. During a trajectory following task, one of the most commonly adopted solutions to ensure additional constraints is to employ redundant robots. A redundant manipulator is a robotic arm that possesses more Degrees of Freedom (DOF) than the number of variables necessary to describe the main task. Numerous studies have been proposed in the literature in order to deal with the problem of designing redundancy resolution schemes able to effectively exploit the redundancy to cope with additional constraints.

The most well-known approach is to employ the self-motion manifold generated by the degrees of redundancy via the Gradient Projection Method (GPM) [12]. The idea is to design a performance criterion to minimize and then project its gradient vector into the null space of the Jacobian of the main task, such that the homogeneous solution optimizes the criterion without affecting the main task. GPM has been employed to achieve joint limits avoidance [13], obstacle avoidance [14], or occlusion avoidance for visual servoing control [15]. In order to include more than one subtask in the control law, it is possible to define a hierarchical prioritized structure of tasks, where the lower priority task is projected into the Jacobian null space of the task with higher priority. Therefore, the performance of the task with the highest priority are always guaranteed, while the task with the lowest one can be performed only if enough degrees of redundancy are still available. This approach was firstly proposed by Nakamura *et al.* [16], Maciejewski and Klein [7], and Konstantinov *et al.* [17], and extended to incorporate singularity robustness by Chiaverini in [18]. Mansard *et al.* proposed in [19] a smooth control law able to cope with unilateral constraints at any level of the prioritized structure. In particular, they employed the inverse operator introduced in [20] to smooth irregularities due to the constraints, and extended its use to a hierarchical prioritized structure of tasks both in the joint space and in the operational space. A supervisory controller has been proposed in [21], to ensure joint limits avoidance even if the number of degrees of redundancy is not enough. The Saturation in the Null Space (SNS) algorithm was developed by Flacco *et al.* in [22] to address the problem of the differential kinematics inversion under hard joint limit constraints. The method firstly computes the Least-Norm (LN), or pseudoinverse, solution and, if necessary, saturates those joints that would overcome the limits by projecting the saturated values into the null space

Daniele Proietti Pagnotta, Andrea Monteriù, Alessandro Freddi, and Sauro Longhi are with the Department of Information Engineering, Università Politecnica delle Marche, Ancona 60131, Italy (e-mails: {d.proietti, a.monteriu, a.freddi, sauro.longhi}@univpm.it). Anthony Maciejewski is with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523, U.S.A. (e-mail: aam@colostate.edu).
* Corresponding author.

D. P. Pagnotta, A. Monteriù, A. Freddi, S. Longhi, and A. Maciejewski, "Redundancy resolution scheme for manipulators subject to inequality constraints," International Journal of Control, Automation and Systems, Springer Nature (Switzerland), vol. 21, no. 2, pp. 575–590, 2023, e-ISSN: 2005-4092.
DOI: 10.1007/s12555-021-0641-8.

of the nonsaturated joints Jacobian. If a solution cannot be found, a task scaling procedure guarantees that at least the directional part of the task will be executed. Moreover, the algorithm enforces a prioritized strategy.

A different approach is to solve the kinematic problem by finding the solution to an optimization problem. In [5], Hollerbach and Suh proposed a method to solve the redundancy while minimizing the joint torques. The authors of [23] proposed a hierarchical framework where each task is solved as a Quadratic Program (QP). In detail, the set of solutions at a specific priority level is always enforced at the levels with lower priority. The tasks are modeled as inequality constraints. In [24], the redundancy resolution problem is formulated as a QP subject to joint angle, velocity, and acceleration constraints with the joint velocity being the decision variable and the joint velocity norm as performance index. On the contrary, the authors of [25] proposed a closed form solution of the inverse kinematics problem. In detail, they developed a multi-objective real-time optimization approach that is able to minimize joint velocities and accelerations, while still avoiding joint limits. A closed form solution to this optimization is derived, and it requires only one tuning parameter.

Predictive approaches to redundancy resolution aim to implement an online predictive strategy to determine the optimal control action to handle the constraints. The authors of [26] considered each joint as a k-th order integral system and a continuous time formulation. As a consequence, longer prediction and control horizons can be used without increasing the computational effort. Joint limits are introduced in the linear model predictive control problem.

Solutions based on the Extended [27] and Augmented [28] Jacobian techniques aim to augment the task vector by considering the additional tasks as a part of the main task. The number of subtasks is limited by the number of degrees of redundancy and, in contrast to the projection into the null space based methods, they can not guarantee completion of the end-effector task. In addition, these techniques introduce algorithmic singularities.

Recently, thanks to the increased efficiency of the parallel computing, inverse kinematics methods based on neural networks have become popular. Authors of [29] exploited a neural network to learn the inverse kinematic function. This approach makes possible to avoid the formal derivation of the inverse kinematic function and, also, to take into account for manufacturing tolerances and other errors due to the differences between the ideal and the real model of the robot. A primal-dual neural network to perform real-time control has been proposed in [30]. This approach allows to directly take noise control into account, making possible to design a controller able to track references in noisy environments. The authors of [31] proposed a position-force controller based on projection recursive neural networks. The proposed control scheme is able to optimize the joint torque and, concurrently, to manage multiple inequality constraints.

Another well-known approach is the Weighted Least Norm (WLN) method. The WLN method employs the weighted pseudoinverse to minimize the weighted norm of the joint velocity. It is possible to force the joint velocity to be zero by appropriately tuning the weighting factor. In [1], Chan and Dubey demonstrated that the WLN technique is effective for the joint limit avoidance problem. More recently, authors of [32] generalized the approach developing the General Weighted Least Norm (GWLN) which is able to cope with constraints modeled as inequalities. However, the number of additional tasks is limited by the DOF of the manipulator.

This work aims to present a general approach to developing a second-order redundancy resolution scheme able to cope with subtasks modeled as unilateral inequalities. The proposed approach represents an alternative to those previously described, and its design is straightforward once the constraints to satisfy are well defined. In particular, thanks to the second-order kinematics, the constraints can be general functions of both joint position and joint velocity variables. The redundancy resolution scheme is developed based on the second-order differential kinematic equation and on the WLN approach in order to take into account the constraints. Subtasks that are functions of joint position are treated separately from those that are functions of joint velocity. Then, a general procedure for mixed constraints is presented. Moreover, the stability of the proposed scheme and the fulfillment of the constraints are demonstrated, as well as performance that is inline, and in some cases even better, with respect to that of state-of-the-art algorithms.

The rest of the paper is organized as follows. Section 2 presents the theoretical background about the basic kinematic control of redundant manipulators and the WLN and GWLN approaches. The redundancy resolution scheme is derived in Section 3. First, the cases for the joint position and velocity constraints are separately investigated and, then, the final version of the scheme for general constraints is explained. Constraints analysis is also presented. Section 4 shows the simulation results for two different case studies. In the former, a single 7-DOF manipulator is employed while, in the latter, a dual-arm system composed of two 7-DOF arms is considered. Conclusions and future work end the paper.

2. BACKGROUND

2.1. Basic Kinematic Control of Redundant Manipulators

Let us consider an n-DOF manipulator performing an m-dimensional task. The forward kinematic function of the manipulator, which represents the mapping from the joint space to the task space, is defined as

$$x = \mathrm{fk}(q) \tag{1}$$

where $q \in \mathbb{R}^n$ is the vector of the robot configuration, $x \in \mathbb{R}^m$ is the pose of the end-effector, while $\mathrm{fk}(\cdot)$ is the nonlinear forward kinematic function.

The first-order differential kinematics relates the robot joint velocities $\dot{q} \in \mathbb{R}^n$ to the end-effector velocity $\dot{x} \in \mathbb{R}^m$, and it can be derived by differentiating (1) with respect to time

$$\dot{x} = J(q)\dot{q} \tag{2}$$

where $J(q) \in \mathbb{R}^{m\times n}$ is the manipulator Jacobian matrix. The inverse kinematics aims to find a joint trajectory solution q such that the trajectory of the end-effector $x = \mathrm{fk}(q)$ follows the desired one x_d as close as possible. The inversion of the kinematic relation (2) provides a unique solution when $m = n$, namely the dimension of the task space equals the number of DOF, when J is of full rank. Otherwise, when $m < n$, there, in general, exists an infinite number of solutions $\dot{q}$ that allow the robot to track the desired task space trajectory x_d. These solutions satisfy (2) and can be expressed as

$$\dot{q} = J^{\dagger}\dot{x}_d + P\dot{q}_0 \tag{3}$$

where $J^{\dagger}$ is the Moore-Penrose pseudoinverse of J, which is defined even when J is not square or it is not full rank [33]. In the case when J is full rank, the pseudoinverse is commonly defined as $J^{\dagger} = J^T(JJ^T)^{-1}$. The matrix $P = I_n - J^{\dagger}J$ is the orthogonal projector on the null space of J (satisfying $JP = 0$), while $\dot{q}_0 \in \mathbb{R}^n$ is an arbitrary joint velocity vector. When $\dot{q}_0 = 0$, (3) provides the LN solution (the solution of minimum norm of joint velocities) to the system (2) or, if $\dot{x}_d$ is not in the column space of J, the solution that minimizes the tracking error $\dot{x} - J\dot{q}$. The vector $\dot{q}_0$ can also be chosen as the gradient vector of a desired performance function in order to obtain the GPM [12] solution. The GPM defines a subtask represented by the minimum point of the performance criterion and performs this subtask in the null space of J. Therefore, there is no guarantee that the subtask is satisfied at each instant of time.

The robot configurations where J is rank deficient are called singularities. When the robot is close to one of these configurations, high joint velocities are required to generate small $\dot{x}$. The damped least square approach [34] overcomes this drawback by introducing a damping factor $\lambda \in \mathbb{R}^+$ that represents a trade-off between the tracking error and the feasibility of the solution. Considering $\dot{q}_0 = 0$, Eq. (3) can be rewritten as

$$\dot{q} = J^T(JJ^T + \lambda^2 I)^{-1}\dot{x}_d \tag{4}$$

which is the solution to the problem

$$\min_{\dot{q}} \quad \|\dot{x}_d - J\dot{q}\|^2 + \lambda^2\|\dot{q}\|^2 \tag{5}$$

If $\lambda = 0$, allowed only if J is of full rank, (4) degenerates into the LN solution.

The second-order differential kinematic equation can be obtained by further differentiating (2) with respect to time

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q,\dot{q})\dot{q} \tag{6}$$

Different from the first-order differential kinematics, Eq. (6) defines a non-homogeneous mapping among the joints and the end-effector acceleration. As for the first-order case, a solution can be expressed in the general form

$$\ddot{q} = J^{\dagger}(\ddot{x}_d - \dot{J}\dot{q}) + P\ddot{q}_0 \tag{7}$$

where $J^{\dagger}$ and P remain the same, $\ddot{q}_0$ is an arbitrary joints acceleration vector, while $\ddot{x}_d$ is the acceleration vector of the desired end-effector trajectory. By choosing $\ddot{q}_0 = 0$, (7) gives the LN acceleration solution.

2.2. General Weighted Least Norm

The WLN method was proposed in [1] in order to achieve joint limits avoidance. The control law at the first-order kinematics

$$\dot{q} = W^{-1}J^T(JW^{-1}J^T)^{-1}\dot{x}_d \tag{8}$$

represents the solution of the following optimization problem

$$\begin{aligned} \min_{\dot{q}} \quad & \frac{1}{2}\dot{q}^T W \dot{q} \\ \text{s.t.} \quad & \dot{x}_d = J\dot{q} \end{aligned} \tag{9}$$

Note that solution (8) minimizes the weighted norm of the joint velocity vector.

Let the performance criterion be defined as

$$H(q) = \sum_{i=1}^{n} \frac{1}{4}\frac{(q^i_{\max} - q^i_{\min})^2}{(q^i_{\max} - q^i)(q^i - q^i_{\min})} \tag{10}$$

where q^i is the i-th joint position, while $q^i_{\max}$ and $q^i_{\min}$ are the upper and the lower bound of the admissible joint range, respectively. Note that function $H(q)$ is normalized with respect to the center of each joint range, and it will tend to infinity at the joint limits. The weighting matrix $W \in \mathbb{R}^{n\times n}$ is diagonal and positive definite, and its diagonal elements w_i are defined as

$$w_i = \begin{cases} 1 + \left|\dfrac{\partial H(q)}{\partial q^i}\right|, & \text{if } \triangle\left|\dfrac{\partial H(q)}{\partial q^i}\right| \geq 0 \\ 1, & \text{if } \triangle\left|\dfrac{\partial H(q)}{\partial q^i}\right| < 0 \end{cases} \tag{11}$$

where $\triangle|\partial H(q)/\partial q^i|$ is the rate of change of $|\partial H(q)/\partial q^i|$.

When the i-th joint is approaching its limit, the term $|\partial H(q)/\partial q^i| \to \infty$ and, thus, $w_i \to \infty$. Therefore, the corresponding element of the inverse weighting matrix W^{-1} approaches zero. From (8), the corresponding joint velocity command $q^i \to 0$, stopping the motion of the i-th joint at its limit. If the joint is moving away from its bound, $\triangle|\partial H(q)/\partial q^i| < 0$ and the movement is not suppressed.

It is worth noting that the matrix $JW^{-1}J^T$ may be ill-conditioned due to weighting factors that are close to zero. If this happens, solution (8) may be unfeasible. A feasible solution can be obtained by introducing the damping factor but allowing a higher tracking error. Eq. (8) becomes

$$\dot{q} = W^{-1}J^T(JW^{-1}J^T + \lambda^2 I)^{-1}\dot{x}_d \tag{12}$$

and the authors of [35] suggest setting the damping factor λ as

$$\lambda^2 = \begin{cases} \left(1 - \left(\frac{\tilde{\sigma}_{\min}}{\varepsilon}\right)^2\right)\lambda^2_{\max}, & \text{if } \tilde{\sigma}_{\min} < \varepsilon \\ 0, & \text{if } \tilde{\sigma}_{\min} \geq \varepsilon \end{cases} \tag{13}$$

where $\tilde{\sigma}_{\min}$ is the minimum singular value of the matrix $JW^{-1/2}$, ε denotes the amplitude of the region where a damping factor different from zero is applied, while $\lambda_{\max}$ represents the maximum admissible value for the damping factor.

The authors of [32] proposed an extension of the WLN, namely the GWLN method, capable of dealing with general subtasks. The idea was exploiting the effectiveness of the WLN in handling joint limits to cope with inequality constraints. They defined the i-th constraint function as

$$H_i(q) \geq h_i \qquad i = 1,\ldots,r \tag{14}$$

where $H_i(q) : \mathbb{R}^n \to \mathbb{R}$, $h_i \in \mathbb{R}$ is the threshold, while $r \leq n$ is the number of constraints. Each constraint $H_i(q)$ was considered as a virtual joint variable and a transformation between the real joint space and the virtual joint space was subsequently defined. In the virtual joint space, the problem of handling constraints expressed by (14) becomes a problem of virtual joint limit avoidance. Thus, the WLN can be used in the virtual joint space to fulfill (14).

By defining the i-th virtual joint variable $q_v^i = H_i(q)$, it is possible to establish a linear map among the real and the virtual joint spaces at the velocity level

$$\dot{q}_v = T(q)\dot{q} \tag{15}$$

The transformation matrix $T(q)$ is derived as

$$T(q) = \begin{bmatrix} \nabla H_1(q) \\ \vdots \\ \nabla H_r(q) \\ K \end{bmatrix} \tag{16}$$

where $T(q) \in \mathbb{R}^{n\times n}$, $\nabla H_i(q)$ is the gradient vector of the constraint function $H_i(q)$, while $K \in \mathbb{R}^{(n-r)\times n}$ is a matrix such that $\nabla H_i(q)K^T = 0 \ \forall i$.

If the matrix $T(q)$ is of full rank, the kinematic equation (2) can be rewritten in the virtual joint space as

$$\dot{x} = J_v(q)\dot{q}_v \tag{17}$$

where $J_v(q) = J(q)T^{-1}(q)$ is the virtual Jacobian matrix. It is now possible to apply WLN to the system (17), which gives the solution

$$\dot{q}_v = W_v^{-1}J_v^T(J_vW_v^{-1}J_v^T)^{-1}\dot{x}_d \tag{18}$$

and, then, transform solution (18) back to the real joint space $\dot{q} = T^{-1}(q)\dot{q}_v$. In Eq. (18), W_v^{-1} is the inverse of the weighting matrix W_v and, because W_v^{-1} is a design parameter, it is possible to define its elements w_{v_i} instead of those of W_v.

3. PROPOSED REDUNDANCY RESOLUTION SCHEME

We first consider constraints of the following form

$$g_i(\dot{q}) \geq \tau_{g_i} \qquad i = 1,\ldots,r \tag{19a}$$

$$f_i(q) \geq \tau_{f_i} \qquad i = 1,\ldots,r \tag{19b}$$

where $f_i(q), g_i(\dot{q}) : \mathbb{R}^n \to \mathbb{R}$, while $\tau_{f_i}, \tau_{g_i} \in \mathbb{R}$ are the constraints thresholds. In this work, we considered a number of constraints $r \leq n$ in order to keep a square mapping matrix. However, it is possible to have more constraints than the number of DOF as explained in Sec. 3.5.

In this section, we first present the derivation of the controller for joint velocity constraints, i.e., (19a). Subsequently, functions of only joint positions, i.e., (19b), are taken into account. Lastly, the general case, namely $c_i(q,\dot{q}) \geq \tau_{c_i} - \tau_i(t)$, $i = 1,\ldots,r$, which simultaneously considers functions of joint positions and velocities with a constant or a time-varying threshold, is presented. The proposed redundancy resolution scheme is developed based on the second-order differential kinematic equation.

3.1. Joint Velocity Constraints $g_i(\dot{q})$

We choose the virtual joint variables as

$$q_v^i = g_i(\dot{q}) \tag{20}$$

which involves the following mapping between the real joint space and the virtual joint space

$$\dot{q}_v = T_g(\dot{q})\ddot{q}. \tag{21}$$

The transformation matrix $T_g(\dot{q})$ is given by

$$T_g(\dot{q}) = \begin{bmatrix} \nabla g_1(\dot{q}) \\ \vdots \\ \nabla g_r(\dot{q}) \\ K_g \end{bmatrix} \tag{22}$$

where $T_g(\dot{q}) \in \mathbb{R}^{n\times n}$, $\nabla g_i(\dot{q})$ is the gradient vector of $g_i(\dot{q})$, while K_g is defined as in (16).

Remark 1: Please note that the method proposed in this paper requires the transformation matrix to be full row rank, i.e. the gradient vectors must be linearly independent over the robot configuration space.

Furthermore, in case of box constraints, rows of T_g become constants and invertibility is always guaranteed as long as the constraints are well-designed.

Remark 2: The choice of the matrix K_g in (22) such that $\nabla g_i(\dot{q})K_g^T$
$= 0\ \forall i$ makes the non-constrained components of the virtual joint velocity vector not influence the constrained ones when the real joint acceleration command is computed.

In order to prove this fact, consider the mapping T_g partitioned as $T_g = [\nabla g^T\ K_g^T]^T$, $\nabla g \in \mathbb{R}^{r\times n}$ and $K_g \in \mathbb{R}^{(n-r)\times n}$, while the inverse mapping as $T_g^{-1} = [P\ Q]$, $P \in \mathbb{R}^{n\times r}$ and $Q \in \mathbb{R}^{n\times(n-r)}$, where $P = K_g^{\perp}(\nabla g K_g^{\perp})^{-1}$ and $Q = \nabla g^{\perp}(K_g \nabla g^{\perp})^{-1}$. The orthogonal matrices $K_g^{\perp}$ and $\nabla g^{\perp}$ are such that $K_g K_g^{\perp} = 0$, and $\nabla g \nabla g^{\perp} = 0$. It follows that the virtual joint velocity vector can be partitioned as $\dot{q}_v = [\dot{q}_{v_c}^T\ \dot{q}_{v_*}^T]^T$, where $\dot{q}_{v_c} \in \mathbb{R}^r$ are the constrained components while $\dot{q}_{v_*} \in \mathbb{R}^{n-r}$ are the non constrained ones.

From (21), it follows that $\ddot{q} = [P\ Q][\dot{q}_{v_c}^T\ \dot{q}_{v_*}^T]^T = P\dot{q}_{v_c} + Q\dot{q}_{v_*}$. Because $\nabla g K_g^T = 0$, the matrix K_g^T is a special $\nabla g^{\perp}$; similarly, ∇g^T is a $K_g^{\perp}$. Therefore, matrices P and Q can be rewritten as $P = \nabla g^T(\nabla g \nabla g^T)^{-1} = \nabla g^{\dagger}$ and $Q = K_g^T(K_g K_g^T)^{-1} = K_g^{\dagger}$, while the real joint acceleration command as $\ddot{q} = \nabla g^{\dagger}\dot{q}_{v_c} + K_g^{\dagger}\dot{q}_{v_*}$. Considering that $\nabla g K_g^{\dagger} = 0$ holds, the matrix $K_g^{\dagger}$ is a projector into the null space of ∇g, thus the actual velocities $\dot{q}_{v_c}$ are not influenced by the non constrained $\dot{q}_{v_*}$.

By substituting (21) in (6), the second order differential kinematic equation can be written in the virtual space as

$$\ddot{x} = J_v \dot{q}_v + \dot{J}\dot{q} \tag{23}$$

where $J_v = JT_g^{-1}$. The WLN solution to the system (23) is

$$\dot{q}_v = W^{-1}J_v^T(J_v W^{-1} J_v^T)^{-1}(\ddot{x}_d - \dot{J}\dot{q}) \tag{24}$$

where $W \in \mathbb{R}^{n\times n}$, while the elements w_i of $W^{-1} = \mathrm{diag}\{w_1, \ldots, w_n\}$ are defined in Sec. 3.4.

Results presented in this paper are based on the assumption that the matrix $J_v W^{-1} J_v^T$ in Eq. (24) is invertible. However, at some time instants, algorithmic singularities may arise and the matrix could lose rank. In this case, the algorithm could be modified by introducing a damping factor (as explained in Sec. 2) to compute a solution, but this may result in constraints violation.

3.2. Joint Position Constraints $f_i(q)$

As seen in the previous section, in order to take into account inequality (19a), the mapping between the real joint space and the virtual joint space is used. This is obtained by applying the time-derivative of the virtual joint variable: this allows us to directly design a second-order redundancy resolution scheme. If the same method is applied to take into account inequality (19b), it is instead not possible to derive a control scheme based on the real joint accelerations, due to the fact that the mapping does not depend on them.

Proposition 1: Inequality (19b) can be written at the velocity level as

$$\dot{f}_i(q,\dot{q}) \geq (\tau_{f_i} - f_i(q))k \tag{25}$$

For continuous systems, whether $k > 0$, $f_i(q_0) > \tau_{f_i}$ (where $q_0 \in \mathbb{R}^n$ is the vector of initial joint positions), and (25) is satisfied $\forall t \in [t_0, t_1]$, then (19b) will hold $\forall t \in [t_0, t_1]$.

Proof: Let $\tilde{f}_i(q) = \tau_{f_i} - f_i(q)$. Thus

$$\dot{\tilde{f}}_i(q,\dot{q}) = -\dot{f}_i(q,\dot{q}) \leq -(\tau_{f_i} - f_i(q))k = -k\tilde{f}_i(q) \quad \Rightarrow \quad \dot{\tilde{f}}_i \leq -k\tilde{f}_i$$

Moreover

$$\tilde{f}_i(q_0) = \tau_{f_i} - f_i(q_0) \rightarrow \tilde{f}_i(q_0) - \tau_{f_i} = -f_i(q_0) \leq -\tau_{f_i} \quad \Rightarrow \quad \tilde{f}_i(q_0) \leq 0$$

From the Comparison Lemma (3.4 of [36]), with $v = \tilde{f}_i(q)$, the following holds

$$\tilde{f}_i(q) \leq \tilde{f}_i(q_0)e^{-k(t-t_0)} \leq 0$$

Thus

$$\tilde{f}_i(q) \leq 0 \rightarrow \tau_{f_i} - f_i(q) \leq 0 \quad \Rightarrow \quad f_i(q) \geq \tau_{f_i}$$

The proof is complete. □

Constraint (25) depends on both joint positions q and joint velocities $\dot{q}$, and the threshold is time-varying, due to the presence of the term $f_i(q)$. This general case is presented in the next subsection.

3.3. General Case: Joint Position and Velocity Constraints

3.3.1 Generic Constraint Function with Constant Threshold

We first consider a generic constraint with a constant threshold

$$c_i(q,\dot{q}) \geq \tau_{c_i} \qquad i = 1,\ldots,r \tag{26}$$

where $c_i(q,\dot{q}) : \mathbb{R}^n \to \mathbb{R}$ is the constraint function of joint positions q and velocities $\dot{q}$, while $\tau_{c_i} \in \mathbb{R}$ is the threshold of the function c_i. The virtual joint variables can be selected as

$$q_v^i = c_i(q,\dot{q}) \tag{27}$$

This leads to a virtual joint velocity of the form

$$\dot{q}_v^i = \nabla c_{q_i}\dot{q} + \nabla c_{\dot{q}_i}\ddot{q} \tag{28}$$

where $\nabla c_{q_i} = \partial c_i(q,\dot{q}) \big/ \partial q$ and $\nabla c_{\dot{q}_i} = \partial c_i(q,\dot{q}) \big/ \partial \dot{q}$. The mapping between the two spaces can be written in compact form as

$$\dot{q}_v = T_c\ddot{q} + k_c(q,\dot{q}) \tag{29}$$

where

$$T_c = \begin{bmatrix} \nabla c_{\dot{q}_1} \\ \vdots \\ \nabla c_{\dot{q}_r} \\ K_c \end{bmatrix} \in \mathbb{R}^{n\times n} \quad k_c(q,\dot{q}) = \begin{bmatrix} k_1(q,\dot{q}) \\ \vdots \\ k_r(q,\dot{q}) \\ 0_{n-r} \end{bmatrix} \in \mathbb{R}^n \tag{30}$$

and

$$k_i(q,\dot{q}) = \begin{cases} \nabla c_{q_i}\dot{q}, & \text{if } c_i(q,\dot{q}) \\ 0, & \text{if } c_i(\dot{q}) \end{cases} \tag{31}$$

The matrix K_c is selected as in (22) and (16), while 0_{n-r} is the zero vector.

Note that mapping (29) is a more general version of (21), because constraint (26) is a generalization of (19a). The term $k_c(q,\dot{q})$ encapsulates the dependency of the constraint function $c_i(q,\dot{q})$ on the joint position variables. If a constraint $c_i(\dot{q})$ does not depend on q, it is possible to derive (21) from (29) by selecting zero elements in (31).

The kinematic equation can be derived by substituting (29) in (6)

$$\ddot{x} = J_v\dot{q}_v - J_v k_c + \dot{J}\dot{q} \tag{32}$$

where $J_v = JT_c^{-1}$, and its WLN solution is

$$\dot{q}_v = W^{-1}J_v^T(J_vW^{-1}J_v^T)^{-1}(\ddot{x}_d + J_v k_c - \dot{J}\dot{q}). \tag{33}$$

3.3.2 Generic Constraint Function with Time-Varying Threshold

Finally we consider the most general case where the threshold of the constraint function is time-varying. This case allows us to solve joint position constraints by transforming them to velocity constraints, namely inequality (25). Inequality (26) is modified by adding a time-dependent term to the threshold

$$c_i(q,\dot{q}) \geq \tau_{c_i} - \tau_i(t) \qquad i = 1,\ldots,r \tag{34}$$

where $\tau_i(t)$ is a generic time-varying scalar function. In order to solve the constraint (25), function $\tau_i(t)$ has to be set as $f_i(q)$, while $c_i(q,\dot{q}) = \dot{f}_i(q,\dot{q})$ (considering $k = 1$).

Virtual joint variables are selected as

$$q_v^i = c_i(q,\dot{q}) + \tau_i(t) \tag{35}$$

from which the virtual joint velocity is

$$\dot{q}_v^i = \nabla c_{q_i}\dot{q} + \nabla c_{\dot{q}_i}\ddot{q} + \dot{\tau}_i \tag{36}$$

where ∇c_{q_i} and $\nabla c_{\dot{q}_i}$ are defined as in (28). Therefore, the mapping between real and virtual joint spaces can be written as

$$\dot{q}_v = T_c\ddot{q} + k_c + k_t \tag{37}$$

where T_c and k_c are exactly the same as in (29), while

$$k_t = \begin{bmatrix} \dot{\tau}_1 \\ \vdots \\ \dot{\tau}_r \\ 0_{n-r} \end{bmatrix} \in \mathbb{R}^n. \tag{38}$$

Similarly to k_c in (29), which encapsulates the dependency of $c_i(q,\dot{q})$ on q, the term k_t incorporates the time dependency of the threshold.

As in the previous cases, the kinematic relation is obtained by substituting (37) to (6)

$$\ddot{x} = J_v\dot{q}_v - J_v k_c - J_v k_t + \dot{J}\dot{q} \tag{39}$$

and the WLN solution is

$$\dot{q}_v = W^{-1}J_v^T(J_vW^{-1}J_v^T)^{-1}(\ddot{x}_d + J_v k_c + J_v k_t - \dot{J}\dot{q}). \tag{40}$$

The final resolution scheme in the real joint space is then derived from (37) and (40)

$$\ddot{q} = T_c^{-1}(W^{-1}J_v^T(J_vW^{-1}J_v^T)^{-1} \\ (\ddot{x}_d + J_v k_c + J_v k_t - \dot{J}\dot{q}) - k_c - k_t). \tag{41}$$

Note that (41) includes all the previous cases, thus it is possible to incorporate different types of constraints, i.e. joint position or velocity constraints, or a combination of them, by selecting the proper terms k_c and k_t.

3.4. Weighting Function and Constraints Analysis

The open loop scheme (41) is modified by employing the Closed Loop Inverse Kinematics (CLIK) scheme [37] in order to avoid errors due to disturbances and discrete implementation. The closed loop scheme can be written as

$$\ddot{q} = T_c^{-1}(W^{-1}J_v^T(J_vW^{-1}J_v^T)^{-1} \\ (\ddot{x}_d + J_vk_c + J_vk_t - \dot{J}\dot{q} + K_D\dot{e} + K_Pe) - k_c - k_t) \quad (42)$$

where $K_D, K_P \in \mathbb{R}^{m\times m}$ are diagonal, positive definite gain matrices, while the tracking error is defined as $e = x_d - x$. The solution (42) applied to system (6) makes the tracking error e converge to zero. To prove this fact, the closed loop dynamic equation of the tracking error $\ddot{e} = -K_D\dot{e} - K_Pe$ can be derived by substituting (42) into (6). Under the assumption of positive definite K_D and K_P, the error e will converge to zero.

The elements of the weighting matrix $W^{-1} = \operatorname{diag}\{w_1, \ldots, w_n\}$ in (42) are set to

$$w_i = \begin{cases} 1 - \arctan\Big(G(q_v^i)\Big)\dfrac{2}{\pi}, & \dot{q}_v^i < 0 \\ 1, & \dot{q}_v^i \geq 0 \end{cases} \quad (43)$$

where

$$G(q_v^i) = \frac{\alpha}{q_v^i - \tau_{c_i}}, \quad \alpha \in \mathbb{R}^+ \quad (44)$$

where α is a parameter to adjust the velocity of which w_i tends to zero. When the i-th virtual joint is moving away from its limit, there is no movement suppression because the weighting factor w_i is set equal to one. As q_v^i starts moving toward the limit τ_{c_i}, the movement is gradually suppressed, reaching maximum suppression before or, in the worst case, at the time instant the constraint limit is reached.

Theorem 1: Assume that the matrix $J_vW^{-1}J_v^T$ is of full rank. Then, constraint (34) is satisfied under the redundancy resolution scheme (42) with the weighting function (43) for the system (6).

Proof: From (35) and (42)

$$\dot{q}_v^i = \dot{c}_i + \dot{\tau}_i \\ = \nabla c_{\dot{q}_i} T_c^{-1} W^{-1} J_v^T (J_vW^{-1}J_v^T)^{-1} \\ (\ddot{x}_d + J_vk_c + J_vk_t - \dot{J}\dot{q} + K_D\dot{e} + K_Pe) \\ - \nabla c_{\dot{q}_i} T_c^{-1}(k_c + k_t) + \nabla c_{q_i}\dot{q} + \dot{\tau}_i$$

From the definition of T_c in (30), it can be derived that $\nabla c_{\dot{q}_i} T_c^{-1} = [0_1, \cdots, 0_{i-1}, 1_i, 0_{i+1}, \cdots, 0_n]$. Thus, $-\nabla c_{\dot{q}_i} T_c^{-1}$ $(k_c + k_t) + \nabla c_{q_i}\dot{q} + \dot{\tau}_i = -k_{c_i} - k_{t_i} + \nabla c_{q_i}\dot{q} + \dot{\tau}_i = 0$, accordingly to (30), (31) and (38).

Moreover, $\nabla c_{\dot{q}_i} T_c^{-1} W^{-1} = [0, \cdots, 0, w_i, 0, \cdots, 0]$. The previous equation can be simplified as

$$\dot{q}_v^i = w_i v_i$$

where v_i is the i-th element of $J_v^T(J_vW^{-1}J_v^T)^{-1}(\ddot{x}_d + J_vk_c + J_vk_t - \dot{J}\dot{q} + K_D\dot{e} + K_Pe)$.

In the following, the reductio ad absurdum approach is adopted. Let us suppose the i-th virtual joint violates the constraint at the time instant t_f, namely $q_v^i(t_f) = \tau_{c_i}$. There exists a small positive constant $\varepsilon \in \mathbb{R}$ such that q_v^i is monotonically decreasing in the time interval $[t_f - \varepsilon, t_f]$. Substituting (43) and (44) leads to

$$\dot{q}_v^i = \left(1 - \arctan\left(\frac{\alpha}{q_v^i - \tau_{c_i}}\right)\frac{2}{\pi}\right) v_i \\ = \left(\arctan\left(\frac{q_v^i - \tau_{c_i}}{\alpha}\right)\frac{2}{\pi}\right) v_i$$

where the property $\arctan\Big(1/x\Big) = \pi/2 - \arctan(x)$, $x > 0$ is used. Because the matrix $J_vW^{-1}J_v^T$ is of full rank, then v_i is bounded. Thus, there exists a positive constant $\eta \in \mathbb{R}$ such that $|v_i| \leq \eta \;\; \forall t \in [t_f - \varepsilon, t_f]$. Furthermore, $-\eta \leq v_i \leq 0$ because q_v^i is monotonically decreasing. There is

$$\dot{q}_v^i \geq -\arctan\left(\frac{q_v^i - \tau_{c_i}}{\alpha}\right)\frac{2\eta}{\pi} \\ \geq -\frac{2\eta}{\pi\alpha}(q_v^i - \tau_{c_i})$$

where $\arctan(x) \leq x$, $x \geq 0$ is used. Defining $d = q_v^i - \tau_{c_i}$, $\dot{d} = \dot{q}_v^i$ holds, leads to

$$\dot{d} \geq -\frac{2\eta}{\pi\alpha} d$$

Because $d(t_f - \varepsilon) > 0$ and $\beta = 2\eta/\pi\alpha > 0$, the Comparison Lemma (3.4 of [36]) allows one to prove that $d(t) > 0 \;\; \forall t \geq t_f - \varepsilon$, but this contradicts the hypothesis $d(t_f) = 0$. □

3.5. Extension to More than n Constraints

In the previous formulation the maximum number of constraints was equal to n in order to keep a square mapping matrix. However, it is possible to handle more constraints than the number of DOF as explained in [38] and [39]. When the number of constraints r is greater than the number of DOF, the matrix $T_c \in \mathbb{R}^{r\times n}$ in (30) becomes rectangular, and the component K_c is not needed. Because the number of real joints n is less than the number of virtual joints r, at most n virtual joints can be exactly tracked at the same time.

Indeed, if a generalized inverse of T_c is considered, namely $T_c^{\#}$ (such that $T_c^{\#}T_c = I$), the virtual joint velocity,

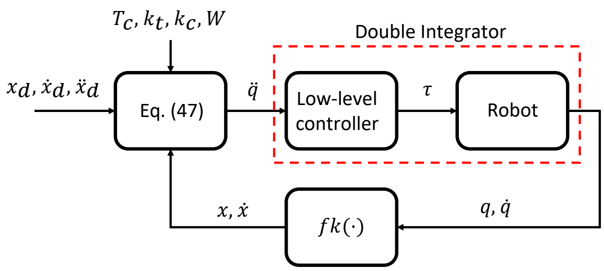

Fig. 1. Schematic of the redundancy resolution scheme.

Table 1. D-H Parameters.

Joint	d[m]	a[m]	α[rad]	offset[rad]
1	0.270	0.069	$-\pi/2$	0
2	0	0	$\pi/2$	$\pi/2$
3	0.364	0.069	$-\pi/2$	0
4	0	0	$\pi/2$	0
5	0.374	0.01	$-\pi/2$	0
6	0	0	$\pi/2$	0
7	0.23	0	0	0

Table 2. Initial Conditions.

Case	Robot	Joint [rad]						
		q_1	q_2	q_3	q_4	q_5	q_6	q_7
I	Single	−0.08	−1.00	−1.19	1.94	0.67	1.03	−0.50
II	A	0.03	−1.01	−1.12	1.91	0.62	1.03	−0.59
	B	0.01	−0.98	1.12	1.89	−0.62	1.02	0.59

Case	Robot	Position [m]			Orientation [rad]		
		x	y	z	ϕ	θ	ψ
I	Single	0.571	0.181	0.271	−2.86	−3.09	−3.13
II	A	0.587	0.257	0.268	π	$-\pi$	$-\pi$
	B	0.592	−0.243	0.265	π	$-\pi$	$-\pi$
	Rel	0	−0.5	0	0	0	0

corresponding to the real joint solution, does not match the desired virtual velocity. Namely, $\dot{q}_v = T_c\dot{q} = T_c T_c^{\#}\dot{q}_{v_{\text{des}}}$ implies that $\dot{q}_v \neq \dot{q}_{v_{\text{des}}}$ because $T_c T_c^{\#} \neq I$. Note that the real joint solution is determined by the inverse mapping from the virtual joint space, where the desired value of the virtual velocity $\dot{q}_{v_{\text{des}}}$, that satisfies the constraints, is computed from (40).

There are many possible ways to choose $T_c^{\#}$, for example the left Moore-Penrose pseudoinverse. Here, the weighted pseudoinverse of T_c has been used, which can be expressed as

$$T_c^{\#} = (T_c^T W_p T_c)^{-1} T_c^T W_p. \tag{45}$$

The weighting matrix $W_p \in \mathbb{R}^{r\times r}$ can be exploited to increase the priority of the critical virtual joints, namely those that are closer to the limits. This choice makes it possible to exactly track the desired virtual trajectory when the virtual joint is going toward the limit, resulting in the limit avoidance. In this work, the elements w_{p_i} of the diagonal matrix W_p have been defined as

$$w_{p_i} = \frac{(1-w_i)^2}{0.1\left(w_i^2 + 8w_i\right)} \tag{46}$$

where w_i is defined in (43).

The inverse mapping $T_c^{\#}$ is then used in the scheme (42) resulting in

$$\begin{aligned} \ddot{q} = {} & T_c^{\#}(W^{-1}J_v^T(J_v W^{-1}J_v^T)^{-1} \\ & (\ddot{x}_d + J_v k_c + J_v k_t - \dot{J}\dot{q} + K_D\dot{e} + K_P e) - k_c - k_t). \end{aligned} \tag{47}$$

A simple schematic of the whole resolution scheme is shown in Fig. 1. In summary, the resolution scheme, represented by eq. (47), is fed with the reference signals $x_d, \dot{x}_d, \ddot{x}_d$, the terms T_c, k_t, k_c, W computed from eqs. (30), (38), and (43) that depend on the constraints, and the feedback $x, \dot{x}$ ($fk(\cdot)$ represents the forward kinematic function). The control signal $\ddot{q}$ generated by the proposed redundancy resolution scheme is supplied to the subsystem low-level controller plus robot, that has been modeled in the simulations as a double integrator chain.

Remark 3: Eq. (47) represents the general closed loop solution when the number of constraints r is greater than the number of degrees of freedom n. Note that, by applying (47) to system (6), the resulting tracking error dynamics can still be written as $\ddot{e} = -K_D\dot{e} - K_P e$ (with the tracking error defined as $e = x_d - x$). Therefore, the position x will tend to the desired x_d asymptotically.

Remark 4: In case of ill-conditioning, generally due to unfeasible tasks or to the weighting factors, solutions (42) and (47) may grow unbounded. The damped least square has to be introduced for guaranteeing numerical stability of the solution [34, 35]. This technique has been also adopted here to carry out the simulations as reported in the next Section.

4. SIMULATION RESULTS

Simulation results of the proposed method are presented in this section. Two different case studies are presented: a single 7-DOF manipulator and a dual-arm system composed of two 7-DOF arms. Both the manipulators include only revolute joints and possess the same D-H parameters, which are shown in Table 1. The proposed solution is compared to the GPM [12] and the priority framework proposed in [23] in order to prove the effectiveness of the method. Simulations are conducted in the Matlab R2018b environment, with a fixed sample time of 0.001 s.

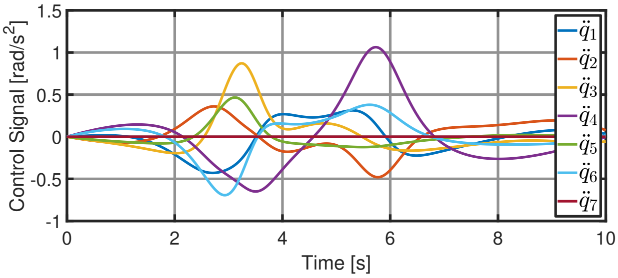

Fig. 2. Case I. Control signal of the proposed solution to constraints (48).

Table 3. Case I, constraint (48): Methods comparison.

	LN	Proposed	GPM	[23]†	[23]‡
Traj. Track. IAE [m]	.0017	.0022	.0026	**.0015**	.0016
Constraints	×	✓	✓	✓	✓

Case I

In this first case, only a single manipulator is employed. The initial joint configuration and the corresponding end-effector task space pose can be found in Table 2. The desired end-effector trajectory consists of a circle of 0.3 m radius in the X-Y plane, centered in $[0.571, -0.119, 0.271]$ m (the Z coordinate remains equal to its initial value), which has to be fulfilled in 10 s. A fifth-order polynomial function is used to plan the trajectory, where the initial and final velocity and acceleration are set to zero. Only the position components are controlled, thus the degree of redundancy is equal to 4. The other parameters are set as follows: the elements of the diagonal feedback gain matrices K_P and K_D to 15 and 7, respectively, the scalar $\alpha = 0.1$ in (44), while the damping factor $\lambda = 1 \times 10^{-6}$.

First, only the following joint position constraints are considered, i.e.,

$$q^i_{\min} \leq f_i(q) = q_i \leq q^i_{\max} \qquad i = 1, \dots, 7 \tag{48}$$

where the vectors $q^i_{\min}$ and $q^i_{\max}$ are equal to $[-2.50, -1.70, -3.05, -1.57, -1.57, -1.05, -3.05]$ rad and $[2.50, 1.70, 3.05, 2.61, 1.57, 3.05, 3.05]$ rad, respectively. Note that the constraint (48) has to be first tranformed as in inequality (34) to be implemented. If we consider the constraint $q_i > q^i_{min}$, then it is sufficient computing the time derivative $\dot{f}_i = \dot{q}_i = c_i$, while $\tau_i = f_i = q_i$, and $\tau_{c_i} = q^i_{\min}$.

Table 3 reports a comparison between the proposed approach, the unconstrained LN, the GPM [12], and the priority framework proposed in [23]. For each approach, the table resumes the trajectory tracking performance, evaluated by the Integral Absolute Error (IAE) of the tracking error, and if the constraint task is satisfied or not.

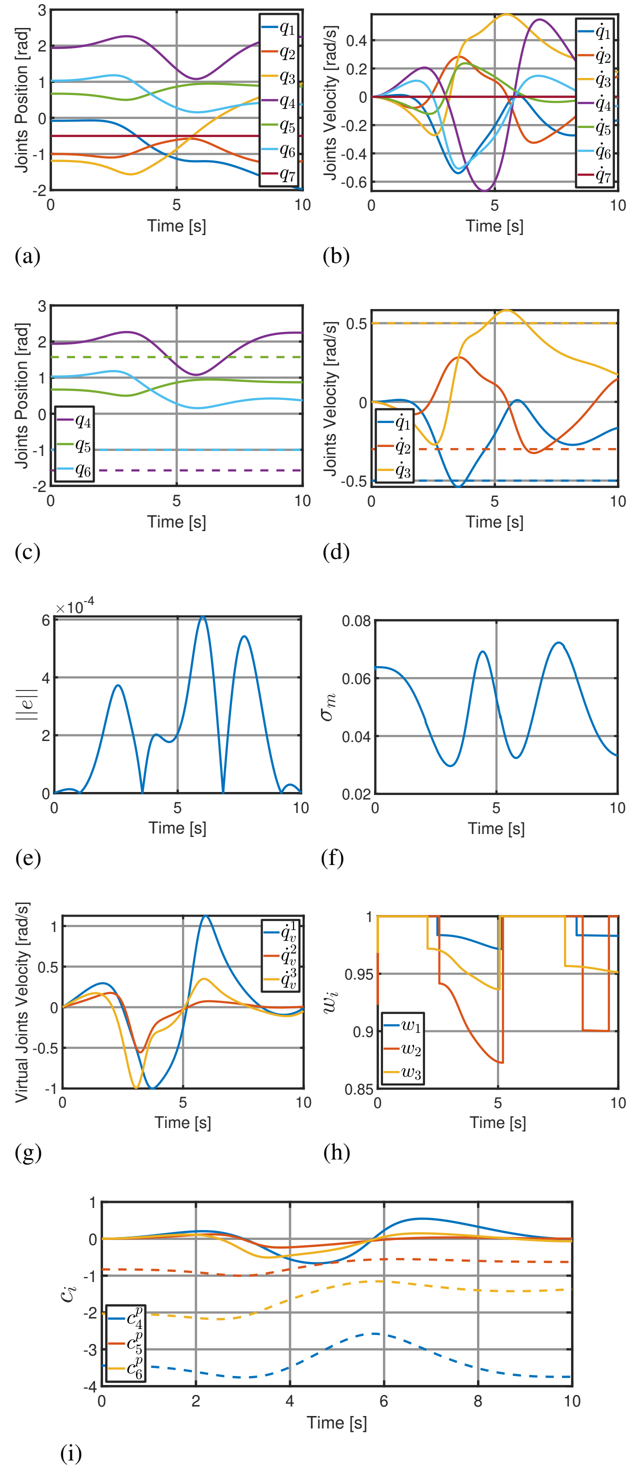

Fig. 3. Case I. Proposed solution to constraints (48). (a) Joint position. (b) Joint velocities. (c) Positions of the joints (solid lines) that previously violated the thresholds (dashed lines) in the LN solution. (d) Velocities of the joints (solid lines) that previously violated the thresholds (dashed lines) in the LN solution. (e) Euclidean norm of the tracking error. (f) Minimum singular value of $J_v W^{-1} J_v^T$. (g) Virtual joint velocities. (h) Weighting factors. (i) Graphical representation of inequality (34): constraint functions c_i (solid lines) and the corresponding thresholds $\tau_{c_i} - \tau_i(t)$ (dashed lines).

Table 4. Case I, constraint (48) and (49): Methods comparison.

	LN	Proposed	GPM	[23]†	[23]‡
Traj. Track. IAE [m]	.0017	.0019	.0022	**.0014**	7.46
Constraints	×	✓	×	×	✓

In detail, the LN pseudoinverse gives a solution that makes the joints q_4, q_5, and q_6 to exceed their limits. The GPM has been employed with a performance function as in (10), while the gain matrix is set as $K_{\mathrm{GPM}} = \mathrm{diag}\{10,\ldots,10\}$. GPM is able to track the desired trajectory and to avoid all the position limits for this specific task. The tracking error has an order of magnitude of 10^{-3}, because the GPM is a projection into the null space based approach. The priority framework proposed in [23] has been employed with both the trajectory tracking task (column [23]† in Tables 3, 4, and 5) and the constraint task (column [23]‡) as the most prioritizing task, respectively. The regularization term ρ^2 has been set equal to 0.01. The priority framework [23] is capable of satisfying both the trajectory tracking and the position limits (48) in this scenario, maintaining an IAE value of the tracking error with an order of magnitude of 10^{-3}.

Fig. 2 depicts the control signal computed by the proposed redundancy resolution scheme to solve the problem of guaranteeing constraints (48). The simulation results obtained by feeding the system with the control signal $\ddot{q}$ are summarized in Fig. 3. In details, it is possible to note in Fig. 3(a) that all the joints remain within their feasible ranges. In particular, recalling that position limits of joints q_4, q_5, and q_6 were not satisfied in the case of LN while now, with the proposed solution, are instead satisfied, as shown in Fig. 3(c). The constraint functions c_i^p, derived in Eq. (34) (the subscript i indicates the joint number, while the superscript p means that the constraint is on the joint positions), are satisfied, too; functions c_4^p, c_5^p, and c_6^p remain above their corresponding thresholds for the entire period (Fig. 3(i)), which means that inequality (48) is ensured. Figs. 3(g) and 3(h) depict the virtual joint velocities, namely the WLN solution in the virtual joint space (Eq. (40)), corresponding to constraints of joints q_4, q_5 and q_6, and the correlated weighting factors (Eq. (43)), respectively. It is worth noting that the weighting factors are discontinuous, the switching law (43) is a discontinuous function, but this does not imply discontinuity in the virtual joint velocities because the switch takes place when $\dot{q}_v^i$ crosses zero. In this way, when the virtual joint q_v^i is moving away from its limit, namely when $\dot{q}_v^i > 0$, the corresponding virtual motion is not suppressed. Because there are four degrees of redundancy, the additional subtasks can be carried out without affecting the performance of the main task. Indeed, the minimum singular value σ_m of the matrix $J_v W^{-1} J_v^T$ (Fig. 3(f)) is sufficiently greater than

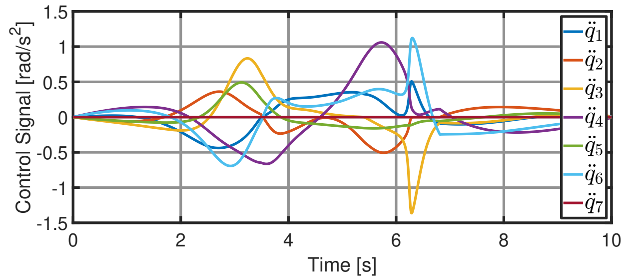

Fig. 4. Case I. Control signal of the proposed solution to constraints (48) and (49).

the damping factor, and this ensures a small tracking error (Fig. 3(e)).

Then, joint velocity constraints are introduced, combined with those in (48)

$$\dot{q}^i_{\min} \le g_i(\dot{q}) = \dot{q}_i \le \dot{q}^i_{\max} \qquad i = 1,\ldots,7 \qquad (49)$$

where $\dot{q}^i_{\min} = [-0.5, -0.3, -1, -1, -1, -1, -1]\,\mathrm{rad/s}$ and $\dot{q}^i_{\max} = [1, 1, 0.5, 1, 1, 1, 1]\,\mathrm{rad/s}$.

Remark 5: Adding the constraint set (49) to (48), the total number of constraints r becomes 14. Therefore, the inversion of the transformation matrix T_c is performed as described in Section 3.5, namely Eq. (45).

Contrary to the previous scenario, the GPM is not able to ensure all the limits for this specific task and constraint set, as reported in Table 4. In order to use the GPM to avoid both the joint position and the joint velocity limits, a possible solution is shaping the velocity constraints as explained in [22].

Then, the shaped constraints can be used in the same performance function of the previous test. The solution obtained from the GPM drives the robot to a position where the main task and the constraints become conflicting and the position/velocity limits can not be avoided. In detail, the joint velocities $\dot{q}_2$ and $\dot{q}_3$ exceed their lower and upper limits, respectively. The priority framework failed to satisfy both the trajectory tracking and the constraints, too. When the first priority level is assigned to the trajectory tracking (column [23]†), the path is correctly followed giving an IAE with an order of magnitude of 10^{-3}, but with some constraints violated. In column [23]‡ the priority levels have been swapped; joint positions and velocities can be kept within the limits but the trajectory can not be properly tracked. Last, it is worth noting that joint velocity limits were violated also in the previous simulation of the proposed approach ($\dot{q}^1_{\min}$ at $t = 3.2\,\mathrm{s}$, $\dot{q}^2_{\min}$ at $t = 6.3\,\mathrm{s}$, and $\dot{q}^3_{\max}$ at $t = 4.7\,\mathrm{s}$ in Fig. 3(d)), where the constraints comprised the position limits only.

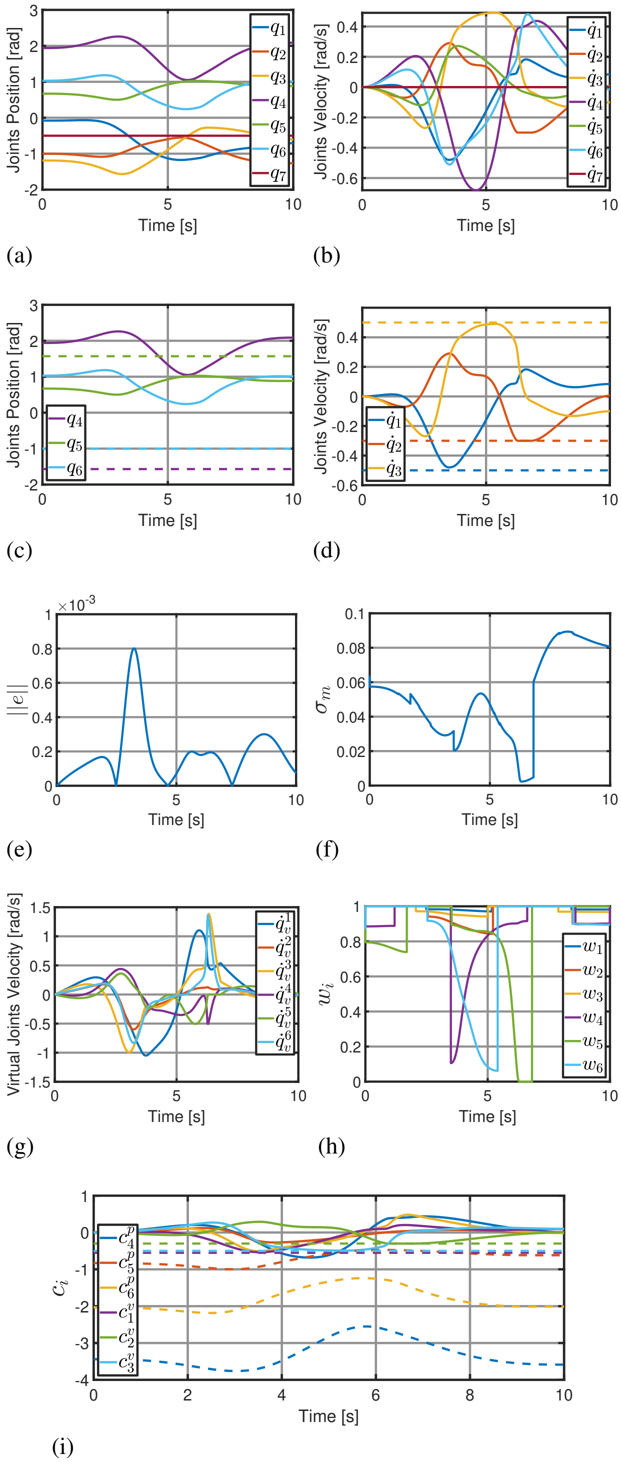

Fig. 5. Case I. Proposed solution to constraints (48) and (49). (a) Joint positions. (b) Joint velocities. (c) Positions of the joints (solid lines) that previously violated the thresholds (dashed lines) in the LN solution. (d) Velocities of the joints (solid lines) that previously violated the thresholds (dashed lines) both in the LN solution and in the first proposed solution (Fig. 3(d)). (e) Euclidean norm of the tracking error. (f) Minimum singular value of $J_v W^{-1} J_v^T$. (g) Virtual joint velocities. (h) Weighting factors. (i) Graphical representation of inequality (34): constraint functions c_i (solid lines) and the corresponding thresholds $\tau_{c_i} - \tau_i(t)$ (dashed lines).

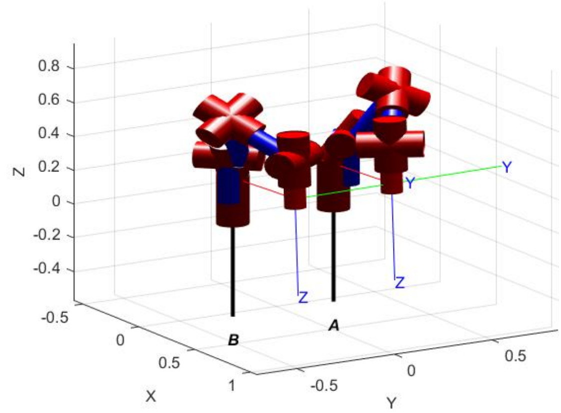

Fig. 6. Robot model.

The control signal generated for the new set of constraints is shown in Fig. 4, while Fig. 5 summarizes the simulation results. All the joint position limits are still satisfied (Fig. 5(a)) and, in particular, those on joints q_4, q_5, and q_6 (Fig. 5(c)). Also the constraint (34) corresponding to the position limits is still ensured because the functions c_4^p, c_5^p, and c_6^p do not violate their thresholds (Fig. 5(i)). In terms of the joint velocities, the limits on $\dot{q}_1$, $\dot{q}_2$, and $\dot{q}_3$ are now satisfied, as is shown in Fig. 5(d). In particular, the most critical joint is q_2, because $\dot{q}_2$ reaches the limit $\dot{q}^2_{\min}$ at $t = 6.3$ s. The constraint functions c_i^v, derived in Eq. (34), corresponding to the velocity constraints (49), are shown in Fig. 5(i) (the subscript i indicates the joint number, while the superscript v means that the constraint is on the joint velocities). As expected, the corresponding constraint function c_2^v comes to its threshold, but it does not exceed it (green line in Fig. 5(i)). Figs. 5(g) and 5(h) show the velocities of the virtual joints and the weighting factors, respectively. The weighting factor corresponding to c_2^v, namely w_5, tends to zero at $t = 6.3$ s in order to stop the motion of the virtual joint q_v^5. During the interval 6.2 s to 6.8 s, q_v^5 is stopped because its velocity $\dot{q}_v^5$ is equal to zero. This makes it possible for the constraint function c_2^v to remain in the feasible range. As soon as the desired trajectory makes q_v^5 start to move away from the limit ($t = 6.8$ s), w_5 is set back to one in order to let the virtual joint freely moving away from the threshold. In terms of the tracking error, the minimum singular value (Fig. 5(f)) decreases to about 0.002 because of the small value assumed by w_5. This value is lower than the case with only joint position limits, but still greater than the damping factor. Thus, the trajectory tracking can still be performed without performance degradation (Fig. 5(e)).

Last, it is worth noting how the priority of the main task and of the subtasks is dynamically adjusted by the rank of the matrix $J_v W^{-1} J_v^T$. In other words, the priority of the subtasks is always greater than or equal to the priority of the main task, thanks to the structure of the redundancy

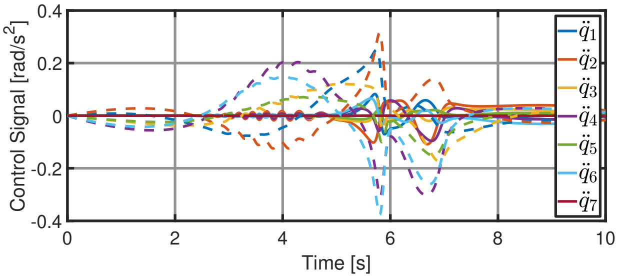

Fig. 7. Case II. Control signal of manipulator A (solid lines) and B (dashed lines).

Table 5. Case II: Methods comparison.

	LN	Proposed	[23]†	[23]‡
Traj. Track. IAE [m]	.048	**.0068**	.397	.389
Constraints	×	✓	×	✓

resolution scheme. This means that, when $J_v W^{-1} J_v^T$ is not rank deficient (as in these two tests because σ_m remains greater than the damping factor), the main task and all the subtasks can be performed without performance degradation. In this case, the main task possesses the same priority as the subtasks. If $J_v W^{-1} J_v^T$ becomes rank deficient, it means that a feasible solution that satisfies both the main task and the constraints at the same time can not be found. Thus, the performance of the main task will be temporarily degraded in order to satisfy the constraints.

Case II

This test is carried out considering the whole dual-arm system. The kinematic controller is built based on the Relative Jacobian approach [21]. Because the Relative Jacobian considers the dual-arm system as a single redundant arm, the proposed approach can be applied in the same way as in the single manipulator case. Table 2 shows the initial joint configuration and the end-effector Cartesian pose for both the manipulators. Note that the relative pose in Table 2 is the position of the end-effector B with respect to A, namely the positions of the two end-effector reference frames shown in Fig. 6. Moreover, manipulators A and B bases are located at $[0.06, 0.26, 0.13]$ m and $[0.06, -0.26, 0.13]$ m, respectively. Similar to Case I, the main task consists of performing a circular path, centered in $[0, -0.6, 0]$ m and with a radius equal to 0.1 m, in the relative X-Y plane in 10 s. Absolute trajectories have not been imposed to the individual arms. The same parameters of Case I have been used. Different from the previous test, a task space constraint is considered in this case. In particular, the absolute motion of the tooltip of the manipulator A must be enclosed in a sphere of radius equal to $r = 0.05$ m and centered at the end-effector initial position.

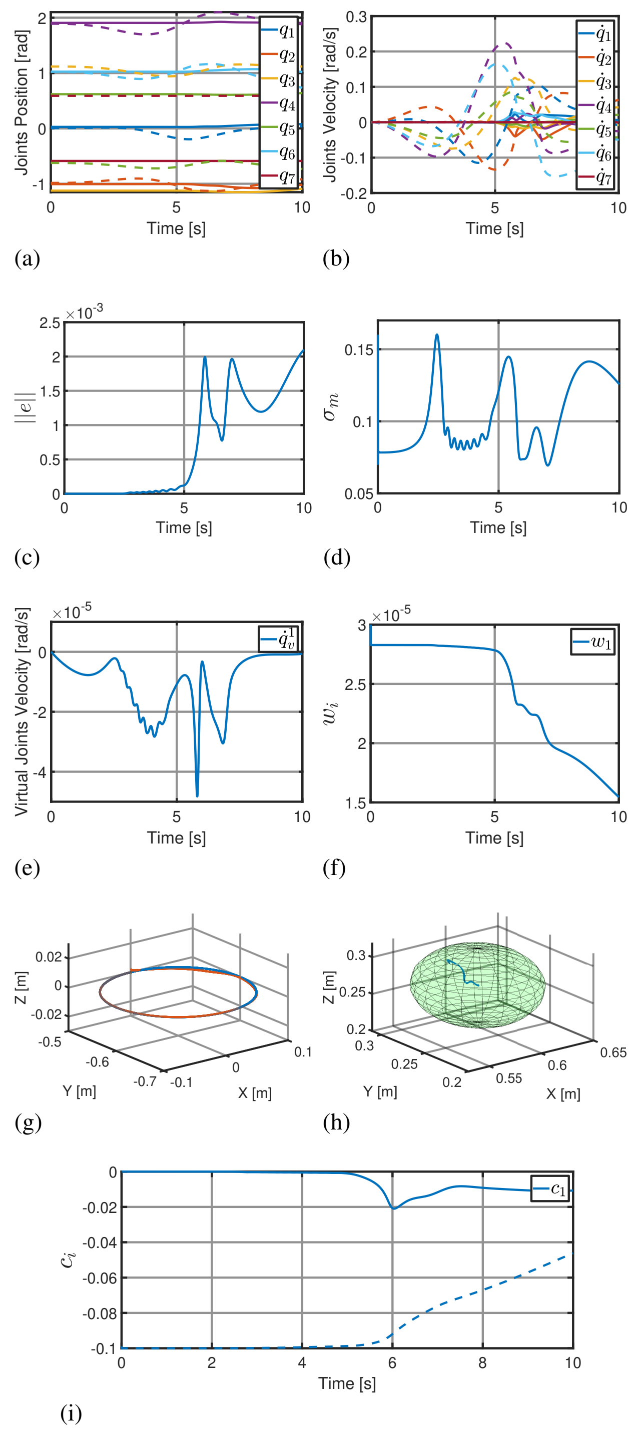

Fig. 8. Case II. Proposed solution. (a) Joint positions of manipulator A (solid lines) and B (dashed lines). (b) Joint velocities of manipulator A (solid lines) and B (dashed lines). (c) Euclidean norm of the tracking error. (d) Minimum singular value of $J_v W^{-1} J_v^T$. (e) Virtual joint velocities. (f) Weighting factors. (g) Cartesian relative trajectory: reference (blue), real (red). (h) Cartesian trajectory of manipulator A: real trajectory (blue line), constraint (green sphere). (i) Graphical representation of inequality (34): constraint function c_i (solid line) and the corresponding threshold $\tau_{c_i} - \tau_i(t)$ (dashed line).

The constraint is written as

$$f_A(q) = d(q) \leq r \tag{50}$$

where $d(q) = \|p(q) - c_S\|$ is the Euclidean distance of the end-effector from the center of the sphere ($p(q)$ is the Cartesian pose of the end-effector while c_S is the center coordinates vector).

Results for this scenario are summarized in Table 5. The LN solution is not capable of keeping the movement of the end-effector A inside of the sphere. The priority framework gives a solution that has an IAE with an order of magnitude of 10^{-1} in both configurations. However, the constraint has been satisfied in configuration [23]‡, while it has not in [23]†.

In contrast, the control signal (Fig. 7) obtained by the proposed scheme makes the constraints satisfied for all time, as Figs. 8(h) and 8(i) illustrate. Indeed, the function c_1 is never less than the threshold and the motion of the end-effector A is always enclosed in the sphere. This results in more regular movements of the joints (Fig. 8(a)) because the manipulator A is almost stationary, while the joints of manipulator B perform sinusoidal-like movements. The more uniform movement in the Cartesian space results in smaller tracking error values (Fig. 8(c)).

5. CONCLUSIONS AND FUTURE WORKS

This work is focused on developing a kinematic controller for redundant manipulators that is able to cope with kinematic constraints. The constraints can be generic linear or nonlinear functions and are modeled as unilateral inequalities. Position and velocity based constraints are first treated separately. Then, a general procedure to also manage mixed constraints, considering both the time independent and the time dependent threshold case, is presented. Simulations are performed in a Matlab-Simulink environment on a 7-DOF arm. In particular, two tests are carried out: the first considers the problem of physical joint limit avoidance for a single 7-DOF manipulator. In the second, a dual-arm system composed of two 7-DOF arms is employed. In this latter case, the main task consists of tracking a relative trajectory by using the Relative Jacobian method, while the constraint is on the absolute Cartesian motion of one of the manipulators. Results show that the proposed method is effective enforcing generic kinematic constraints.

REFERENCES

[1] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286–292, 1995.

[2] F. Chaumette and É. Marchand, "A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 719–730, 2001.

[3] Yunong Zhang, Jun Wang, and Youshen Xia, "A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 658–667, 2003.

[4] Y. Yavin, "Control of a three-link manipulator with a constraint on the velocity of its end-effector," *Computers & Mathematics with Applications*, vol. 40, no. 10, pp. 1263–1273, 2000.

[5] J. Hollerbach and K. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 308–316, 1987.

[6] Yunong Zhang, S. S. Ge, and Tong Heng Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2126–2132, 2004.

[7] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.

[8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. Springer, 1986, pp. 396–404.

[9] K. N. Groom, A. A. Maciejewski, and V. Balakrishnan, "Real-time failure-tolerant control of kinematically redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 6, pp. 1109–1115, 1999.

[10] J. D. English and A. A. Maciejewski, "Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 566–575, 1998.

[11] C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 4, pp. 622–629, 1997.

[12] A. Liegeois, "Automatic supervisory control of the configuration and behaviour of multibody mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[13] H. Zghal, R. Dubey, and J. Euler, "Efficient gradient projection optimization for manipulators with multiple degrees of redundancy," in *IEEE International Conference on Robotics and Automation*, 1990, pp. 1006–1011.

[14] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy," in *Robotics research: the first international symposium*, 1984, pp. 735–747.

[15] N. Mansard and F. Chaumette, "Directional redundancy for robot control," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1179–1192, 2009.

[16] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[17] M. Konstantinov, M. Markov, and D. Nenchev, "Kinematic control of redundant manipulators," in *11th International Symposium on Industrial Robots*, 1981, pp. 561–568.

[18] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

[19] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.

[20] N. Mansard, A. Remazeilles, and F. Chaumette, "Continuity of varying-feature-set control laws," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2493–2505, 2009.

[21] G. Foresi, A. Freddi, V. Kyrki, A. Monteriù, R. Muthusamy, D. Ortenzi, and D. Pagnotta, "An avoidance control strategy for joint-position limits of dual-arm robots," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1056–1061, 2017.

[22] F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 637–654, 2015.

[23] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[24] Y. Zhang, S. Li, J. Gui, and X. Luo, "Velocity-level control with compliance to acceleration-level constraints: A novel scheme for manipulator redundancy resolution," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 921–930, 2017.

[25] W. Wiedmeyer, P. Altoé, J. Auberle, C. Ledermann, and T. Kröger, "A real-time-capable closed-form multi-objective redundancy resolution scheme for seven-dof serial manipulators," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 431–438, 2020.

[26] M. Faroni, M. Beschi, L. M. Tosatti, and A. Visioli, "A predictive approach to redundancy resolution for robot manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8975–8980, 2017.

[27] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 1985, pp. 722–728.

[28] O. Egeland, "Task-space tracking with redundant manipulators," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 471–475, 1987.

[29] A. Csiszar, J. Eilers, and A. Verl, "On solving the inverse kinematics problem using neural networks," in *24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 2017, pp. 1–6.

[30] S. Li, M. Zhou, and X. Luo, "Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4791–4801, 2017.

[31] Z. Xu, S. Li, X. Zhou, S. Zhou, T. Cheng, and Y. Guan, "Dynamic neural networks for motion-force control of redundant manipulators: an optimization perspective," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 1525–1536, 2020.

[32] J. Xiang, C. Zhong, and W. Wei, "General-weighted least-norm control for redundant manipulators," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 660–669, 2010.

[33] J. C. A. Barata and M. S. Hussein, "The moore–penrose pseudoinverse: A tutorial review of the theory," *Brazilian Journal of Physics*, vol. 42, no. 1-2, pp. 146–165, 2012.

[34] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

[35] S. Chiaverini, O. Egeland, and R. Kanestrom, "Achieving user-defined accuracy with damped least-squares inverse kinematics," in *5th International Conference on Advanced Robotics*. IEEE, 1991, pp. 672–677.

[36] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[37] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, ser. Advanced Textbooks in Control and Signal Processing. Springer London, 2010.

[38] S. Huang, J. Xiang, W. Wei, and M. Z. Chen, "On the virtual joints for kinematic control of redundant manipulators with multiple constraints," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 65–76, 2017.

[39] D. Proietti Pagnotta, A. Freddi, S. Longhi, and A. Monteriù, "Dual-arm manipulators kinematic control under multiple constraints via virtual joints approach," in *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2019, pp. 1–6.

Daniele Proietti Pagnotta received his Ph.D. in computer and automation engineering from Università Politecnica delle Marche in 2019, where he is currently a PostDoc Researcher. His research interests are in robotics, system and control theory and fault diagnosis, and the main application field consists in redundant and/or cooperative systems.

Andrea Monteriù is Associate Professor at Università Politecnica delle Marche. His main research interests include fault diagnosis, fault tolerant control, nonlinear, dynamics and control, periodic and stochastic system control, applied in different fields including aerospace, marine and robotic systems.

Alessandro Freddi is Assistant Professor at Università Politecnica delle Marche and is founding member of "Syncode", a startup operating in the field of industrial automation. His main research activities cover fault diagnosis and fault-tolerant control with applications to robotics, and development and application of assistive technologies.

Sauro Longhi is Full Professor of Automation Engineering Università Politecnica delle Marche, where he acted as Rector from 2013 to 2019. Since May 2014 he is President of the GARR Consortium, the Italian national computer network for universities and research. Since September 2019 he is also President of SIDRA, the Italian Society of Researchers and Professors of Automation.

Anthony Maciejewski is Full Professor and Head of the Department of Electrical and Computer Engineering (ECE) at Colorado State University. In 2018-2019 he served as President of the ECE Department Head's Association. His research has been supported by NSF, Sandia Nat'l Lab, Oak Ridge Nat'l Lab, DARPA, NASA, Nat'l Imagery and Mapping Agency, Missile Defense Agency, Non-lethal Technology Innovation Center, the NEC Corporation, Caterpillar, AT&T, H-P, Intel, Chrysler, Wolf Robotics, and the TRW Foundation.