

Inverse kinematics of redundant manipulators with dynamic bounds on joint movements

Marco Faroni¹, Manuel Beschi², Nicola Pedrocchi¹

¹Istituto di Sistemi e Tecnologie Industriali Intelligenti per il Manifatturiero Avanzato
Consiglio Nazionale delle Ricerche – Milano, Italy

²Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia – Brescia, Italy
Email: marco.faroni@stiima.cnr.it, manuel.beschi@unibs.it, nicola.pedrocchi@stiima.cnr.it

Abstract—Redundant manipulators are usually required to perform tasks in the operational space, but collision-free path planning is computed in the configuration space. Limiting the deviation with respect to the collision-free configuration-space trajectory may allow the robot to avoid collisions without modifying the primary task. This paper proposes a method to guarantee that the solution of the inverse kinematic problem deviates from the nominal joint-space trajectory less than a desired threshold. The excursion limitation is ensured by means of linear constraints and the automatic regulation of the weights of secondary tasks. Numerical and experimental results prove the validness of the proposed approach.

I. INTRODUCTION

Kinematically redundant manipulators have high versatility and dexterity thanks to the additional degrees of freedom they have with respect to the task they are required to perform. Extra degrees of freedom can be exploited to endow the robot with desired secondary behaviors. To name a few examples, the redundancy can be exploited to: maximize the manipulability along selected axis [1], [2], keep the joint configuration far from the mechanical limits [3], mimic anthropomorphic motion [4], activate the least number of actuators [5], avoid collisions [6], [7], or maximize the distance from obstacles [8].

The problem of exploiting the redundancy of the robot is typically addressed when solving the Inverse Kinematic (IK) problem: a primary task is given in the operational space (usually, the Cartesian space) and, for all points, the joint coordinates are computed so that the primary task is met and the secondary behavior is performed at best [9]. The most widespread approach consists in solving the IK problem at differential level to exploit the linear mapping between joint and task velocities given by the task Jacobian [10]. Joint configurations are then obtained via forward integration and numerical drifts are recovered by means of a closed-loop IK algorithm [10]. Up to few years ago, this was done by pseudo-inverting the Jacobian to find the minimum-norm solution to the problem and by projecting the velocity related to the secondary objective into the null space of the Jacobian [11]. Recently, Quadratic Programming (QP) approaches are preferred because they allow for considering robot limitations explicitly (for example, joint configuration, velocity, and acceleration limits can be written as linear constraints in the QP problem) [12], [13], [14]. In both cases, the result is that the robot modifies its posture according to the secondary

objective, as shown for example in Figure 1. On the one hand, this approach has the great advantage that the IK problem can be solved at runtime, by taking into account the current state of the system. For example, the robot can adapt its posture to the position of the operator without modifying the primary task [8]. On the other hand, the main drawback is the lack of a non-collision guarantee. As a matter of fact, the new robot posture is usually very different from the initial one and it might be in conflict with obstacles (see Figure 1). Most works on redundancy handling take into account the collision avoidance problem by maximizing the distance from the closest obstacle (assumed to be a 3D point or a simple 3D shape) [6], [15]. This may work in simplified laboratory examples, but it is hardly scalable to real-case scenarios. For example, industrial robots are often required to work in very cluttered environments or to approach objects with complex surfaces. In these cases, re-configuration based on the closest-obstacle criterion is prone to failure.

Most times, collision-free path planning is indeed performed in the configuration space. The most widespread methods are sampling-based (e.g., RRT and its variants [16], [17], and probabilistic roadmaps [18]) and local algorithms that search in the neighborhood of an initial solution (e.g., CHOMP [19] and STOMP [20]). Notice that this planning approach does not explicitly exploit redundancy. As a matter of fact, the presence of redundant degrees of freedom just results in a larger search space, but secondary objectives can not be specified. Moreover, the robot motion can not be modified at high sampling rates during the execution, which is the main advantage of online IK schemes.

Based on these considerations, [21] suggested to set a collision-free joint-space trajectory computed by an online path planner as low-priority task of the IK problem. In this way, the robot should follow the joint-space reference when it is not in conflict with other tasks. In that case, however, the collision-free trajectory results in a soft constraint that is soon violated to meet higher-priority tasks (as also shown in Section IV-A of this paper). Notice that imposing hard constraints on the joint deviation with respect to the reference trajectory would result in a possible failure of the QP problem. This is a well-known issue in online IK schemes, which require additional *viability* constraints to ensure the feasibility of the QP problem when constant position limits are considered [22], [23]. To the best of

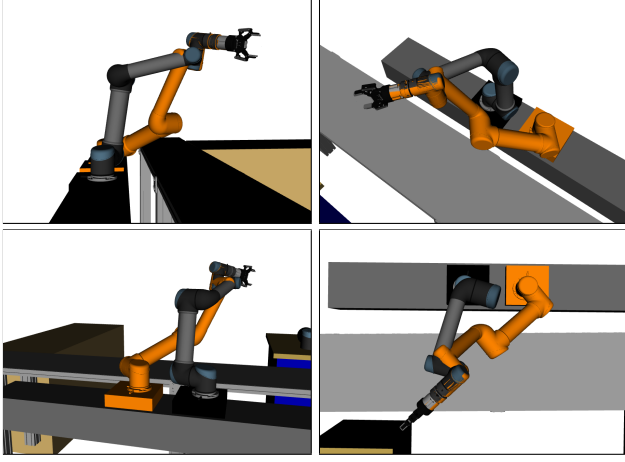


Fig. 1: Effect of usual IK schemes. Orange robot: initial posture; dark robot: final posture that optimizes secondary tasks. Robot posture may vary significantly with respect to the initial one and collision check is no longer guaranteed.

the authors' knowledge, however, no prior works address viability of time-varying position limits.

This paper proposes a method to ensure that the IK solution, computed online, has a bounded deviation with respect to a given time-varying signal. By using a configuration-space collision-free trajectory as reference signal, it is possible to avoid collisions with the surrounding environment. To do so, the paper devises a viability constraint valid for time-varying position bounds by extending the viability conditions of constant bounds [22], [23], and by using a task-scaling IK approach [24], [25]. Moreover, it devises a closed-loop algorithm that automatically adapts the weights of the secondary tasks, in order to regulate the deviation of the joint position to a given value. Compared to soft constraint-based approaches (e.g., [3], [8]), the joint movement is guaranteed to be limited to the maximum range also for time-varying bounds. The paper also analytically derives a linear approximation of the new viability constraints, taking out the need of time-consuming numerical approaches (e.g., [23] derives viable polygons for constant bounds by means of nonlinear programming). Numerical and experimental results on a 7-degree-of-freedom manipulator show that the proposed approach is able to hardly limit the excursion with respect to the nominal joint trajectory where soft-constraint approaches fail.

The paper is organized as follows. Section II deals with the formulation of IK problems as QP problems and the related feasibility issues. Proposed method is described in Section III; simulation and experimental results are in Section IV. Conclusions are drawn in Section V. Appendix A derives the polygonal viable set in closed form.

II. PRELIMINARIES

A. QP-based inverse kinematic problem

Recently, the IK problem of robot manipulators has been addressed by means of quadratic programming to cope with robot limits and/or other limitations that can be modeled as

optimization constraints. As an example, consider a manipulator to which a primary task $A_1\dot{q} + b_1 = 0$ and a secondary task $A_2\dot{q} + b_2 = 0$ are assigned. The primary task is usually the desired Cartesian trajectory; that is, A_1 is the end-effector Jacobian and $-b_1$ is the desired twist at each time step. Closed-loop IK schemes modifies b_1 to recover from position errors [10]. Secondary tasks are soft constraints the robot should fulfill at best (e.g., manipulability maximization [2] or maximization of the human-robot distance [6]).

At each sampling time $t = kT$ (where T is the sampling period of the system), the joint velocity is chosen by solving the following hierarchical QP problem:

$$\begin{aligned} \dot{q} = \underset{u}{\operatorname{argmin}} \quad & \|A_2u + b_2\|^2 \\ \text{subject to} \quad & A_1u = A_1u_1 \\ & u_1 = \underset{u'}{\operatorname{argmin}} \|A_1u' + b_1\|^2 \\ & u \in \mathcal{S} \end{aligned} \quad (1)$$

where \mathcal{S} is a set of linear inequalities that account for the robot limits (details on the derivation of \mathcal{S} are given in Section II-C). Problem (1) means that the secondary task (i.e., the cost function) is optimized by searching in the null space of the primary task (i.e., first and second constraints in (1)). This approach has been extended to any number of prioritized tasks and to inequality tasks in [13].

Hierarchical QP problems in the form of (1) can be solved by means of a nested approach, that is: i) solve the inner QP problem and find $u_1 = \underset{u'}{\operatorname{argmin}} \|A_1u' + b_1\|^2$; ii) use u_1 to solve the outer QP problem (i.e., $\dot{q} = \underset{u}{\operatorname{argmin}} \|A_2u + b_2\|^2$ subject to $A_1u = A_1u_1, u \in \mathcal{S}$). Although useful to have an insight of QP-based IK, this approach is computationally inefficient; dedicated solvers that exploit the geometric properties of hierarchical QPs are preferred [14], [26], [27].

Methods based on (1) are referred to as *local*, as they only take into account tasks and constraints at the current time instant. Look-ahead and model predictive control approaches have been also proposed to overcome the drawbacks of local methods [28], [29], [30].

B. Task scaling inverse kinematics

Recent approaches exploit the path-velocity decomposition of the trajectory to preserve the geometry of the path by acting on the timing law when the nominal trajectory would exceed the robot limits [24], [31]. Problem (1) is modified by adding a scaling variable $s \in [0, 1]$ that scales the primary task velocity b_1 so that the geometric direction is not changed. The resulting hierarchical QP is:

$$\begin{aligned} \begin{pmatrix} \dot{q} \\ s \end{pmatrix} = \underset{u, s}{\operatorname{argmin}} \quad & \|A_2u + b_2\|^2 + \mu(1 - s)^2 \\ \text{subject to} \quad & A_1u + b_1s = A_1u_1 + b_1s_1 \\ & \begin{pmatrix} u_1 \\ s_1 \end{pmatrix} = \underset{u', s'}{\operatorname{argmin}} \|A_1u' + b_1s'\|^2 \\ & u \in \mathcal{S} \\ & 0 \leq s \leq 1 \end{aligned} \quad (2)$$

where $\mu \gg 1$. Problem (2) is similar to (1), but when the primary task is not feasible with respect to the robot limits, s results to be smaller than one and the trajectory automatically slows down to preserve the desired geometrical path.

A receding horizon approach to this problem was also proposed in [25] to overcome the drawbacks of local methods.

C. Viability and feasibility of the inverse kinematic problem

Referring to (1) and (2), \mathcal{S} is the set of admissible velocities that satisfies the robot limits. Usually, IK schemes take into account joint configuration, velocity, and acceleration limits such that:

$$\begin{aligned} q_{\min} &\leq q \leq q_{\max} \\ -\dot{q}_{\max} &\leq \dot{q} \leq \dot{q}_{\max} \\ -\ddot{q}_{\max} &\leq \ddot{q} \leq \ddot{q}_{\max} \end{aligned} \quad (3)$$

where q_{\min} and q_{\max} are the minimum and maximum joint configuration limits, \dot{q}_{\max} is the maximum joint velocity vector, and \ddot{q}_{\max} is the maximum joint acceleration vector. To ensure the feasibility of the QP problem for all possible states in \mathcal{S} , an additional viability constraint has to be added as follows [22]:

$$q + \frac{\dot{q}^2}{2\ddot{q}_{\max}} - q_{\max} \leq 0 \quad (4)$$

$$q + \frac{\dot{q}^2}{2\ddot{q}_{\max}} - q_{\min} \geq 0 \quad (5)$$

Note: products, quotients, and inequalities between vectors are intended as component-wise.

Equations (4)(5) determine the maximal viable set for a double integral system with bounded configuration, velocity, and acceleration. The equation is quadratic in \dot{q} . A viable set given by a set of linear constraint is helpful to include such conditions in the online control problem in case of optimization-based algorithms (e.g., model predictive control [25], [32]). In [25], a viable set composed of one linear constraint was derived. The result is that (4)(5) can be substituted by the following linear constraints:

$$\ddot{q}_{\max} q + \dot{q}_{\max} \dot{q} - \ddot{q}_{\max} q_{\max} \leq 0 \quad (6)$$

$$\ddot{q}_{\max} q + \dot{q}_{\max} \dot{q} - \ddot{q}_{\max} q_{\min} \geq 0 \quad (7)$$

The problem is analyzed more in details in [23], where a numerical method to obtain the maximal polygon given the number of sides is proposed.

Equations (4)–(7) are valid if position constraints are static; thus, if q_{\min} or q_{\max} varies in time, they are no longer sufficient. The case of time-varying position limits is addressed by our method in the next section.

III. PROPOSED METHOD

Consider a joint-space nominal trajectory given by a curve $q_{\text{nom}}(s)$ and a timing law $s(t)$. At sampling time $t = kT$, we aim to ensure that:

$$q_{\text{nom}}(t) - \Delta q_{\max} \leq q(t) \leq q_{\text{nom}}(t) + \Delta q_{\max} \quad \forall t \geq kT \quad (8)$$

where Δq_{\max} is the vector of maximum allowed joint displacement with respect to the nominal trajectory.

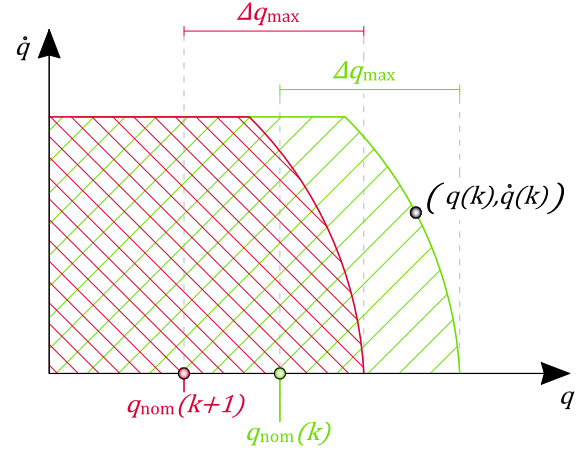


Fig. 2: Viability regions for time-varying joint bounds (only the first quarter is shown). Green: viability set at time k . Purple: viability set at time $k+1$. At each time instant, the viability region is determined by an interval of amplitude $2\Delta q_{\max}$ centered in the current nominal joint trajectory q_{nom} .

We address this issue by adding a time-varying constraint to the optimization problems (1) and (2) and by adjusting the secondary task cost function dynamically.

A. Constraint-based limitation of the joint excursion

Equation (8) can be considered as a time-varying joint-configuration constraint to be added to (1) and (2). Its addition requires a further viability analysis of the admissible set. Similarly to (4), the following viability constraint may be added to the QP problem (for the sake of brevity only the right side of (8) is considered):

$$q(k+1) + \frac{\dot{q}(k+1)^2}{2\ddot{q}_{\max}} - q_{\text{lim}} \leq 0 \quad (9)$$

where $q_{\text{lim}} = q_{\text{nom}}(k+1) + \Delta q_{\max}$ (hereafter, dependency on $k+1$ will be omitted unless necessary).

Feasibility of the new problem depends on the value of $q_{\text{nom}}(k+1)$. To clarify this, consider the example in Figure 2, in which joint state (q, \dot{q}) is on the boundary of (4). If $q_{\text{nom}}(k+1) < q_{\text{nom}}(k)$, then the viable set shifts to the left and the current state does not belong to the viable set anymore. Nonetheless, feasibility can be strongly guaranteed when a task-scaling IK scheme such as (2) is implemented. In this case, the desired configuration at time $k+1$ depends on the scaling factor s , that is:

$$q_{\text{nom}}(k+1) = q_{\text{nom}}(k) + \Delta q_{\text{nom}}(k) s \quad (10)$$

from which:

$$q_{\text{lim}} = q_{\text{nom}}(k) + s \Delta q_{\text{nom}}(k) + \Delta q_{\max} \quad (11)$$

The decision variable s therefore can slow down the original trajectory to ensure the feasibility of the problem. Notice that in the worst case described in Figure 2, the solution $s = 0$ does not change the viable set from cycle k to cycle $k+1$ (red and green areas coincide), keeping the state within the viable set.

1) *Linear approximation of the viable set:* Constraint (9) is nonlinear. Nonetheless, the approximation of the viable set recalled in Section II-C still holds. For example, the single-segment constraint (6) becomes:

$$\ddot{q}_{\max} q + \dot{q}_{\max} \dot{q} - \ddot{q}_{\max} q_{\lim} \leq 0 \quad (12)$$

which is linear in q , \dot{q} , and s .

A less conservative viable set can be obtained by using more linear constraints. Given the number of constraints N , the points P_j , $j = 0, \dots, N$, that delineate a viable polygonal set for each joint can be obtained as follows:

$$P_j = \left(\frac{\dot{q}_{\max}}{N} j, q_{\max} - \frac{\dot{q}_{\max}^2}{2N^2 \ddot{q}_{\max}} j(j+1) \right) \quad (13)$$

The derivation of (13) is given in Appendix A.

B. Self-regulating weighting of secondary tasks

The mere use of (9) (or its linear approximations) ensures the viability of the admissible set of QP problem (2). However, the IK problem is unaware of the nominal joint-space trajectory and therefore computes the joint variables by only considering the secondary task(s). As a result, the joint configurations are likely to reach the maximum excursion Δq_{\max} and the task-scaling mechanism would slow down the robot to respect the excursion limit (even up to the stop of the task, as in the previous example). To avoid this situation, the IK problem should be aware of the desired joint-space trajectory. In [8], the joint-space nominal trajectory was considered as the lowest-priority task in the QP problem. Here, we prefer to adopt a different approach where the priority of the joint-space nominal trajectory changes dynamically. Changing priorities between tasks on the fly is not trivial because an instantaneous swap of the priority of two tasks generates discontinuities in the joint commands and abrupt motions of the robot [33]. This issue can be addressed by gradually swapping the tasks in a weighted fashion during a transition phase [34], [35]. This is useful especially when the priorities of the stack of tasks are strict (e.g., humanoid robots). In industrial robotics, strict priority is usually only between the primary (*i.e.*, the Cartesian trajectory) and the successive tasks, whereas secondary tasks are usually objective behaviors such as dexterity, human-robot distance maximization, etc. For this reason, we use a mixed approach where the secondary task $A_2 \dot{q} + b_2 = 0$ and the joint-space nominal trajectory are handled in a weighted fashion. The hierarchical QP problem (2) therefore becomes:

$$\begin{aligned} \begin{pmatrix} \dot{q} \\ s \end{pmatrix} &= \underset{u,s}{\operatorname{argmin}} \quad \|q_{\text{nom}} - q\|^2 + \lambda \|A_2 u + b_2\|^2 + \mu(1-s)^2 \\ &\text{subject to } A_1 u + b_1 s = A_1 u_1 + b_1 s_1 \\ \begin{pmatrix} u_1 \\ s_1 \end{pmatrix} &= \underset{u',s'}{\operatorname{argmin}} \|A_1 u' + b_1 s'\|^2 \\ u &\in \mathcal{S} \\ 0 &\leq s \leq 1 \end{aligned} \quad (14)$$

where λ is a non-negative variable that can be adjusted to regulate the importance of the joint-space trajectory with

respect to the secondary task. The automatic choice of λ can be addressed as a regulation problem such that:

$$\max_i \left| \frac{\Delta q_i}{\Delta q_{\text{ref},i}} \right| \rightarrow 1 \quad (15)$$

where $\Delta q = q_{\text{nom}}(k) - q(k)$ and Δq_{ref} is the vector of reference values for each joint excursion such that $|\Delta q_{\text{ref},i}| \leq \Delta q_{\max,i}$. Variable λ can be therefore considered as a control input to drive Δq to a reference value Δq_{ref} based on the current value of Δq . In this work, we used a discrete-time proportional-integral-derivative control law defined as:

$$\lambda_{\text{PID}}(k) = K_P y(k) + K_I \sum_{j=0}^k y(k-j) + K_D (y(k) - y(k-1)) \quad (16)$$

where $y = 1 - \max_i \left| \frac{\Delta q_i}{\Delta q_{\text{ref},i}} \right|$ and K_P , K_I , K_D are the proportional, integral, and derivative gains respectively. The output is saturated between 0 and λ_{\max} in such a way that:

$$\lambda(k) = \begin{cases} 0 & \text{if } \lambda_{\text{PID}}(k) < 0 \\ \lambda_{\max} & \text{if } \lambda_{\text{PID}}(k) > \lambda_{\max} \\ \lambda_{\text{PID}}(k) & \text{otherwise} \end{cases} \quad (17)$$

A conditioned integration anti-windup strategy have been implemented to avoid windup issues due to saturation.

IV. RESULTS

We performed simulation and experimental tests on a 7-degree-of-freedom robot composed of a Universal Robots UR5 manipulator mounted on an actuated linear track. We consider conservative limits with respect to datasheet values because of the presence of cabling required by the end-effector, namely:

$$\begin{aligned} q_{\min} &= (q_{\min}^{\text{track}}, q_{\min}^{\text{robot}}) \\ q_{\max} &= (q_{\max}^{\text{track}}, q_{\max}^{\text{robot}}) \\ \dot{q}_{\max} &= (\dot{q}_{\max}^{\text{track}}, \dot{q}_{\max}^{\text{robot}}) \end{aligned} \quad (18)$$

where $q_{\min}^{\text{track}} = 0$ m, $q_{\max}^{\text{track}} = 1.55$ m, $\dot{q}_{\max}^{\text{track}} = 0.5$ m/s and

$$\begin{aligned} q_{\min}^{\text{robot}} &= (-2.5, -2.2, -2.5, -\pi, -\pi, -3.5) \text{ rad} \\ q_{\max}^{\text{robot}} &= (2.5, 0, 2.5, 2, \pi, 3.5) \text{ rad} \\ \dot{q}_{\max}^{\text{robot}} &= (1.4, 1.4, 1.4, 1.4, 1.4, 1.4) \text{ rad/s} \end{aligned} \quad (19)$$

Acceleration limits have been chosen based on empirical tests and correspond to:

$$\ddot{q}_{\max} = (\ddot{q}_{\max}^{\text{track}}, \ddot{q}_{\max}^{\text{robot}}) \quad (20)$$

where $\ddot{q}_{\max}^{\text{track}} = 1.0$ m/s² and

$$\ddot{q}_{\max}^{\text{robot}} = (1.57, 1.57, 1.57, 1.57, 1.57, 1.57) \text{ rad/s}^2.$$

The task-scaling IK algorithm used in the tests is the look-ahead method proposed in [25] with a predictive horizon equal to 0.2 s. Regarding the proposed method, the parameters to be chosen are Δq_{\max} and Δq_{ref} , where $\Delta q_{\text{ref}} < \Delta q_{\max}$. Their value is directly related to the maximum deviation with respect to the collision-free trajectory: the smaller are Δq_{\max} and Δq_{ref} , the smaller is the deviation due to the redundancy.

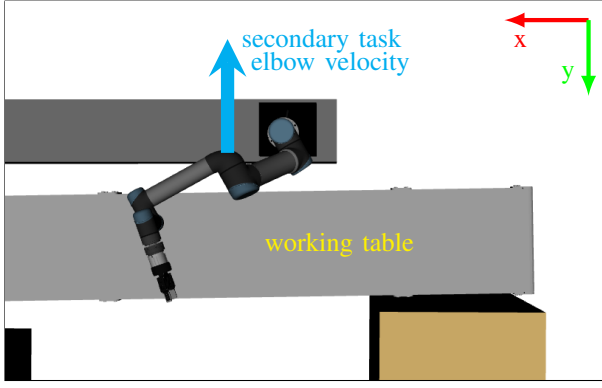


Fig. 3: Simulation scenario: the robot elbow should point toward the opposite direction with respect to the working table.

A. Simulations

Simulation tests have been performed by using a constrained kinematic model of the robot in ROS (Robot Operating System). The tests consist in the execution of a point-to-point Cartesian trajectory. A joint-space path is planned by means of the sampling-based algorithm *BiTRRT-Connect* available in *MoveIt!*. This path is collision free and is taken as the nominal joint-space task q_{nom} . The corresponding Cartesian path represents the primary task to be followed by the robot. As secondary task, the robot should keep the elbow link as far as possible from the working table positioned in front of it. This is formalized by setting A_2 equal to the second row of the robot elbow Jacobian and $-b_2$ equal to a velocity directed toward the direction $-y$ of the world frame. The scenario is shown in Figure 3.

For the purpose of the present work, the actual joint trajectory q should deviate from q_{nom} less than a given threshold Δq_{max} . Four different cases have been analyzed:

- Case A: typical IK scheme with constant joint bounds $[q_{\text{min}}, q_{\text{max}}]$ [22]; no deviation with respect to a joint-space reference is considered;
- Case B: soft-constraint approach [8]; the joint-space reference trajectory is set as lowest-priority task in the IK scheme;
- Case C: the self-regulating mechanism described in Section III-B is used, with $\Delta q_{\text{ref}} = 0.15$ for all joints.
- Case D: the constraint-based method described in Section III-A and the self-regulating method described in Section III-B are used, with $q_{\text{max}} = 0.2$ and $\Delta q_{\text{ref}} = 0.15$ for all joints.

In Cases C and D, the following values of the PID parameters have been used: $K_P = 1$, $K_I = 10$, and $K_D = 20$.

Figure 4 shows the resulting joint trajectories of Joints 2 and 6, and the ratio $\max_i |\Delta q_i / \Delta q_{\text{max},i}|$. Such ratio has to be less than or equal to one to satisfy the maximum deviation Δq_{max} . As expected, in Case A, the joint trajectory deviates from the nominal one without control. The soft-constraint approach (Case B) reduces the deviation from the nominal joint-space reference (that is set as lowest-priority

task) but the joint trajectories drift away from the reference as the elbow task has strict higher priority than the joint-space reference. When the adaptive weighting mechanism is activated (Case C), the deviation with respect to the nominal trajectory decreases and eventually converges to Δq_{ref} , but it is not possible to guarantee a maximum deviation value. As a matter of fact, although $\Delta q_{\text{ref}} = 0.75 \Delta q_{\text{max}}$, the maximum deviation value exceeds Δq_{max} during part of the task (at around 10.5 s). On the contrary, Case D always ensures the strict satisfaction of the maximum deviation: the joint trajectory never exits the tube $q_{\text{nom}} \pm \Delta q_{\text{max}}$ and the ratio is always less than or equal to one. As a drawback, the scaling mechanism has to be activated to recover feasibility and this results in the slowdown of the task. As shown in the figure, the time required to complete the task is around 1.5 times larger than the time taken by Cases A, B, and C. This phenomenon is relevant especially when Δq_{max} is very small or when $\Delta q_{\text{ref}} / \Delta q_{\text{max}}$ is close to one, for the joints are often close to their maximum allowed excursion. It is also evident that the limitation of the joint range of motion comes at the cost of a reduced satisfaction of the secondary task; the smaller is the allowed range of motion, the smaller is the secondary task satisfaction. In this scenario, the secondary task minimizes the y -component of the elbow position. Its level of satisfaction for the different cases is shown in the following table:

Case	A	B	C	D
2 nd task satisfaction	100%	58%	39%	27%

where the percentage of the secondary task satisfaction (compared with the unbounded Case A) is computed as:

$$2^{\text{nd}} \text{ task satisfaction} = 100 \cdot \frac{y_i^{\text{elb}} - y_0^{\text{elb}}}{y_A^{\text{elb}} - y_0^{\text{elb}}} \quad (21)$$

where y_i^{elb} is the average value of the elbow y -coordinate during the test, $i \in \{A, B, C, D\}$ denotes the case and y_0^{elb} refers to the average value of the elbow y -coordinate corresponding to the joint-space nominal trajectory.

B. Experiments

A human-robot cooperation application has been implemented on the real 7-degree-of-freedom platform composed of a Universal Robots model UR5, version 3.2, mounted on a linear track actuated via a linear motor and an Elmo Gold Cello servo drive. The proposed IK scheme (Case D in Section IV-A) has been implemented in ROS-control framework as an intermediate controller between *MoveIt!* and the robot controller. The IK scheme generates the reference signal for the robot controller, which is a cascade of a joint-position and a joint-velocity control loops. The position controller is a proportional gain equal to 7, the velocity controller is the built-in control system of the robot. The IK output is the reference of the joint-position control loop, whose control action is sent to the robot velocity controller via TCP/IP with a frequency of 125 Hz (maximum allowed by the UR5 controller interface). Similarly, the linear track

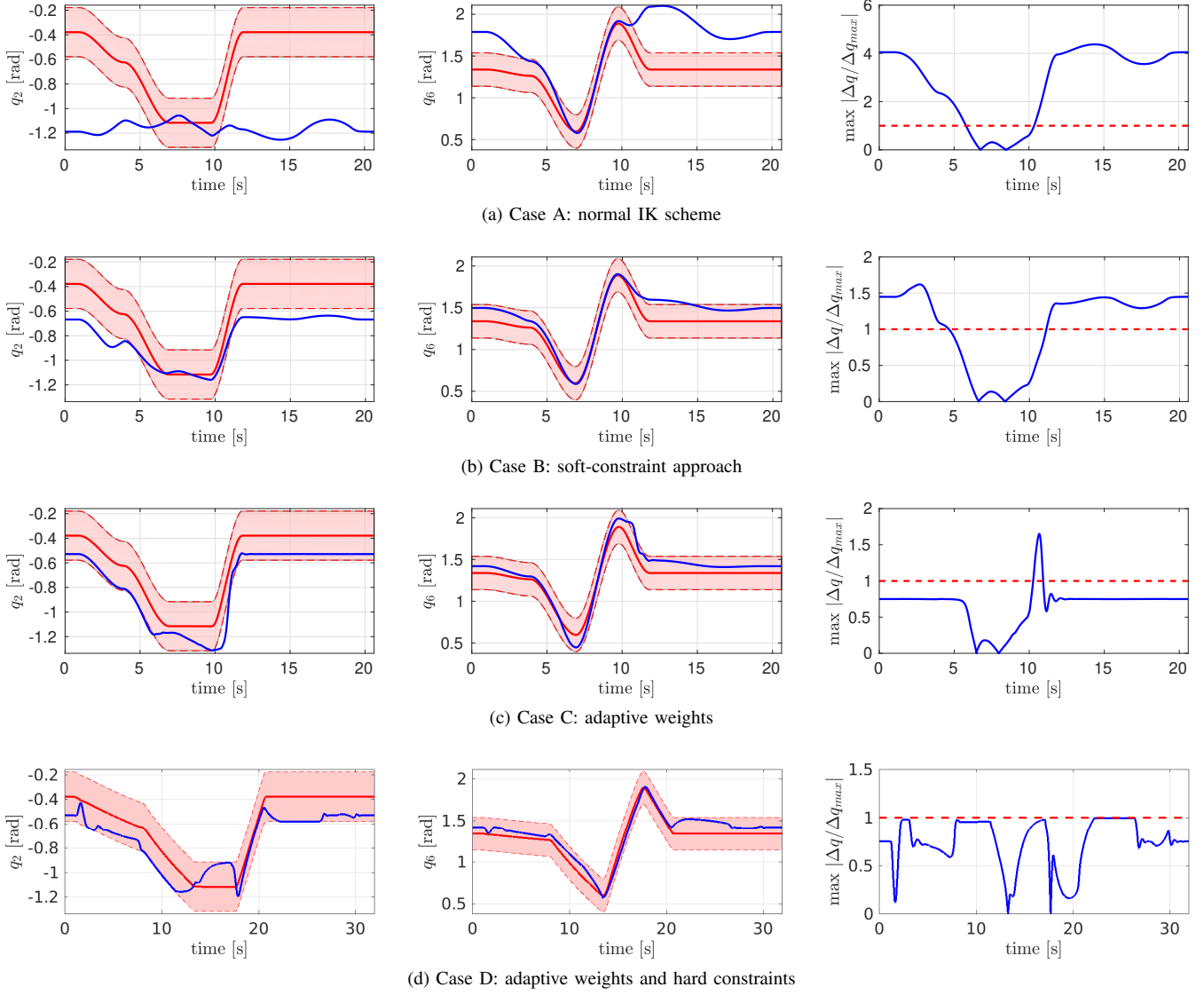


Fig. 4: Simulation results: joint configurations of Joints 2 and 6 (respectively, UR5's base and wrist2 joints, according to the manufacturer naming) and ratio $\max_i |\Delta q_i / \Delta q_{max,i}|$ for Cases A (first row), B (second row), C (third row), and D (fourth row). Solid blue: actual joint trajectories; solid red: nominal configuration-space trajectories q_{nom} ; rose area: allowed joint deviations $q_{nom} \pm \Delta q_{max}$. Cases A and B cannot limit the joint deviation; in Case C the joint deviation converges to Δq_{ref} but the hard constraint is violated during transients (the ratio is greater than 1 at time 10.5 s); in Case D the joint trajectories are always within the allowed tube (the ratio is always less than or equal to 1).

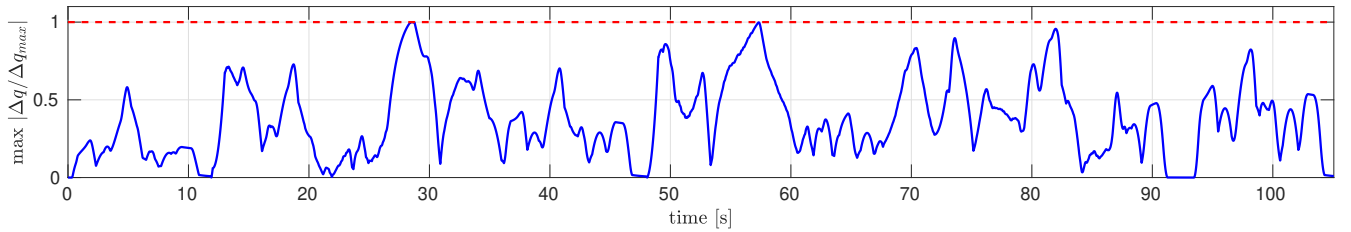


Fig. 5: Experimental results: ratio $\max_i |\Delta q_i / \Delta q_{max,i}|$ for Case D (solid blue). Ratio equal to 1 (dashed red) represents the maximum allowed deviation. Thanks to the proposed method (Case D), the ratio is always less than or equal to 1, that is, the joint trajectories are always within the allowed tube $q_{nom} \pm \Delta q_{max}$.

receives the control action with a frequency of 1 kHz by means of a Ethercat field-bus. The real-time position of the humans is acquired by means of two Realsense cameras, model d435. Static obstacles are removed from the point cloud signal by means of the ROS robot self-filter. The filter output is re-sampled by using 10 cm^3 voxels and sent to the IK solver with a frequency of 30 Hz.

The robot has to repeat a point-to-point motion in the Cartesian space, sharing the workspace with two operators. The secondary task consists in the movement of the robot elbow in the opposite direction with respect to the position of the closest body part. Hence, A_2 coincides with the first two rows of the elbow Jacobian and b_2 is equal to the unit vector from the elbow to the human position, multiplied by a gain equal to 0.4. The maximum and reference joint excursions are $\Delta q_{\max} = 0.4$ and $\Delta q_{\text{ref}} = 0.3$. To avoid collisions with the operators, the speed of the robot is reduced by a factor between 0 and 1 based on the human-robot relative distance as in [8]. The speed reduction factor is equal to 1 up to a distance equal to 1 m and is equal to 0 when the distance is equal to 0.3 m. Intermediate values are obtained by interpolating via a third-order polynomial.

Figure 5 shows that the ratio $\max_i |\Delta q_i / \Delta q_{\max, i}|$ is always less than or equal to one, which means that the joint trajectories always lay within the allowed tube $q_{\text{nom}} \pm \Delta q_{\max}$, according to the theoretical and simulation results.

V. CONCLUSIONS

This paper studied the time-varying limitation of the joint movements in the inverse kinematic problem of redundant manipulator. The motivation of the work came from the need of ensuring a bounded deviation of the joint movement with respect to a given nominal configuration-space trajectory. The paper showed that such limitation can be strictly ensured by using a task-scaling inverse kinematics and by deriving linear constraints with respect to the joint command variables. Moreover, a self-regulating weighting strategy was implemented to keep the robot joint close to the reference signal, so that the scaling mechanism activation is reduced. The validity of the proposed approach was proved via simulations and experiments on a 7-degree-of-freedom industrial manipulator.

APPENDIX

A. Viable polygonal sets for constrained double integral systems

Imposing the viability of the admissible set is necessary to ensure the feasibility of the optimal control problem for all possible states of the system [36]. The equation of the maximal viable set for a double integral system is quadratic in the system states. Polygonal viable sets are useful because they can be easily implemented in optimization-based control algorithms (*e.g.*, model predictive control [25], [32]). In [23], a polygonal viable set with a given number of sides is obtained by setting up an optimization problem that maximizes the area of the polygon. The output of the optimization is the sequence of points that delineate the polygon. The numerical

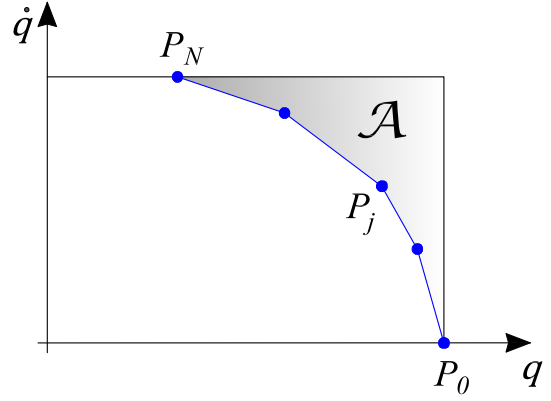


Fig. 6: Graphical representation of the polygonal viable set derivation. The first quarter of the polygon is delineated by the points $\{P_0, \dots, P_N\}$ where $P_j = (x_j, y_j)$.

approach is time consuming and a change in the joint limits would require the re-computation of the set. In this work we derive a viable polygonal set in closed-form.

To this purpose, consider the portion of the convex polygon in the first quarter delineated by the sequence of points $\{P_0, \dots, P_N\}$, where $P_j(x_j, y_j)$ (results for the third quarter of the plane can be derived straightforwardly by mirroring the solution). Starting from point $P_0(0, q_{\max})$, the maximal viable polygon can be constructed by maximizing the slope of the segments in the plane $q\dot{q}$ (see also Figure 6). The maximum slope at $\dot{q} = x_j$ depends on the maximum acceleration \ddot{q}_{\max} and is equal to $-x_j/\ddot{q}_{\max}$. Thus, the successive points can be derived with respect to their x_j coordinate, for example:

$$\begin{aligned} P_1 &= (x_1, q_{\max} - x_1^2/\ddot{q}_{\max}) \\ P_2 &= (x_2, q_{\max} - (x_2 - x_1)x_2/\ddot{q}_{\max}) \end{aligned} \quad (22)$$

For a generic point P_j it results:

$$P_j = \left(x_j, q_{\max} - \frac{1}{\ddot{q}_{\max}} \sum_{i=1}^j (x_i - x_{i-1}) x_i \right) \quad (23)$$

where $x_0 = 0$. The approach in [23] sets up an optimization problem to minimize the area \mathcal{A} in Figure 6. An analytical solution to the problem is doable only for small values of N ($N \leq 3$). Consider the simpler problem where only the y coordinate of point P_N is maximized, that is:

$$\begin{aligned} \underset{x_i}{\text{maximize}} \quad & q_{\max} - \frac{1}{\ddot{q}_{\max}} \sum_{i=1}^N (x_i - x_{i-1}) x_i \\ \text{subject to} \quad & x_0 = 0; \quad x_N = \dot{q}_{\max}. \end{aligned} \quad (24)$$

Notice that this problem is a simplification of the original one, but the approximation is reasonable also in view of the fact that the maximization of y_N tends to maximize the region where the robot is allowed to work at high speed. Problem (24) can be solved analytically by imposing $\delta y_N / \delta x_j = 0$, that is:

$$-\frac{1}{\ddot{q}_{\max}} (2x_j - x_{j-1} - x_{j+1}) = 0 \quad (25)$$

that can be rewritten as

$$x_{j+1} - x_j = x_j - x_{j-1} \quad (26)$$

with $x_N = \dot{q}_{\max}$. It follows that the points have to be equally spaced along the axis \dot{q} . This can be summarized as:

$$x_j = \frac{\dot{q}_{\max}}{N} j \quad \text{with } j = 1, \dots, N. \quad (27)$$

Finally, the points of the polygon can be found by inserting (27) into (23) to obtain:

$$P_j = \left(\frac{\dot{q}_{\max}}{N} j, q_{\max} - \frac{\dot{q}_{\max}^2}{2N^2 \ddot{q}_{\max}} j(j+1) \right) \quad (28)$$

It is interesting to note that:

$$\lim_{N \rightarrow \infty} y_N = q_{\max} - \frac{\dot{q}_{\max}^2}{2\ddot{q}_{\max}} \quad (29)$$

which means that the polygonal viable set tends to the maximal viable set given by (4), for $N \rightarrow \infty$.

ACKNOWLEDGMENT

The research leading to these results was partially funded by the European Union H2020-ICT-2017-1 – Pickplace: Flexible, safe and dependable robotic part handling in industrial environment (grant agreement: 780488).

REFERENCES

- [1] L. Jin, S. Li, H. M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4710–4720, 2017.
- [2] G. Huber and D. Wollherr, "Efficient closed-form task space manipulability for a 7-dof serial robot," *Robotics*, vol. 8, no. 4, 2019.
- [3] A. Atawneh, D. Papageorgiou, and Z. Doulgeri, "Kinematic control of redundant robots with guaranteed joint limit avoidance," *Rob. Auton. Syst.*, vol. 79, pp. 122–131, 2016.
- [4] C. Lamperti, A. M. Zanchettin, and P. Rocco, "A redundancy resolution method for an anthropomorphic dual-arm manipulator based on a musculoskeletal criterion," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Rob. Syst.*, Hamburg (Germany), 2015, pp. 1846–1851.
- [5] S. Tarbouriech, B. Navarro, P. Fraisse, A. Crosnier, A. Cherubini, and D. Sallé, "Dual-arm relative tasks performance using sparse kinematic control," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Rob. Syst.*, Madrid (Spain), 2018, pp. 6003–6009.
- [6] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *Proc. of the IEEE Int. Conf. on Rob. Autom.*, Saint Paul (USA), 2012, pp. 338–345.
- [7] M. Jin, Q. Liu, B. Wang, and H. Liu, "An efficient and accurate inverse kinematics for 7-dof redundant manipulators based on a hybrid of analytical and numerical method," *IEEE Access*, vol. 8, pp. 16 316–16 330, 2020.
- [8] M. Faroni, M. Beschi, and N. Pedrocchi, "An MPC framework for online motion planning in human-robot collaborative tasks," in *Proc. IEEE Int. Conf. on Emerging Tech. Factory Autom.*, Zaragoza (Spain), 2019.
- [9] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *Trans. Syst. Man Cybern.*, vol. 7, pp. 868–871, 1977.
- [10] S. Chiaverini, G. Oriolo, and A. A. Maciejewski, "Redundant robots," in *Springer Handbook of Robotics*. Springer, 2016, pp. 221–242.
- [11] B. Siciliano and J.-J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. of the Int. Conf. on Advanced Robotics*, Pisa (Italy), 1991, pp. 1211–1216.
- [12] F.-T. Cheng, T.-H. Chen, and Y.-Y. Sun, "Resolving manipulator redundancy under inequality constraints," *IEEE Trans. Rob. Autom.*, vol. 10, pp. 65–71, 1994.
- [13] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks," *IEEE Trans. Rob.*, vol. 27, pp. 785–792, 2011.
- [14] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *Int. J. Rob. Res.*, vol. 33, pp. 1006–1028, 2014.
- [15] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 882–893, 2016.
- [16] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.
- [17] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proc. of the IEEE Int. Conf. on Rob. Autom.*, vol. 2, San Francisco (USA), 2000, pp. 995–1001.
- [18] L. E. Kavraki, P. Svestka, J. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Rob. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.
- [19] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *Proc. of the IEEE Int. Conf. on Rob. Autom.*, Kobe (Japan), 2009, pp. 489–494.
- [20] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proc. of the IEEE Int. Conf. on Rob. Autom.*, Shanghai (China), 2011, pp. 4569–4574.
- [21] M. Faroni, D. Gorni, and A. Visioli, "Energy minimization in time-constrained robotic tasks via sequential quadratic programming," in *Proc. IEEE Int. Conf. on Emerging Tech. Factory Autom.*, Torino (Italy), 2018.
- [22] A. Del Prete, "Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators," *IEEE Rob. Autom. Lett.*, vol. 3, pp. 281–288, 2018.
- [23] M. Faroni, M. Beschi, N. Pedrocchi, and A. Visioli, "Viability and feasibility of constrained kinematic control of manipulators," *Robotics*, vol. 7, no. 3, pp. 1–19, 2018.
- [24] F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Trans. Rob.*, vol. 31, pp. 637–654, 2015.
- [25] M. Faroni, M. Beschi, N. Pedrocchi, and A. Visioli, "Predictive inverse kinematics for redundant manipulators with task scaling and kinematic constraints," *IEEE Trans. Rob.*, vol. 35, no. 1, pp. 278–285, 2018.
- [26] M. Liu, Y. Tan, and V. Padois, "Generalized hierarchical control," *Autonomous Robots*, vol. 40, pp. 17–31, 2016.
- [27] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. G. Tsagarakis, "Opensot: a whole-body control library for the compliant humanoid robot Coman," in *Proc. of the IEEE Int. Conf. on Rob. Autom.*, Seattle (USA), 2015, pp. 6248–6253.
- [28] M. Faroni, M. Beschi, A. Visioli, and L. Molinari Tosatti, "A predictive approach to redundancy resolution for robot manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8975–8980, 2017.
- [29] C. Schuetz, T. Buschmann, J. Baur, J. Pfaff, and H. Ulbrich, "Predictive online inverse kinematics for redundant manipulators," in *Proc. of the IEEE Int. Conf. on Rob. Autom.*, Hong Kong (China), 2014, pp. 5056–5061.
- [30] A. Zube, "Cartesian nonlinear Model Predictive Control of redundant manipulators considering obstacles," in *Proc. of the IEEE Int. Conf. on Industrial Technology*, Seville (Spain), 2015, pp. 137–142.
- [31] G. Antonelli, S. Chiaverini, and G. Fusco, "Kinematic control of redundant manipulators with on-line end-effector path tracking capability under velocity and acceleration constraints," in *Proc. of the IFAC Symposium on Robot Control*, Vienna (Austria), 2000, pp. 183–188.
- [32] M. Faroni, M. Beschi, C. Guarino Lo Bianco, and A. Visioli, "Predictive joint trajectory scaling for manipulators with kinodynamic constraints," *Control Engineering Practice*, vol. 95, p. 104264, 2020.
- [33] H. Han and J. Park, "Robot control near singularity and joint limit using a continuous task transition algorithm," *Int. J. Adv. Rob. Syst.*, vol. 10, no. 10, p. 346, 2013.
- [34] M. Liu, S. Hak, and V. Padois, "Generalized projector for task priority transitions during hierarchical control," in *Proc. of the IEEE Int. Conf. on Rob. Autom.*, Seattle (USA), 2015, pp. 768–773.
- [35] S. Kim, K. Jang, S. Park, Y. Lee, S. Y. Lee, and J. Park, "Continuous task transition approach for robot controller based on hierarchical quadratic programming," *IEEE Rob. Autom. Lett.*, vol. 4, no. 2, pp. 1603–1610, 2019.
- [36] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, pp. 1747–1767, 1999.