Mechanism and Machine Theory 226 (2026) 106477

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmt

Research paper

Kinematic redundancy resolution in robotic systems for energy efficiency

Giuliano Fabris [a], Lorenzo Scalera [a,*], Andrea Giusti [b], Alessandro Gasparetto [a]

[a] Polytechnic Department of Engineering and Architecture, University of Udine, via delle Scienze 206, Udine, 33100, Italy
[b] Faculty of Engineering, Free University of Bozen-Bolzano, NOI Techpark, via Bruno Buozzi, 1, Bolzano, 39100, Italy

ARTICLE INFO

Keywords:
Energy efficiency
Kinematic redundancy
Robotics
Trajectory planning
Uncertain dynamics

ABSTRACT

Increasing energy efficiency in automation can lead to reduced operating costs and enhanced production sustainability. In this paper, we present a novel approach for energy efficiency in redundant robotic systems. The proposed scheme aims at determining the weighting matrix used to compute the minimum-energy solution of the inverse kinematics through the weighted Jacobian pseudo-inverse. Two different solutions for computing the energy-efficient weights are proposed, one independent and the other dependent on the desired robot end-effector trajectory. The proposed approach also accounts for robot dynamics uncertainties. The performance of the approach is validated on a robotic system with 8 degrees of freedom, composed of a manipulator mounted on a linear axis. The results of extensive numerical simulations and bespoke experimental tests demonstrate the effectiveness of the proposed approach in reducing the mechanical energy consumption with respect to a state-of-the-art approach. A reduction of the energy expenditure up to 97.5% is found in the numerical tests, and up to 97.2% in the experiments, while guaranteeing robustness to imperfect knowledge of the robot dynamics parameters.

1. Introduction

Robotic systems are fundamental to modern industry, where they can often replace human operators in repetitive, non-ergonomic, and hazardous tasks. They also play a crucial role in enhancing precision, repeatability, and overall productivity [1,2]. For these reasons, the demand for robotic installations is growing in the industrial sector. However, this trend has been accompanied by increasing energy consumptions [3]. In this context, achieving energy efficiency in robotic systems is crucial for sustainability, operational cost reduction, and performance optimization [4].

In existing work, various solutions are provided to promote energy efficiency in robotic systems. Possible strategies include the adoption of regenerative drives [5], the design and optimization of the robot structure [6] and its electronics [7], as well as the proper selection of joint motors and gearboxes [8]. Furthermore, other approaches to reduce energy consumption include the exploitation of natural motion by adding compliant elements to the robotic system [9], the optimal placement of the robot with respect to the task to be executed [10–12], the optimization of motion time [13,14], and the optimal trajectory planning [15,16]. The optimization of the robot trajectory is an extensively investigated solution, since it can be widely applied, enhancing adaptability and production flexibility, without the need to modify the structure and the actuators of existing systems. Trajectory planning algorithms that consider

* Corresponding author.
E-mail address: lorenzo.scalera@uniud.it (L. Scalera).

https://doi.org/10.1016/j.mechmachtheory.2026.106477
Received 19 December 2025; Received in revised form 9 April 2026; Accepted 1 May 2026
Available online 9 May 2026

Nomenclature

A_{bounds}	Area inside the linear axis friction bounds
$\boldsymbol{C}$	Coriolis and centrifugal contributions matrix
$\boldsymbol{D}$	Experimental data for computing linear axis friction bounds
DOF	Degree of freedom
E	Mechanical energy
E_{exp}	Mechanical energy in the experimental tests
E_{num}	Mechanical energy in the numerical tests
$\Delta E_{\lvert num-exp \rvert}$	Mechanical energy difference between simulations and experiments
$F_{c,low}$	Lower bound of the Coulomb friction coefficient
$F_{c,up}$	Upper bound of the Coulomb friction coefficient
$F_{v,low}$	Lower bound of the viscous friction coefficient
$F_{v,up}$	Upper bound of the viscous friction coefficient
F_s	Friction coefficient accounting for velocity uncertainty at low speed
H	Turns of the helical path
IA-RNEA	Interval arithmetic-based Newton-Euler algorithm
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{J}^{\#\boldsymbol{W}}$	Weighted right pseudo-inverse of the Jacobian matrix
J_g	Gearbox inertia
J_m	Motor inertia
J_p	Pulley inertia
$\boldsymbol{K_p}$	Control gain matrix
L	Length of linear axis
$\boldsymbol{M}$	Mass matrix
N	Number of the robot degrees of freedom
$\boldsymbol{P}$	Mechanical power
R	Dimension of the task space
ROS	Robot Operating System
SNS	Saturation in the Null Space
SQP	Sequential quadratic programming
S1	Trajectory-independent strategy for weights selection
S2	Trajectory-dependent strategy for weights selection
T	Total execution time
T_t	Total execution time of the trapezoidal trajectory
$\boldsymbol{W}$	Weighting matrix
$\boldsymbol{W_{heur}}$	Heuristic energy-efficient weighting matrix
$2a_f$	Major axis of the final ellipse of the helical trajectory
$2a_i$	Major axis of the initial ellipse of the helical trajectory
a_1	Coulomb friction coefficient
a_2	Viscous friction coefficient
a_3	First low-velocity friction coefficient
a_4	Second low-velocity friction coefficient
$2b_f$	Minor axis of the final ellipse of the helical trajectory
$2b_i$	Minor axis of the initial ellipse of the helical trajectory
$\boldsymbol{e}$	End-effector pose error
$\boldsymbol{e_o}$	End-effector orientation error
$\boldsymbol{e_p}$	End-effector position error
f	Sampling frequency
f_s	Sampling frequency of the re-sampled trajectory
$\boldsymbol{g}$	Gravitational contribution
i_r	Gear ratio
m_{arm}	Mass of the robotic arm
m_{cart}	Mass of the cart
n	Number of points of the trajectory
n_s	Number of points of the re-sampled trajectory
p	Sampling percentage
pct_{exp}	Percentage energy reduction in the experimental tests
pct_{num}	Percentage energy reduction in the numerical tests
p	Sampling percentage

$\boldsymbol{q}$	Joint positions
$\boldsymbol{q}_{\boldsymbol{lim}}^{-}$	Lower joint position limits
$\boldsymbol{q}_{\boldsymbol{lim}}^{+}$	Upper joint position limits
$\boldsymbol{q}_{\boldsymbol{start}}$	Initial joint configuration
$\Delta q_{i,a}$	Space covered during the acceleration phase
$\dot{\boldsymbol{q}}$	Joint velocities
$\dot{\boldsymbol{q}}_{\boldsymbol{lim}}$	Joint velocity limits
$\ddot{\boldsymbol{q}}$	Joint accelerations
$\ddot{\boldsymbol{q}}_{\boldsymbol{lim}}$	Joint acceleration limits
r	Pulley radius
rms	Root mean square
t	Time
t_a	Acceleration time
t_c	Time spent at constant velocity
Δt_s	Sampling time of the re-sampled trajectory
w	Manipulability index
$\boldsymbol{x}$	Robot end-effector pose
$\boldsymbol{x}_{\boldsymbol{d}}$	Robot desired end-effector pose
$\dot{\boldsymbol{x}}$	Robot end-effector velocities
$\dot{\boldsymbol{x}}_{\boldsymbol{d}}$	Robot desired end-effector velocities
$\ddot{\boldsymbol{x}}$	Robot end-effector accelerations
x_b	x-axis of the robot base reference frame
x_e	y-axis of the robot base reference frame
Δx_f	Horizontal dimension of the final rectangle of the helical trajectory
Δx_i	Horizontal dimension of the initial rectangle of the helical trajectory
x_s	x coordinate of the starting point of the helical trajectory
y_b	z-axis of the robot base reference frame
y_e	x-axis of the robot end-effector reference frame
y_s	y coordinate of the starting point of the helical trajectory
Δy	Movement along the y-axis during the helical trajectory
z_b	y-axis of the robot end-effector reference frame
z_e	z-axis of the robot end-effector reference frame
Δz_f	Vertical dimension of the final rectangle of the helical trajectory
Δz_i	Vertical dimension of the initial rectangle of the helical trajectory
z_s	z coordinate of the starting point of the helical trajectory
$\boldsymbol{\Delta}$	Robot dynamics parameters
$\boldsymbol{\kappa}$	Forward kinematics
$\boldsymbol{\tau}$	Joint torques
θ	Pitch angle
θ_{max}	Maximum pitch angle
θ_{min}	Minimum pitch angle
$\underline{\boldsymbol{\tau}}$	Infimum of the interval for the joint torques
$\overline{\boldsymbol{\tau}}$	Supremum of the interval for the joint torques
$\tau_{f,exp}$	Experimental friction torque
$\boldsymbol{\tau}_{\boldsymbol{f}}$	Joint friction torques
$\boldsymbol{\tau}_{\boldsymbol{lim}}$	Joint torque limits
$\boldsymbol{\tau}_{\boldsymbol{max}}$	Maximum absolute value of the joint torques
ϕ	Yaw angle
ψ	Roll angle
ψ_{max}	Maximum roll angle
ψ_{min}	Minimum roll angle
$round(\cdot)$	Round operator
$sign(\cdot)$	Sign operator

energy efficiency for robotic systems can provide significant energy savings and operational efficiency [17]. Several examples of energy-optimal trajectory planning approaches exist, based for instance on the selection and optimization of the motion law [18,19], and its associated parameters [20].

Energy efficiency through trajectory planning can also be achieved by exploiting kinematic redundancy. A kinematically redundant robotic system has a number of degrees of freedom (DOFs) greater than the number of variables needed to perform the desired task, potentially leading to infinite inverse kinematics solutions. Redundant robots offer greater flexibility and enhanced capabilities compared to non-redundant ones [21]. Indeed, the redundant DOFs can be exploited for achieving a secondary objective through redundancy resolution, such as respecting joint limits or avoiding collisions, without affecting the motion of the end-effector [22,23].

Most existing works focus on solving redundancy through optimal control, with the aim of maximizing manipulability, avoiding collisions and singularities. For instance, Vu et al. [24] aim to find the inverse kinematics solution that maximizes the manipulability index $w = \sqrt{\det(\boldsymbol{J}\boldsymbol{J}^T)}$, with $\boldsymbol{J}$ being the manipulator Jacobian matrix, through an artificial neural network. Hong et al. [25] propose to include an interference vector, determined through reinforcement learning, in the computation of the pseudo-inverse of the Jacobian for self-collision avoidance, whereas Pei et al. [26] optimize the robot configuration during the trajectory, relying on velocity and force ellipses. Various approaches exploit the null space of the Jacobian for pursuing secondary tasks. For example, the authors in [27,28] present two different approaches for avoiding singularities by exploiting null space projection. The former exploits the use of differential dynamic programming to perform a global optimization based on the gradient projection method, whereas the latter solves the inverse kinematic problem locally. Monari et al. [29] propose a method based on null space projection to avoid collisions, whereas the authors in [30] consider null space projection to achieve different secondary tasks, i.e., maximization of manipulability together with collision and singularity avoidance.

Null space projection can also be considered for respecting joint limits. To this end, the Saturation in the Null Space (SNS) approach is introduced in [31] at velocity level, which accounts for joint position and velocity limits as hard constraints. For respecting these constraints, the initial pseudo-inverse-based solution is progressively adjusted by iteratively clamping the most violating joint velocity to its saturation limit. These joint velocities are then projected into the null space of the task-Jacobian associated with the remaining active (i.e., unsaturated) joints (obtained by zeroing the task Jacobian columns corresponding to the saturated joints), thereby ensuring task feasibility while respecting joint limits. If the task is still infeasible, the SNS algorithm adopts a task-scaling strategy to ensure motion feasibility under the inequality constraints, while maintaining the geometric characteristics of the primary task, i.e., the trajectory to be executed by the end-effector. The SNS approach has been extended in [32–35]. In these works, the authors aim at enhancing computational efficiency, dealing with multiple tasks in a hierarchical way, and accounting for Cartesian constraints. A different method for avoiding joint limits violations is proposed in [36], where the authors consider a self-organizing map network and a weighting matrix that penalizes joint motion for computing the inverse kinematics.

The redundancy of robotic systems can also be exploited for reducing energy consumption. Doan et al. [37] consider a modified particle swarm optimization scheme to find the optimal end-effector torch angle during a welding task for minimizing the consumed energy, avoiding collisions and joint limits. Ruiz et al. [38] propose to use model predictive control to find the position of the redundant joints that reduces the energy consumption, keeping that position fixed during the trajectory. Similarly, the authors in [39] present an approach for energy efficiency that optimizes the position of selected redundant joints at each of the way points that define a pick-and-place trajectory. Boscariol et al. [40] leverage the dynamic and electric models of a UR5 robot for optimizing the end-effector orientation during the trajectory in order to minimize the energy consumption. Furthermore, the authors in [41] enhance the energetic performance by optimizing the Cartesian inertia of the robot, whereas Li et al. [42] present an approach for energy efficiency that employs a neural network in order to find the energy-optimal inverse kinematics solution.

In this work, we present a novel approach for energy efficiency in redundant robotic systems. Our proposed scheme aims at optimizing the solution of the inverse kinematics in order to minimize the energy consumption along given end-effector trajectories, also considering uncertain dynamics of the robotic system. This method is based on the SNS algorithm, which allows the execution of the predefined task, while respecting the joint limits. The SNS algorithm is selected as base approach with respect to dynamic optimization methods for its reduced computational demands [34,35]. Other approaches exist that consider constrained optimal control problems for energy efficiency purposes. However, these are often restricted to simple cases, as for instance planar robots with two revolute joints [43]. Furthermore, approaches based on optimal control can require up to two orders of magnitude more computation time to find a solution [44]. The proposed approach aims at defining a suitable weighting matrix for the inverse kinematics solution through the weighted Jacobian pseudo-inverse [45] that provides a minimum-energy solution to the inverse differential kinematics problem. We propose two solutions, which provide complementary pros and cons, one independent and the other dependent on the desired trajectory. With our proposed approach we show that an optimized choice of the weighting matrix has a significant impact on the energy consumption, and its optimization to minimize energy-related costs has previously remained unexplored, to the best of our knowledge. Furthermore, uncertainty in the robot dynamics parameters, which affects the proper computation of joint torques and consumed energy, is here considered through interval arithmetic [46]. We evaluate the effectiveness of the proposed approach on a 8-DOF robotic system, considering two different test cases. The results of extensive simulations and experiments highlight the mechanical energy-saving capabilities of our proposed approach with respect to a state-of-the-art approach [31] with percentage reduction up to 97.5% in the numerical tests, and up to 97.2% in the experiments. These results show the feasibility of the proposed approach in reducing significantly the mechanical energy consumption while respecting the joint limits even in case of imperfect knowledge of the robot dynamics parameters.

Table 1
Overview on redundancy resolution approaches for energy-efficiency.

Reference	Robotic system DOFs	Tested degrees of redundancy	Redundancy resolution	Energy-related cost function	Dynamic model
[37]	6	1	Particle swarm optimization	$P = k \sum_{i=1}^{6} \tau_i \omega_i$	Nominal
[38]	6	1, 2, 3	Optimization of redundant joints position	$f(\boldsymbol{q}) = \boldsymbol{q}^T \boldsymbol{H} \boldsymbol{q}$	None (kinematics only)
[39]	8	2	Optimal inverse kinematics at way points	$E = \int_0^T \sum_{i=1}^{N} \tau_i \dot{q}_i \, dt \Big\vert_{\tau_i \dot{q}_i > 0}$	Nominal
[40]	6	2	Optimization of end-effector angles	$E = \int_0^T \sum_{i=1}^{N} V_i \, I_i dt$	Nominal
[41]	7	1	Optimization of robot Cartesian inertia	$f(\boldsymbol{q}) = \frac{1}{2} \boldsymbol{n}^T (\boldsymbol{J} \boldsymbol{M}^{-1} \boldsymbol{J}^T)^{-1} \boldsymbol{n}$	Nominal
Proposed approach	8	2	Weighted Jacobian pseudoinverse optimization with SNS	$E = \int_0^T \boldsymbol{\tau}^T \dot{\boldsymbol{q}} \, dt$	Nominal and uncertain

In summary, the main contributions of this work are:

1. an approach for kinematic redundancy resolution that reduces the mechanical energy consumption in redundant robotic systems, while also avoiding joint limits violations;
2. the consideration of the uncertainty in the robot dynamics parameters in the redundancy resolution;
3. the results of the extensive numerical and experimental validation of the proposed approach that quantify the actual energy savings with respect to a state-of-the-art technique.

Table 1 summarizes the essential features of the proposed strategy with respect to prior works that use mass-matrix weighting or other energy-related cost functions. In [37], k is a weight to normalize the computed power in the range $[0, 1]$. In [38], $\boldsymbol{H}$ is needed to write the cost function in a matrix form. In [40], V_i and I_i are the electrical voltage and current of the ith motor. In [41], $\boldsymbol{n}$ is a vector defining the direction of motion. Remarkably, our approach is the first that considers both nominal and uncertain dynamics, and leverages the benefits of the SNS approach for handling hard joint constraints.

The remainder of the paper is organized as follows: the problem statement is presented in Section 2, whereas Section 3 describes the proposed approach for minimizing energy consumption. Section 4 illustrates the simulation testbed and the simulation results, Section 5 describes the experimental setup, whereas the experimental results are detailed in Section 6. Finally, Section 7 concludes the work.

2. Problem statement

The problem addressed in this work is the kinematic redundancy resolution for maximizing energy efficiency in robotic systems, with or without uncertain dynamics. Given a serial kinematic chain of rigid bodies and kinematic pairs that provide an open-chain robotic system with N DOFs and a R-dimensional task (with $N > R$), the relation between the joint variables $\boldsymbol{q} \in \mathbb{R}^N$ and the task coordinates $\boldsymbol{x} \in \mathbb{R}^R$ is given by the forward kinematics equation:

$$\boldsymbol{x} = \boldsymbol{\kappa}(\boldsymbol{q}). \tag{1}$$

By differentiating Eq. (1) with respect to time, we get:

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

where $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\kappa}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ is the $R \times N$ Jacobian matrix. Assuming that the robot is controlled by means of a kinematic control law at the velocity level, the following inverse differential kinematics relation can be considered [45]:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#\boldsymbol{W}}(\boldsymbol{q}) \dot{\boldsymbol{x}} \tag{3}$$

where

$$\boldsymbol{J}^{\#\boldsymbol{W}}(\boldsymbol{q}) = \boldsymbol{W}^{-1} \boldsymbol{J}^T(\boldsymbol{q}) (\boldsymbol{J}(\boldsymbol{q}) \boldsymbol{W}^{-1} \boldsymbol{J}^T(\boldsymbol{q}))^{-1} \tag{4}$$

is the right pseudo-inverse of $\boldsymbol{J}(\boldsymbol{q})$, weighted through $\boldsymbol{W}$, i.e., a $N \times N$ suitable, symmetric, positive definite matrix.

When solving the inverse differential kinematics as in Eq. (3), the choice of the weighting matrix $\boldsymbol{W}$ influences the resulting joint velocities, privileging the movement of some joints with respect to others. A possible selection of the weighting matrix $\boldsymbol{W}$ is the mass matrix $\boldsymbol{M}$, which ensures dynamical consistency of the generated robot motion and penalizes joints experiencing higher reflected inertia [47]. However, from an energy perspective, choosing $\boldsymbol{W}$ equal to $\boldsymbol{M}$ only reduces the system kinetic energy, since the resulting weighted Jacobian pseudo-inverse $\boldsymbol{J}^{\#\boldsymbol{W}}$ is the one that minimizes $\frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{M} \dot{\boldsymbol{q}}$ [48]. Furthermore, this choice does not take

into account that the total energy consumption is influenced by other components, such as Coriolis and centrifugal forces, gravity, and friction effects. To solve this problem, optimal weights $\boldsymbol{W}$ that reduce the total energy consumption for a given end-effector task can be defined.

Knowing the joint velocity $\dot{\boldsymbol{q}}$ from the inverse differential kinematics, the joint positions $\boldsymbol{q}$ can be calculated by numerical integration using the forward Euler method, whereas the joint accelerations $\ddot{\boldsymbol{q}}$ can be obtained by numerical differentiation. Then, the joint torques $\boldsymbol{\tau}$ needed by the robot to execute the trajectory $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, where the time dependence of the variables is omitted for compactness, can be computed through inverse dynamics as:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q}, \boldsymbol{\Delta})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\Delta})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\Delta}) + \boldsymbol{\tau}_f(\dot{\boldsymbol{q}}, \boldsymbol{\Delta}). \tag{5}$$

$\boldsymbol{M}(\boldsymbol{q}, \boldsymbol{\Delta})$ represents the mass matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\Delta})\dot{\boldsymbol{q}}$ includes the Coriolis and centrifugal terms, the vector $\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\Delta})$ considers the gravitational contribution, and $\boldsymbol{\tau}_f(\dot{\boldsymbol{q}}, \boldsymbol{\Delta})$ accounts for the friction effects. The robot dynamics parameters are denoted by $\boldsymbol{\Delta}$.

The considered optimization problem for a robot trajectory executed in a total time T can be formulated as follows:

$$\min_{\boldsymbol{W}} \; E \tag{6}$$

where

$$E = \int_0^T \boldsymbol{\tau}^T \dot{\boldsymbol{q}}\, dt = \int_0^T \boldsymbol{\tau}^T \boldsymbol{J}^{\#\boldsymbol{W}}(\boldsymbol{q})\dot{\boldsymbol{x}}\, dt \tag{7}$$

subject to Eq. (3) with Eq. (4) and:

$$\begin{cases} \boldsymbol{q}_{\boldsymbol{lim}}^- < \boldsymbol{q} < \boldsymbol{q}_{\boldsymbol{lim}}^+ \\ |\, \dot{\boldsymbol{q}}\, | < \dot{\boldsymbol{q}}_{\boldsymbol{lim}} \\ |\, \ddot{\boldsymbol{q}}\, | < \ddot{\boldsymbol{q}}_{\boldsymbol{lim}} \\ |\, \boldsymbol{\tau}\, | < \boldsymbol{\tau}_{\boldsymbol{lim}} \end{cases} \tag{8}$$

where the inequalities are applied element-wise, $\boldsymbol{q}_{\boldsymbol{lim}}^-$ and $\boldsymbol{q}_{\boldsymbol{lim}}^+$ are the vectors of lower and upper joint position limits, whereas $\dot{\boldsymbol{q}}_{\boldsymbol{lim}}$, $\ddot{\boldsymbol{q}}_{\boldsymbol{lim}}$, and $\boldsymbol{\tau}_{\boldsymbol{lim}}$ are the vectors of joint velocity, joint acceleration, and joint torque limits, respectively.

Since $\boldsymbol{W}$ influences all the kinematics and dynamics variables during the trajectory, it is included in both the cost function and the constraints. The mechanical energy E required by the robot to complete the desired trajectory is computed by integrating the mechanical power $\boldsymbol{P}_i = \boldsymbol{\tau}_i \dot{\boldsymbol{q}}_i$ provided by all N joints over the execution time T, as in Eq. (7). To respect the limits of the robot joints during the trajectory, we leverage the results in [31].

Please note that computing the mechanical energy consumption as in Eq. (7) assumes that the considered robotic system is able to recover energy during braking phases. In case this is not possible, a different cost function may be considered, e.g.,

$$E = \int_0^T \sum_{i=1}^{N} |\, \tau_i\, \dot{q}_i\, |\, dt. \tag{9}$$

3. Proposed approach

In this section, we first present two strategies for minimizing the energy consumption in redundant robotic systems. Next, we describe the strategy adopted for handling the uncertainties in the dynamic parameters of the robot.

3.1. Energy consumption minimization

We propose two different solutions for selecting the energy-efficient weights: (S1) a choice that is independent from the trajectory; (S2) a choice which is trajectory-dependent. The proposed S1 and S2 approaches are then compared with the SNS state-of-the-art reference approach in [31].

S1 relies on the considered robotic system only and allows us to determine a heuristic energy-efficient weighting matrix $\boldsymbol{W}$ with a simple approach. To this end, the following motion, composed of two trapezoidal trajectories, is first executed:

$$q_i(t) = \begin{cases} q_{i,lim}^- + \frac{1}{2}\ddot{q}_{i,lim} t & 0 \le t \le t_a \\ q_{i,lim}^- + \Delta q_{i,a} + \dot{q}_{i,lim}(t - t_a) & t_a < t \le t_a + t_c \\ q_{i,lim}^+ - \frac{1}{2}\ddot{q}_{i,lim}(T_t - t) & t_a + t_c < t \le T_t \\ q_{i,lim}^+ - \frac{1}{2}\ddot{q}_{i,lim}(t - T_t) & T_t \le t \le T_t + t_a \\ q_{i,lim}^+ - \Delta q_{i,a} - \dot{q}_{i,lim}(t - T_t - t_a) & T_t + t_a < t \le T_t + t_a + t_c \\ q_{i,lim}^- + \frac{1}{2}\ddot{q}_{i,lim}(2T_t - t) & T_t + t_a + t_c < t \le 2T_t \end{cases} \tag{10}$$

where $t_a = \dot{q}_{i,lim}/\ddot{q}_{i,lim}$ is the acceleration (and deceleration) time, $\Delta q_{i,a}$ indicates the space covered during the acceleration (and deceleration) phase, whereas $t_c = (q_{i,lim}^+ - q_{i,lim}^-)/\dot{q}_{i,lim} - \dot{q}_{i,lim}/\ddot{q}_{i,lim}$ is the time spent at constant velocity, and $T_t = 2t_a + t_c$ represents the total time needed to execute a single trapezoidal trajectory. This motion is performed for a ith joint with $1 < i < N$, while all the other jth joints with $1 < j < N \wedge j \neq i$ remain fixed in their center position $q_j = (q_{j,lim}^- + q_{j,lim}^+)/2$. Then, the mechanical energy E_i

required to execute this motion is estimated as in Eq. (7), where the joint torques are computed from Eq. (5). This procedure is then repeated for all the N joints. Finally, the weights W_i with $i \in [1, ..., N]$ are obtained as:

$$W_i = \frac{E_i}{E_{min}} \tag{11}$$

where $E_{min} = \min_j E_j$ is the smallest energy consumption among all the considered motions. With this procedure, a diagonal heuristic energy-efficient weighting matrix $\boldsymbol{W_{heur}}$ is determined, where each W_i represents the ith diagonal element. This approach is simple and penalizes the movement of the most energy-demanding joints.

In the strategy S2 the weights are optimized considering both the robotic system and the specific trajectory to execute, i.e., taking as input the velocity $\dot{\boldsymbol{x}}$ of the end-effector of the redundant robotic system during the trajectory, discretized with a sampling rate f, and the robot dynamics parameters $\boldsymbol{\Delta}$. The weighting matrix $\boldsymbol{W}$ is optimized so as to find the energy-optimal solution of the inverse differential kinematics, computed as in Eq. (3), whereas the mechanical energy consumption is obtained from Eq. (7). To speed up the optimization process one can conveniently re-sample the trajectory as follows:

$$n_s = \text{round}\left(\frac{p\,n}{100}\right) \tag{12}$$

where $p \in (0, 100]$ indicates a sampling percentage, $n = f\,T$ represents the number of points of the original trajectory, whereas n_s is the number of points considered for the re-sampled trajectory, and $round(\cdot)$ is the operator that rounds the argument to the nearest integer. This is aimed at finding a trade-off between energy savings and computational time. The sampling process of the end-effector trajectory $(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}})$ is performed uniformly over time. In this manner, the sampling rate of the re-sampled trajectory becomes $f_s = (n_s/n)\,f$.

Once a sampling percentage p and an initial guess for the weighting matrix $\boldsymbol{W}$ have been selected, the joint velocities $\dot{\boldsymbol{q}}$ are computed using Eq. (3) for each point of the re-sampled trajectory, whereas the joint positions $\boldsymbol{q}$ can be calculated by numerical integration. In order to properly follow the desired end-effector trajectory $\boldsymbol{x_d}$, a feedback action is added to the desired end-effector velocity $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$ as:

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{K_p e} = \dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{K_p}(\boldsymbol{x_d} - \boldsymbol{\kappa}(\boldsymbol{q})) \tag{13}$$

where $\dot{\boldsymbol{x}}$ is the current end-effector velocity to be considered in Eq. (3). $\boldsymbol{K_p}$ is a $R \times R$ diagonal, positive definite control gain matrix, $\boldsymbol{e}$ represents the end-effector pose error, and $\boldsymbol{\kappa}(\boldsymbol{q})$ indicates the forward kinematics equation (Eq. (1)). The joint accelerations $\ddot{\boldsymbol{q}}$ are then obtained through numerical differentiation, and the joint torques are computed with Eq. (5). In case that the robot dynamics parameters are uncertain, the approach described in Section 3.2 can be adopted. Finally, the considered measure of the mechanical energy consumption is obtained as:

$$E = \sum_{i=1}^{n_s} \boldsymbol{\tau_i}^T \dot{\boldsymbol{q}}_{\boldsymbol{i}}\, \Delta t_s \tag{14}$$

where $\Delta t_s = 1/f_s$ is the sampling time of the re-sampled trajectory.

3.2. Handling robot dynamics uncertainty

Both proposed strategies S1 and S2 require the computation of the inverse dynamics to obtain the joint torques for Eq. (7) and Eq. (14). In Section 3.1, the nominal dynamics parameters of the robot are assumed to be accurately known. In this subsection, the dynamics parameters of the redundant robotic system are considered to be uncertain within known bounds. To describe the set of torques from inverse uncertain-dynamics, a multidimensional interval is adopted:

$$[\boldsymbol{\tau}] := [\underline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}}],\ \ \underline{\boldsymbol{\tau}} \in \mathbb{R}^N,\ \ \overline{\boldsymbol{\tau}} \in \mathbb{R}^N,\ \ \underline{\tau_i} < \overline{\tau_i},\ \ \forall\, i = 1, ..., N \tag{15}$$

where $\underline{\boldsymbol{\tau}}$ and $\overline{\boldsymbol{\tau}}$ indicate the infimum and supremum of the interval vector for the joint torques, respectively. By assuming that the dynamics parameters are uncertain, we introduce the corresponding multidimensional interval $[\boldsymbol{\Delta}]$ (defined as $[\boldsymbol{\tau}]$ in Eq. (15), *mutatis mutandis*), so that the joint torques can be computed using an interval arithmetic-based NE algorithm (IA-RNEA) as in [49,50]:

$$[\underline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}}] = \text{IA-RNEA}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, [\boldsymbol{\Delta}]). \tag{16}$$

With this algorithm, the greatest lower bound and the least upper bound of the joint torques set can be computed. Similarly to the standard NE approach, this algorithm preserves linear computational complexity as a function of the number of robot DOFs and can be extended to take into account the uncertain kinematic states of the robot base [51]. Please note that this algorithm provides the same results of the standard NE approach if the interval vector of dynamics parameters is degenerate, i.e., its infimum coincides with the supremum and contains the nominal values. Therefore, our proposed approach can be used both for the case in which the dynamics parameters can be considered as nominal and the case in which intervals should be used instead for obtaining more robust results.

When considering the uncertainty in the robot dynamics parameters, the maximum absolute value of the joint torque $\boldsymbol{\tau_{max}}$ can be computed from its bounds as:

$$\boldsymbol{\tau_{max}} = \max(|\,\underline{\boldsymbol{\tau}}\,|, |\,\overline{\boldsymbol{\tau}}\,|) \tag{17}$$

where the *max* operator is applied element-wise. Then, $\boldsymbol{\tau_{max}}$ can be included in Eq. (8), allowing us to optimize the weighting matrix $\boldsymbol{W}$ while respecting the joint limits even in the worst-case scenario.

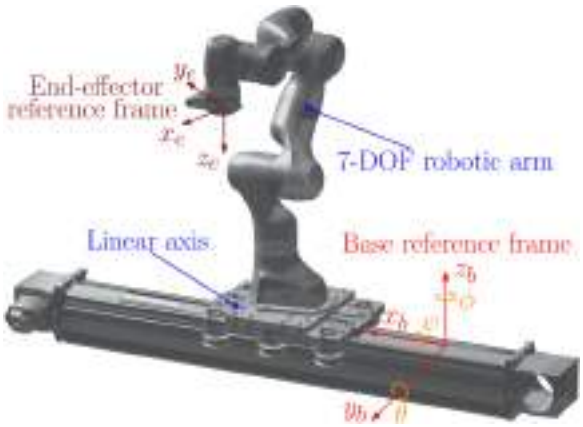

Fig. 1. Representation of the considered redundant robotic system with 8 DOFs.

Table 2
Nominal geometrical and dynamical parameters of the linear axis.

Parameter	Symbol	Value	Parameter	Symbol	Value
Motor inertia	J_m	$9.3 \cdot 10^{-5}$ kg m^2	Pulley radius	r	0.045 m
Gearbox inertia	J_g	$7.0 \cdot 10^{-5}$ kg m^2	Axis length	L	0.8 m
Pulley inertia	J_p	$2.1 \cdot 10^{-3}$ kg m^2		a_1	0.0 N
Cart mass	m_{cart}	18.69 kg	Friction	a_2	351.28 Ns/m
Arm mass	m_{arm}	17.80 kg	coefficients	a_3	197.35 N
Gear ratio	i_r	10		a_4	159.43

4. Simulation results

The performance of the proposed approach is evaluated on a 8-DOF robotic system, composed of a 7-DOF arm (Panda robot by Franka Emika GmbH) mounted on a linear axis, shown in Fig. 1. The robotic arm has an operational reach equal to 855 mm, whereas the total stroke of the cart is equal to 800 mm. The robot base (x_b, y_b, z_b) and end-effector (x_e, y_e, z_e) reference frames are shown in Fig. 1. The x-axis of the base reference frame is aligned with the linear axis, while its origin corresponds to the center of the cart when it is placed at the right end of the linear axis.

The joint torques and the energy consumption of the considered robotic system are computed considering the nominal dynamics parameters of the linear axis identified in [52] (Table 2), and the nominal dynamic model of the robotic arm identified in [53].

From the nominal dynamics parameters, bounds of ±2.5% for the center of mass, and ±5% for mass and inertia are applied to all components to consider uncertainties, as in [51,54]. We adopted the uncertainty bounds that have been considered for the same manipulator in [50], where a robust control approach was developed and tested. The accuracy of the conservative estimates may improve by implementing a dedicated identification procedure. A simple yet sufficiently overapproximative estimate of the friction model is considered for the interval arithmetic computations for both the linear axis and the manipulator joints, including only the Coulomb and viscous friction contributions. For the joint viscous and Coulomb friction coefficients, bounds of ±10% are applied for the robotic arm, whereas the bounds of the friction coefficients for the linear axis are retrieved from experimental data. Moreover, friction bounds modelling includes a coefficient that accounts for velocity uncertainty around zero, as in [55]. More details on the determination of the friction bounds of the linear axis are reported in Appendix A.

We verify the effectiveness of our proposed strategies S1 and S2 introduced in Section 3 with two tests each. The first one (Test 1) is a helical path comprising a rectangular and an elliptical part, repeated five times (named Test 1.1, 1.2, 1.3, 1.4, and 1.5) moving the path within the robot workspace, as it can be seen in Fig. 2. Dividing the trajectory of Test 1 into a rectangular and an elliptical part allows us to design a highly variable test trajectory, composed of linear and curved sections. Before performing each Test 1.k with $k \in [1, ..., 5]$, the robot starts from an initial joint configuration and reaches the starting point of the considered trajectory with a linear path (dashed lines in Fig. 2).

In contrast, in Test 2 we used 100 linear trajectories in Cartesian space (Fig. 3). The initial and final positions of each linear trajectory are randomly selected in the robot workspace for generalization purposes. The robot starts each linear trajectory with the same joint configuration for all compared approaches (the proposed S1 and S2 strategies and the reference SNS approach). The 100 random trajectories of Test 2 have been designed to exceed the workspace of the 7-DOF manipulator, so as to explore the 3D space that can be reached by also moving the linear axis. Please note that in the vicinity of boundaries of the workspace one can expect a reduction of the energy saving performance due to the limited possibility for the robot to beneficially exploit the degrees of redundancy.

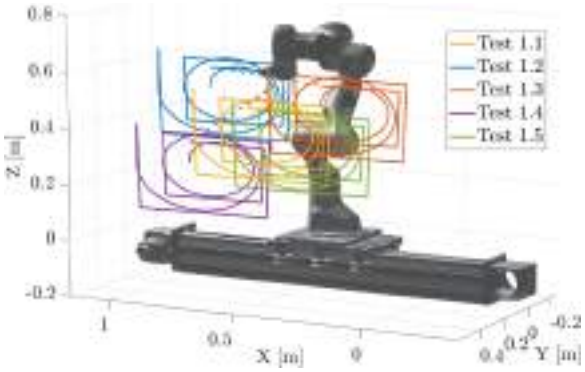

Fig. 2. Representation of the 3D path of the robot end-effector in the Cartesian space planned for the five repetitions of Test 1. The dashed lines represent the path of the robot end-effector from the initial joint configuration to the starting point of each trajectory.

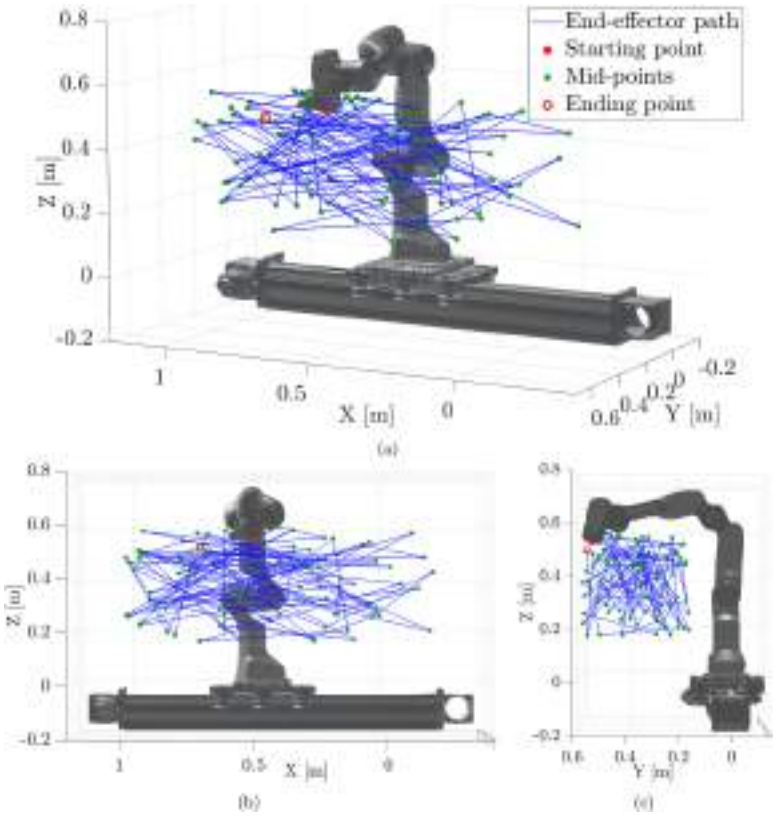

Fig. 3. Representation of the 3D path of the robot end-effector in the Cartesian space planned for Test 2: 3D (a), frontal (b), and lateral (c) views.

For both Tests 1 and 2, the end-effector orientation is changed during the trajectory to increase the motion variability. In particular, the rotation of the end-effector reference frame with respect to the base frame is expressed as roll-pitch-yaw (ψ, θ, ϕ) angles. With this convention, ψ, θ, and ϕ indicate the rotation about the x-, y-, and z-axis, respectively (see Fig. 1). In Test 1 the weighting matrix $\boldsymbol{W}$ is unique and constant during the trajectory and is computed as described in Section 3. In Test 2 $\boldsymbol{W}$ is optimized for each linear trajectory as described in Section 3. Finally, the original trajectory is sampled with a frequency of 1000 Hz for both tests. More details on the two considered test cases are reported in Appendix B.

For all compared approaches, the optimization problem in Eq. (6) with Eq. (8) is implemented in Matlab 2020b using *fmincon* and solved by means of the sequential quadratic programming (SQP) algorithm, using a computer running Windows 10 Pro with an Intel Core i5-10600K CPU @ 4.10 GHz and 32.0 GB of RAM. The maximum number of iterations is set equal to 500, and the maximum function evaluations to 5000. The SQP algorithm is an iterative, gradient-based method used to solve constrained, non-linear optimization problems. This optimization algorithm was chosen in preliminary tests for its limited computational times with respect

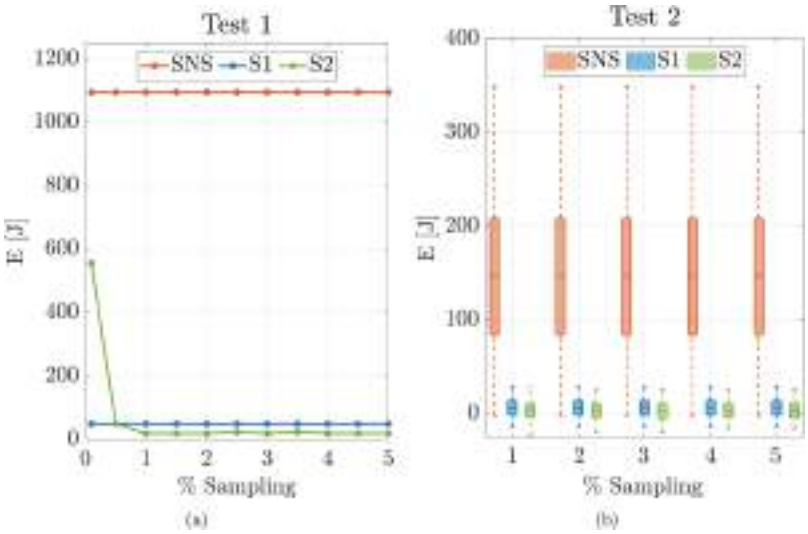

Fig. 4. Mean of the mechanical energy consumed with the three considered approaches during the five trajectories considered for Test 1 as a function of the sampling percentage (a). Boxplots of the mechanical energy consumption with the three considered approaches during the 100 point-to-point motions for Test 2 as a function of the sampling percentage (b).

to alternative solutions, as for instance, the interior-point method [56]. The $\boldsymbol{W}$ matrix is initialized using the weights determined with the heuristic trajectory-independent approach S1 ($\boldsymbol{W} = \boldsymbol{W}_{heur} = \text{diag}([974.4,\ 2.8,\ 2.4,\ 3.3,\ 3.3,\ 4.3,\ 1,\ 2.9])$). By defining a fixed initial guess we ensure the reproducibility of the results. In preliminary analysis we assessed that this initial condition yields to better results in most cases than a random choice of the initial weight matrix.

The end-effector pose error $\boldsymbol{e}$ in Eq. (13) is composed of two parts, a first one that considers the position error $\boldsymbol{e}_p$, and a second one that includes the orientation error $\boldsymbol{e}_o$. The position error $\boldsymbol{e}_p$ is simply derived from the difference between the desired and the actual positions. For the orientation part, we first convert the orientation from the roll-pitch-yaw representation to the axis-angle representation. Then, the orientation error $\boldsymbol{e}_o$ is obtained correspondingly as in [45]. This procedure allows us to avoid problems due to representation singularities for the orientation.

4.1. Results with nominal dynamics

For both the considered tests, we compute the mechanical energy required to complete the trajectories with the three considered approaches (the proposed S1 and S2, and the state-of-the-art SNS reference approach), taking into account different sampling percentages for S2. The results are shown in Fig. 4, where the energy efficiency benefits of the proposed strategies clearly appear with respect to the reference approach. Fig. 4a reports the mean mechanical energy consumption with the three compared approaches (SNS, S1, and S2) during the five trajectories of Test 1, as a function of the sampling percentage. Furthermore, Fig. 4b represents the boxplots of the mechanical energy consumed with the three considered approaches during the 100 point-to-point motions for Test 2, as a function of the sampling percentage. Considering Test 1, the adoption of S1 leads to a mean energy saving equal to 95.54% with respect to the reference approach for the five trajectories of Test 1. A further energy expenditure reduction can be obtained by considering S2. Taking into account a sampling percentage of 2%, the mean energy saving is equal to 60.45% with respect to the S1 strategy, and to 98.24% with respect to the reference approach.

Similar results are obtained in Test 2. In this case, the mean energy consumption reduction during the 100 point-to-point motions with S1 is equal to 95.95% with respect to adopting the basic SNS algorithm. Taking into account S2 and a sampling percentage of 2%, the mean energy saving is equal to 58.49% with respect to the S1 strategy, and to 98.32% with respect to the reference approach. Furthermore, Fig. 4 shows that S2 consistently provides better results with respect to S1 for sampling percentages of 1% and above. Therefore, a sampling percentage equal to 2% is considered next in S2.

It is important to mention that the strategy S2 requires the computation of the weight matrix before each new trajectory is executed. The computational time that is required depends on the percentage p used in Eq. (12) and the specific trajectory. We have performed extensive simulations to estimate the computational cost of this strategy by using the 100 trajectories of Test 2 and varying the value of the parameter p. The system on which we run the optimizations is described in Section 4. The results are collected in Fig. 5 in terms of boxplot of the computational time. These results show the practicability of the approach, as the maximum computation time we experienced is less than 130 seconds for a value of $p = 2\%$, which does not introduce significant deterioration of the performance, as shown in Fig. 4b.

On the other hand, strategy S1 can also be applied when online redundancy resolution is required (e.g., in visual-servoing scenarios), since the weighting matrix is independent of the trajectory and needs to be computed only once for the specific robotic system. Under comparable conditions, the online computational cost of strategy S1 with respect to the number of degrees of freedom is

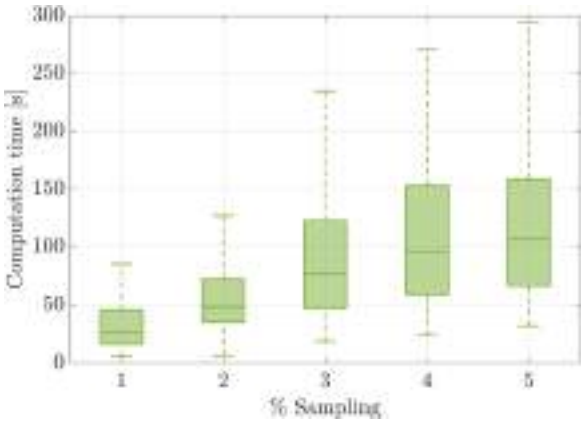

Fig. 5. Boxplot of the computational time required for optimizing the weights for S2 for each one of the 100 trajectories of Test 2, as a function of the sampling percentage.

dominated by the pseudoinverse of the Jacobian matrix $\boldsymbol{J}$. This operation has complexity $O(RN^2)$ [57], where R is the dimension of the task, and N the number of degrees of freedom of the robotic system ($R < N$ in our considered cases). However, this cost is the same for all Jacobian-pseudoinverse-based schemes. In addition to these costs, when considering strategy S2 instead, the computational cost as a function of the DOFs increases with the number of parameters of the optimization, i.e., the weights of $\boldsymbol{W}$ to be optimized, which is one for each degree of freedom. In that case the additional offline computational burden depends on the specific algorithm chosen for the optimization.

4.2. Results with uncertain dynamics

The inclusion of the robot dynamics uncertainty in the optimization ensures a safety margin in the trajectory computation, complying with torque limits and minimizing the energy consumption in the worst-case scenario. In the following, for the linear axis we consider the angular position and velocity of the driving pulley and its corresponding torque. Fig. 6 shows the joint positions, velocities and torques for Test 1.1 with the three compared approaches, whereas Fig. 7 shows the Euclidean norm of joint velocities, torques, and power for the same trajectories. These figures highlight that the proposed strategies S1 and S2 reduce the demanded joint velocities with respect to the SNS reference approach. More specifically, the motion of the linear axis is significantly reduced with the two proposed approaches with respect to the SNS strategy (Fig. 6a). Moreover, the torque required with the S2 approach is almost always lower than the one needed by the SNS and S1 strategies, as it can be seen from Fig. 7. Therefore, the combination of reduced velocity and torque leads to a lower mechanical power and consequently a reduced energy consumption at equal motion of the robot end-effector.

The differences in the robot motion for Test 1.1 among the three considered approaches are also visible in Fig. 8, which depicts the robot configurations at various time instants. Fig. 9 shows the mechanical energy consumed over time for Test 1.1 with the three considered approaches. The plot reported in Fig. 9 accounts for uncertainties of dynamics parameters (including friction uncertainties) in the shaded area. This can also be considered as an estimate of the sensitivity of the energetic savings with respect to uncertain dynamics parameters. From Fig. 9, it can be seen that the proposed approaches always require less energy than the reference one.

5. Experimental setup

The results obtained from the simulations are experimentally validated on a real 8-DOF robotic system composed of a Panda arm by Franka Emika GmbH mounted on a linear axis, available at the Robotics and Mechatronics Laboratory of the University of Udine (Italy) (Fig. 10). The robotic arm exhibits a pose repeatability of ± 0.1 mm, a payload capacity of 3 kg, and a maximum reach equal to 855 mm. The linear axis adopts a toothed belt made of polyurethane with steel strands for moving the cart. The system is actuated by a brushless motor HDT SR08L, connected to the driven pulley of the linear axis through a planetary gearbox with reduction ratio equal to 10.

The considered robotic system is controlled with a computer equipped with 32 GB RAM and an Intel Core i9 processor running Ubuntu 20.04. The linear axis is driven from a Docker container using ROS 2 (Robot Operating System) Galactic Geochelone, whereas another Docker container with ROS Noetic Ninjemys is adopted for controlling the manipulator. Both containers are implemented

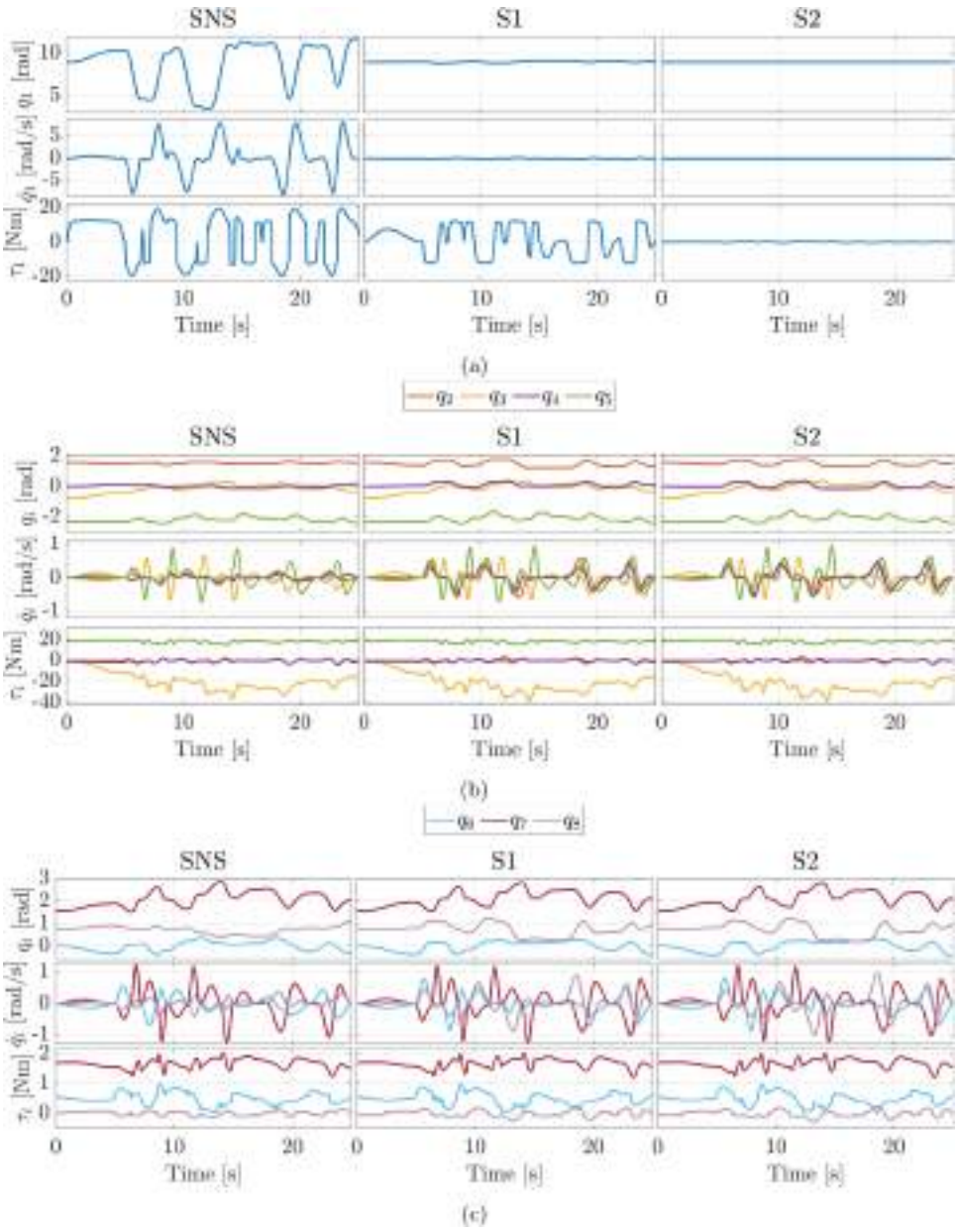

Fig. 6. Joint positions, velocities and torques for Test 1.1 with the three considered approaches, taking into account the uncertain dynamics and a sampling percentage equal to 2% for S2.

in Python 3.8, and the communication between them is performed using a shared network. The *ros1_bridge* package is used for transmitting messages between the two different ROS versions. Furthermore, the motion of the linear axis and the robotic arm is defined and synchronized in the ROS 2 environment with a sampling rate of 100 Hz.

6. Experimental results

The reference and optimized trajectories of Test 1.1 and Test 2 obtained in simulation with uncertain dynamics (Section 4.2) are executed on the real robotic system. Fig. 11 shows the desired and the experimental joint positions and velocities for Test 1.1 with the three considered approaches. From the figure it can be seen that the experimental trajectories accurately follow the desired ones for all the considered motions. Fig. 12 depicts the experimental joint torques of the first five joints for the same trajectory. We report only the joint torques of the first five joints since they provide a consistently higher torque with respect to the last three. The joint torque bounds computed in the simulations are represented as shaded areas. Fig. 12 highlights that the experimental friction

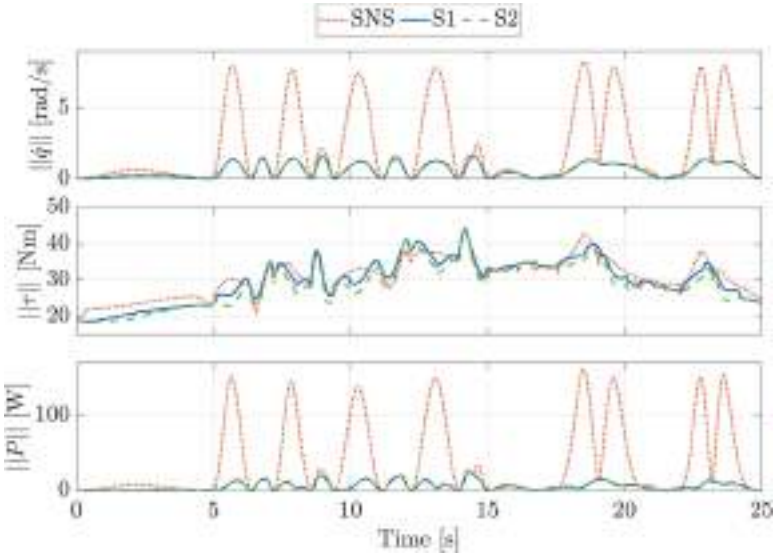

Fig. 7. Euclidean norm of joint velocities, torques, and power for Test 1.1 with the three considered approaches, taking into account the uncertain dynamics and a sampling percentage equal to 2% for S2.

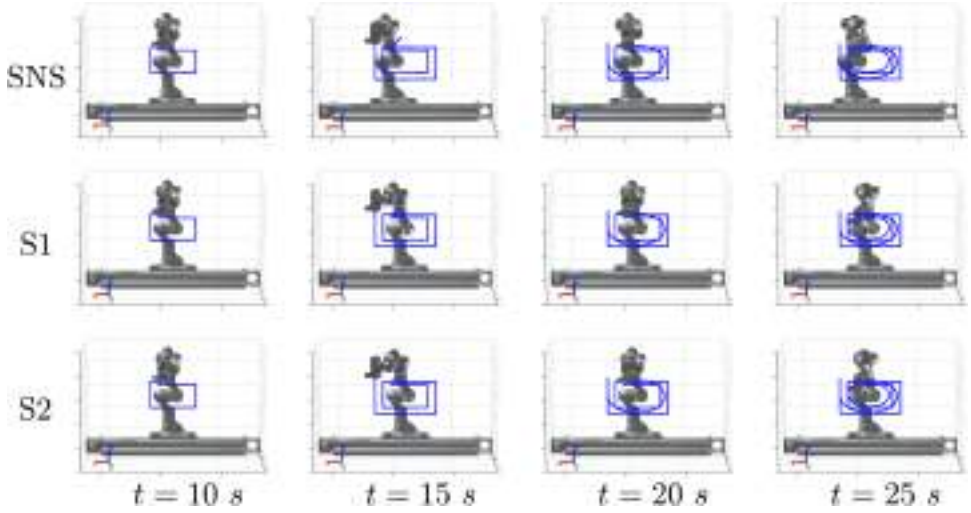

Fig. 8. Robot configurations at different time instants during Test 1.1 for the three considered approaches.

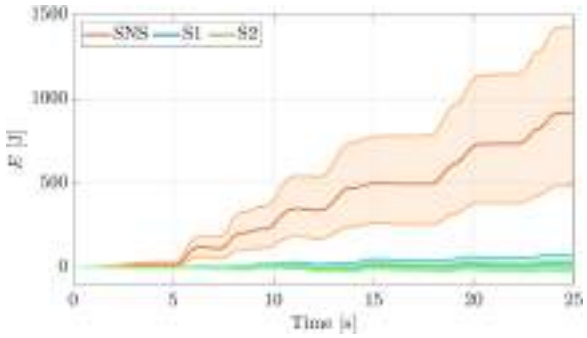

Fig. 9. Mechanical energy consumed for Test 1.1 with the three considered approaches, taking into account the uncertain dynamics and a sampling percentage equal to 2% for S2. The energy consumption bounds are represented as shaded areas.

Fig. 10. Considered redundant robotic system.

Table 3
Energy consumption for the three considered approaches during Test 1.1 in the numerical (E_{num}) and experimental (E_{exp}) trials; the lower and upper limits for the energy consumption in the numerical tests are computed accounting for the uncertain dynamics pct_{num} and pct_{exp} indicate the percentage energy saving of the two proposed approaches (S1 and S2) with respect to the reference one (SNS), for the numerical and experimental tests, respectively; $\Delta E_{|num-exp|}$ is the absolute value of the difference between the energy consumed in the numerical and experimental tests.

	E_{num} [J]	pct_{num} [%]	E_{exp} [J]	pct_{exp} [%]	$\Delta E_{\|num-exp\|}$ [J]
SNS	$917.2^{+509.7}_{-438.6}$	–	874.0	–	43.2
S1	$22.1^{+47.5}_{-43.6}$	97.6	21.5	97.5	0.6
S2	$6.5^{+35.4}_{-35.4}$	99.3	6.8	99.2	0.3

Table 4
Energy consumption for the three considered approaches during Test 2 in the numerical (E_{num}) and experimental (E_{exp}) trials; the lower and upper limits for the energy consumption in the numerical tests are computed accounting for the uncertain dynamics pct_{num} and pct_{exp} indicate the percentage energy saving of the two proposed approaches (S1 and S2) with respect to the reference one (SNS), for the numerical and experimental tests, respectively; $\Delta E_{|num-exp|}$ is the absolute value of the difference between the energy consumed in the numerical and experimental tests.

	E_{num} [J]	pct_{num} [%]	E_{exp} [J]	pct_{exp} [%]	$\Delta E_{\|num-exp\|}$
SNS	$13731.8^{+6468.7}_{-6177.7}$	-	12941.3	-	790.5
S1	$871.6^{+1049.3}_{-928.1}$	93.7	912.5	92.9	40.9
S2	$345.5^{+848.4}_{-790.6}$	97.5	361.7	97.2	16.2

torques are always within their limits, highlighting the performance of the proposed approach in minimizing energy consumption while respecting the joint limits even in the worst-case scenario. To better appreciate the experimental tests, a video is available as supplementary material and online.[1] Table 3 reports the energy consumption for the three approaches considered during Test 1.1 in the numerical (with bounds accounting for the uncertain dynamics) and experimental tests, the corresponding percentage reductions with respect to the reference case, and the absolute value of the difference between the energy consumed in the numerical and experimental tests.

Table 3 also reports the mechanical energy consumption for the three considered approaches during Test 1.1, both from the simulation and experimental tests, and the percentage energy saving of the two proposed approaches (S1 and S2) with respect to the reference one. From the table it can be noticed that the energy consumption reduction with respect to the reference approach is up 99.3% when S2 is considered in the numerical simulation, whereas with S2 it reaches to 99.2% in the experimental tests. Moreover, Table 3 indicates the lower and upper limits for the energy expenditure computed in simulation, which are obtained accounting for the uncertain dynamics. As it can be seen, the energy consumed during the experiments is inside the bounds obtained from the simulations for all the three considered approaches. This verifies that the proposed approach accounts for deviations of the robot

[1] https://www.youtube.com/watch?v=EyanlYu6mz0

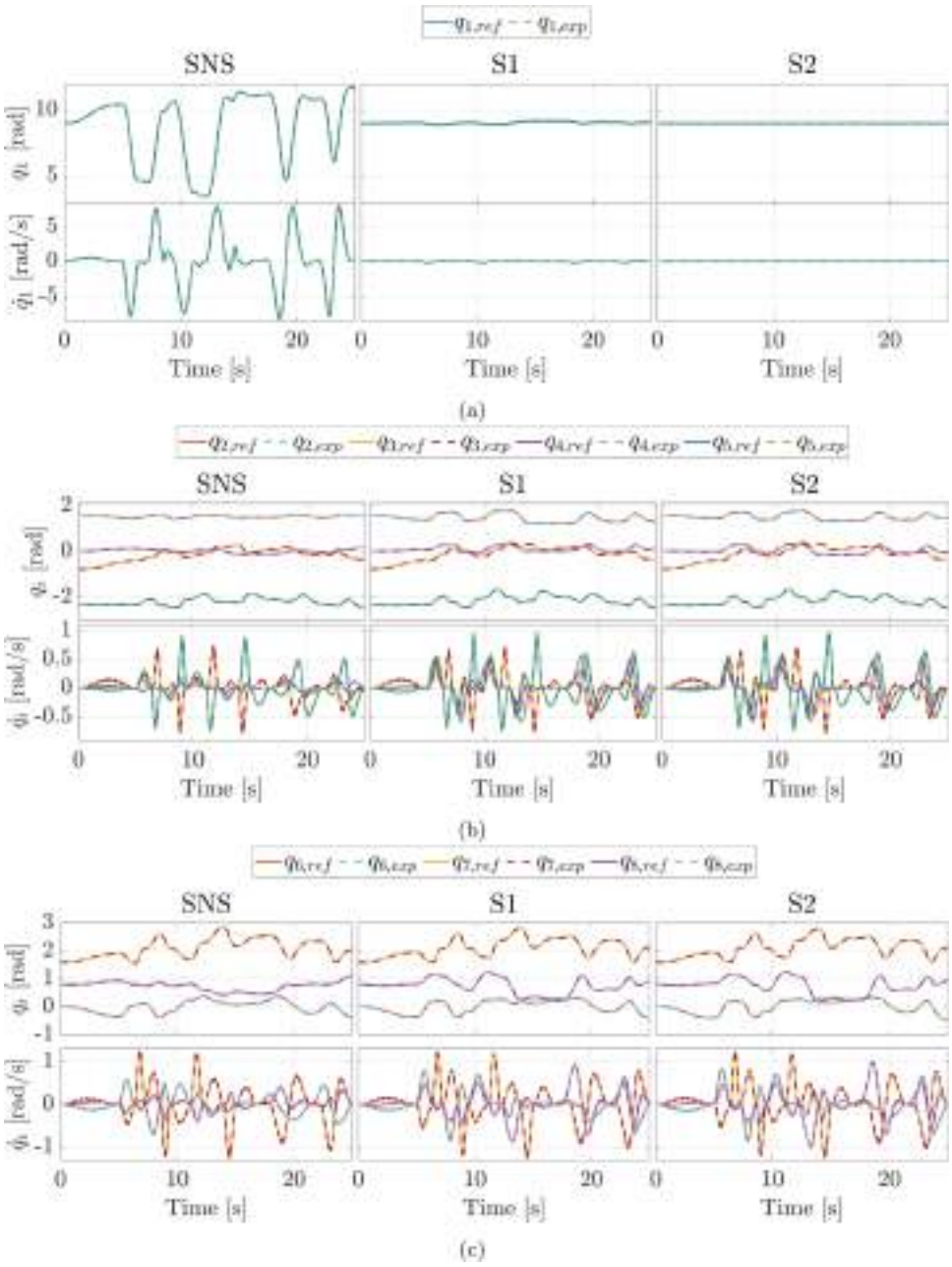

Fig. 11. Desired and experimental joint positions and velocities for Test 1.1 with the three considered approaches, taking into account the uncertain dynamics and a sampling percentage equal to 2% for S2.

dynamics parameters from the nominal ones. Finally, Table 3 compares the values of the energy consumed in the numerical and experimental tests. The results show a difference lower than 5% between the two tests.

Fig. 13 shows the boxplots of the root-mean-square (*rms*) value of the joint velocities during the 100 point-to-point motions for the three considered approaches, whereas Fig. 14 reports the boxplots of the *rms* value of the joint torques for the same trajectories. From Figs. 13 and 14 it can be seen that both velocity and torque of the linear axis decrease consistently moving from SNS to S1 and S2, while the joint velocities and torques of the robotic manipulator do not change appreciably. This trend is reflected on the required mechanical power and consequently on the mechanical energy consumption, shown in Figs. 15 and 16, respectively. More in detail, Fig. 15 depicts the boxplots of the *rms* value of the joint mechanical power during the 100 point-to-point motions for the three considered approaches, whereas Fig. 16 reports the boxplots of the mechanical energy consumed by the different joints during the same trajectories. It can be noted that in the reference approach the linear axis is responsible for the largest amount of the total

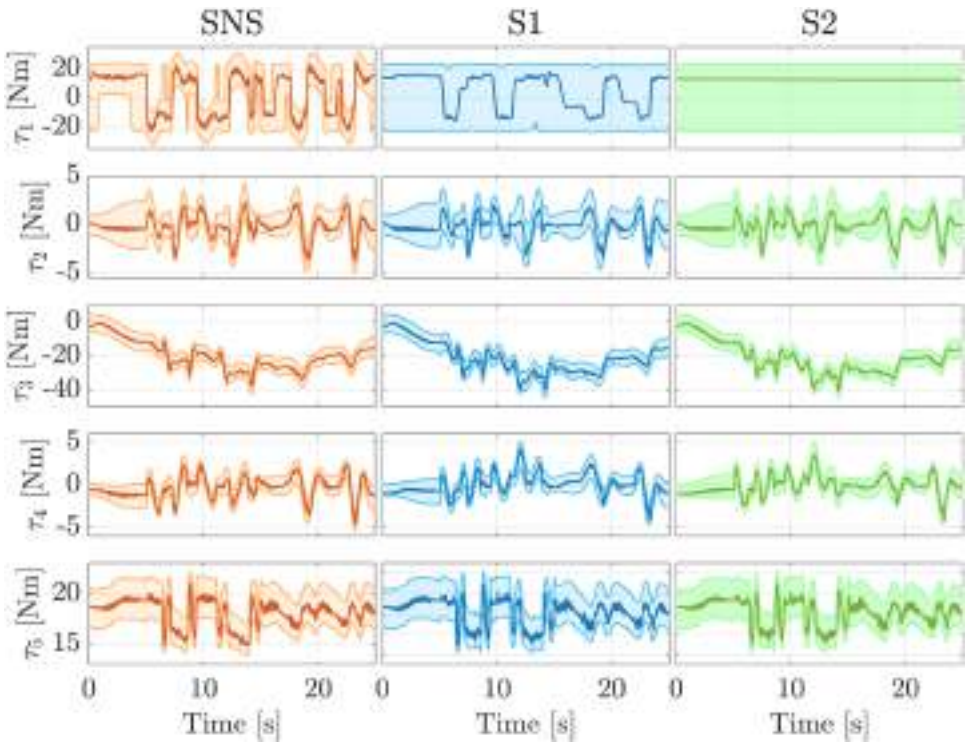

Fig. 12. Experimental joint torques for Test 1.1 with the three considered approaches. Uncertain torque bounds are represented as shaded areas.

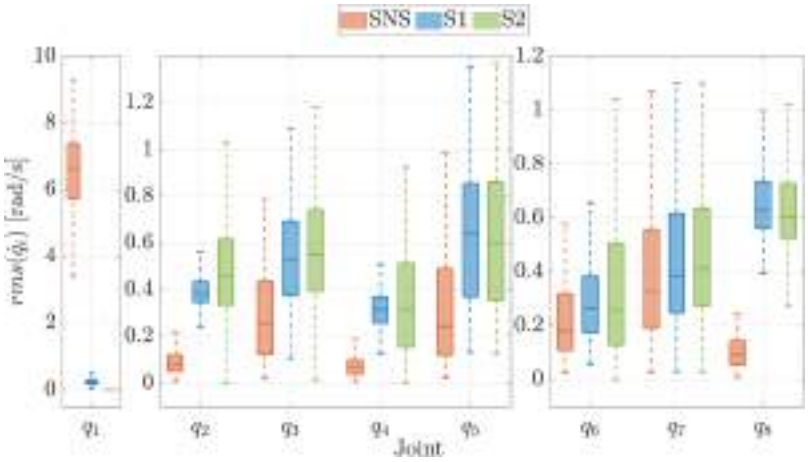

Fig. 13. Boxplots of the *rms* value of the joint velocities during the 100 point-to-point motions for the three considered approaches.

energy needed to execute the 100 point-to-point trajectories, limiting the energy required by the other joints. Differently, the proposed approaches (S1 and S2) are able to redistribute the required motion needed by different joints to keep the end-effector on the desired trajectory, consistently decreasing the energy required by the linear axis while slightly increasing the energy consumption of the other joints.

Table 4 shows the mechanical energy consumption for the three considered approaches during all the 100 trajectories of Test 2, both from the simulation and experimental tests, and the percentage energy saving of the two proposed approaches (S1 and S2) with respect to the reference one (SNS). In Test 2, the energy saving with respect to the reference approach obtained by considering the S2 strategy is equal to 97.5%. Regarding the experimental tests, the adoption of S2 leads to a energy consumption reduction of 97.2% with respect to the reference approach. Furthermore, Table 4 reports the lower and upper bounds of the energy consumed in the numerical simulations, which are computed taking into account the uncertain dynamics. Similarly to the results obtained for Test 1.1, from Table 4 it can be noted that the energy needed to complete all the trajectories in the experiments is still within the limits obtained from the simulations for all the three considered approaches. Finally, Table 4 compares the values of the energy consumed

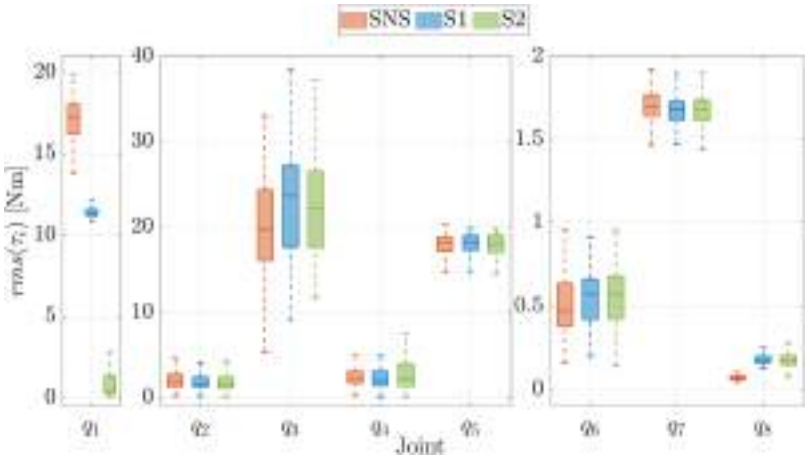

Fig. 14. Boxplots of the *rms* value of the joint torques during the 100 point-to-point motions for the three considered approaches.

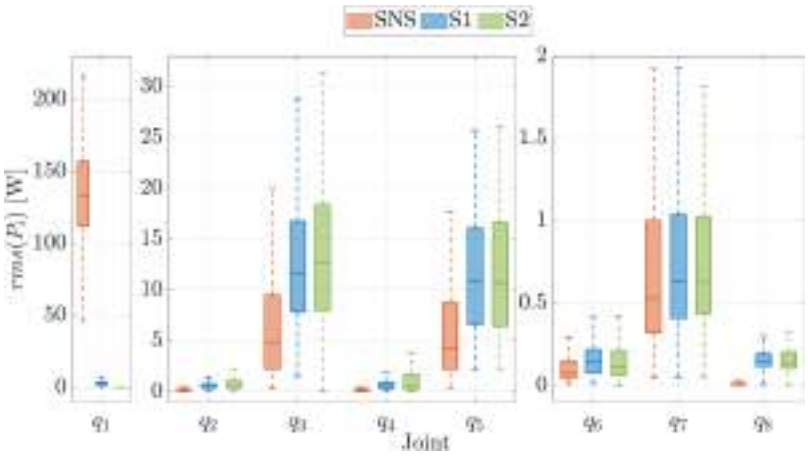

Fig. 15. Boxplots of the *rms* value of the joint mechanical power during the 100 point-to-point motions for the three considered approaches.

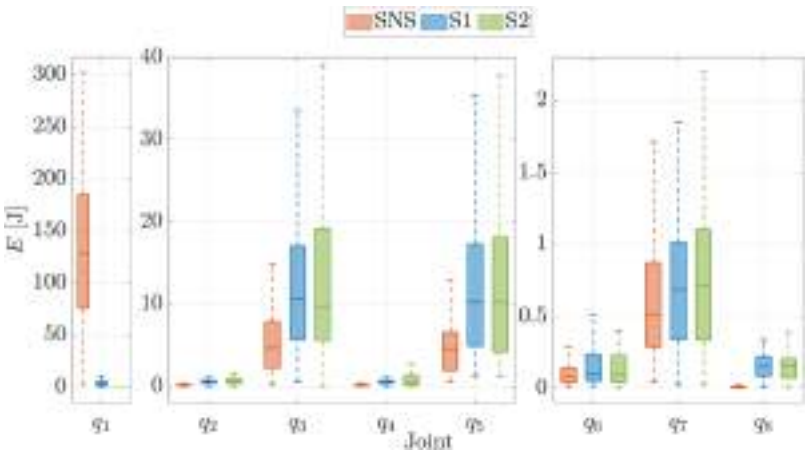

Fig. 16. Boxplots of the mechanical energy consumed by the different joints during the 100 point-to-point motions for the three considered approaches.

Table 5
Energy consumption for the three considered approaches during Test 2 in the numerical (E_{num}) and experimental (E_{exp}) trials obtained by considering a non-regenerative robotic system; the lower and upper limits for the energy consumption in the numerical tests are computed accounting for the uncertain dynamics pct_{num} and pct_{exp} indicate the percentage energy saving of the two proposed approaches (S1 and S2) with respect to the reference one (SNS), for the numerical and experimental tests, respectively; $\Delta E_{|num-exp|}$ is the absolute value of the difference between the energy consumed in the numerical and experimental tests.

	E_{num} [J]	pct_{num} [%]	E_{exp} [J]	pct_{exp} [%]	$\Delta E_{\|num-exp\|}$
SNS	$15836.6^{+5908.7}_{-6279.5}$	-	14375.8	-	1460.8
S1	$6460.2^{+2847.8}_{-2437.3}$	59.2	6127.6	57.4	332.6
S2	$4195.4^{+1387.8}_{-1185.8}$	73.5	4372.2	69.6	176.8

in the numerical and experimental tests. The results show a difference of about 5% between the two tests, which is consistent with the result obtained numerically.

To check the statistical differences between the three considered approaches (SNS, S1, and S2), we first tested the normality of the consumed energy data using the Kolmogorov-Smirnov test [58]. Then, since all three sets of data are found to be non-normally distributed, the Kruskal-Wallis H-test is applied to check statistical differences [59]. A p-value < 0.05 is obtained in the comparison between the reference (SNS) and each one of the two proposed (S1 and S2) approaches, demonstrating the significant difference between the baseline and the proposed solutions.

Finally, we assessed the energy consumption reduction obtained if the considered robotic system is not able to recover energy during braking phases. For this evaluation, as cost function we considered Eq. (9) in place of Eq. (7). Table 5 reports the energy consumption for the three considered approaches during all the 100 trajectories of Test 2, both from the simulation and experimental tests, and the percentage energy saving of the two proposed approaches (S1 and S2) with respect to the reference one (SNS). These results highlight the energy saving capabilities of the proposed approaches even applied to non-regenerative robotic systems, obtaining a percentage energy consumption reduction up to 73.5% in the numerical tests, and up to 69.6% in the experiments.

Our approach targets directly the mechanical energy expenditure of redundant robots. However, we also evaluated its benefits when considering the electrical energy required to power the system for a more comprehensive and intuitive overall evaluation of energy-reduction effectiveness. To this end, we measured the electrical power consumption of the robotic system for 144 repetitions of the trajectories of Test 1.1. A Nanovip Plus power and harmonic analyzer by Elcontrol Energy was used for the power measurement starting from voltage and current signals of the Franka robot and the linear axis together. The electrical energy consumption of the whole 8-DOF robotic system when executing the trajectory planned with the SNS approach results equal to $0.78\ kWh$, with the S1 approach to $0.72\ kWh$, and with the S2 approach to $0.70\ kWh$. For completeness, we also measured the idle energy expenditure to be $0.66\ kWh$ for the same amount of time of the above-mentioned tests with the robot standing still when brakes are disabled in the starting configuration of Test 1.1 (see Fig. 2). The results show that the proposed approach can provide a significant improvement also when considering the overall electrical energy. Further electrical energy savings may be achieved by accounting for the idle losses and the electrical efficiency of the drive.

7. Conclusions

In this paper, a novel approach has been presented to enhance energy efficiency in redundant robotic systems by optimizing the weighting matrix used to solve the inverse kinematics through the weighted Jacobian pseudo-inverse. Two different solutions for computing the energy-efficient weights have been proposed, one independent and the other dependent on the desired robot end-effector trajectory. The proposed approach also considers uncertainties in the robot dynamics parameters. The performance of the proposed approach has been validated on a 8-DOF robotic system. Extensive numerical simulations and experimental tests have been performed to check the feasibility of the proposed approach and quantify the actual energy savings with the optimized joint motion profiles. The results highlight the energy saving capabilities of the proposed approach with respect to a state-of-the-art approach based on the saturation in the null space, obtaining an energy consumption reduction up to 97.5% in the numerical tests, and up to 97.2% in the experiments. The results demonstrate the effectiveness of the proposed approach in reducing the energy requirement while meeting kinematics and dynamics joint limits, even in the case of imperfect knowledge of the robot dynamics parameters.

The proposed approach is general and can also be applied to redundant robotic systems with a different number of DOFs and degrees of redundancy. Furthermore, the proposed strategy can be adopted in several industrial tasks where the motion of the end-effector is determined (e.g., finishing, spray painting, welding), allowing for more efficient and sustainable manufacturing processes.

Direct extensions of this work will include the development of a procedure for identifying the bounds for the dynamics parameters, as well as the consideration of electrical energy contributions for the optimization. Furthermore, an interesting direction for future development, which requires reformulating the system dynamics, constraints, and weighting matrices, may consider the extension to mechanisms with closed chains, non-holonomic mobile manipulators, and soft robotic systems.

CRediT authorship contribution statement

Giuliano Fabris: Writing – original draft, Software, Methodology, Investigation, Data curation; **Lorenzo Scalera:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Conceptualization; **Andrea Giusti:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization; **Alessandro Gasparetto:** Writing – review & editing, Supervision, Project administration.

Data availability

Data will be made available on request.

Funding

The first author acknowledges the support from the European Social Fund Plus of the Autonomous Region of Friuli Venezia Giulia, Italy (FVG FSE+ Regional Programme 2021-2027, PPO 2023, Specific Programme No. 22/23, Support for advanced training within the regional university system, Line A, CUP G23C23001130008). This work was developed within the Laboratory for Big Data, IoT, Cyber Security (LABIC) funded by the Friuli Venezia Giulia region (Italy), and the Laboratory for AI for Human-Robot Collaboration (AI4HRC) funded by Fondazione Friuli (Italy). This study was carried out within the Interconnected Nord-Est Innovation Ecosystem and received funding from the European Union Next-GenerationEU (National Recovery and Resilience Plan, Mission 4, Component 2, Investment 1.5, D.D. 1058 23/06/2022, ECS00000043). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We would like to thank the anonymous Reviewers for the constructive comments that helped us to improve this manuscript. We also thank prof. Sandro Calligaro and prof. Roberto Petrella for their help with the electrical power measurements, and Dr. Federico Lozer and Mr. Franci Rrapi for their assistance with the experimental tests.

Supplementary material

Supplementary material associated with this article can be found in the online version at 10.1016/j.mechmachtheory.2026.106477.

Appendix A. Optimization of the friction bounds of the linear axis

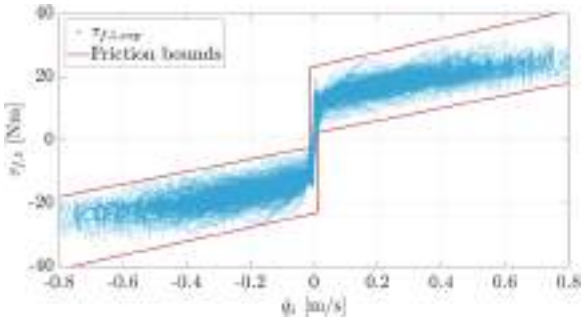

Fig. A.1. Experimental friction torque and computed friction bounds for the linear axis.

The formulation of the friction model adopted to describe friction effects in the linear axis is the following:[2]

$$[\tau_f(\dot{q}, \boldsymbol{\Delta})] = [F_c] \odot \text{sign}(\dot{q} \ominus [F_s]) \oplus [F_v] \odot \dot{q}$$

[2] Given $[x] \in \mathbb{R}$ and $[y] \in \mathbb{R}$ (sets of scalar intervals), the result of the binary operations $* \in \{+, -, \cdot\}$ is defined as $[x] \circledast [y] := \{x * y \mid x \in [x], y \in [y]\}$ [46].

where $[F_c] = [F_{c,low}, F_{c,up}]$ is the interval that accounts for Coulomb friction torque uncertainties, $[F_s] = [-F_s, F_s]$ is an interval that accounts for velocity uncertainty near zero, and $[F_v] = [F_{v,low}, F_{v,up}]$ is the interval that accounts for viscous friction uncertainties. The considered friction formulation with bounded parameters is a simple model that allows the inclusion of intervals in order to create bounds that embrace the experimental data. Alternative friction formulations can be found in [52,60]. However, these consider nominal friction parameters only.

The friction bounds for the linear axis are defined to minimize the enclosed area A_{bounds} while still containing 99.9% of the experimental data D. The experimental data consist of pairs $(\dot{q}_1, \tau_{f,1,exp})$, where $\dot{q}_1$ is the linear axis velocity, whereas $\tau_{f,1,exp}$ represents the associated experimental friction torque. The optimization problem to define the friction bounds of the linear axis is formulated as follows:

$$\min_{F_{c,low}, F_{c,up}, F_{v,low}, F_{v,up}, F_s} A_{bounds} \tag{A.1}$$

subject to:

$$0.999 \cdot n(D) \in A_{bounds} \text{ where } D = \{(\dot{q}_1, \tau_{f,1,exp})\} \tag{A.2}$$

The experimental friction torque and the computed friction bounds for the linear axis are shown in Fig. A.1.

Appendix B. Planning of the test trajectories

In this appendix, a detailed description of the trajectories planned for Tests 1 and 2 is reported. As mentioned in Section 4, the trajectory considered for Test 1 is a helical path comprising two parts, as it can be seen in Fig. 2. The first part is composed of linear rest-to-rest motions that describe a rectangle parallel to the xz plane, which size progressively increases while it is moved along the y-axis. These motions are executed adopting a fifth-degree polynomial motion law. In the second section of Test 1 the helix is elliptical, which width and height are progressively reduced while it is shifted in the opposite direction to the y-axis. The fifth-degree polynomial motion law is also considered for the second part of Test 1. The total duration of the trajectory is set equal to 20 s. Furthermore, the linear motion from the initial joint configuration to the starting point of the trajectory is executed in a total time equal to 5 s, using a fifth-degree polynomial motion law. In Table B.1, the initial joint configuration, the starting points, and the initial and final dimensions of both the rectangle and the ellipse are reported.

Table B.1
Parameters considered for the trajectories of Test 1.

Parameter	Symbol	Value	Unit
Total execution time	T	20	s
Initial rectangle dimensions	$\Delta x_i \times \Delta z_i$	0.250×0.150	m
Final rectangle dimensions	$\Delta x_f \times \Delta z_f$	0.500×0.300	m
Initial ellipse dimensions	$2a_i \times 2b_i$	0.500×0.300	m
Final ellipse dimensions	$2a_f \times 2b_f$	0.250×0.150	m
Movement along y-axis	Δy	0.150	m
Helical turns	H	2	-
ψ angle limits	ψ_{min}, ψ_{max}	$\frac{11}{12}\pi, \frac{13}{12}\pi$	rad
θ angle limits	$\theta_{min}, \theta_{max}$	$-\frac{\pi}{12}, +\frac{\pi}{12}$	rad
ϕ angle	ϕ	$\frac{\pi}{4}$	rad
1st starting point	$x_{s,1}, y_{s,1}, z_{s,1}$	0.525, 0.425, 0.500	m
2nd starting point	$x_{s,2}, y_{s,2}, z_{s,2}$	0.800, 0.200, 0.600	m
3rd starting point	$x_{s,3}, y_{s,3}, z_{s,3}$	0.300, 0.300, 0.575	m
4th starting point	$x_{s,4}, y_{s,4}, z_{s,4}$	0.800, 0.350, 0.350	m
5th starting point	$x_{s,5}, y_{s,5}, z_{s,5}$	0.425, 0.325, 0.450	m
Initial joint configuration	$\boldsymbol{q}_{start}$	$\left[0.4, \frac{\pi}{2}, -\frac{\pi}{4}, 0, -\frac{3}{4}\pi, 0, \frac{\pi}{2}, \frac{\pi}{4}\right]^T$	m, rad

For Test 2, we randomly selected 101 waypoints in the robot workspace, connecting them with linear paths in the Cartesian space (Fig. 3). These 100 trajectories are performed using fifth-degree polynomial motion laws. Moreover, the execution time is chosen as the maximum between the time needed to reach a maximum linear speed of 0.85 m/s and the time required to obtain a maximum linear acceleration of 6.5 m/s^2.

References

[1] A. Brunello, G. Fabris, A. Gasparetto, A. Montanari, N. Saccomanno, L. Scalera, A survey on recent trends in robotics and artificial intelligence in the furniture industry, Robot. Computer-Integrated Manuf. 93 (2025) 102920.
[2] V. De Simone, V. Di Pasquale, V. Giubileo, S. Miranda, Human-robot collaboration: an analysis of worker's performance, Procedia Comput. Sci. 200 (2022) 1540–1549.
[3] E.-Z. Wang, C.-C. Lee, Y. Li, Assessing the impact of industrial robots on manufacturing energy intensity in 38 countries, Energy Econ. 105 (2022) 105748.
[4] J. Petersen, A. Nourmohammadi, M. Fathi, M. Ghobakhloo, M. Tavana, Line balancing for energy efficiency in production: a qualitative and quantitative literature analysis, Comput. Industr. Eng. (2025) 111144.
[5] P. Khalaf, H. Richter, Trajectory optimization of robots with regenerative drive systems: Numerical and experimental results, IEEE Trans. Robot. 36 (2) (2019) 501–516.

[6] M. Althoff, A. Giusti, S.B. Liu, A. Pereira, Effortless creation of safe robots from modules through self-programming and self-verification, Sci. Robot. 4 (31) (2019) eaaw1924.
[7] J. Heredia, R.J. Kirschner, C. Schlette, S. Abdolshah, S. Haddadin, M.B. Kjægaard, ECDP: energy consumption disaggregation pipeline for energy optimization in lightweight robots, IEEE Robot. Autom. Lett. 8 (10) (2023) 6107–6114.
[8] P. Boscariol, D. Richiedei, Energy optimal design of servo-actuated systems: a concurrent approach based on scaling rules, Renew. Sustain. Energy Rev. 156 (2022) 111923.
[9] L. Scalera, G. Carabin, R. Vidoni, T. Wongratanaphisan, et al., Energy efficiency in a 4-DOF parallel robot featuring compliant elements, Int. J. Mechan. Control 20 (02) (2019) 49–57.
[10] F. Vidussi, P. Boscariol, L. Scalera, A. Gasparetto, Local and trajectory-based indexes for task-related energetic performance optimization of robotic manipulators, J. Mechan. Robot. 13 (2) (2021) 021018.
[11] G. Fabris, L. Scalera, A. Gasparetto, Dynamic modelling and energy-efficiency optimization in a 3-DOF parallel robot, Int. J. Adv. Manuf. Technol. 132 (5) (2024) 2677–2699.
[12] S. Ghungrad, A. Haghighi, Three-dimensional spatial energy-quality map construction for optimal robot placement in multi-robot additive manufacturing, Robot. Computer-Integrated Manuf. 88 (2024) 102735.
[13] X. Li, Y. Lan, P. Jiang, H. Cao, J. Zhou, An efficient computation for energy optimization of robot trajectory, IEEE Trans. Industr. Electr. 69 (11) (2021) 11436–11446.
[14] P. Jiang, Z. Wang, X. Li, X.V. Wang, B. Yang, J. Zheng, Energy consumption prediction and optimization of industrial robots based on LSTM, J. Manuf. Syst. 70 (2023) 137–148.
[15] Z. Liu, Y. Huang, D. Liu, X. Guo, K. Wang, J. Tan, Trajectory planning of large redundant manipulator considering kinematic constraints and energy efficiency, Robotica 41 (11) (2023) 3524–3540.
[16] A.T. Elgohr, H.A. Khater, M.A.A. Mousa, Trajectory optimization for 6 DOF robotic arm using WOA, GA, and novel WGA techniques, Res. Eng. 25 (2025) 104511.
[17] M. Soori, B. Arezoo, R. Dastres, Optimization of energy consumption in industrial robots, a review, Cognit. Robot. 3 (2023) 142–157.
[18] R. De Laet, N. Van Oosterwyck, L. Scalera, A. Cuyt, A. Gasparetto, S. Derammelaere, Energy-efficient motion planning for robotic systems using polynomials in the Chebyshev basis, Robot. Auton. Syst. (2025) 105051.
[19] H. Mohamad, S. Ozgoli, Online and multi-objective trajectory planner for robotic systems, Disc. Appl. Sci. 7 (1) (2025) 53.
[20] Y. Fang, C. Gu, Y. Zhao, W. Wang, X. Guan, Smooth trajectory generation for industrial machines and robots based on high-order S-curve profiles, Mechan. Mach. Theory 201 (2024) 105747.
[21] A.A. Maciejewski, B. Xie, Redundant robots, in: Robotics Goes MOOC: Design, Springer, 2025, pp. 39–87.
[22] A. Albu-Schäffer, A. Sachtler, Redundancy resolution at position level, IEEE Trans. Robot. 39 (6) (2023) 4240–4261.
[23] M. Fabregat-Jaén, A. Peidró, M. Colombo, P. Rocco, Ó. Reinoso, Topological and spatial analysis of self-motion manifolds for global redundancy resolution in kinematically redundant robots, Mechan. Mach. Theory 210 (2025) 106020.
[24] M.N. Vu, F. Beck, M. Schwegel, C. Hartl-Nesic, A. Nguyen, A. Kugi, Machine learning-based framework for optimally solving the analytical inverse kinematics for redundant manipulators, Mechatronics 91 (2023) 102970.
[25] T. Hong, W. Li, K. Huang, A reinforcement learning enhanced pseudo-inverse approach to self-collision avoidance of redundant robots, Front. Neurorobot. 18 (2024) 1375309.
[26] Y. Pei, Y. Ivanov, Planning With Purpose: Task-Specific Trajectory Optimization, IEEE Robot. Autom. Lett. 9 (5) (2024) 4551–4558.
[27] J. Cavacanti Santos, M. Martins da Silva, Redundancy resolution of kinematically redundant parallel manipulators via differential dynamic programing, J. Mechan. Robot. 9 (4) (2017) 041016.
[28] M. Shahabi, H. Ghariblu, Optimal joint motion for complicated welding geometry by a redundant robotic system, Eng. Optimiz. 52 (5) (2020) 875–895.
[29] E. Monari, Y. Chen, R. Vertechy, On locally optimal redundancy resolution using the basis of the null space, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 9665–9671.
[30] L.A. Nguyen, K.D. Le, T.L. Harman, Kinematic redundancy resolution for Baxter robot, in: 2021 7th International Conference on Automation, Robotics and Applications (ICARA), IEEE, 2021, pp. 6–9.
[31] F. Flacco, A. De Luca, O. Khatib, Motion control of redundant robots under joint constraints: Saturation in the null space, in: 2012 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 285–292.
[32] F. Flacco, A. De Luca, O. Khatib, Control of redundant robots under hard joint constraints: Saturation in the null space, IEEE Trans. Robot. 31 (3) (2015) 637–654.
[33] A. Ziese, M.D. Fiore, J. Peters, U.E. Zimmermann, J. Adamy, Redundancy resolution under hard joint constraints: a generalized approach to rank updates, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 7447–7453.
[34] A. Kazemipour, M. Khatib, K. Al Khudir, C. Gaz, A. De Luca, Kinematic control of redundant robots with online handling of variable generalized hard constraints, IEEE Robot. Autom. Lett. 7 (4) (2022) 9279–9286.
[35] M.D. Fiore, G. Meli, A. Ziese, B. Siciliano, C. Natale, A general framework for hierarchical redundancy resolution under arbitrary constraints, IEEE Trans. Robot. 39 (3) (2023) 2468–2487.
[36] S.K. Kuri, K. Halder, M.F. Orlando, SOM network with weighted least norm matrix based redundancy resolution for a 5-DOF spatial robotic manipulator, in: 2023 7th International Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA), IEEE, 2023, pp. 1–6.
[37] N.C.N. Doan, P.Y. Tao, W. Lin, Optimal redundancy resolution for robotic arc welding using modified particle swarm optimization, in: 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), IEEE, 2016, pp. 554–559.
[38] A.G. Ruiz, J.C. Santos, J. Croes, W. Desmet, M.M. da Silva, On redundancy resolution and energy consumption of kinematically redundant planar parallel manipulators, Robotica 36 (6) (2018) 809–821.
[39] G. Fabris, L. Scalera, A. Gasparetto, Leveraging kinematic redundancy for energy efficiency in a 8-DOF robotic system, in: Carbone, G., Quaglia, G. (eds) Proceedings of I4SDG Workshop 2025 - IFToMM for Sustainable Development Goals. I4SDG 2025. Mechanisms and Machine Science, 179, Springer, Cham, 2025, pp. 340–348.
[40] P. Boscariol, R. Caracciolo, D. Richiedei, A. Trevisani, Energy optimization of functionally redundant robots through motion design, Appl. Sci. 10 (9) (2020) 3022.
[41] S. Sutjipto, J. Woolfrey, M.G. Carmichael, G. Paul, Cartesian inertia optimization via redundancy resolution for physical human-robot interaction, in: 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), IEEE, 2021, pp. 570–575.
[42] X. Li, H. Liu, M. Dong, A general framework of motion planning for redundant robot manipulator based on deep reinforcement learning, IEEE Trans. Industr. Inf. 18 (8) (2021) 5253–5263.
[43] J. Gregory, A. Olivares, E. Staffetti, Energy-optimal trajectory planning for robot manipulators with holonomic constraints, Syst. Control Lett. 61 (2) (2012) 279–291.
[44] E. Ferrentino, H.J. Savino, A. Franchi, P. Chiacchio, A dynamic programming framework for optimal planning of redundant robots along prescribed paths with kineto-dynamic constraints, IEEE Trans. Autom. Sci. Eng. 21 (4) (2023) 6744–6757.
[45] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Differential kinematics and statics, Robot. Modell. Plann. Control (2009) 105–160.
[46] R.E. Moore, R.B. Kearfott, M.J. Cloud, Introduction to Interval Analysis, Society for Industrial and Applied Mathematics, 2009.
[47] V. Schettino, M.D. Fiore, C. Pecorella, F. Ficuciello, F. Allmendinger, J. Lachner, S. Stramigioli, B. Siciliano, Geometrical interpretation and detection of multiple task conflicts using a coordinate invariant index, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 6613–6618.
[48] M.D. Fiore, F. Allmendinger, C. Natale, A general constraint-based programming framework for multi-robot applications, Robot. Computer-Integrated Manuf. 86 (2024) 102665.

[49] A. Giusti, M. Althoff, Efficient computation of interval-arithmetic-based robust controllers for rigid robots, in: 2017 First IEEE International Conference on Robotic Computing (IRC), IEEE, 2017, pp. 129–135.
[50] C. Nainer, A. Giusti, Automatically deployable robust control of modular reconfigurable robot manipulators, IEEE Robot. Autom. Lett. 7 (2) (2022) 5286–5293.
[51] A. Giusti, C. Nainer, Inverse uncertain-dynamics of robot manipulators using interval arithmetic, in: In: Niola, V., Gasparetto, A., Quaglia, G., Carbone, G. (eds) Advances in Italian Mechanism Science. IFToMM Italy 2022. Mechanisms and Machine Science, 122, Springer, Cham, 2022, pp. 661–668.
[52] G. Fabris, L. Scalera, A. Gasparetto, Online optimization of minimum-time and minimum-energy trajectories for a 1-DOF belt-driven robotic system, Robotica 43 (6) (2025) 2030–2045.
[53] C. Gaz, M. Cognetti, A. Oliva, P.R. Giordano, A. De Luca, Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization, IEEE Robot. Autom. Lett. 4 (4) (2019) 4147–4154.
[54] L. Scalera, C. Nainer, A. Giusti, A. Gasparetto, Robust safety zones for manipulators with uncertain dynamics in collaborative robotics, Int. J. Comput. Integr. Manuf. 37 (7) (2024) 887–899.
[55] M. Wagner, S. Liu, A. Giusti, M. Althoff, Interval-arithmetic-based trajectory scaling and collision detection for robots with uncertain dynamics, in: 2018 Second IEEE International Conference on Robotic Computing (IRC), IEEE, 2018, pp. 41–48.
[56] J. Nocedal, S.J. Wright, Numerical Optimization, Second Edition, Springer Series in Operations Research, Springer Verlag, 2006.
[57] L.N. Trefethen, D. Bau, Numerical Linear Algebra, SIAM, 2022.
[58] F.J. Massey, Jr, The Kolmogorov-Smirnov test for goodness of fit, J. Am. Stat. Assoc. 46 (253) (1951) 68–78.
[59] W.H. Kruskal, W.A. Wallis, Use of ranks in one-criterion variance analysis, J. Am. Stat. Assoc. 47 (260) (1952) 583–621.
[60] G. Fabris, L. Scalera, P. Boscariol, A. Gasparetto, Static and dynamic friction models for robotic manipulators: state of the art and experimental comparison, J. Intell. Robot. Syst. 112 (41) (2026).