The International Journal of Advanced Manufacturing Technology (2024) 132:2677–2699
https://doi.org/10.1007/s00170-024-13496-w

ORIGINAL ARTICLE

Dynamic modelling and energy-efficiency optimization in a 3-DOF parallel robot

Giuliano Fabris[1] · **Lorenzo Scalera**[1] · **Alessandro Gasparetto**[1]

Received: 10 January 2024 / Accepted: 20 March 2024 / Published online: 3 April 2024

Abstract
Energy efficiency is a challenging and relevant research field in modern manufacturing industries, where robotic systems play an essential role in the automation of several industrial operations. In this paper, we present an approach for the energy-efficiency optimization of a 3-DOF parallel robot. The proposed strategy leverages the task placement, the execution time, and the length of the robot lower arms to minimize the energy consumption for the execution of a predefined high-speed pick-and-place operation. To evaluate the actuators energy consumption, the kinematic, dynamic and electro-mechanic mathematical models, as well as an equivalent multibody model, of the parallel robot are implemented. The results of extensive numerical simulations show that the proposed strategy provides notable improvements in the energy efficiency of the parallel robot, with respect to alternative approaches. Starting from a pick-and-place task with optimal task placement with a consumption of 38.2 J (with a cycle time of 0.4 s), the energy expenditure can be reduced to 3.75 J (with a cycle time of 1.86 s), with a reduction percentage of 90.2%, by additionally optimizing the execution time, and the length of the robot lower arms. These results lead to a reduction from 5733 J/min (for 150 cycles/min) to 121 J/min (for 32 cycles/min), allowing to choose the best trade-off between robot productivity and consumed energy.

Keywords Parallel robot · Energy efficiency · Multibody dynamics · Trajectory planning · Optimization

1 Introduction

In the last years, the growing energy demand, especially in the manufacturing industry, has been accompanied by an increased environment awareness. Therefore, the focus of the design and control of robotic and mechatronic systems has moved towards the investigation of energy-efficient and cost-effective solutions. This trend is also encouraged by the European Union policies, which plan to decrease the energy consumption by 32.5% by 2030 [1]. Several studies have demonstrated that an intelligent use of industrial robots can reduce the energy consumption while increasing production rate [2–4]. Nevertheless, the industrial manufacturing requires high volumes of production and high-speed operations. These requests lead the industry sector to be responsible for a large percentage of the total global energy consumption and consequently of the CO_2 emissions (in 2022, a quarter of the CO_2 emissions of all energy systems are attributable to the industrial sector [5]). In this context, energy efficiency is becoming a challenging research topic in both industry and academia.

Different solutions and optimization methods to achieve energy efficiency in automatic machines and robots are provided in the present literature, which includes both theoretical and experimental investigations. Possible approaches to reach this goal comprise the (re)design of the robot structure and the use of lightweight materials to reduce motors effort [6, 7], and the adoption of regenerative drives in order to recover the energy generated during braking phases [8]. Another solution to enhance energy efficiency of robotic systems relies on the proper choice of electric motor and gearbox during the design of the system [9]. A particular strategy is based on the exploitation of the natural motion, defined as the system dynamic response due to the conversion of potential elastic energy into kinetic energy [10]. This approach can

✉ Lorenzo Scalera
lorenzo.scalera@uniud.it

Giuliano Fabris
giuliano.fabris@uniud.it

Alessandro Gasparetto
alessandro.gasparetto@uniud.it

[1] Polytechnic Department of Engineering and Architecture, University of Udine, 33100 Udine, Italy

be implemented adding compliant elements to the robotic system, beneficial especially in cyclic tasks [11].

Nevertheless, these methods require that the mechanical system has to be modified, introducing or substituting physical components. Such operations may not always be easy to implement in an existing system, especially if already integrated in a manufacturing workcell. Further strategies for energy efficiency in industrial robots that can overcome these issues are the re-scheduling of the operations [12], the proper selection and tuning of control strategy [13], the optimization of motion time [14], the optimal task placement in the robot workspace [15–17], as well as advanced solutions for the trajectory planning [18–20] in order to limit the energy or torque expenditure of the motors. Examples of trajectory planning strategies include the study of standard primitives or polynomials for point-to-point motion [21, 22], multi-point trajectories [23], and the optimization of motion parameters [16, 24].

Focusing on parallel robots, the industrial applications in which they offer the best performance are high-frequency pick and place and manipulation of low-payload objects in industries such as electronics, food processing, and pharmaceuticals [25, 26]. Their rapid and precise movements allow for efficient handling of small components or products on assembly lines, as well as packaging products into containers and sorting items on conveyor belts [27]. These operations are often cyclical and repeated for large volumes of products. Furthermore, parallel robots are used in measurement and inspection processes that require repetitive movements [28]. In all these applications, parallel robots optimize throughput and minimize downtime, thus enhancing overall production efficiency. In addition, parallel robots are more and more used in additive manufacturing and 3D printing, machine tooling, as well as in metrology and calibration [29, 30]. Therefore, promoting energy saving in these systems can lead to a substantial reduction in the energy consumption and, consequently, in CO_2 emissions and operating costs. For instance, the authors in [31] exploit the natural dynamics and optimize torsional spring parameters for the energy consumption reduction in a four-degree-of-freedom (DOF) parallel robot, where each kinematic chain is characterized by a revolute joint and two subsequent universal joints (4-revolute-universal-universal, 4-RUU). The manipulator executes predefined pick-and-place trajectories, achieving a reduction of energy expenditure up to 67.8% with respect to the reference case without springs, with numerical tests. In [10] Carabin et al. apply the same concept to a 3-DOF Linear Delta robot, optimizing the spring parameters and position for performing a 3D printing operation, obtaining a reduction of energy consumption up to almost 50%, with both numerical and experimental tests. Another example of energy efficiency exploiting the natural dynamics of the system can be found in [32], where an energy consumption reduction up to 70% is achieved equipping a 3-DOF Delta robot with torsional variable stiffness springs (VSSs), according to numerical results. Differently from previous works that optimize spring parameters only for a fixed task, the authors in [33] first optimize spring parameters for 2-DOF planar parallel robot performing a nominal pick-and-place task. Then, the variation of energy consumption is evaluated with respect to the change of the predefined task, optimizing spring preload for each task while spring stiffness remains fixed. Both numerical and experimental tests demonstrate that an energy consumption reduction up to 51.3% can be achieved even when the task to perform significantly differs from the nominal one.

Alternative approaches for achieving energy efficiency in parallel robots can be found in [34] and [35], where the authors optimize the parameters of the functions that define the motion profile, respectively B-spline and Lamé curves (the latter are used for rounding the corners of a pick-and-place trajectory), achieving a substantial energy saving. In [36] and [37] the authors address the problem of reducing the energy expenditure of a 4-DOF parallel manipulator (4-RUU) by optimizing only the placement of a linear movement and a pick-and-place operation, respectively, with respect to the robot base reference frame. Furthermore, Liu et al. [38] evaluate the energy efficiency of parallel robots based on the kinetic energy change rate. Such procedure can be applied for optimizing trajectories and structural and process parameters. Lee et al. [39] achieve energy efficiency for a 2-DOF parallel mechanism through redundant actuation, using three motors instead of two. This method enables a reduction of electrical energy consumption of the actuators up to 45% compared to the corresponding non-redundant actuated version of the mechanism, with experimental tests. An overview on energy-efficiency strategies for parallel robots is summarized in Table 1. To the best of the authors' knowledge, no example of optimization approaches that combine the task placement, the execution time, and the geometry of robot parts to minimize the energy consumption for the execution of a predefined operation can be found in the present literature.

In this paper, we present an approach for the energy-efficiency optimization of a 3-DOF parallel robot. We focus on parallel manipulators since, compared to serial robots, they offer notable advantages in terms of velocity and acceleration, stiffness, rigidity, payload capacity, and reduced inertia [40]. For these reasons, they are employed in many advanced manufacturing industrial processes, as mentioned above. The proposed strategy leverages the task placement, the execution time, and the length of the robot lower arms to minimize the energy consumption for the execution of a predefined high-speed pick-and-place operation. To evaluate the actuators energy consumption, the kinematic, dynamic and electromechanic mathematical models, as well as an equivalent

Table 1 Overview on energy-efficiency strategies for parallel robots

First author	Year	Ref.	Considered robot	Optimization approach
Balderas Hill	2021	[32]	3-DOF Delta robot with VSSs	Exploiting natural dynamics
Carabin	2021	[10]	3-DOF Linear Delta robot with linear springs	Optimization of spring no-load length and stiffness
Chen	2019	[34]	6-UPU parallel robot	Optimization of B-spline motion profile parameters
Lee	2015	[39]	2-DOF parallel robot	Exploiting redundant actuation
Liu	2019	[38]	5-DOF parallel robot	Exploiting kinetic energy change rate
Mora	2022	[33]	2-DOF parallel robot with torsional springs	Optimization of trajectory, spring stiffness and preload
Scalera	2019	[31]	4-DOF Delta robot with torsional springs	Optimization of spring preload and stiffness
Scalera	2020	[36]	4-DOF Delta robot	Task placement analysis
Scalera	2021	[37]	4-DOF Delta robot	Optimization of task placement
Zhang	2019	[35]	4-DOF Delta robot	Optimization of Lamé curve parameters
Proposed approach			3-DOF Delta robot	Optimization of task placement, execution time, and length of the robot lower arms

multibody model of the parallel robot are implemented. The results of extensive numerical simulations show that the proposed strategy provides notable improvements in the energy efficiency of the parallel robot, with respect to alternative approaches.

The paper is organized as follows: Sect. 2 reports the kinematic, dynamic and electro-mechanic models of the robotic system, and Sect. 3 introduces the compared approaches. Section 4 describes the implementation of the optimization problems and the multibody model of the robot, whereas Sect. 5 reports simulation results. Finally, Sect. 6 concludes the work.

2 Mathematical model of a 3-DOF parallel robot

The robot considered in this paper is a 3-DOF Delta robot, a type of parallel robot whose platform can only perform translations in the 3D space. As it can be seen in Fig. 1a, its structure is composed of two platforms, one fixed (that acts as a base) and one mobile (to which the end-effector is attached). These are connected by three identical kinematic chains. Each chain consists of two links, an upper arm and a pair of lower arms, that are linked together by means of spherical joints. The upper arm is connected to the fixed platform by a revolute joint, whereas each pair of lower arms is connected to the mobile platform through the other two spherical joints. It can be noticed that the pairs of lower arms cannot rotate around their central axis, due to the kinematics of the manipulator. This feature allows each pair to be simplified as a single arm parallel to the original ones and passing through the center of the pair (i.e., points C_i and B_i, with $i = 1, 2, 3$). In the following, the kinematic, dynamic, and electro-mechanic models of the considered manipulator are described. The strategy for developing the kinematic and dynamic models is a generic method that can be adapted to parallel robots with a similar structure.

2.1 Kinematics

In this section, the inverse kinematics of the 3-DOF parallel manipulator is described, which aims at calculating the position of the joint variables $\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$ starting from the pose of the end-effector $\boldsymbol{X} = \begin{bmatrix} x & y & z \end{bmatrix}^T$. The single joint variable is obtained by finding the intersection between the arc of the circumference representing the possible positions of the upper arm end not fixed to the base (point $C_{i,upper\ arm}$) and the positions reachable by the lower arm end not linked to the mobile platform (point $C_{i,lower\ arm}$), as shown in Fig. 1b. This can lead to three situations:

- zero intersections: no solutions found, the chosen pose is outside the robot workspace;
- one intersection: the chosen pose is at the limits of the workspace;
- two intersections: only one of them is acceptable, namely the one that leads to a smaller q_i, as it is illustrated in Figs. 1b and 2b.

The kinematic relationship that corresponds to this consideration can be obtained starting from the closing equation of the mechanism:

$$\boldsymbol{b_i}^T \boldsymbol{b_i} = b^2 \tag{1}$$

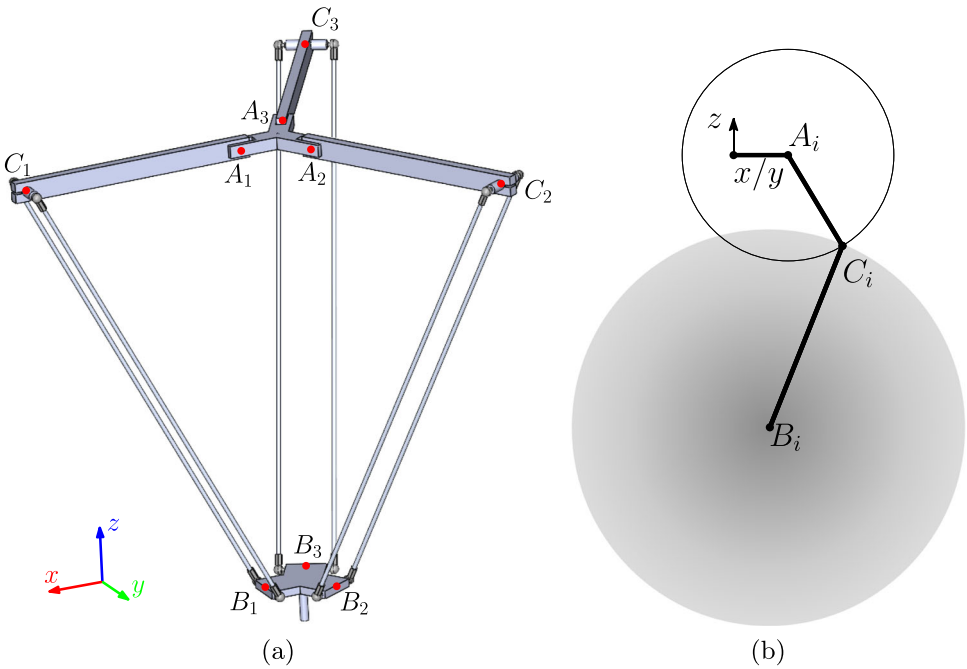


Fig. 1 3D model of the robot (a); search for the intersection between upper and lower arms (b)

where:

$$\boldsymbol{b_i} = \boldsymbol{B_i} - \boldsymbol{C_i} = \begin{bmatrix} b_{ix} & b_{iy} & b_{iz} \end{bmatrix}^T \tag{2}$$

The positions of points $\boldsymbol{C_i}$ are derived by knowing the coordinates of points $\boldsymbol{A_i}$, which are fixed and given by the robot geometry, as function of joint variables q_i and length of upper arm a (Fig. 2). The coordinates of points $\boldsymbol{B_i}$ can be obtained starting from the end-effector pose $\boldsymbol{X}$ and by taking the geometric constants r_b and h into account.

Substituting expression of $\boldsymbol{B_i}$ and $\boldsymbol{C_i}$ in Eq. 2 and applying Eq. (1) leads to the following system of equation:

$$I_i \cos(q_i) + L_i \sin(q_i) + K_i = 0\,, \quad i = 1, 2, 3 \tag{3}$$

where I_i, L_i e K_i are constants that are function of the robot geometry and the end-effector pose $\boldsymbol{X}$. Solving Eq. (3), the solution of the inverse kinematic problem for the i-th joint variable can be written as follows:

$$q_i = 2 \arctan \frac{-I_i - \sqrt{I_i^2 + L_i^2 - K_i^2}}{K_i - L_i}\,, \quad i = 1, 2, 3 \tag{4}$$

It can be noticed that these solutions are real only if the radicands are greater or equal to zero; if one of them becomes negative, it means that the robot pose $\boldsymbol{X}$ is outside the robot workspace. Therefore, by imposing this condition, the robot workspace can be easily derived.

The inverse differential kinematics of the parallel robot can be obtained by differentiating Eq. (1) with respect to time and it is given by:

$$\dot{\boldsymbol{q}} = {\boldsymbol{J_q}}^{-1} \boldsymbol{J_x} \dot{\boldsymbol{X}} = \boldsymbol{J}^{-1} \dot{\boldsymbol{X}} \tag{5}$$

where:

$$\boldsymbol{J_q} = \begin{bmatrix} \boldsymbol{u_1}(\boldsymbol{a_1} \times \boldsymbol{b_1}) & 0 & 0 \\ 0 & \boldsymbol{u_2}(\boldsymbol{a_2} \times \boldsymbol{b_2}) & 0 \\ 0 & 0 & \boldsymbol{u_3}(\boldsymbol{a_3} \times \boldsymbol{b_3}) \end{bmatrix} \tag{6}$$

and

$$\boldsymbol{J_x} = \begin{bmatrix} x_{b1} & y_{b1} & z_{b1} \\ x_{b2} & y_{b2} & z_{b2} \\ x_{b3} & y_{b3} & z_{b3} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b_1} & \boldsymbol{b_2} & \boldsymbol{b_3} \end{bmatrix}^T \tag{7}$$

$\boldsymbol{J}$ is the Jacobian matrix, the vector $\boldsymbol{u_i}$ is the generic unit vector parallel to motor axis (Fig. 2a), whereas $\boldsymbol{a_i} = \boldsymbol{C_i} - \boldsymbol{A_i}$ is the vector of the upper arm.

Finally, the inverse acceleration kinematics can be derived by differentiating Eq. (5) with respect to time, obtaining:

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{-1} \ddot{\boldsymbol{X}} + {\boldsymbol{J_q}}^{-1} (\dot{\boldsymbol{J}}_{\boldsymbol{x}} - \dot{\boldsymbol{J}}_{\boldsymbol{q}} \boldsymbol{J}^{-1}) \dot{\boldsymbol{X}} \tag{8}$$

2.2 Dynamics

The dynamics of the 3-DOF parallel robot can be solved adopting the Lagrangian approach restricted to some simpli-

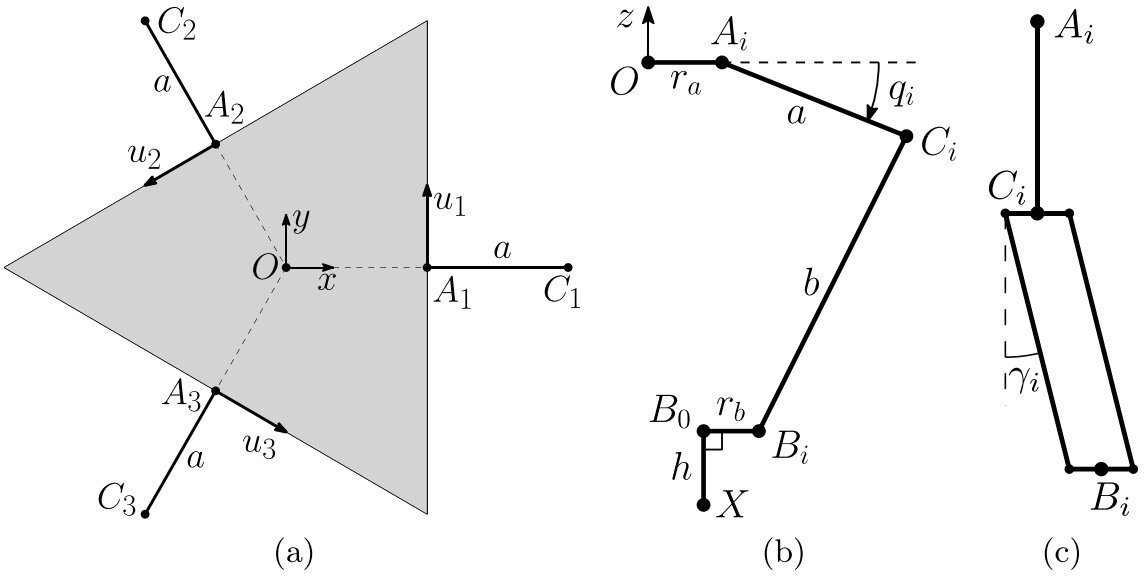


Fig. 2 Notation used in the model of the manipulator: top view (a), lateral view (b), and particular of lower arms movement (c)

fying assumptions, necessary to facilitate its implementation and resolution. As main hypothesis of the dynamic model, it is supposed that the inertia of the lower arms is small enough to be neglected, and the distributed mass of each pair of lower arms is approximated by two equal point masses positioned at the extremities of the link, as shown in Fig. 3.

Another simplification is to reduce the mobile platform to a mass concentrated at the point $\boldsymbol{B_0}$ (Fig. 2b). This is a reasonable approximation, since the mobile platform can only translate, so its motion is the same of its center of mass (approximated to the point $\boldsymbol{B_0}$) and therefore it can be considered acceptable to neglect its rotational inertia and assimilate it to a point body. Aerodynamics effects are also neglected.

The purpose of the inverse dynamics is to compute the torque needed for the movement of the parallel robot by knowing the position, velocity and acceleration of the joint variables. The total torque $\boldsymbol{\tau} = \begin{bmatrix}\tau_1 & \tau_2 & \tau_3\end{bmatrix}^T$ needed to steer the robot during its motion can be obtained as sum of the torques required to move the different components of the system, i.e., $\boldsymbol{\tau_{act}}$ for the actuating system (actuators and upper arms), $\boldsymbol{\tau_{plat}}$ for the mobile platform, and $\boldsymbol{\tau_{load}}$ for the payload. The first torque contribution can be written as follows:

$$\boldsymbol{\tau_{act}} = \boldsymbol{I_{act}}\ddot{\boldsymbol{q}} + g\boldsymbol{M_{arm}}\cos(\boldsymbol{q}) + \boldsymbol{f_s} + \boldsymbol{F_v}\dot{\boldsymbol{q}} \tag{9}$$

The first term is the inertia term, proportional to the angular acceleration of the joints $\ddot{\boldsymbol{q}}$. $\boldsymbol{I_{act}} = \text{diag}(\begin{bmatrix}I_{eq} & I_{eq} & I_{eq}\end{bmatrix})$ is

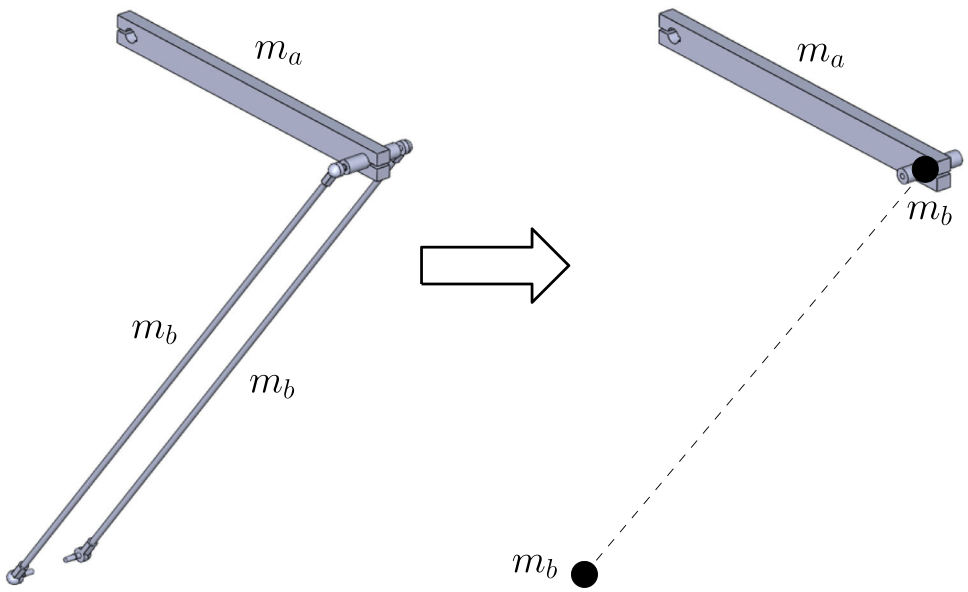


Fig. 3 Simplification of the lower arms

the inertia matrix, where $I_{eq} = i_{rid}^2 I_{act} + I_a + I_b$ represents the equivalent inertia of a single arm evaluated with respect to the corresponding point A_i, i_{rid} is the gear ratio, I_{act} is the inertia of the motor, $I_a = \frac{1}{3} m_a a^2$ is the inertia of the upper arm and $I_b = m_b a^2$ is the inertia of the point mass m_b. The second term is the gravitational term, given by the upper arm mass m_a and by the point mass m_b located at its extremity. These contribution are comprised in matrix $\boldsymbol{M_{arm}} = \text{diag}([m_{eq}\ m_{eq}\ m_{eq}])$, where $m_{eq} = \frac{1}{2} m_a a + m_b a$, whereas g is the gravitational acceleration. Furthermore, the third and fourth terms of Eq. 9 take into account the contribution of Coulomb and viscous friction, respectively: $\boldsymbol{f_s} = f_s \tanh(\dot{\boldsymbol{q}})$ is the static friction torque, $\boldsymbol{F_v} = \text{diag}([f_v\ f_v\ f_v])$ and f_v is the viscous friction coefficient. The definition of Coulomb friction permits to avoid discontinuities in the torque profile as the sign of the joint velocity changes [41]. By substituting $\ddot{\boldsymbol{q}}$ with Eq. 8, Eq. 9 can be rewritten as function of the velocity $\dot{\boldsymbol{X}}$ and acceleration $\ddot{\boldsymbol{X}}$ of the end-effector as:

$$\boldsymbol{\tau_{act}} = \boldsymbol{I_{act}} \boldsymbol{J}^{-1} \ddot{\boldsymbol{X}} + \boldsymbol{J_q}^{-1} (\dot{\boldsymbol{J}}_{\boldsymbol{x}} - \dot{\boldsymbol{J}}_{\boldsymbol{q}} \boldsymbol{J}^{-1}) \dot{\boldsymbol{X}} + g \boldsymbol{M_{arm}} \cos(\boldsymbol{q}) + \boldsymbol{f_s} + \boldsymbol{F_v} \dot{\boldsymbol{q}} \tag{10}$$

The second torque contribution analyzed is the one given by the mobile platform. From the principle of virtual work, the required torque is given by:

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{F} \tag{11}$$

where $\boldsymbol{F}$ is the vector of forces acting on the mobile platform. Since it can only translate, the acting forces are only the inertia force and the force of gravity, which can be written as:

$$\boldsymbol{F} = \boldsymbol{M_{plat}} (\ddot{\boldsymbol{X}} + \boldsymbol{G}) \tag{12}$$

where $\boldsymbol{M_{plat}} = \text{diag}([m_p\ m_p\ m_p])$, $m_p = m_{plat} + 3m_b$, where m_{plat} is the mass of the mobile platform and $\boldsymbol{G} = [0\ 0\ -g]^T$. It follows that the torque required by the motors due to the mobile platform is:

$$\boldsymbol{\tau_{plat}} = \boldsymbol{J}^T \boldsymbol{M_{plat}} (\ddot{\boldsymbol{X}} + \boldsymbol{G}) \tag{13}$$

The last torque contribution is given by the possible payload connected to the end-effector. Since this is connected to the mobile platform, its movement will be the same as the latter and the required torque will be calculated in the same way. Therefore, the torque required by the motors due to the payload is:

$$\boldsymbol{\tau_{load}} = \boldsymbol{J}^T \boldsymbol{M_{load}} (\ddot{\boldsymbol{X}} + \boldsymbol{G}) \tag{14}$$

where $\boldsymbol{M_{load}} = \text{diag}([m_{load}\ m_{load}\ m_{load}])$ and m_{load} is the mass of the payload.

Finally, the resulting expression for the torque needed for the movement of the parallel robot can be expressed by:

$$\begin{aligned} \boldsymbol{\tau} = [\boldsymbol{I_{act}} \boldsymbol{J}^{-1} + \boldsymbol{J}^T (\boldsymbol{M_{plat}} + \boldsymbol{M_{load}})] \dot{\boldsymbol{X}} + \boldsymbol{J}^T (\boldsymbol{M_{plat}} + \boldsymbol{M_{load}}) \boldsymbol{G} + \\ \boldsymbol{I_{act}} \boldsymbol{J_q}^{-1} (\dot{\boldsymbol{J}}_{\boldsymbol{x}} - \dot{\boldsymbol{J}}_{\boldsymbol{q}} \boldsymbol{J}^{-1}) \dot{\boldsymbol{X}} + g \boldsymbol{M_{arm}} \cos(\boldsymbol{q}) + \boldsymbol{f_s} + \boldsymbol{F_v} \dot{\boldsymbol{q}} \end{aligned} \tag{15}$$

2.3 Electro-mechanic model

The electro-mechanic model of robot actuators is needed to assess energy consumption and it is developed by considering equivalent brushless DC motors, as in [36]. The i-th motor armature current $i_i(t)$ can be computed as function of the torque provided by the motor $\tau_{m,i} = \tau_i / i_{rid}$ and the motor torque constant K_t as follows:

$$i_i(t) = \frac{\tau_{m,i}}{K_t}, \quad i = 1, 2, 3 \tag{16}$$

Then, the voltage drop $v_i(t)$ across the i-th motor can be expressed as follows:

$$v_i(t) = R\, i_i(t) + K_e\, \dot{q}_{m,i}(t), \quad i = 1, 2, 3 \tag{17}$$

where R is the resistance of the motor windings, K_e is the back-emf constant, whereas $q_{m,i} = q_i i_{rid}$ is the velocity of the i-th motor shaft. Considering the driver efficiency η_d, the voltage-current product gives the instantaneous electric power drawn by the i-th robot actuator:

$$P_{e,i}(t) = \frac{v_i(t)\, i_i(t)}{\eta_d}, \quad i = 1, 2, 3 \tag{18}$$

When the actuators absorb energy from the drive unit, the electric power $P_{e,i}$ has positive values. Vice versa, when the energy takes the opposite path going from the actuators to the drive unit, $P_{e,i}$ takes negative values. However, the drives considered for this system are not regenerative. Consequently, the total energy consumption E_c is calculated taking into account only the power consumed by the three actuators $P_{c,i}(t)$ and integrating this over the time period T, obtaining the following equation:

$$\begin{aligned} E_c &= \sum_{i=1}^{3} \int_0^T P_{c,i}(t) dt \qquad \text{where} \\ P_{c,i}(t) &= \begin{cases} P_{e,i}(t) & \text{if } P_{e,i}(t) \geq 0 \\ 0 & \text{if } P_{e,i}(t) < 0 \end{cases} \end{aligned} \tag{19}$$

3 Compared approaches

In this paper, we compared three approaches to minimize the total energy consumption of the 3-DOF parallel robot, during

a defined task. The compared approaches are described as follows:

- Approach (1) finds the optimal placement of a pick-and-place task within the robot workspace that minimizes energy consumption;
- Approach (2) plans time-energy optimal trajectories by evaluating the proper execution times for the different path sections.
- Approach (3) is the strategy proposed in this paper, which leverages the task placement, the execution time, and the length of the robot lower arms to minimize the robot energy consumption.

Three different cases are considered for the proposed approach: first only the length of the robot lower arms is optimized (Approach (3a)), then, the length and the task placement are considered (Approach (3b)). Finally, the task placement, the execution time, and the length of the robot lower arms are optimized together to reduce the energy consumption (Approach (3c)).

In all the compared approaches, we take into account an ordinary pick-and-place task composed of two vertical and one horizontal movements. However, these methods can be applied to other desired paths as well. The "434" spline algorithm is adopted as base for motion planning [42]. The trajectory is defined using nine way points in the operational space as shown in Fig. 4. The path for the robot end-effector consists of a vertical ascent of 25 mm, a horizontal translation of 305 mm, a vertical descent of 25 mm, and return along the same path. Path corners are blended with a radius of 5 mm.

In all the considered scenarios, the problem that we aim to solve is a bounded constrained nonlinear optimization problem, which can be formulated as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\boldsymbol{\chi}} f(\boldsymbol{x}) \\ &\boldsymbol{\chi} = \left[\boldsymbol{x}^L\ \boldsymbol{x}^U\right]^T \\ &\text{subject to } \boldsymbol{H}(\boldsymbol{x}) \geq \boldsymbol{c} \end{aligned} \tag{20}$$

where $\boldsymbol{x}$ is the vector of optimization variables, $f(\boldsymbol{x})$ is the cost function that has to be minimized, $\boldsymbol{\chi}$ indicates the lower ($\boldsymbol{x}^L$) and upper ($\boldsymbol{x}^U$) bounds for the design domain, whereas $\boldsymbol{H}(\boldsymbol{x})$ and $\boldsymbol{c}$ define the inequality constraints. While $\boldsymbol{x}$, $f(\boldsymbol{x})$ and $\boldsymbol{\chi}$ depend on the approach, the inequality constraints are the same for both approaches and are defined as follows:

$$\begin{cases} I_i^2 + L_i^2 - K_i^2 \geq 0 \\ q_{\min} < q_i < q_{\max} \\ |\ \gamma_i\ | < \gamma_{\max} \\ |\ \dot{q}_i\ | < \dot{q}_{\max} \\ |\ \tau_i\ | < \tau_{\max} \end{cases} \quad \text{for } i = 1, 2, 3 \tag{21}$$

The first condition derives from Eq. (4) and imposes the task to be within the robot workspace. $q_{\min}$, $q_{\max}$ and $\gamma_{\max}$ are defined from robot kinematics, whereas $\dot{q}_{\max}$ and $\tau_{\max}$ are imposed by motor limits. γ represents the rotation angle of spherical joints from their neutral position as shown in Fig. 2c and it is limited due to their physical limits. The geometric, dynamic and electro-mechanic parameters of the manipulator considered in this work are reported in Table 2.

3.1 Approach (1): optimal task placement

Approach (1) aims at evaluating the optimal positioning of the pick-and-place task inside the robot workspace in order to minimize the total energy consumption, by fixing the time distance between each pair of consecutive way points. The path of the robot is parameterized as in [37] (Fig. 5):

The position of the way points can be defined using four parameters, i.e., the distance d and the angle ϕ, which are the polar coordinates of the mid-point of the trajectory $\boldsymbol{M}$ in the xy plane, θ, that measures the orientation of the trajectory with respect to the direction of $\boldsymbol{OM}$, and z, which is the vertical coordinate of point $\boldsymbol{M}$. This leads to the following optimization problem:

$$\min_{d,\phi,\theta,z} E_c \tag{22}$$

subject to the constraints in Eq. 21. The angles ϕ and θ are limited above respectively to $2\pi/3$ and π. This choice allows us to consider all the possible solutions without repetitions

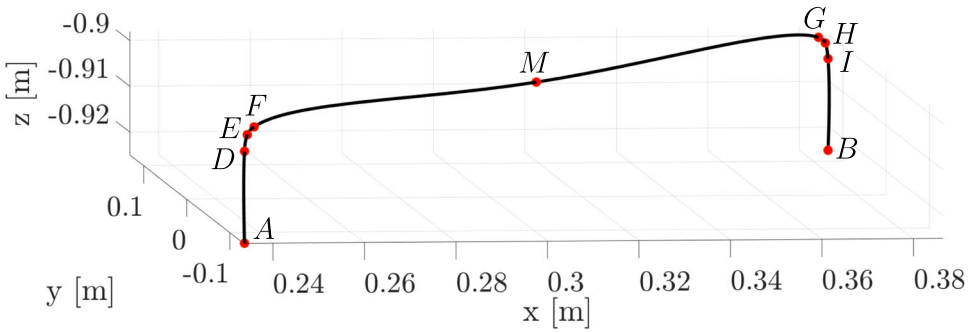


Fig. 4 Path of the robot end-effector in the 3D space

Table 2 Geometric, dynamic and electro-mechanic parameters of the robotic system considered for the numerical simulations

Parameter	Value	Parameter	Value
a	0.450 m	g	$-9.81\ m/s$
b	1.030 m	f_s	2.40 Nm
r_a	0.091 m	f_v	0.24 Nms/rad
r_b	0.073 m	I_{act}	$3.52 \cdot 10^{-4}\ kg\ m^2$
h	0.078 m	i_{rid}	16
m_a	1.115 kg	K_t	1.04 Nm/A
m_b	0.165 kg	K_e	0.70 Vs/rad
m_{plat}	0.846 kg	R	2.40 Ω
m_{load}	0.200 kg	η_d	0.9

due to the symmetry of the structure. For the execution of the different path sections (Fig. 4), a time of 0.025 s for the vertical sections (t_1, sections $A-D$ and $I-B$), 0.005 s for half of the blend sections (t_2, sections $D-E$, $E-F$, $G-H$, and $H-I$) and 0.065 s for half of the horizontal sections (t_3, sections $F-M$ and $M-G$) are set.

3.2 Approach (2): optimal time distances between way points

Approach (2) aims at optimizing the execution times of different path sections in order to plan time-energy optimal trajectories, fixing the task position. In general, decreasing the duration of the trajectory increases velocities, accelerations, torques and consumed power. A trade-off between duration and energy consumption can be found, depending on which of the two contributions is preferred. Therefore, the optimization problem that we aim to solve is the following:

$$\min_{t_i}\left[\omega \cdot \sum_i t_i + (1-\omega)\cdot E_c\right] \tag{23}$$

subject to the constraints in Eq. 21. The cost function in Eq. 23 weighs both the sum of t_i, which is equal to the total duration T, and the total energy consumption E_c. The trade-off between the two items of the cost function is set by means of the parameter ω, which can vary from 0 to 1. A small value equal to 0.001 s for the lower bound of execution times is adopted in order to ensure numerical stability. The position of the task is defined by $d = 0.135\,m, \phi = \pi/3\,rad, \theta = 0\,rad$, and $z = -0.964\ m$. For Approach (2) we evaluate how the optimal solution changes as the value of ω varies, as well as the influence of the discretization (i.e., the sampling rate of the trajectory) on computation times and the optimal solution found. Finally, we also test how the solution is affected by the optimization algorithm adopted for solving the optimization problem.

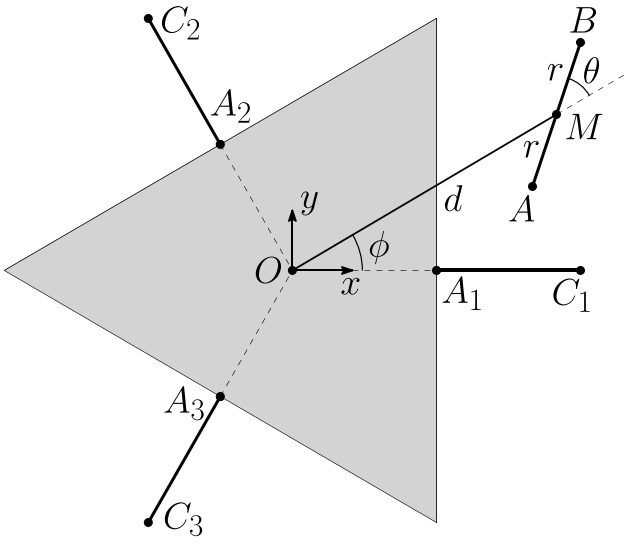


Fig. 5 Top view of the parametrization of the path

3.3 Approach (3): proposed strategy

The approach proposed in this paper leverages the task placement, the execution time, and the length b of the robot lower arms to minimize the robot energy consumption. Three different cases are considered for the proposed approach:

- Approach (3a) finds the value of b that minimizes the total energy consumption E_c, fixing the task position and the execution time;
- Approach (3b) evaluates the optimal combination of b and task placement inside robot workspace to minimize the total energy consumption E_c, fixing the execution time;
- Approach (3c) finds the optimal combination of b, task placement inside robot workspace and execution times of different path sections to planning time-energy optimal trajectories.

We have chosen to optimize the length of the lower arms, usually composed of lightweight aluminum or carbon fiber tubes, since they are one of the easiest and cheapest components of the robot structure to replace. The lower arms can indeed be substituted one by one without disassembling the whole robot, nor disconnecting the robot from its support structure. Furthermore, only the rod has to be replaced without changing the couplings with the upper arms and the mobile platform. To evaluate the mass change of lower arms with their length, a lightweight aluminum alloy is considered

as rod material. The length of the rod l_{rod} can be related to the length of entire lower arm knowing the length of coupling parts (see Fig. 6):

$$l_{rod} = b - 2 \cdot l_{coupling} \tag{24}$$

Therefore, the value of m_b as a function of b can be computed as follows:

$$m_b = m_{coupling} + m_{rod} = m_{coupling} + \pi \ \rho_{rod} \ r_{rod}^2 \ (b - 2 \cdot l_{coupling}) \tag{25}$$

where $m_{coupling}$ is the coupling parts mass, m_{rod} is the rod mass, r_{rod} is its external radius, and ρ_{rod} is the density of the aluminum alloy. Table 3 reports the main parameters values of lower arms.

We underline the main features of the different scenarios. In Approach (3a), the optimization problem can be formulated as follows:

$$\min_{b} E_c \tag{26}$$

The task placement is defined by $d = 0.21\ m$, $\phi = \pi/3\ rad$, $\theta = 0.0\ rad$ and $z = -0.8\ m$, whereas the execution times considered are $t_1 = 0.050\ s$, $t_2 = 0.010\ s$ and $t_3 = 0.140\ s$.

For Approach (3b), the optimization problem is:

$$\min_{b,d,\phi,\theta,z} E_c \tag{27}$$

The execution times considered are $t_1 = 0.050\ s$, $t_2 = 0.010\ s$ and $t_3 = 0.140\ s$.

Approach (3c) can be seen as a combination of all previous approaches described, through which we can find the best configuration for executing the desired task with the minimum energy consumption. Therefore, the optimization problem assumes the following form:

$$\min_{b,d,\phi,\theta,z,t_i} \left[\omega \cdot \sum t_i + (1 - \omega) \cdot E_c\right] \tag{28}$$

The optimization problems described in Eqs. 26, 27 and 28 are all subject to the constraints in Eq. 21.

4 Simulation test bed

In the following, we describe the simulated test bed. First, the implementation and resolution methods for the optimization problems are outlined. Then, we describe how the multibody model is built. Finally, the method adopted for solving the optimization problem integrating the multibody model in the co-simulation one is illustrated.

4.1 Mathematical model

The mathematical model of the 3-DOF parallel robot presented in Sect. 2 is implemented in Matlab 2020b, using a computer running Windows 10 Pro with an Intel Core i5-10600K CPU @ 4.10 GHz and 32.0 GB of RAM. The optimization problem is solved using two different Matlab functions, the constrained nonlinear multivariable function *fmincon*, and the *ga* function. With the *fmincon* function, the optimization problem is solved implementing two different algorithms: the interior-point method and the sequential quadratic programming (SQP) method.

The interior-point optimization algorithm transforms the original inequality constrained problem in a sequence of approximate equality constrained problems, that are easier to solve. As optimization parameters, we choose a function and a constraint tolerance of $1.0 \cdot 10^{-6}$, and a step tolerance

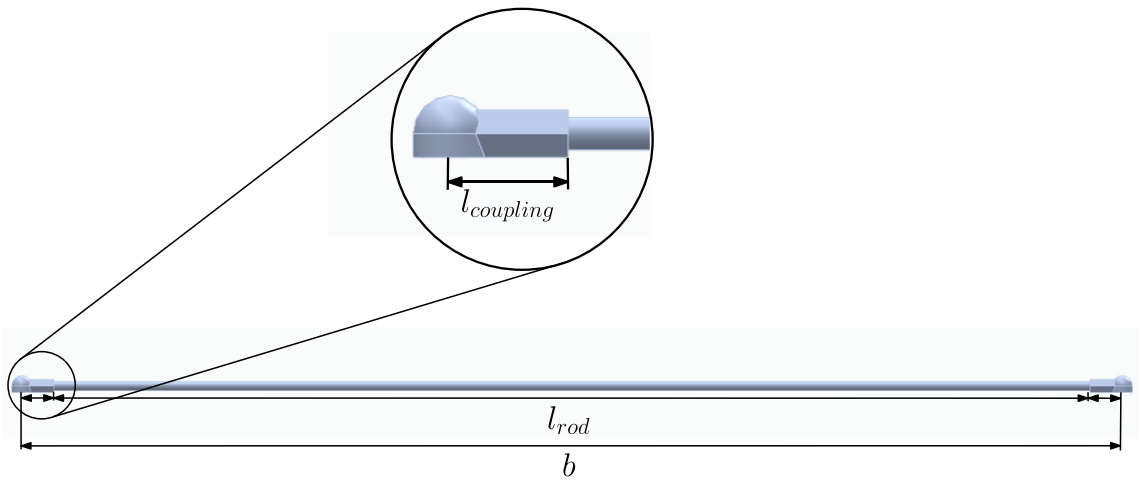


Fig. 6 Lower arm parts and detail of coupling part

Table 3 Main parameters of the lower arms

Parameter	Value
r_{rod}	0.004 m
$l_{coupling}$	0.030 m
$m_{coupling}$	0.039 kg
ρ_{rod}	2584 kg/m^3

of $1.0 \cdot 10^{-10}$. The maximum number of iterations is set to 2500, whereas the maximum number of function evaluations is set to 10000. Moreover, we perform a multi-start optimization, in order to better investigate the whole design domain χ. 50 random starting points are chosen for Approaches (1), (2) and (3b), 20 for Approach (3a) and 100 for Approach (3c), due to the increased complexity of the problem.

The SQP method closely mimics the Newton's method for a constrained optimization, as it is made for an unconstrained optimization. At each main iteration, an approximation of the Hessian of the Lagrangian function is used to define a search direction. As optimization parameters, we choose a function and a constraint tolerance of $1.0 \cdot 10^{-6}$, and a step tolerance of $1.0 \cdot 10^{-6}$. The maximum number of iterations is set to 400, whereas the maximum number of function evaluations is set to 100 multiplied for the number of variables. A multi-start optimization is performed also with the SQP algorithm, choosing 30 random starting points for Approaches (1), (2) and (3b), 10 for Approach (3a) and 50 for Approach (3c).

Furthermore, the *ga* function finds the minimum of a function using a genetic algorithm, based on a natural selection process that mimics the biological evolution. As optimization parameters, we choose a function tolerance of $1.0 \cdot 10^{-6}$, and a constraint tolerance of $1.0 \cdot 10^{-3}$. The maximum number of generations is set to 100 multiplied for the number of variables n, whereas the population size N_p is determined as $N_p = \min(\max(10 \cdot n, 40), 100)$.

For each approach and algorithm adopted, we report the mean solution time, the minimum energy consumption of the parallel robot for the tested trajectory, as well as the optimal parameters found. The energy is computed sampling positions, velocities, accelerations and torques during the trajectories with a frequency of 1000 Hz.

4.2 Multibody model

The mathematical dynamic model of the 3-DOF parallel robot includes some simplifying hypothesis that approximate the actual torques that the motors have to provide, as mentioned in Sect. 2.2. To assess the reliability of the mathematical model, we implement a multibody model of the manipulator using MSC Adams 2022.4. This software allows one to simulate the kinematics and dynamic behavior of a mechanical system with an acceptable error with respect to the real one. In this way, the theoretical dynamic model is validated and the differences in the results provided by the mathematical and the multibody dynamic models can be assessed.

The 3D model of the robot is designed using SolidWorks and then imported in Adams. Subsequently, the required constraints between different structure elements are imposed. First, the base of the 3-DOF parallel robot is fixed to the ground. Then, the revolute joints between the base and the three upper arms are created (Fig. 8a), choosing the correct origin and direction for the measure of joint variable position in order to make them consistent with those measured in Matlab. Furthermore, friction is imposed. In Adams, choosing the *Original* formulation with the parameters reported in Fig. 7 permits to make friction effects comparable with those obtained in Matlab. Differently from the Matlab model, in the Adams one also the transition velocity and the transition velocity coefficient have to be specified. These two parameters have a similar function to that of the $\tanh(\dot{\boldsymbol{q}})$ included in

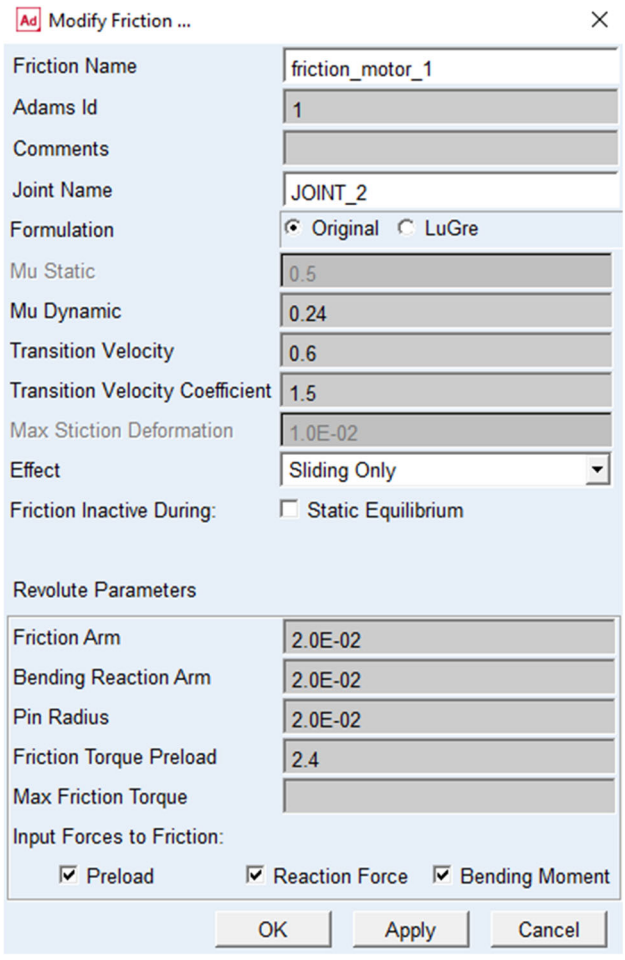


Fig. 7 Parameters chosen for friction description in the Adams software

the definition of friction in Matlab. Friction coefficients are reported in Table 2 and static friction torque f_s in Eq. 9 is implemented in Adams as a friction torque preload.

Then, the spherical joints between upper arms and lower arms and between the mobile platform and the lower arms are implemented (Fig. 8b and c), imposing non-penetration constraints between solids in order to avoid unrealistic movements of the lower arms.

Additionally, a payload is added to the model by creating a small body and fixing it to the end-effector, as illustrated in Fig. 8d. This definition can be considered a suitable choice, since the considered payload is lightweight (see Table 2) and generally small for parallel manipulators employed in fast pick-and-place operations.

Finally, the dynamic properties of the components are specified, i.e., mass and inertia parameters (see Table 2). Inertia parameters are computed from geometric properties of components, imposing weights and materials.

4.3 Co-simulation model

The co-simulation using Adams and Matlab/Simulink software allows to control a mechanical system built in Adams with the input information from Matlab/Simulink. This second resolution approach allows to solve the dynamics of the manipulator with the multibody model, while the optimization process still runs in Matlab. Then, the optimization results obtained with the mathematical and the multibody models can be compared.

Using the co-simulation, the motor joint position profiles are sent from Matlab/Simulink to Adams, where the dynamics of the parallel robot is solved. Then, the joint torque profiles are sent back to Matlab and used for the computation of the energy consumption. Therefore, in Adams the joints position are set as the input, whereas the torque is selected as the output (Fig. 9a). Figure 9b illustrates model used for co-simulation in Simulink, with mechanical system (represented by the *adams_sub* block), the input and output variables as blocks.

5 Simulation results

This section presents the simulation results obtained from the numerical simulations with both the mathematical and multibody models. It should be noted that the mathematical and the multibody models adopted in the optimization problems solution have some differences that may lead to slight differences in the results. These differences can be related to:

- friction formulation: the friction models implemented in the mathematical and in the multibody models are similar but not the same, as described in Sect. 4.2; consequently, these differences could produce slight variations in the computed torques;
- masses and inertia: the mathematical model includes some simplifying hypothesis in the mass distribution and inertia of the robot components (see Sect. 2.2), whereas the multibody one considers a more realistic geometry and mass distribution;
- joints: in the mathematical model the joints are idealized, since their behavior is represented through equations that model only the relative movement between links, but not consider real constraints like non-penetration ones; differently, the multibody model considers joints with friction and non-penetration constraints, making the behavior of the manipulator more similar to the one of the real robot;

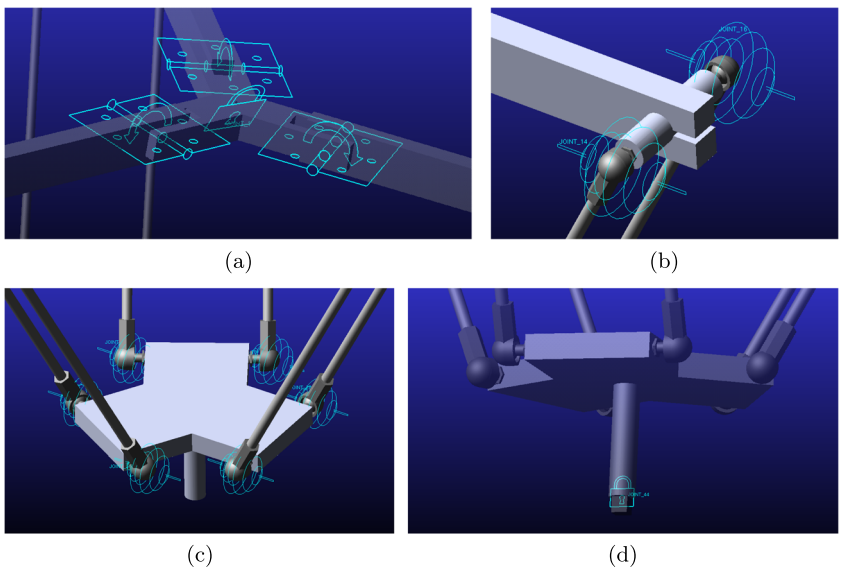


Fig. 8 Detail of fixed and revolute joints (a), of spherical joints between upper arms and lower arms (b), of spherical joints between mobile platform and lower arms (c), and of the payload and how it is fixed (d)

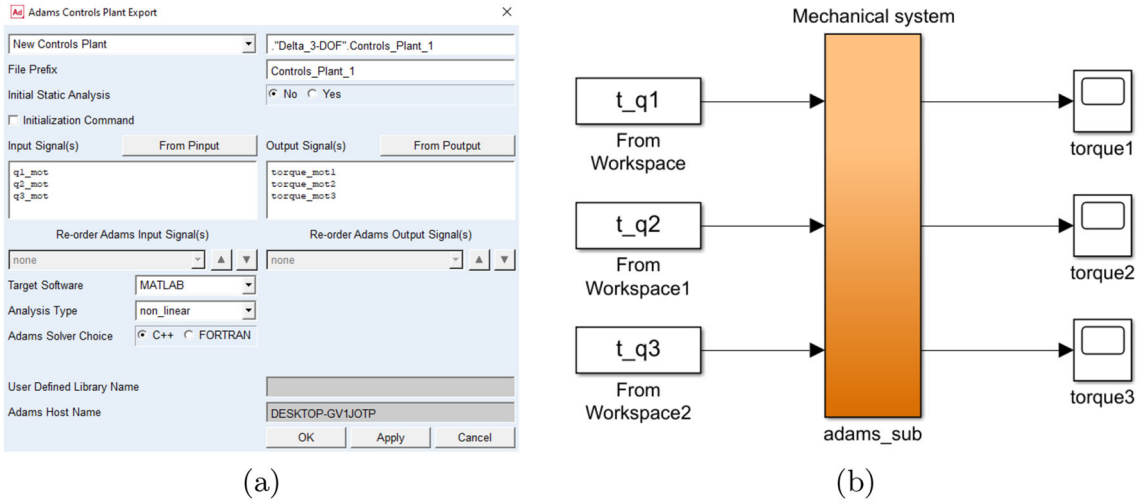


Fig. 9 Parameters selected in Adams for the co-simulation (a), and block diagram of the model used for co-simulation with input and output variables (b)

- dynamics solver: in the multibody model the velocity and acceleration profiles may have non-real oscillations, that could affect the torques computation; differently, the mathematical model does not show this issue;
- linearity: in the mathematical model the behavior of some components is simplified in order to easier solve the dynamics equations and find a single solution; however, these simplifications eliminate certain non-linearities that are still inherent in the multibody model; due to non-linearities, for a same trajectory the torque profile computed with the multibody model could not be always exactly the same.

However, these slight differences between the two models should not produce significant changes in the results. Figure 10a depicts a comparison between the joints torques calculated by Matlab and Adams for the same exemplary trajectory, whereas Fig. 10b provides a visualization of the differences between the computed joints torques. The average error is $2.3 \pm 2.5\ Nm$, highlighting that the differences between the torques computed by the mathematical and the multibody model are minimal.

5.1 Approach (1): optimal task placement

Approach (1) aims at evaluating the optimal positioning of the pick-and-place task inside the robot workspace in order to minimize the total energy consumption, fixing the time distance between each pair of consecutive way points. Approach (1) can be suitable if there are time limits (for instance for production requirements) but the robot can be placed freely with respect to the task to execute. The results obtained adopting the mathematical model for the three different algorithms are reported in Table 4, whereas the ones obtained with the multibody model in co-simulation are reported in Table 5. Figure 11 shows a comparison between the optimal trajectories in the operative space computed by the two different models using the SQP algorithm. Figure 12 represents the same trajectories in the joints space. Finally, Fig. 13 depicts the main phases of the task executed in Adams.

From Table 4 it can be seen that the optimal solutions found with the mathematical model and adopting the three algorithms are basically the same, both for the minimum energy consumption (the differences in minimum energy consumption computed by the three algorithms is lower than 0.1%) and for optimal task placement. Instead, the computation times for the different optimization algorithms are different, especially the one needed by the *ga* function is much greater. In cases with a small number of variables like this, the *fmincon* function (in particular implementing the SQP method) is preferred, since it leads to the same solution in a shorter time. Differently, Table 5 shows that the minimum energy consumption found with the co-simulation model and adopting the three algorithms are very similar (the differences among the optimum solutions found by the three algorithms is lower than 0.2%), but the combinations of parameters that lead to them are quite different. This can be related to the fact that the multibody model simulates the dynamics of the robot with less approximations with respect to the real one, also including some non-linearities. Moreover, the computation times taken by the co-simulation model are higher than the ones needed for the mathematical model. This is due to how the co-simulation works, since for each function evaluation it has to simulate the entire task in Adams.

Comparing the results from Tables 4 and 5, it can be seen that, although the optimum solutions found are slightly different, the consumed energy is almost the same (38 J for the mathematical model and 35 J for the multibody model), with

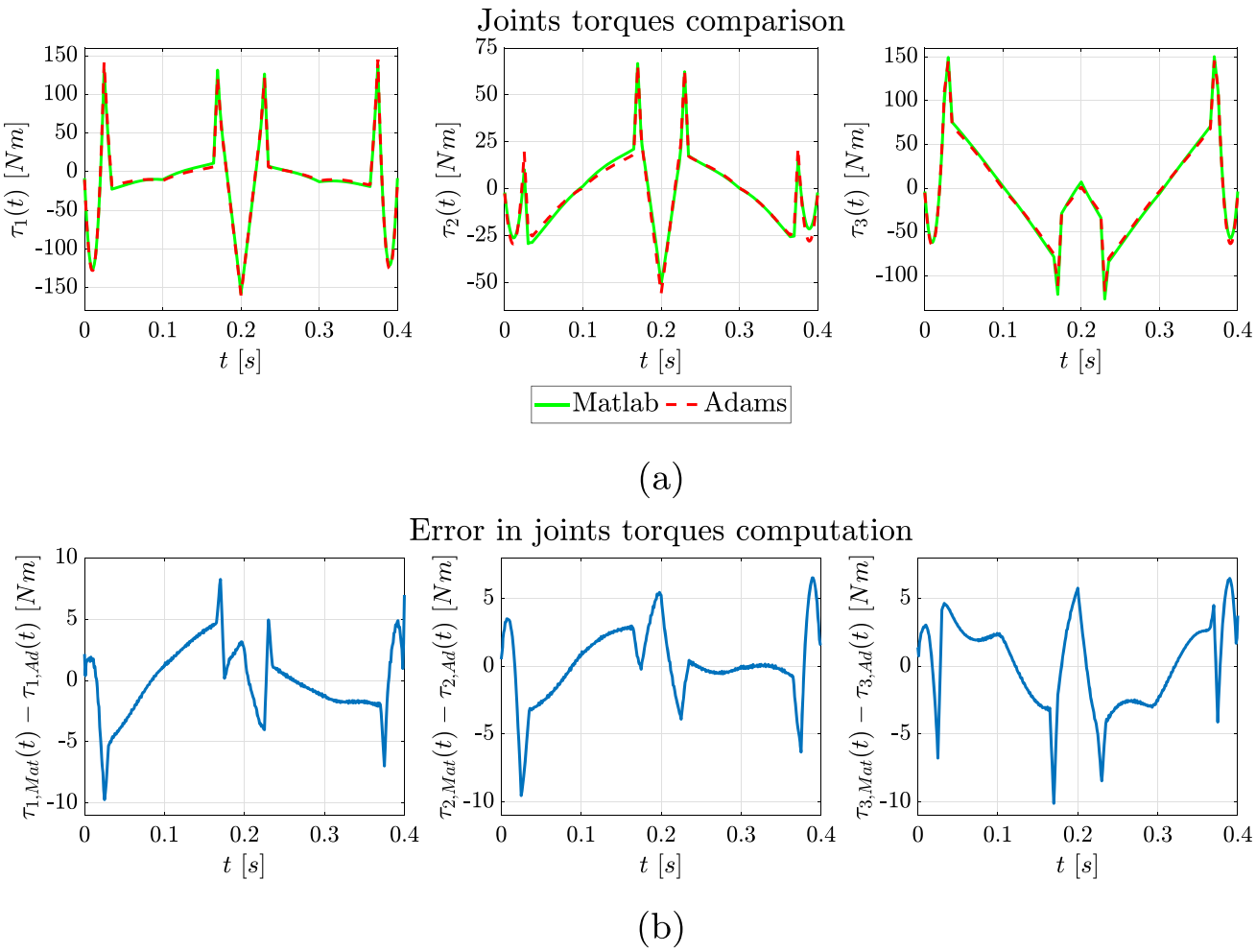


Fig. 10 Example of the joints torques computed by Matlab and Adams for the same exemplary trajectory (a), and difference in computed joints torques (b)

a difference lower than 10%. This can be also highlighted by observing Fig. 12b and d, where optimal velocity and torque profiles obtained with the two models are very similar. However, differences in the position profiles (Figs. 11a and 12a) are present since the mathematical and the multibody models have found slightly different optimal configurations, as it can be seen from Tables 4 and 5. Finally for both models the lowest minimum of energy consumption is obtained with the SQP algorithm.

5.2 Approach (2): optimal time distances between way points

Approach (2) aims at optimizing the execution times of different path sections in order to plan time-energy optimal

Table 4 Approach (1): computation times and results for different algorithms in optimizing task placement in the robot workspace with mathematical model

Algorithm	Interior-point	SQP	GA
$t_{comp,mean}$ $[s]$	64.035	12.150	152.556
$E_{c,\min}$ $[J]$	38.220	38.220	38.222
d $[m]$	0.176	0.176	0.179
ϕ $[rad]$	1.019	1.019	1.076
θ $[rad]$	1.777	1.777	1.723
z $[m]$	−1.052	−1.052	−1.052

Table 5 Approach (1): computation times and results for different algorithms in optimizing task placement in the robot workspace with multibody model

Algorithm	Interior-point	SQP	GA
$t_{comp,mean}$ $[s]$	27922.080	3888.943	344959.975
$E_{c,\min}$ $[J]$	35.330	35.329	35.391
d $[m]$	0.050	0.132	0.000
ϕ $[rad]$	0.124	0.758	0.680
θ $[rad]$	3.115	2.063	0.467
z $[m]$	−1.054	−1.054	−1.053

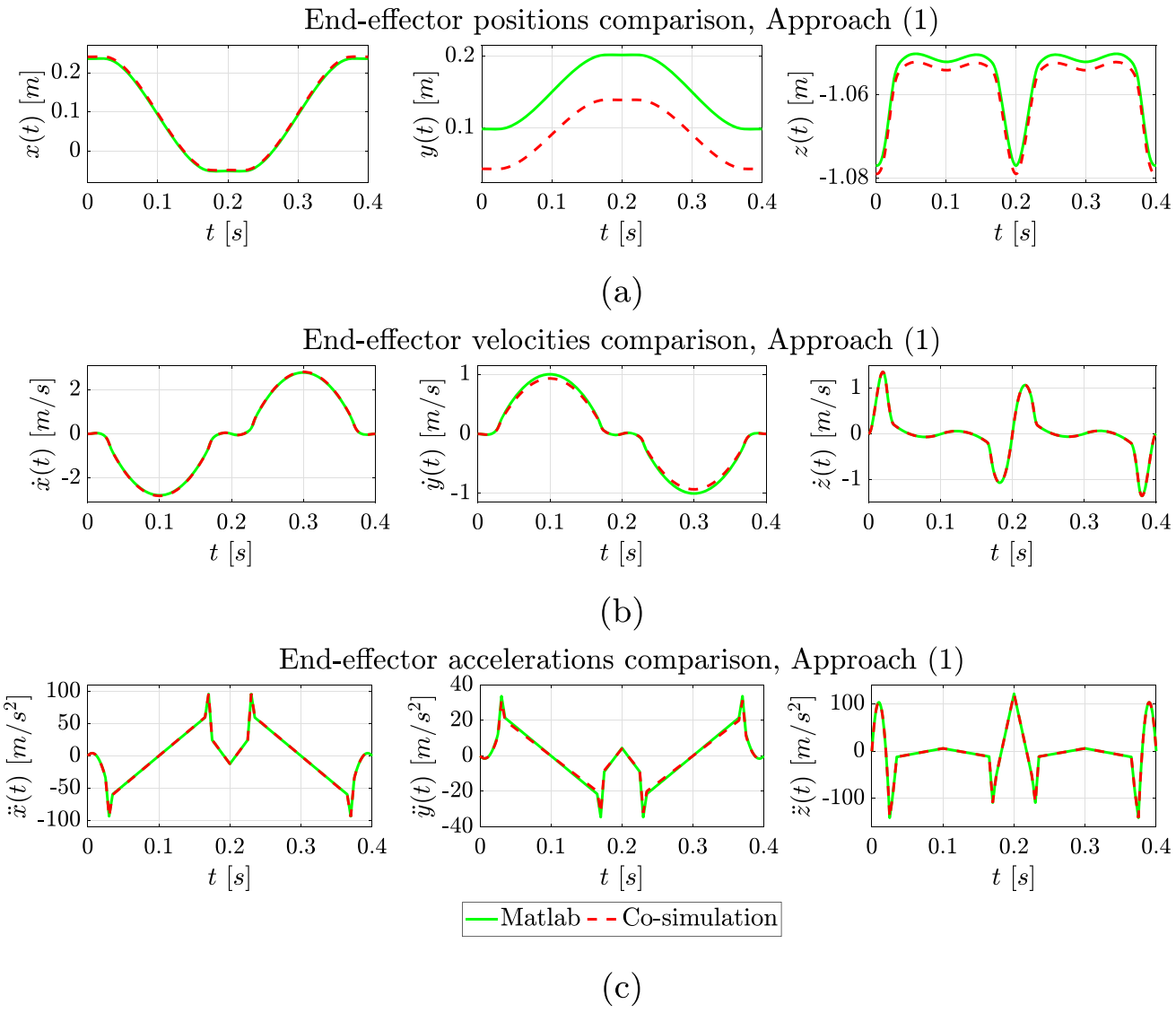


Fig. 11 Comparison between the optimal trajectory in the operative space calculated by the two models for Approach (1): positions (a), velocities (b), and accelerations (c)

trajectories, by keeping the task position fixed. Approach (2) can be exploited when robot placement relative to the task is fixed (for example for space or layout constraints) but the execution times can be modified. Figure 14 depicts how the minimum energy consumption $E_{c,\min}$ changes with respect to the execution time T. This curve presents a minimum in correspondence to the execution time that maximizes the energy saving.

The optimization problem is first solved varying the value of the weight ω in Eq. 23 from 0 to 1. This procedure allows to highlight how the optimal solution changes as the weight given to the energy and time contributions varies. In order to reduce calculation times, the SQP algorithm is implemented for solving the optimization problem.

The results showed in Table 6 evidence that as the ω value increases, the energy consumed increases, while the execution time decreases. The case with $\omega = 0$ equals to the minimum energy consumption without requirements for the execution time. Contrarily, the case with $\omega = 1$ corresponds to the shortest execution time without considering the energy consumption, taking into account especially the constraints for torques and velocities. For example, with $\omega = 1$ the execution time decreases by 85% with respect to the case with $\omega = 0$, but the energy consumed increases by 1105%. Therefore, in this way we can find a trade-off between productivity (i.e., minimum execution time) and operating costs (i.e., energy consumed), varying depending on which of the two requirements is to be favored.

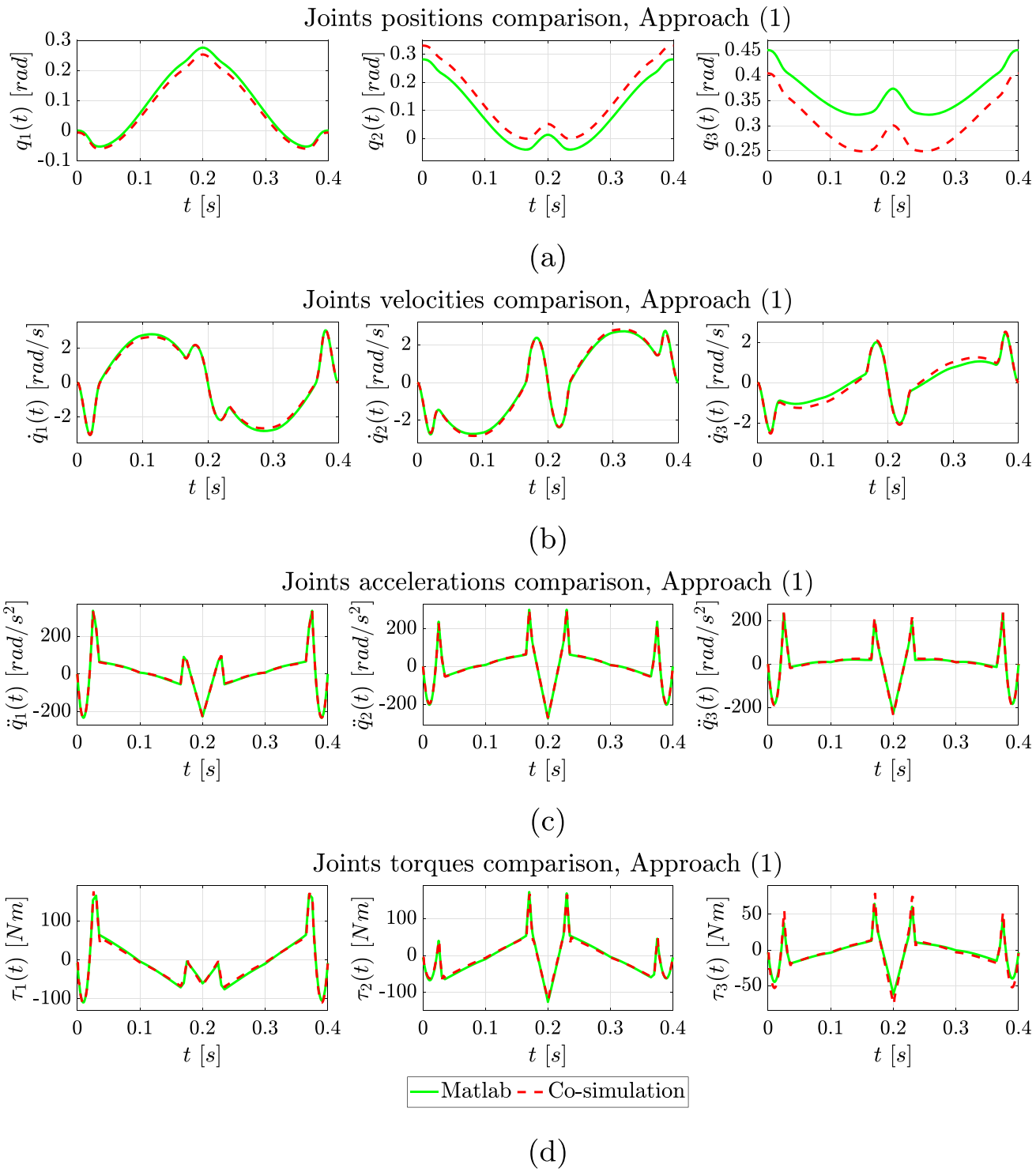
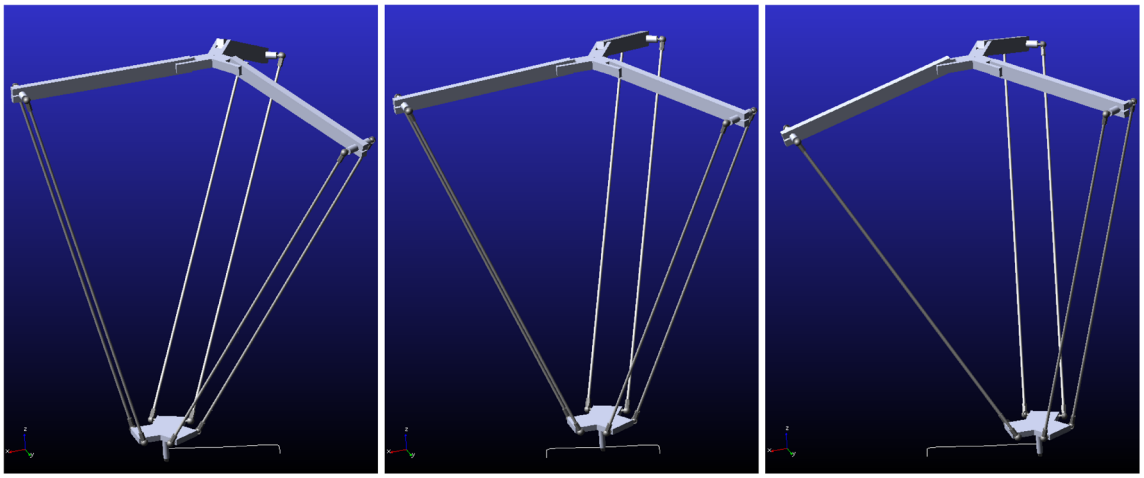


Fig. 12 Comparison between the optimal trajectories in the joint space calculated by the two models for Approach (1): positions (a), velocities (b), accelerations (c), and torques (d)

With the aim of reducing the computation times, an evaluation on how the minimum energy changes by reducing the sampling rate of the trajectories (i.e., by varying the dynamics discretization) is performed. The optimization problem is solved adopting the SQP algorithm and setting $\omega = 0.5$ to equally weight the time duration and the energy consumption of the robot trajectory. The results are reported in Table 7.



(a) $t = 0.000\ s.$ (b) $t = 0.100\ s.$ (c) $t = 0.200\ s.$

Fig. 13 Main task phases for Approach (1): pick the piece (a), move horizontally (b), and place the piece (c)

From Table 7 it can be observed that the dynamics discretization considerably decreases the computation times (with a sampling frequency of 200 Hz the computation time is reduced by 33% compared with the reference case at 1000 Hz), without visibly affecting the optimal solution computed. A slight difference is present with a sampling frequency of 200 Hz, due to the less precise energy calculation, which is done through integration of the consumed power over time (Eq. (19)). However, the optimal solution found with a sampling frequency of 200 Hz is similar to the one obtained with a sampling frequency of 1000 Hz, with differences in minimum energy consumption and execution time of 0.2% and 1%, respectively.

Finally, the influence of the optimization algorithm on the optimization problem is evaluated. Also in this case, the value of ω is set to 0.5 to equally weight the time duration and the energy consumption of the robot trajectory. The results obtained adopting the mathematical model for the

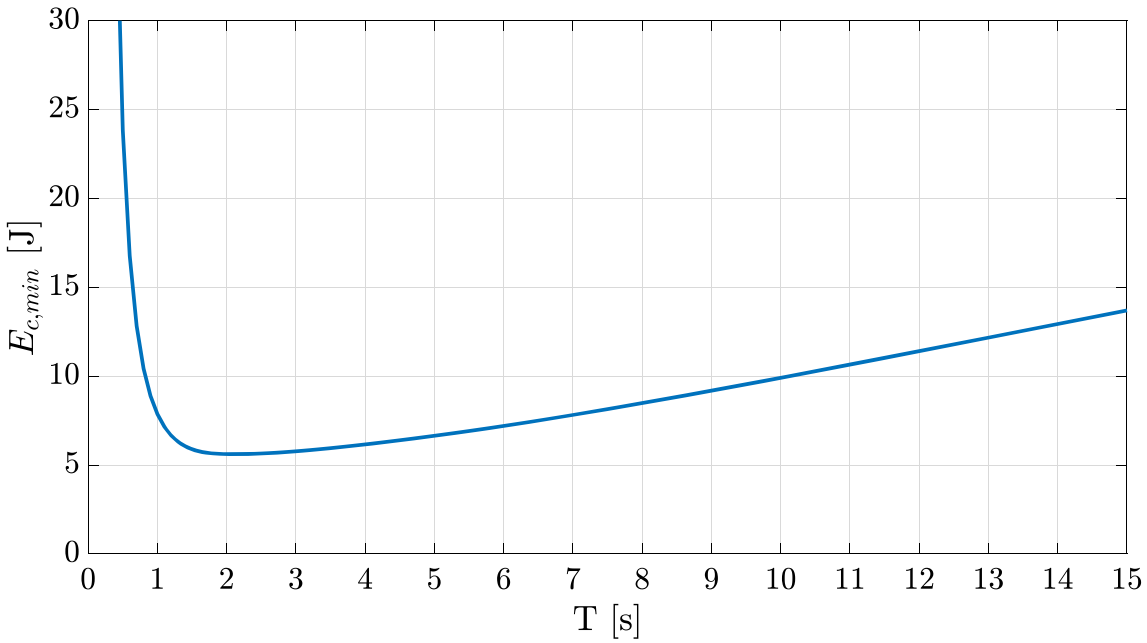


Fig. 14 Minimum energy consumption $E_{c,\min}$ with respect to execution time T

Table 6 Approach (2): Calculation times and results by changing ω in optimizing time distance between way points

ω	$E_{c,min}$ $[J]$	T $[s]$	t_1 $[s]$	t_2 $[s]$	t_3 $[s]$
0.0	5.613	2.124	0.138	0.029	0.335
0.1	5.620	1.992	0.133	0.028	0.309
0.2	5.645	1.852	0.126	0.026	0.285
0.3	5.689	1.720	0.118	0.025	0.262
0.4	5.721	1.660	0.112	0.023	0.257
0.5	5.786	1.584	0.105	0.022	0.247
0.6	5.903	1.488	0.099	0.020	0.233
0.7	6.128	1.372	0.091	0.019	0.214
0.8	6.615	1.212	0.081	0.016	0.190
0.9	8.022	0.980	0.067	0.012	0.154
1.0	67.652	0.316	0.022	0.005	0.047

three different algorithms are reported in Table 8, whereas the ones obtained with the multibody model in co-simulation are reported in Table 9.

Table 8 shows that the optimal solutions for the energy consumption found with the mathematical model and adopting the three algorithms are almost the same, both for the minimum of energy expenditure (the differences in minimum energy consumption computed by the three algorithms is lower than 0.8%) and for the combination of parameters (t_1, t_2, t_3) that leads to it. Regarding the calculation times, the considerations are the same made for Approach (1). Likewise, from Table 9 it can be noticed that the minimums found with the multibody model and adopting the two methods of *fmincon* function are again quite close, both for the minimum energy consumption (the differences among the optimum solutions found by the three algorithms is lower than 1.5%) and for the optimal execution times. Similarly to Approach (1) the computation times taken by the co-simulation model are higher than the one required for the mathematical model, still caused by the co-simulation working. No simulations have been performed in co-simulation with the genetic algorithm due to the high computational times.

Comparing the results from Tables 8 and 9, the differences between minimum energy consumption is about 17%, a value slight higher than the Approach (1) but still acceptable. The optimal parameters computed by the two models

Table 8 Approach (2): Computation times and results for different algorithms in optimizing time distance between the way points with mathematical model

Algorithm	**Interior-point**	**SQP**	**GA**
$t_{comp,mean}$ $[s]$	49.198	9.168	553.684
$E_{c,\min}$ $[J]$	5.784	5.786	5.739
T $[s]$	1.588	1.584	1.648
t_1 $[s]$	0.106	0.105	0.111
t_2 $[s]$	0.022	0.022	0.025
t_3 $[s]$	0.247	0.247	0.251

are also quite similar, differently from Approach (1). Similarly to Approach (1), for both models the lowest minimum is obtained with the SQP algorithm. Furthermore, as it can be seen from Tables 8 and 9, the optimal execution time computed by the mathematical and the multibody models is equal to 1.584 s and 1.452 s, respectively, with a percentage difference of 8%.

5.3 Approach (3): Proposed approach

The results obtained from Approaches (1) and (2) have validated the mathematical model, showing minimal variations between the optimal trajectories computed by the mathematical and the multibody models. Therefore, although there are differences between the mathematical and the multibody models, with the mathematical one energy optimal trajectories can be calculated in a shorter computation time. Then, for Approach (3) only the mathematical model is used in order to limit the computation times.

5.3.1 Approach (3a): Optimizing only lower arms length

Approach (3a) aims at finding the value of b that minimizes the total energy consumption E_c, fixing the task position and the execution time. Approach (3a) can be useful when the robot position with respect to the task and the execution time are predefined (for example for space and productivity constraints), but the robot structure can be modified. For this optimization problem, the *ga* function is not used to solve the optimization problem, since there is only one variable and a

Table 7 Approach (2): computation times and results by varying dynamics discretization in optimizing time distance between way points

Sampling frequency	1000 Hz	500 Hz	333 Hz	250 Hz	200 Hz
$t_{comp,mean}$ $[s]$	10.162	7.958	7.382	7.076	6.764
$E_{c,\min}$ $[J]$	5.786	5.786	5.786	5.789	5.775
T $[s]$	1.584	1.584	1.584	1.580	1.600
t_1 $[s]$	0.105	0.105	0.105	0.104	0.108
t_2 $[s]$	0.022	0.022	0.022	0.022	0.022
t_3 $[s]$	0.247	0.247	0.247	0.247	0.248

Table 9 Approach (2): computation times and results for different algorithms in optimizing time distance between way points with multibody model

Algorithm	Interior-point	SQP
$t_{comp,mean}$ $[s]$	169712.573	44930.65
$E_{c,\min}$ $[J]$	4.685	4.754
T $[s]$	1.660	1.452
t_1 $[s]$	0.113	0.098
t_2 $[s]$	0.026	0.021
t_3 $[s]$	0.250	0.223

genetic algorithm would not be well exploited. The results obtained adopting the mathematical model for the SQP and the interior-point algorithms are illustrated in Table 10. The optimal solutions for the energy consumption found adopting the two algorithms are the same, both for the minimum energy consumption and for the lower arms length. The percentage of energy consumption reduction with respect to the case with the original lower arms length ($b = 1.030\,m, E_c = 12.592\,J$) is 26.6%. Furthermore, the computation times are shorter than the previous approaches, since in this case there is only one variable to optimize.

Previously, the energy saving performance of three different methods for achieving energy efficiency has been evaluated, i.e., optimize the task position within the robot workspace (Approach (1)), plan time-energy optimal trajectories (Approach (2)), and find the proper length of the lower arms (Approach (3a)). In the following, the different methods are applied together with the aim at enhancing energy efficiency in a parallel robot.

5.3.2 Approach (3b): Optimizing lower arms length and task placement

Approach (3b) has the purpose to evaluate the optimal combination of b and task placement inside robot workspace to minimize the total energy consumption E_c, fixing the execution time. Approach (3b) can be applied when there are productivity constraints (i.e., the execution time is limited), but the robot can be moved with respect to the task and its structure can be modified. The optimization problem is solved by varying the optimization algorithm adopted. The

Table 10 Approach (3a): calculation times and results for lower arms length optimization

Algorithm	Interior-point	SQP
$t_{comp,mean}$ $[s]$	2.934	2.348
$E_{c,\min}$ $[J]$	9.286	9.286
b $[m]$	0.808	0.808

Table 11 Approach (3b): computation times and results for different algorithms in optimizing lower arms length and task placement in the robot workspace

Algorithm	Interior-point	SQP	GA
$t_{comp,mean}$ $[s]$	76.532	18.882	461.174
$E_{c,\min}$ $[J]$	8.180	8.180	8.183
b $[m]$	0.527	0.526	0.533
d $[m]$	0.148	0.148	0.150
ϕ $[rad]$	0.004	0.004	0.006
θ $[rad]$	1.570	1.570	1.567
z $[m]$	−0.818	−0.818	−0.823

results depicted in Table 11 show that the optimal solutions computed with the mathematical model and adopting the three algorithms are almost the same, both for the minimum energy consumption (the differences in minimum energy consumption computed by the three algorithms is lower than 0.05%) and for the task placement and the lower arms length. The better solution for minimum energy consumption is obtained with the SQP and the interior-point algorithms. The evaluations on computation times are the same made for Approaches (1) and (2). Furthermore, Approach (3b) achieves a reduction of energy consumption of 12% with respect to the results obtained for Approach (3a), where the execution time is the same but the task placement is not optimized.

5.3.3 Approach (3c): Optimizing lower arms length, task placement and time distance between way points

After demonstrating that combining two different energy efficiency methods can further reduce energy consumption, it can be evaluated whether combining all three previously

Table 12 Approach (3c): computation times and results for different algorithms in optimizing lower arms, task placement in the robot workspace and time execution of different sections

Algorithm	Interior-point	SQP	GA
$t_{comp,mean}$ $[s]$	232.983	39.234	966.904
$E_{c,\min}$ $[J]$	3.772	3.767	3.751
b $[m]$	0.760	0.735	0.927
d $[m]$	0.232	0.242	0.110
ϕ $[rad]$	0.001	0.000	1.013
θ $[rad]$	1.562	1.571	0.044
z $[m]$	−1.131	−1.101	−1.320
T $[s]$	1.776	1.764	1.856
t_1 $[s]$	0.128	0.127	0.130
t_2 $[s]$	0.026	0.025	0.026
t_3 $[s]$	0.264	0.264	0.282

described methods can significantly reduce energy consumption compared to the minimums found with the previous approaches. Therefore, in Approach (3c) the goal is to find the optimal combination of b, task placement inside robot workspace and execution times of different path sections to planning time-energy optimal trajectories. Approach (3c) can be useful when robot and task configuration can be modified (in this case, task placement with respect to the robot, time execution and robot structure). The optimization problem is solved by varying the optimization algorithm adopted, whereas the value of ω is chosen equal to 0.5 in order to equally weight the energy consumption and the time duration of the robot trajectory. The results obtained adopting the mathematical model for the three different algorithms are shown in Table 12. Figure 15 shows the optimal trajectory in the operative space computed adopting the SQP algorithm, whereas Fig. 16 represents the same trajectory in the joints space.

The time-energy optimal trajectories computed with the three algorithms similar, both for the minimum of energy consumption (the differences in minimum energy expenditure computed by the three algorithms is lower than 0.6%) and for the combination of parameters (b, d, ϕ, θ, z, t_1, t_2, t_3) that leads to it. The computation times are higher than the ones needed by the other approaches, mainly due to the higher number of considered optimization variables. Similarly to the previous approaches, the lowest minimum is obtained adopting the SQP algorithm.

As illustrated in Approach (2), by varying the value of the weight ω in Eq. 28 the relevance of the two contributions (energy and time) can be modified, allowing planning trajectories that give greater priority to one of the two requirements.

Approach (3c) highlights that applying together different energy efficiency strategies can further improve the energy saving for the executed task. Indeed, the optimization of all the eight parameters considered (b, d, ϕ, θ, z, t_1, t_2, t_3) gives

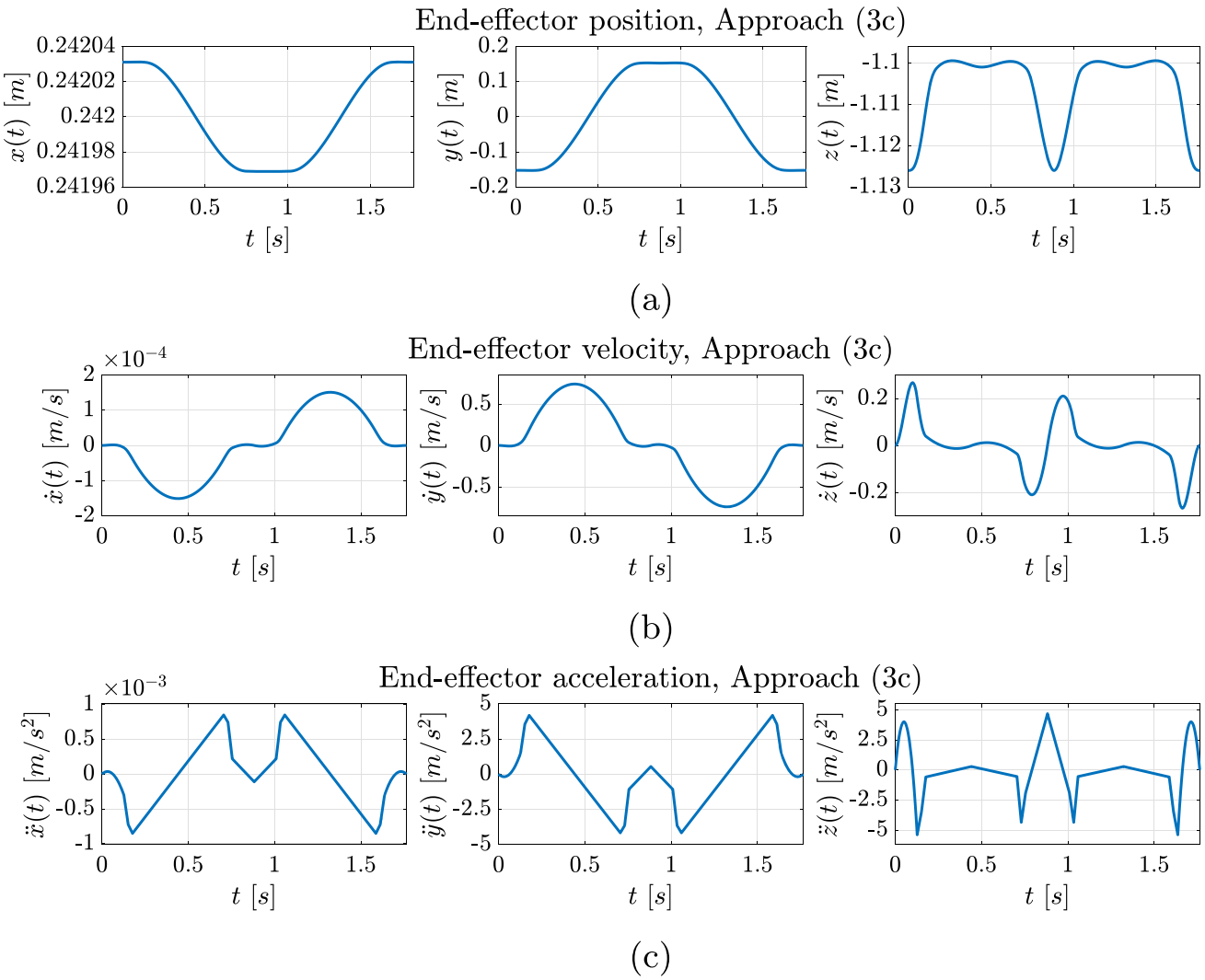


Fig. 15 Optimal trajectory in the operative space computed with the mathematical model for Approach (3c): positions (a), velocities (b), and accelerations (c)

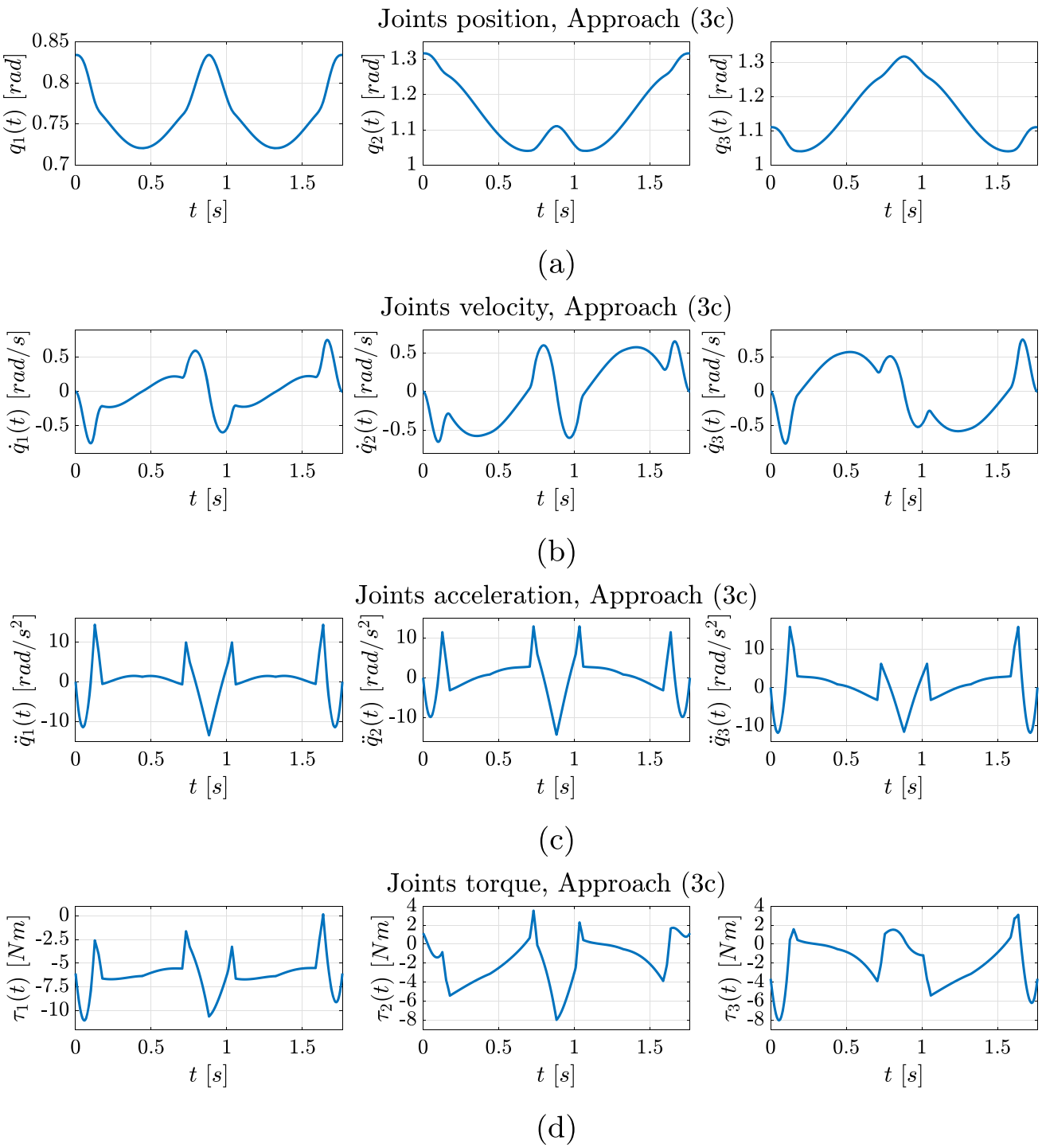


Fig. 16 Optimal trajectory in the joint space computed with the mathematical model for Approach (3c): positions (a), velocities (b), accelerations (c), and torques (d)

good results and the task is executed by the robot with an energy consumption of only 3.767 J. For instance, with respect to optimizing only the execution times (Approach (2)) the energy expenditure decreases by 35%, whereas the execution time remains almost the same. This reduction in energy consumption can lead to sustainability of industrial

manufacturing processes and to lower energy costs especially in robots that continuously perform high-speed tasks in the industry environment.

6 Conclusions

In this paper, we presented an approach for the energy-efficiency optimization of a 3-DOF parallel robot. The proposed strategy leverages the task placement, the execution time, and the length of the robot lower arms to minimize the energy consumption for the execution of a predefined high-speed pick-and-place operation. To evaluate the actuators energy consumption, the kinematic, dynamic and electro-mechanic mathematical models, as well as an equivalent multibody model, of the parallel robot have been implemented. The results of extensive numerical simulations showed that the proposed strategy provides notable improvements in the energy efficiency of the parallel robot, with respect to alternative approaches.

More in detail, starting from a pick-and-place task with optimal task placement with a consumption of 38.2 J (with a cycle time of 0.4 s), the energy expenditure can be reduced to 3.75 J (with a cycle time of 1.86 s), with a reduction percentage of 90.2%, by additionally optimizing the execution time, and the length of the robot lower arms. These results lead to a reduction from 5733 J/min (for 150 cycles/min) to 121 J/min (for 32 cycles/min). Furthermore, compared to Approach (2), which only considers the execution time, the energy expenditure found with the proposed strategy decreases by 34.8%, whereas the execution time increases by 11.8%. With respect to Approach (3a), which considers only the length of the lower arms, the energy consumption decreases by 59.4% and the execution time raises by 110.0%. Finally, compared to Approach (3b) that optimizes task placement and the length of the lower arms, Approach (3c) reduces the energy consumption by 53.9%, with an increase of the execution time of 110.0%. These comparisons show that the energy saving with Approach (3c) is higher than both approaches that exploit only one energy efficiency strategy (Approaches (1), (2), and (3a)) and Approach (3b), which optimizes both the task placement and the length of the lower arms. However, the execution time of the pick-and-place trajectory increases. The results of this work highlight the importance of properly setting the execution time of the task, in addition to optimizing the task placement and the robot physical structure, in order to further reduce the energy consumption, allowing to choose the best trade-off between robot productivity and consumed energy.

The limitations of the proposed approach include the need of an optimization routine that can result in high computational times to solve the energy minimization problem. Furthermore, the proposed approach does not only require the modification of the software trajectory parameters, but also the replacement of a physical component of the robotic system, i.e., the lower robot arms. However, this operation can be performed without disassembling the whole robot, nor disconnecting the robot from its support structure.

In future work, we plan to experimentally validate the results of the proposed optimization approach with consumed energy measurements on real parallel robotic systems in which voltage and current data are available. Furthermore, we will also explore alternative strategies for energy efficiency in parallel robots, such as exploitation of the natural dynamics, and the optimization of additional physical components.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Giuliano Fabris and Lorenzo Scalera. The first draft of the manuscript was written by Giuliano Fabris, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Università degli Studi di Udine within the CRUI-CARE Agreement. This research has been developed within the Laboratory for BigData, IoT, Cyber Security (LABIC) funded by Friuli Venezia Giulia region, and the Laboratory for Artificial Intelligence for Human-Robot Collaboration (AI4HRC) funded by Fondazione Friuli. This study was carried out within the Interconnected Nord-Est Innovation Ecosystem (iNEST) and received funding from the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) - Mission 4.2, Investment 1.5 - D.D. 1058 23/06/2022, ECS 00000043). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Availability of data and materials Not applicable.

Code availability Not applicable.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Conflict of interest Not applicable.

References

1. Faure C, Guetlein M-C, Schleich J, Tu G, Whitmarsh L, Whittle C (2022) Household acceptability of energy efficiency policies in the European Union: Policy characteristics trade-offs and the role of trust in government and environmental identity. Ecol Econ 192:107267
2. Wang E-Z, Lee C-C, Li Y (2022) Assessing the impact of industrial robots on manufacturing energy intensity in 38 countries. Energy Econ 105:105748
3. Chen Y, Cheng L, Lee C-C (2022) How does the use of industrial robots affect the ecological footprint? International evidence. Ecol Econ 198:107483
4. Xiao W, Han G, Ally AS, Chen X (2023) Energy consumption modeling and parameter identification based on system decomposition of welding robots. Int J Adv Manuf Technol 1–16
5. International Energy Agency: IEA, Energy system, Industry. https://www.iea.org/energy-system/industry. Accessed 21 July 2023
6. Bugday M, Karali M (2019) Design optimization of industrial robot arm to minimize redundant weight. Engineering Science and Technology, an International Journal 22(1):346–352
7. Yin H, Liu J, Yang F (2019) Hybrid structure design of lightweight robotic arms based on carbon fiber reinforced plastic and aluminum alloy. IEEE Access 7:64932–64945
8. Khalaf P, Richter H (2019) Trajectory optimization of robots with regenerative drive systems: numerical and experimental results. IEEE Transactions on Robotics. 36(2):501–516
9. Boscariol P, Richiedei D (2022) Energy optimal design of servo-actuated systems: a concurrent approach based on scaling rules. Renew Sustain Energy Rev 156:111923
10. Carabin G, Scalera L, Wongratanaphisan T, Vidoni R (2021) An energy-efficient approach for 3D printing with a Linear Delta Robot equipped with optimal springs. Robotics and Computer-Integrated Manufacturing 67:102045
11. Bettega J, Richiedei D, Tamellin I, Trevisani A (2023) Reducing energy consumption and driving torque in an underactuated robotic arm through natural motion. In: International workshop IFToMM for sustainable development goals, pp 89–96. Springer
12. Bukata L, Šůcha P, Hanzálek Z, (2019) Optimizing energy consumption of robotic cells by a Branch & Bound algorithm. Comput Oper Res 102:52–66
13. Benotsmane R, Kovács G (2023) Optimization of energy consumption of industrial robots using classical PID and MPC controllers. Energies 16(8):3499
14. Althoff M, Giusti A, Liu SB, Pereira A (2019) Effortless creation of safe robots from modules through self-programming and self-verification. Sci Robot 4(31):1924
15. Vidussi F, Boscariol P, Scalera L, Gasparetto A (2021) Local and trajectory-based indexes for task-related energetic performance optimization of robotic manipulators. J Mech Robot 13(2):021018
16. Zhang M, Yan J (2021) A data-driven method for optimizing the energy consumption of industrial robots. J Clean Prod 285:124862
17. Ghungrad S, Mohammed A, Haghighi A (2023) Energy-efficient and quality-aware part placement in robotic additive manufacturing. J Manuf Syst
18. Stan L, Nicolescu AF, Pupăză C, Jiga G (2023) Digital twin and web services for robotic deburring in intelligent manufacturing. J Intell Manuf 34(6):2765–2781
19. Zhang S, Xia Q, Chen M, Cheng S (2023) Multi-objective optimal trajectory planning for robotic arms using deep reinforcement learning. Sensors 23(13):5974
20. Dona' D, Lenzo B, Rosati G (2023) Planning real-time energy efficient trajectories for a two degrees of freedom balanced serial manipulator. In: International workshop IFToMM for sustainable development goals, pp 81–88. Springer
21. Carabin G, Scalera L (2020) On the trajectory planning for energy efficiency in industrial robotic systems. Robotics 9(4):89
22. Van Oosterwyck N, Vanbecelaere F, Knaepkens F, Monte M, Stockman K, Cuyt A, Derammelaere S (2022) Energy optimal point-to-point motion profile optimization. Mechanics Based Design of Structures and Machines 1–18
23. Boscariol P, Richiedei D (2019) Energy-efficient design of multi-point trajectories for Cartesian robots. Int J Adv Manuf Technol 102:1853–1870
24. Gadaleta M, Pellicciari M, Berselli G (2019) Optimization of the energy consumption of industrial robots for automatic code generation. Robotics and Computer-Integrated Manufacturing 57:452–464
25. Li Y, Huang T, Chetwynd DG (2018) An approach for smooth trajectory planning of high-speed pick-and-place parallel robots using quintic B-splines. Mechanism and Machine Theory 126:479–490
26. Mo J, Shao Z-F, Guan L, Xie F, Tang X (2017) Dynamic performance analysis of the X4 high-speed pick-and-place parallel robot. Robotics and Computer-Integrated Manufacturing 46:48–57
27. Leveziel M, Laurent GJ, Haouas W, Gauthier M, Dahmouche R (2022) A 4-DoF parallel robot with a built-in gripper for waste sorting. IEEE Robotics and Automation Letters 7(4):9834–9841
28. Aliakbari M, Mahboubkhah M (2020) An adaptive computer-aided path planning to eliminate errors of contact probes on free-form surfaces using a 4-DOF parallel robot CMM and a turn-table. Measurement 166:108216
29. Song X, Pan Y, Chen Y (2015) Development of a low-cost parallel kinematic machine for multidirectional additive manufacturing. J Manuf Sci Eng 137(2):021005
30. Qin X, Li Y, Feng G, Bao Z, Li S, Liu H, Li H (2024) A novel surface topography prediction method for hybrid robot milling considering the dynamic displacement of end effector. The International Journal of Advanced Manufacturing Technology, 1–14
31. Scalera L, Carabin G, Vidoni R, Wongratanaphisan T (2019) Energy efficiency in a 4-DOF parallel robot featuring compliant elements. International Journal of Mechanics and Control. 20(2):49–57
32. Balderas Hill R, Briot S, Chriette A, Martinet P (2021) Minimizing the energy consumption of a Delta robot by exploiting the natural dynamics. In: ROMANSY 23-robot design, dynamics and control: Proceedings of the 23rd CISM IFToMM Symposium 23, pp 213–221. Springer
33. Mora JP, Barreto JP, Rodriguez CF (2022) Energy optimization of a parallel robot in pick and place tasks. In: Multibody Mechatronic Systems: MuSMe 2021 7, pp 191–200. Springer
34. Chen D, Li S, Wang J, Feng Y, Liu Y (2019) A multi-objective trajectory planning method based on the improved immune clonal selection algorithm. Robotics and Computer-Integrated Manufacturing. 59:431–442
35. Zhang X, Ming Z (2019) Trajectory planning and optimization for a Par4 parallel robot based on energy consumption. Appl Sci 9(13):2770
36. Scalera L, Boscariol P, Carabin G, Vidoni R, Gasparetto A (2020) Enhancing energy efficiency of a 4-DOF parallel robot through task-related analysis. Machines. 8(1):10
37. Scalera L, Boscariol P, Carabin G, Vidoni R, Gasparetto A (2021) Optimal task placement for energy minimization in a parallel manipulator. In: New advances in mechanisms, mechanical transmissions and robotics: MTM & robotics 2020 2, pp 12–22. Springer
38. Liu X, Bi W, Xie F (2019) An energy efficiency evaluation method for parallel robots based on the kinetic energy change rate. Sci China Technol Sci 62:1035–1044

39. Lee G, Park S, Lee D, Park FC, Jeong JI, Kim J (2015) Minimizing energy consumption of parallel mechanisms via redundant actuation. IEEE/ASME Transactions on Mechatronics 20(6):2805–2812
40. Merlet J-P (2006) Parallel Robots, vol 128. Springer, Dordrecht, NL
41. Hashlamon I (2020) Adaptive Disturbance Estimation and Compensation for Delta Robots. Jordan Journal of Mechanical & Industrial Engineering 14(4)
42. Cook C, Ho C (1984) The application of spline functions to trajectory generation for computer-controlled manipulators. Computing Techniques for Robots. Springer, Boston, MA, pp 101–110