International Journal on Interactive Design and Manufacturing (IJIDeM)
https://doi.org/10.1007/s12008-025-02443-y

ORIGINAL ARTICLE

An integrated engineering approach for the preliminary design and synthesis of delta robots

Antonio Pandolfi[1,2] · **Pietro Bilancia**[2] · **Marcello Pellicciari**[2]

Received: 24 January 2024 / Revised: 10 October 2025 / Accepted: 1 November 2025
© The Author(s) 2025

Abstract

The design of delta robots poses significant challenges as their mechanical behavior depends on a high number of dimensional parameters and dynamic factors. This is further compounded by the presence of demanding performance requirements, particularly in terms of position accuracy during high-dynamics motion tasks. By leveraging theoretical models, dynamic optimization techniques and advanced simulations, the present paper aims to streamline the design process, providing a structured engineering method and tool to address the dimensional synthesis of delta robots, encompassing kinematics, dynamics, link flexibility, and ball joint clearance. The systematic design process incorporates user requirements, including bounding box specifications, cycles per minute for pick-and-place operations, end-effector accuracy tolerance, maximum static payload, and cost minimization. The methodology involves an initial dynamic optimization phase employing a genetic algorithm to derive optimal dimensional parameters. Analytical models implemented in Matlab expedite the iterative optimization process. Then, the optimized design is virtually prototyped in RecurDyn flexible multibody simulation tool for validation by including the link flexibility and the effect of ball joint clearances. The iterative approach ensures that the final design aligns with user expectations. Additionally, the paper addresses motor selection based on torque requirements and proposes an approach for evaluating the robot performance in terms of maximum end-effector acceleration and payload. Finally, the efficacy of the tool is evaluated through a case study focused on designing a manipulator as an integral part of a collaborative research project with an industrial partner.

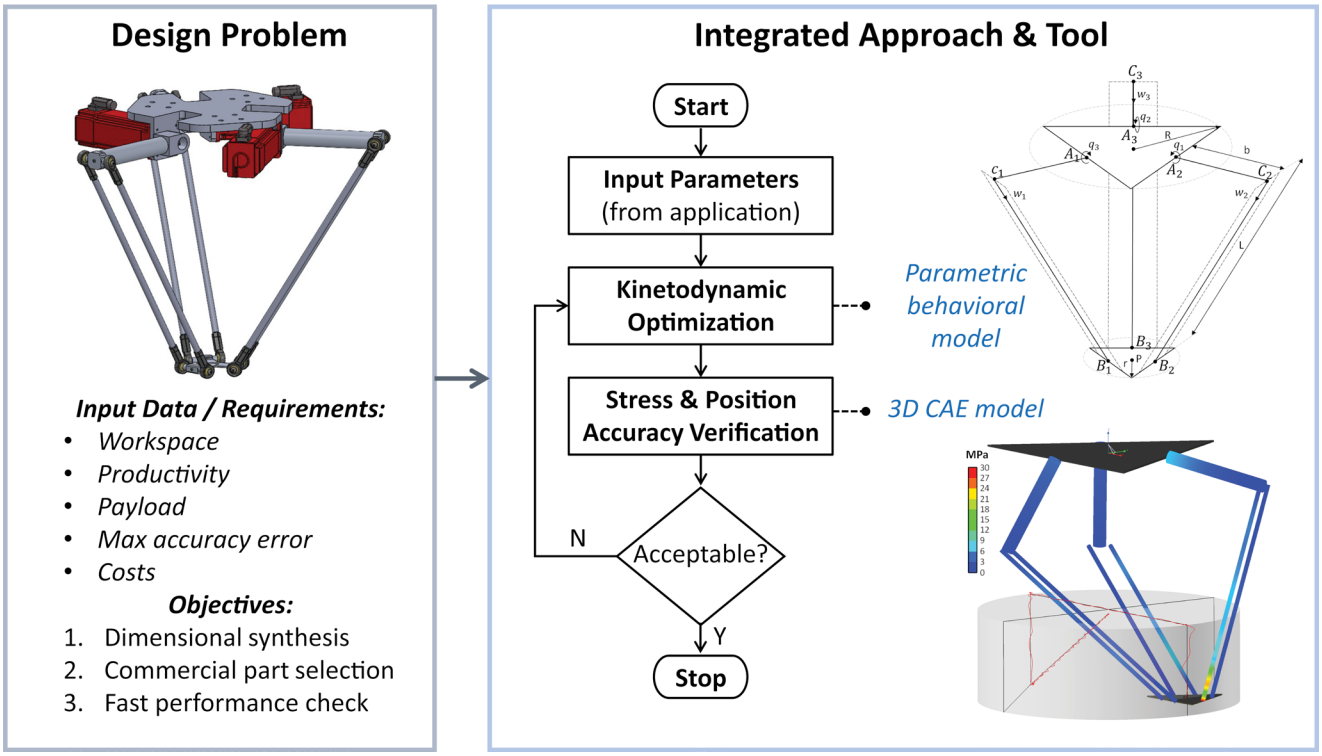
✉ Pietro Bilancia
pietro.bilancia@unimore.it

[1] K-LOOPS S.R.L, Modena 41125, Italy

[2] Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Reggio Emilia 42122, Italy

Published online: 28 November 2025

Graphical Abstract



Keywords Industrial robotics · Delta robot · Design tool · Optimization · Multibody simulation

1 Introduction

In the context of I4.0, industrial robotics has become one of the most valuable keys for factory progress and industrial innovations [1]. Several types of robotic tools and devices have played a crucial role in enhancing industrial efficiency, reducing losses and providing the adaptability required to navigate the ever-evolving landscape of market dynamics and competition. Industrial robots are generally divided into two main groups based on their kinematic architecture: serial robots and parallel robots. The former presents a single chain of joints, offering great versatility and dexterity, while the latter has multiple chains of joints. In recent years, significant research efforts have been dedicated to the design and advancement of parallel robots, driven by their superior attributes compared to serial manipulators, namely high dynamics and productivity [2, 3]. One of the most famous parallel manipulators is the Clavel Delta, which has been widely recognized for its performance in tasks such as fast pick-and-place operations [4]. Originally designed as a manipulator for handling lightweight and small objects, the Clavel has been successfully adopted in many applications across industries. In general, delta robots have found use in many domains, i.e. from driving simulators and medical devices to deployment in entertainment industries and vibration isolators [5]. However, due to their light structure and high speed, the operational efficiency of the delta robots can be compromised by various kinematic and dynamic effects arising from factors such as joint clearances, friction or link flexibility [6]. Their dynamic performance and position accuracy are heavily influenced by the mutual interaction between the mechanical structure and the control system. As a result, the state-of-the-art separated mechanical design and control development approaches could be expensive and may not result in significant improvements even after numerous iterations. A more effective approach would incorporate changes in the control system during the mechanical design, by developing and integrating advanced control modules with predictive capabilities within CAD-based environments. This would allow for fast assessment and correction of manipulator errors from the earliest design stages, offering a more cost-effective and efficient method for enhancing its overall performance. By creating a comprehensive virtual prototype that combines both mechanical design and control systems, this approach not only optimizes the design but also facilitates the creation of a high-fidelity framework for control system development, deployment, and continuous optimization. Modern

commercial mechanical simulation tools, such as RecurDyn or ADAMS, offer direct interfaces with MATLAB/Simulink or Python, facilitating the co-simulation of the robot's mechanical behavior alongside its control system, as detailed in [7–12]. This integration ensures that both systems are optimized together, enabling the implementation of predictive control strategies that adapt to real-world conditions, thereby minimizing errors and enhancing performance throughout the robot's operational lifecycle. Then, considering that integrating innovative control modules into existing commercial robot controllers may pose challenges, during commissioning it becomes necessary to explore open and programmable commercial solutions where novel custom control modules can be implemented [13–15].

The scientific literature presents various methodologies for achieving an optimal design of parallel kinematic manipulators and, in particular, delta robots. A common approach addresses the drawback of a limited workspace in parallel manipulators. Such methods seek to identify the manipulator geometric dimensions that align with the prescribed workspace while also avoiding kinematic singularities [16–18]. Although the workspace is one of the most crucial features for a robot, it is not the only parameter to consider during the optimization process. Indeed, other proposed approaches are based on kinematic optimization to reduce the end-effector positioning errors [19, 20], dynamic optimizations centered on the minimization of manipulator torques and motors' power [21], and kineto-static optimization, which incorporates considerations of both kinematics and link flexibility within static conditions [22]. It can be observed that, despite the well-known substantial effects on the manipulator accuracy caused by the link flexibility in dynamic conditions and the clearance of the joints highlighted by numerous studies [6, 23–25], there is still no design methodology that allows for the consideration of these effects in the design phase due to their inherent analytical complexity.

Owing to these considerations, the purpose of this paper is to describe an integrated method and tool for determining the optimal dimensions of a delta manipulator, considering not only kinematics and dynamics aspects but also accounting for the flexibility of the links and the clearance of the ball joints. Analytical models are developed and implemented within Matlab scripts, which are then utilized for behavioral analysis and optimization purposes. The advantage of employing analytical models lies in their ability to speed up the optimization process and their easy integration into the future control system. Furthermore, their versatility enables application for various purposes, as demonstrated in the paper, such as evaluating the dynamic performance of the manipulator. Finally, the identified optimal candidate is verified using a commercial flexible multibody solver (RecurDyn), where a dynamic simulation is run to assess both the overall end-effector position accuracy error and the structural integrity with an imposed motion profile. Apart from its advanced simulation features, e.g. the possibility to combine both rigid and flexible bodies in a single dynamic simulation environment, RecurDyn has been adopted in this study due to its ability to be executed in batch mode from external environments (e.g. Matlab) [26, 27], making it particularly advantageous for future applications, including the testing of novel control algorithms.

The remaining of the paper is organized as follows: Sect. 2 outlines the conceptual steps leading to the definition of the design method, Sect. 3 reports the analytical modeling of the kinematics and dynamics of the manipulator, Sect. 4 presents the implementation of the proposed design method, whereas Sect. 5 shows its application on a case study. The concluding remarks are given in Sect. 6.

2 Delta robot design overview and methodology

The 3D CAD drawing of a delta robot is depicted in Fig. 1(a), which consists of a base, a moving platform and three identical kinematic chains that establish a connection between the base and the moving platform (end-effector). Each kinematic chain is composed of three links, namely an active arm and two passive arms arranged in a parallelogram configuration. The movement of the platform is constrained to pure translation, facilitated by ball joints connecting the active arm to the passive arms and the passive arms to the moving platform, respectively.

The active arms are actuated by servomotors, which are fixed to the upper base. Typically, the links are made from lightweight materials such as carbon fiber, reinforced plastics or aluminum alloys to minimize inertias, thereby enhancing the dynamic performance of the robot. Then, depending on the application, a specific tool is mounted on the moving platform. Figure 1(b) shows a schematic representation of the manipulator including the main dimensions considered during the kinematic synthesis. The base and the moving platform can be conceptualized as two triangles inscribed in circumferences with radii R and r, respectively. L represents the length of the passive arm, b is the length of the active arm, and H is the distance between the workplane and the base. The passive arms typically feature a cylindrical design, whereas the active arms may present a variety of different shapes.

Achieving an optimal design is a complex process due to the high number of involved parameters. The size and geometry of the manipulator are intricately linked to the intended workspace as this defines the volume in which

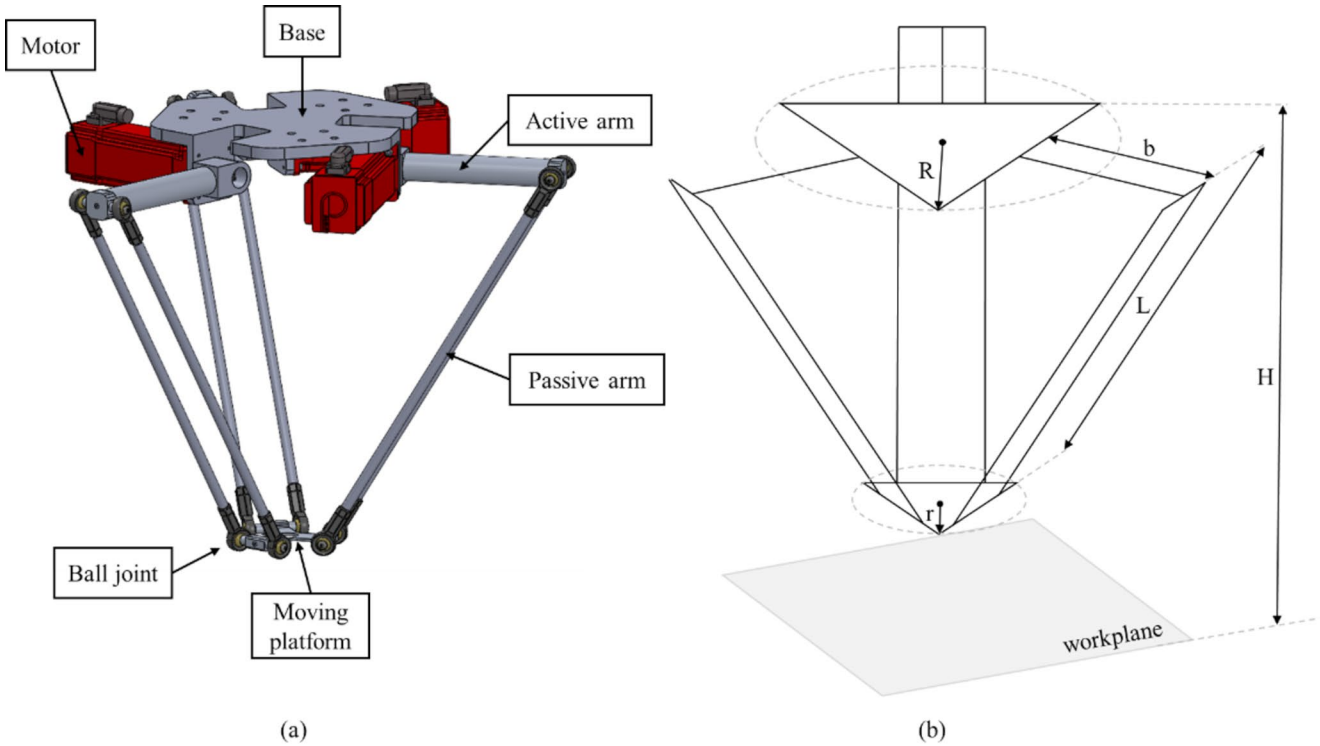


Fig. 1 Delta manipulator: (**a**) 3D embodiment design, (**b**) schematic representation

the end-effector operates. Therefore, its synthesis involves a comprehensive assessment of reachability and singularity avoidance for all points within the workspace. Moreover, the dimensions of the manipulator are notably influenced by the specific task requirements and the maximum acceleration assigned to the end-effector. This becomes particularly critical for a delta robot with demanding operability requirements (cycles/min). Under these circumstances, there is a potential risk of oversizing the motors, which could result in increased costs. On the other hand, reducing the cross-section of the links may lead to significant deformations during operation. In fact, coupled with the unavoidable clearance in the ball joints, these significantly compromise the end-effector position accuracy.

Given the high number of interconnected parameters, it is crucial to establish a systematic process for the mechanical design of the delta robot. Such process inevitably starts with an in-depth analysis of the user requirements, which typically include:

- **Bounding box**: The spatial constraints within which the robot is intended to operate. Users are required to specify the dimensions of the box ($\mathrm{A} \times \mathrm{B} \times \mathrm{C}$ mm).
- **Cycles Per Minutes (CPM)**: Pertaining to pick-and-place operations, users specify the desired cycle rate, also evaluating a specific payload (including the tool attached to the end-effector).
- **Maximum static payload [kg]**: The maximum load that the manipulator can carry under static conditions, providing crucial information for structural considerations.
- **Cost minimization**.
- **Maximum end-effector accuracy error e [mm]**: Based on the accuracy requirements of the intended process/application, users define the maximum allowable error for the robot end-effector.

Following the flowchart shown in Fig. 2, once the end-effector workspace is defined, by taking into consideration the CPM that the manipulator must provide with a specified payload, trajectories for pick-and-place operations can be generated, e.g. with their initial and final points placed on the workspace borders. This circumstance leads to maximum accelerations and is further compounded by the manipulator being in an unfavorable position in terms of payload capability [4]. The computation of mobile platform trajectories allows for the consideration of dynamic effects of the manipulator, enabling the calculation of motor torques required to achieve the desired motion. The most favorable configuration will be the one that, while maintaining the same end-effector trajectory, yields lower torque profiles compared to other configurations. This choice aims to minimize energy consumption and motor size, meeting the cost reduction requirement. Since torque profiles are often irregular, direct comparison becomes challenging.

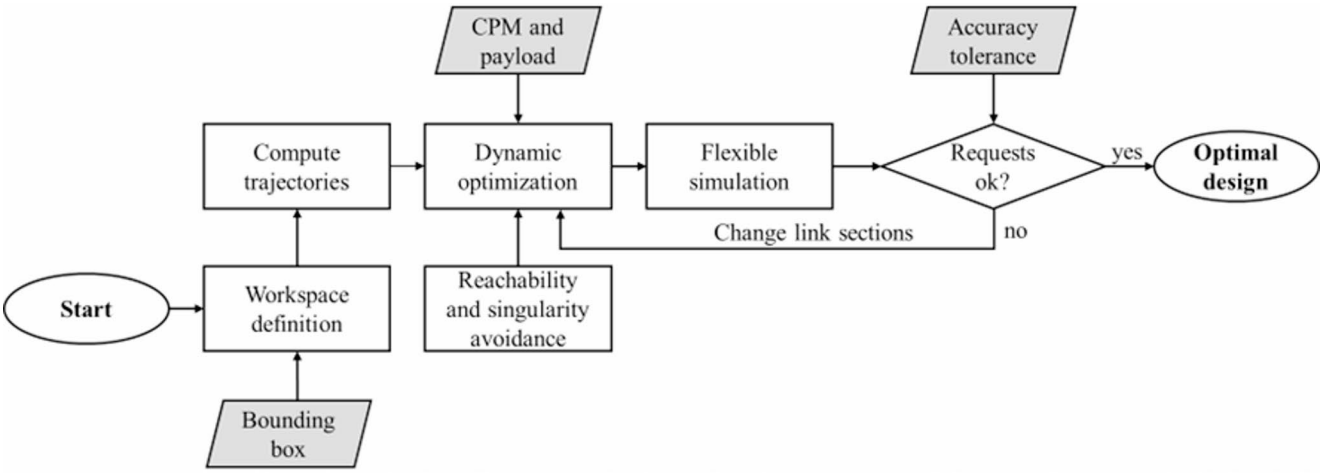


Fig. 2 Proposed flow chart for the optimal design of a delta robot

Consequently, an evaluation parameter, such as the torque Root Mean Square (RMS), is used for a more straightforward comparison. Specifically, the RMS is calculated for all motors and the highest value (indicating the most stressed motor) is selected as the comparison index for the optimization. It is evident that this optimization should also consider the kinematic aspects in order to enable the traversal of all points in the workspace without encountering singular configurations and avoid impacts with the frame. Thus, the configurations that do not satisfy these conditions are no further considered.

The sizing process outlined so far enables the derivation of a parameter set that meets almost all the user inputs listed above, except the last one. To bridge this gap and conclude the process, the obtained configuration must undergo structural verification accounting for the flexibility of the links and the clearance of the ball joints. This verification is crucial to prevent structural failures during high dynamic work cycles and ensure adherence to the specified position accuracy requirements. If the tests yield positive results, the optimization procedure is completed. Otherwise, adjustments are to be made to the link sections before relaunching the optimizer. The final outputs are the optimal kinematics parameters, the link sections and the workspace.

Given the iterative nature of the proposed methodology, depending solely on a commercial multibody solver may become laborious and time-consuming [28]. Therefore, in order to maximize both versatility and calculation efficiency, theoretical formulations of the delta kinematics and dynamics have been developed in the following section. As previously discussed, computationally efficient models prove valuable not only for the mechanical design but also for future implementation in developing the motion controller.

3 Analytical modeling

In this section, an analytical model covering the inverse kinematics and dynamics of the delta manipulator is discussed. Subsequently, the results obtained with the same model replicated in RecurDyn are reported and compared for validation. To facilitate the reproduction of the described approaches and support future developments, all the utilized models used are provided in the supplementary material section.

3.1 Inverse kinematics and singularity avoidance

Figure 3 depicts the kinematic scheme of the delta robot, where three identical kinematic chains are equally arranged at angles of 120° from one another. Each of these chains is identified by an index, denoted as i, where $\mathrm{i} = 1,\ 2,\ 3$. The active arms are connected to the base through revolute joints $\mathrm{A_i}$, which represent the abstraction of the motors, positioned at the center of each triangle side. Points $\mathrm{C_i}$ denote the ends of the active arms, whereas points $\mathrm{B_i}$ denote the centers of each side of the moving platform triangle. The connection between the moving platform and the active arms is established through the linkage of points $\mathrm{B_i}$ and $\mathrm{C_i}$ via the passive arms. The rotation angles of each revolute joint are denoted as $\mathrm{q_i}$, and their configuration determines the position of the end-effector in the workspace. Conversely, when the position of the moving platform is known, the inverse kinematics is solved to obtain $\mathrm{q_1}$, $\mathrm{q_2}$ and $\mathrm{q_3}$ [29].

This problem can be approached geometrically by intersecting geometric entities to determine the spatial position of points $\mathrm{C_i}$, as in [30]. Notably, deducing the angles $\mathrm{q_i}$ from the positions of points $\mathrm{C_i}$ in the space is a straightforward process. Indeed, by considering only one kinematic chain (i.e. $\mathrm{A_i - C_i - B_i}$), the position of point $\mathrm{B_i}$ is known

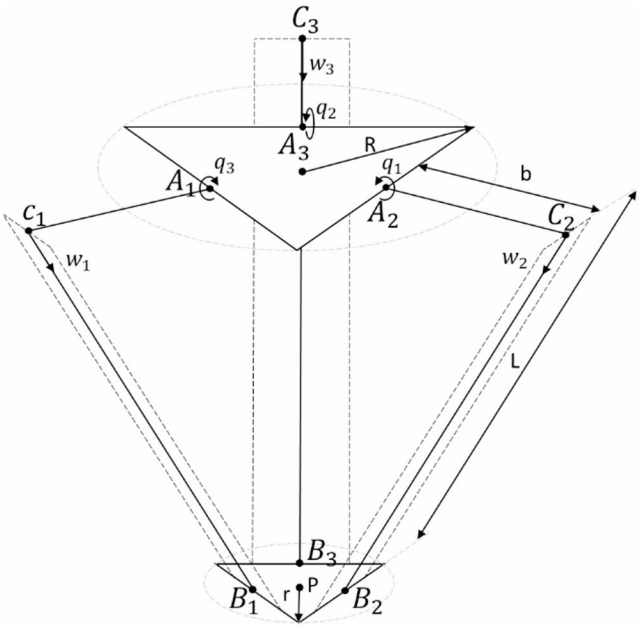


Fig. 3 Kinematic model of the delta robot

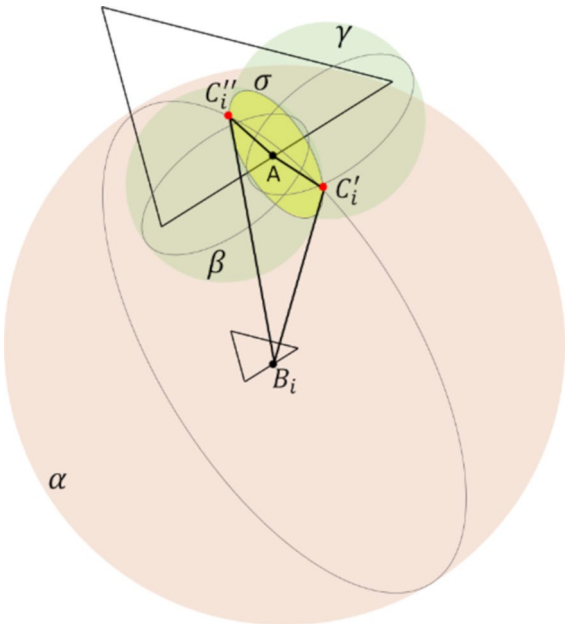


Fig. 4 Inverse kinematics with intersection of three spheres

as it belongs to the moving platform. The point C_i must be located on the sphere α with center B_i and radius L. Also, C_i must belong to the circumference σ of radius b and center A_i, perpendicular to the axis of the revolute joint. The coordinates of C_i can therefore be computed as the intersection of the sphere α and the circumference σ. With reference to Fig. 4, to simplify the resolution of this analytical task, it is expedient to introduce two supplementary spheres, named β and γ. These have identical radii, arbitrarily selected but greater than b, and both centered symmetrically along the axis of the revolute joint with respect the circumference σ. The points of intersection between these two spheres define the circumference σ. This strategic approach leads to the establishment of three spheres (α, β, γ) whose collective intersection precisely determines the spatial coordinates of point C_i. More precisely, the intersection of three spheres results in two outcomes, namely C_i' and C_i'', as shown in Fig. 4. Naturally, as the procedure is repeated for all kinematic chains, eight potential configurations emerge for the manipulator, and only one is to be selected. Once the coordinates of C_i are found, the relative angle q_i is simply determined as the inclination of segment $A_i - C_i$.

The solution of the inverse kinematics problem may lead to the manipulator reaching a singularity for a specific assumed position. Indeed, when the passive arms are nearly coplanar, the robot transmission behavior degrades, approaching the kinematic singular configuration. As discussed in [21], to mitigate this unwanted situation one possible strategy is to constrain the transmission angles (shown in Fig. 5) within specific limits:

$$45^\circ \leq \{\delta_{1i}, \delta_{2i}\} \leq 135^\circ \tag{1}$$

In addition, it is possible to use the volume of the tetrahedron formed by the unit vectors of the passive arms ($\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ in Fig. 3) as a metric to assess the linkages transmission behavior:

$$V = \frac{1}{6}\det([\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3]) \geq V_{\min} = \frac{0.372}{6} \tag{2}$$

where $V_{\min}$ is the volume of the tetrahedron when the angles formed by the intersection of each couple of passive arms equals 40°.

3.2 Rigid body dynamic model

To evaluate the motors' torque needed to obtain a desired motion of the end-effector, the dynamics of the manipulator must be considered. For a delta robot, this has been addressed by applying the virtual work principle to a simplified model with lumped parameters [31]. By condensing distributed masses and inertias into discrete, lumped values, this method reduces computational complexity while preserving the essential characteristics of the system's dynamics. This is particularly advantageous in applications requiring rapid evaluations, such as iterative design processes and real-time control loops. In the current model, the following assumptions are made:

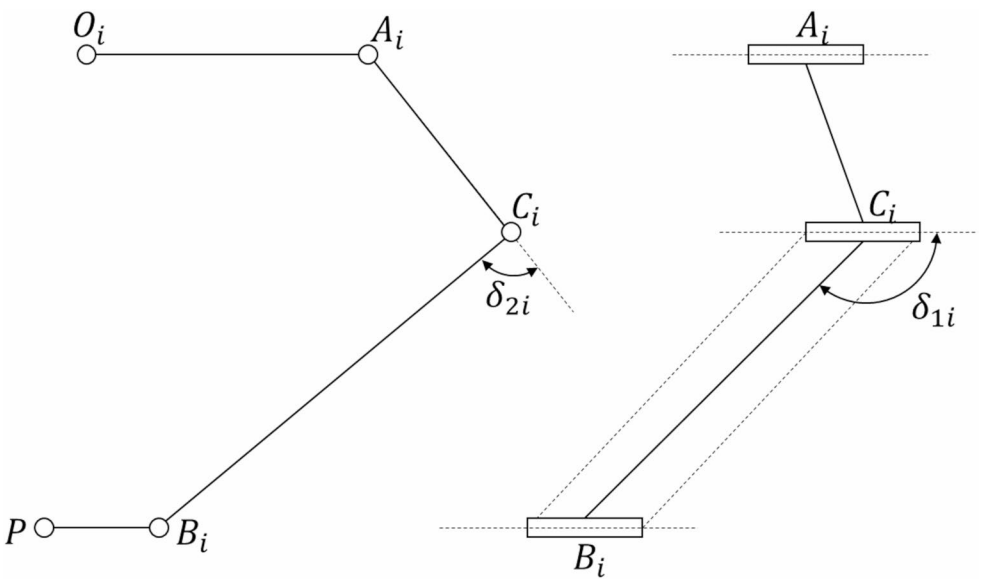


Fig. 5 Transmission angles of the kinematic chain

- **Rotational inertias of passive links are neglected**: this is justified as passive links are typically slender and constructed from lightweight materials, and the forces exchanged with the mobile platform align closely with their orientation. As a result, their rotational inertias have minimal impact on the system's dynamics and can be safely omitted.
- **Masses of passive links are lumped**: 2/3 of the mass is concentrated at the ends of active links, and 1/3 is assigned to the mobile platform. This approximation is based on the fact that the inertia of a bar of length L and mass m is $\mathrm{mL}^2/3$, which is equivalent to the inertia of $1/3$ of the mass m placed at the mobile end of the active link. This simplification not only reduces the complexity of the equations but also facilitates the computational efficiency required for practical uses.
- **Friction and deformability of bodies are neglected**.

From these assumptions, it is possible to simplify the manipulator by "removing" the passive links and separating the upper group, consisting only of active links and actuated joints, from the lower group that constitutes the mobile platform. The closure of the kinematic chain will be achieved analytically by leveraging the Jacobian matrix of the robot. Again, due to the triple axial symmetry, one kinematic chain at a time can be considered and analyzed. The resulting mass of the mobile platform is:

$$m_{nt} = m_{payload} + m_n + 3\,(1-x)\,m_{ab} \tag{3}$$

In this equation, $\mathrm{m_{payload}}$ is the mass of the load to be moved, $\mathrm{m_n}$ is the mass of the mobile platform, whereas $\mathrm{m_{ab}}$ is the mass of the passive link. The parameter x identifies the portion of the passive link's mass placed at the end

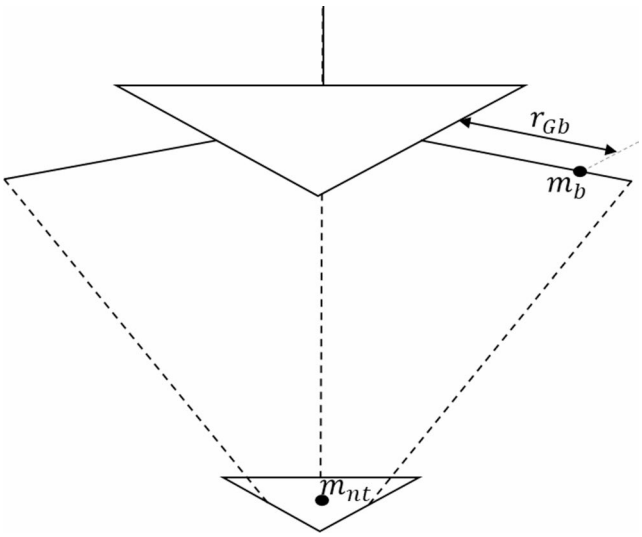


Fig. 6 Dynamic model with lumped mass parameters

of the active link, which in this case becomes $2/3$. Regarding the active link, the following expression holds:

$$r_{Gb} = b\frac{\frac{1}{2}m_{br} + m_c + xm_{ab}}{m_b} \tag{4}$$

where

$$m_b = m_{br} + m_c + xm_{ab} \tag{5}$$

Here, $\mathrm{m_{br}}$ is the mass of the active link, $\mathrm{m_c}$ is the mass of the elbow (connection between active and passive links), and $\mathrm{r_{Gb}}$ is the position of the final center of mass. The manipulator schematic and its geometric parameters are illustrated in Fig. 6. The final inertia of each active link, $\mathrm{I_{bi}}$, is the sum of the motor inertia, $\mathrm{I_m}$, and the arm inertia, $\mathrm{I_{br}}$:

$$I_{bi} = I_m + I_{br} \tag{6}$$

$$I_{br} = b^2 \left(\frac{1}{3} m_{br} + m_c + x m_{ab} \right) \tag{7}$$

By considering the introduced simplifications, along with the separation between the mobile platform and the active links, and leveraging the Jacobian matrix, the forces acting on both the motors and the mobile platform can be defined. In particular, gravitational forces $\mathbf{G_n}$ and inertial forces $\mathbf{F_n}$ act on the mobile platform. These are defined as follows:

$$\boldsymbol{G_n} = m_{nt} \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T \tag{8}$$

$$\boldsymbol{F_n} = m_{nt} \ddot{\boldsymbol{X}}_{\boldsymbol{n}} \tag{9}$$

The contribution of these two forces on each motor can be calculated by multiplying with the transpose of the Jacobian matrix, i.e.:

$$\boldsymbol{\Gamma}_{\boldsymbol{n}} = \boldsymbol{J^T F_n} = \boldsymbol{J^T} m_{nt} \ddot{\boldsymbol{X}}_{\boldsymbol{n}} \tag{10}$$

$$\boldsymbol{\Gamma}_{\boldsymbol{Gn}} = \boldsymbol{J^T G_n} = \boldsymbol{J^T} m_{nt} \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T \tag{11}$$

In accordance with the principle of virtual work, the contribution of all non-inertial forces must be equal to the contribution of all inertial forces. Applying this principle at the joint level yields:

$$\boldsymbol{\Gamma} = \boldsymbol{I_b} \ddot{\boldsymbol{q}} + \boldsymbol{J^T F_n} - \boldsymbol{J^T G_n} - \boldsymbol{\Gamma}_{\boldsymbol{Gb}} \tag{12}$$

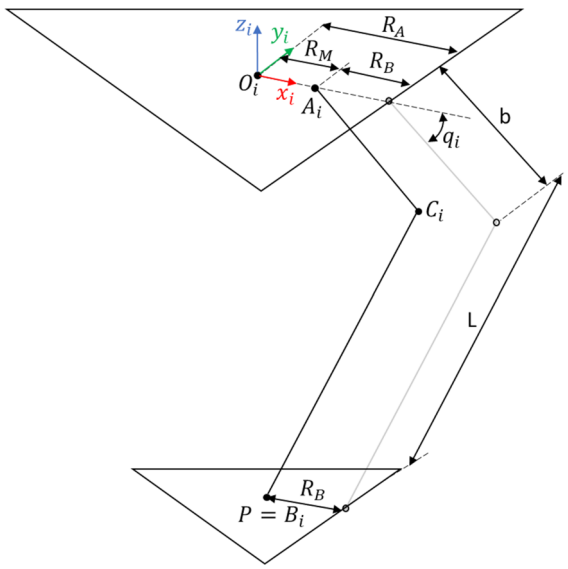


Fig. 7 Single chain and kinematics simplifications

where $\boldsymbol{\Gamma}$ is the vector of torques that must be applied by the motors, and $\boldsymbol{\Gamma}_{\mathbf{Gb}}$ is the torque generated by the gravitational forces acting on the active links, given by:

$$\boldsymbol{\Gamma}_{\boldsymbol{Gb}} = m_b r_{Gb} g \begin{bmatrix} cosq_1 & cosq_2 & cosq_3 \end{bmatrix}^T \tag{13}$$

$\mathbf{I_b}$ is the inertia matrix of the active links in the joint space, defined as:

$$\boldsymbol{I_b} = \begin{bmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{bmatrix} \tag{14}$$

To solve Eq. (12), the Jacobian matrix of the system must first be computed. In the following, the methodology for its assessment is presented. Each kinematic chain is identified in the same reference system $\{\mathbf{R}\}$ but rotated along z -axis by angles $\theta_i = 0°, 120°, 240°$ for frames 1, 2, and 3, respectively. Hence, the transformation matrix between **{Ri}** and **{R}** is defined as:

$${}_{\mathbf{i}}^{\mathbf{R}}\mathbf{R} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

Furthermore, since the mobile platform can only translate, it is possible to consider the distance of the reference system $\{\mathrm{R}\}$ from the motor as $\mathrm{R_M} = \mathrm{R_A} - \mathrm{R_B}$ and to have $\mathrm{P} = \mathrm{B_i}$ (as shown in Fig. 7, where a single kinematic chain is considered). These considerations will significantly simplify the derivation of the Jacobian matrix $\mathbf{J}$ of the manipulator, which expresses the correlation between the velocity in the workspace $\dot{\mathbf{X}}$ and the velocity in the joint space $\dot{\mathbf{q}}$:

$$\dot{\mathbf{X}} = \mathbf{J}\dot{\mathbf{q}} \tag{16}$$

The following constraint must be imposed to ensure that the length of the passive links cannot change:

$$\|\mathbf{C_i B_i}\|^2 = \mathrm{L}^2 \tag{17}$$

which can be rewritten as:

$$\mathbf{s_i^T} \cdot \mathbf{s_i} - \mathrm{L}^2 = 0 \tag{18}$$

where.

$$\mathbf{s_i} = \mathbf{O_i B_i} - (\mathbf{O_i A_i} + \mathbf{A_i C_i}) = \begin{bmatrix} \mathrm{x_n} \\ \mathrm{y_n} \\ \mathrm{z_n} \end{bmatrix} - {}_{\mathbf{i}}^{\mathbf{R}}\mathbf{R} \left(\begin{bmatrix} \mathrm{R_M} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathrm{bcosq_i} \\ 0 \\ -\mathrm{bsinq_i} \end{bmatrix} \right) \tag{19}$$

Deriving Eq. (18) results in:

$$\mathbf{s_i^T} \cdot \dot{\mathbf{s}}_\mathbf{i} = 0 \tag{20}$$

where $\dot{\mathbf{s}}_\mathbf{i}$ can be expressed as:

$$\dot{\mathbf{s}}_\mathbf{i} = \begin{bmatrix} \dot{x}_n \\ \dot{y}_n \\ \dot{z}_n \end{bmatrix} + {}_\mathbf{i}^\mathbf{R}\mathbf{R} \begin{bmatrix} b\sin q_i \\ 0 \\ b\cos q_i \end{bmatrix} \dot{q}_i = \dot{\mathbf{X}}_\mathbf{n} + \mathbf{b_i}\dot{q}_i \tag{21}$$

Then, after rewriting considering Eqs. (20)-(21) in vector form, one could obtain:

$$\begin{bmatrix} \mathbf{s_1^T} \\ \mathbf{s_2^T} \\ \mathbf{s_3^T} \end{bmatrix} \dot{\mathbf{X}}_\mathbf{n} + \begin{bmatrix} \mathbf{s_1^T} \cdot \mathbf{b}_1 & 0 & 0 \\ 0 & \mathbf{s_2^T} \cdot \mathbf{b}_2 & 0 \\ 0 & 0 & \mathbf{s_3^T} \cdot \mathbf{b}_3 \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{22}$$

where $\dot{\mathbf{q}} = [\ \dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3\ ]^T$ is the vector of joint space velocities. Finally, recalling Eq. (16), the Jacobian matrix is determined as:

$$\mathbf{J} = -\begin{bmatrix} \mathbf{s_1^T} \\ \mathbf{s_2^T} \\ \mathbf{s_3^T} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{s_1^T} \cdot \mathbf{b}_1 & 0 & 0 \\ 0 & \mathbf{s_2^T} \cdot \mathbf{b}_2 & 0 \\ 0 & 0 & \mathbf{s_3^T} \cdot \mathbf{b}_3 \end{bmatrix} \tag{23}$$

It is observed that this formula encompasses all the main parameters related to the dimensions and masses of the manipulator, such as gravitational and inertial forces. Therefore, knowing the masses of the links, the mobile platform, and the accelerations of the active arms and the end-effector, it is possible to uniquely define the torques profile exerted by the motors. From this result it is possible to calculate the RMS torque value of each motor.

Table 1 Parameters used for model validation

Parameter	Value	Parameter	Value
b	220 mm	m_{br}	0.054 kg
L	550 mm	m_c	0 kg
R	210 mm	I_m	2.6 kgmm^2
r	100 mm	m_n	0.070 kg
m_{ab}	0.030 kg	$m_{payload}$	0.1 kg

3.3 Model validation

The analytical model has been coded and solved numerically in Matlab. To check its correctness and accuracy, a virtual prototype of the delta robot is also realized and simulated in RecurDyn [32]. To keep consistency among the models, in this preliminary analysis the links flexibility has been neglected in RecurDyn. Furthermore, the spherical joints have been idealized without friction, clearance, or mass to simplify the model. The tests are conducted considering an aluminum alloy and the parameters listed in Table 1.

A pick-and-place path on the XZ plane followed using double-S motion profiles suggested by an industrial partner is enforced in a series of discrete points (700 for each trajectory segment) in both the simulation environments. Figure 8 shows the 3D model of the manipulator simulated in recurdyn and also the motion law imposed at the end-effector. For each point of the trajectory, inverse kinematics and dynamics are solved to obtain the rotation angles of the revolute joints and the driving torques. The theoretical models are validated through a comparative analysis of values obtained from matlab and recurdyn. The plots reported in Fig. 9 illustrate good agreement between the results obtained, with a maximum deviation of 1.2%, thereby confirming the correctness of the proposed model.

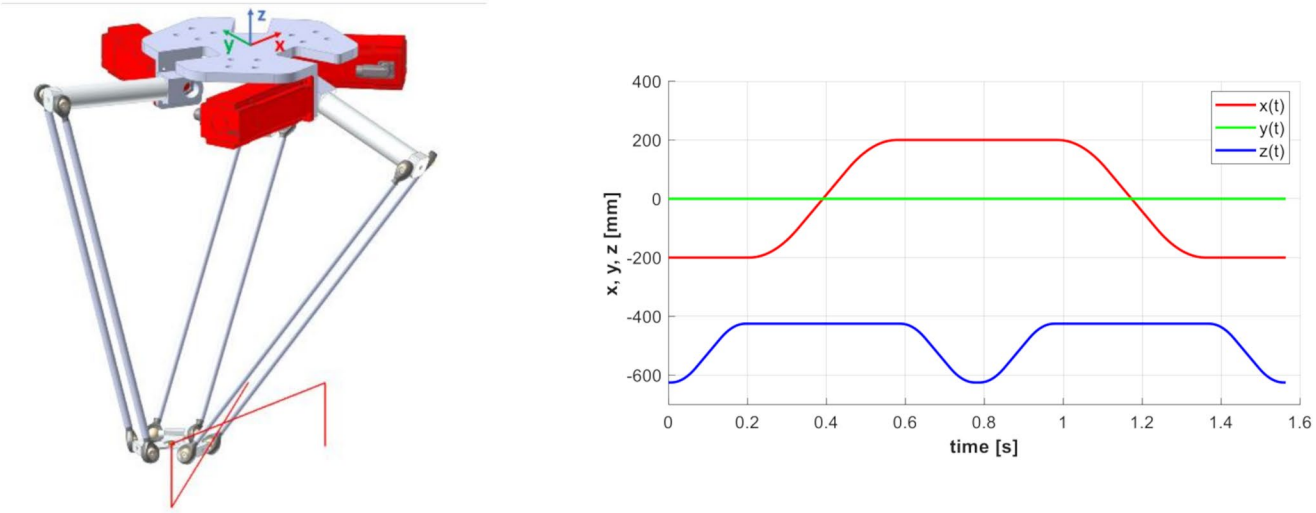


Fig. 8 Imposed pick-and-place motion: RecurDyn 3D path and input motion profiles

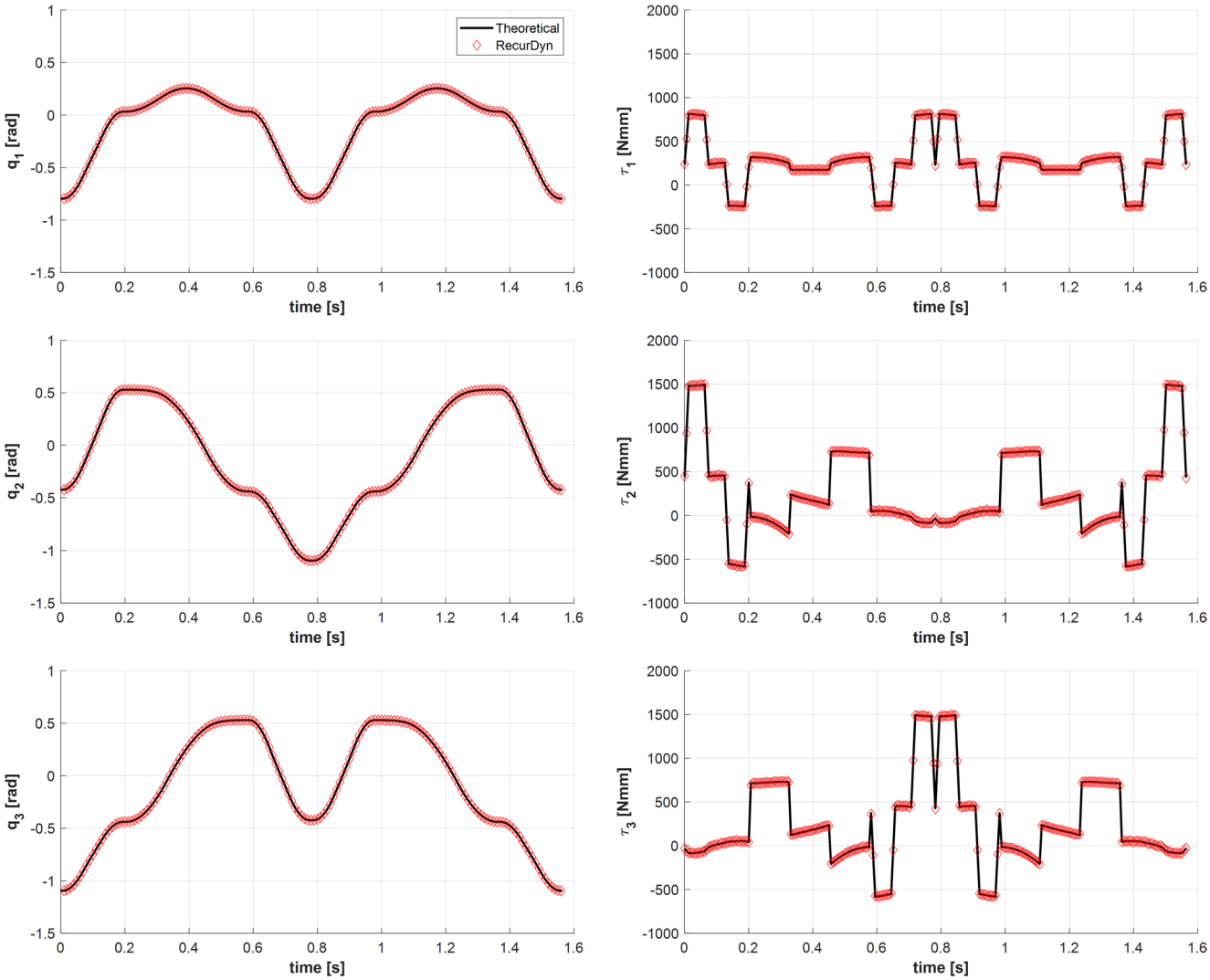


Fig. 9 Comparison between theoretical and RecurDyn dynamic models

4 Design tool

The analytical model has been leveraged within the framework of a novel design tool specifically developed for optimizing industrial delta robots. The tool, implemented in a main Matlab script, is organized in 5 different sections, as shown in Fig. 10. The first section involves user input, including the bounding box, end-effector accuracy error, and the required CPM with a specified payload for pick-and-place operations. The script processes these inputs to define the workspace and the manipulator's optimal dimensions by executing a dynamic optimization with a genetic algorithm (i.e. *ga* routine from the Matlab library). After the optimization, the Matlab script interfaces with the multibody simulator RecurDyn, transmitting the parameters of the best candidate in order to execute flexible multibody batch simulations enabling the evaluation of potential structural failure and the prediction of the end-effector trajectory while considering the influence of link flexibility and joints clearance. In this way, the deviation of the end-effector position with respect to the ideal path is computed and compared with the maximum error tolerance specified by the user. If the manipulator is structurally verified and the prescribed accuracy is achieved, the optimization process concludes successfully. However, if the criteria are not met, the iterative process is repeated, exploring alternative section dimensions of the links.

After completing the optimization process, another Matlab script is used to determine the maximum torque required to support the maximum allowable payload under static conditions within the workspace. The static torques are subsequently compared to those required under dynamic conditions, guiding the appropriate sizing of the motors. Subsequently, the manipulator performance is assessed in terms of maximum end-effector accelerations based on

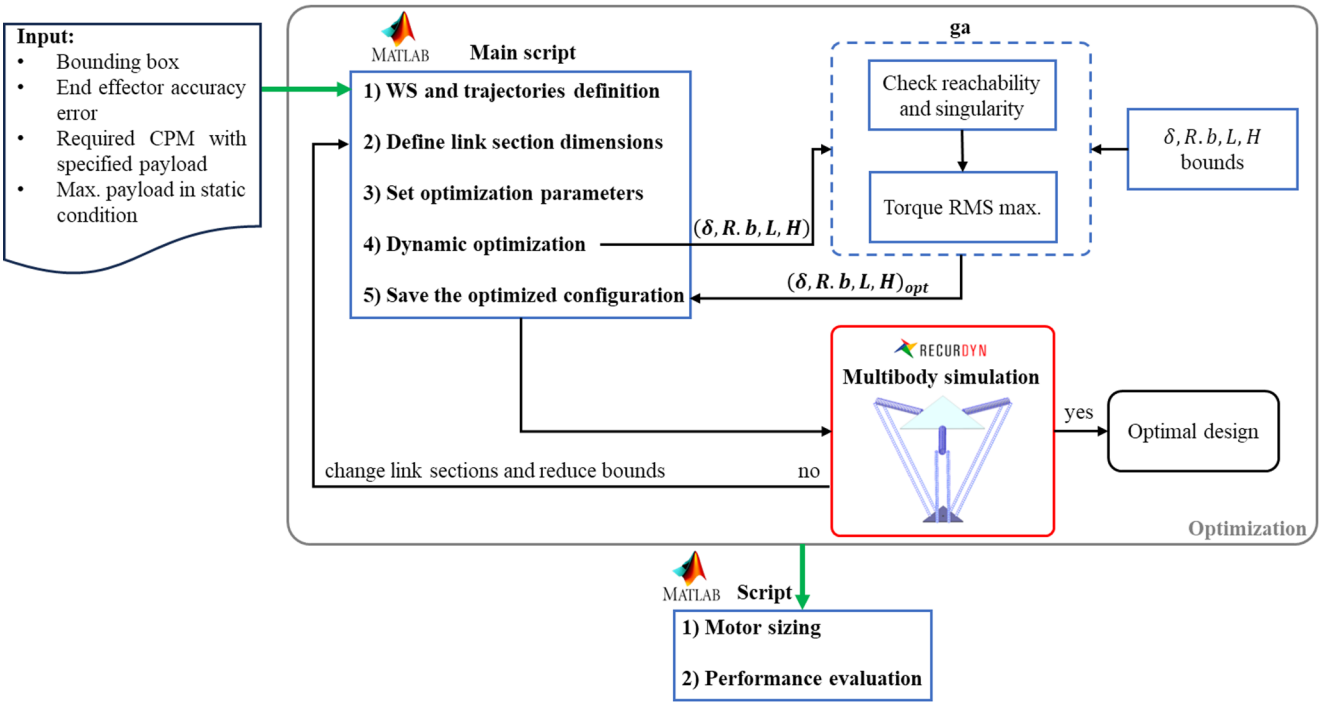


Fig. 10 Structure of the implemented design tool

the payload for a pick-and-place trajectory chosen within the workspace and suggested by an industrial partner. This assessment is significant because the manipulator is sized considering critical trajectories that start and end at the edges of the workspace, which may not be utilized in typical operating conditions. In addition, the user may also decide to use a different payload. To perform this calculation, once again, a Matlab script is employed, implementing the manipulator dynamics model detailed in Sect. 3.2. More details regarding all the process steps are given in the remaining of this section.

4.1 Optimization problem

As outlined in Sect. 2, optimizing a delta robot involves a multitude of parameters. Specifically, dynamic optimization not only involves the kinematic parameters but also considers the masses of the robot components, influenced by their cross-sectional areas and material properties. Given the density of the chosen material, the dimensions of the sections cannot be directly fed into the optimization algorithm with all the other parameters as they would tend towards zero, aiming for the most dynamically favorable condition (where null or nearly null link masses result in nearly null motor torques). However, this leads to a manipulator that is excessively compliant and would inevitably deform excessively or even break. For this reason, it has been decided to initially set tentative values for the links' cross section area and then perform dynamic optimization based solely on the kinematics parameters (δ , R, b, L, H), where the parameter δ is used to ensure a good proportion of the manipulator [21]:

$$\delta = \frac{\frac{R}{2} - \frac{r}{2} + b}{\frac{d}{2}} \quad \text{where} \quad 0.8 \leq \delta \leq 1.3 \tag{24}$$

From this relation, it is possible to calculate the value of the radius r of the moving platform as a function of the other optimization parameters. Aluminum links with hollow circular cross section have been chosen and their initial dimensions are set based on design experience. For the active arm, the inner diameter d_{bi} is set at 24 mm, whereas the outer diameter d_{be} is 26 mm. The passive arms have an inner diameter d_{Li} of 6 mm and an outer diameter d_{Le} of 8 mm. At last, the thickness of the moving platform s is set equal to 3 mm. Starting from these considerations, the optimization problem can be formalized as follows:

$$\textbf{Min.}\ \Gamma_{RMS} = \max\left(\frac{1}{k}\sum_{k=1}^{n}\boldsymbol{\Gamma}_k\right)$$

$$\textbf{Constrain} \rightarrow\ r_{min} \leq 2\left(\frac{R}{2} + b - \delta\frac{d}{2}\right)$$

$$\textbf{Design var.} \rightarrow \begin{cases} H \in [2\mathrm{h},\ \mathrm{C}] \\ \delta \in [0.8,\ 1.3] \\ \mathrm{b} \in \left[\mathrm{b_{min}},\ \frac{\min(\mathrm{A},\mathrm{B})}{2} - \frac{\mathrm{R_{min}}}{2}\right] \\ \mathrm{L} \in \left[\sqrt{\left(0.8\frac{\mathrm{d}}{2}\right)^2 + \mathrm{h}^2},\ \sqrt{\left(\frac{\min(\mathrm{A},\mathrm{B})}{2}\right)^2 + \mathrm{C}^2}\right] \\ R \in \left[\mathrm{R_{min}},\ 2\left(\frac{\min(\mathrm{A},\mathrm{B})}{2} - \mathrm{b_{min}}\right)\right] \end{cases}$$

where Γ_{RMS} is the highest value of the RMS torques calculated for the three motors (i.e. active joints), evaluated in k incremental steps for each trajectory segment. The trajectories are based on the specified CPM and define a cylindrical workspace from the bounding box. The diameter d is set at approximately 75% of the minimum value between the lateral sides of the box, ensuring large space on the sides and avoiding collisions with the lateral wall. Maintaining practical proportions, the workspace height h is set at around 35% of the diameter. The border trajectories, as illustrated in Fig. 11, must be executed while adhering to the specified CPM, i.e. ensuring compliance with the cycle time. It is assumed that the cycle time is partitioned, with 50% allocated to the diametrical path and the remaining 50% evenly distributed between the pick-and-place actions. Double-S timing laws are employed, divided into three equal time periods, two with constant acceleration and one with constant velocity. It is also assumed that the end-effector starts to travel each segment from a stationary position and comes to a stop at the conclusion. The imposed constraint in the optimization problem ensures a minimum value of the radius r, enabling the proper mounting of a tool

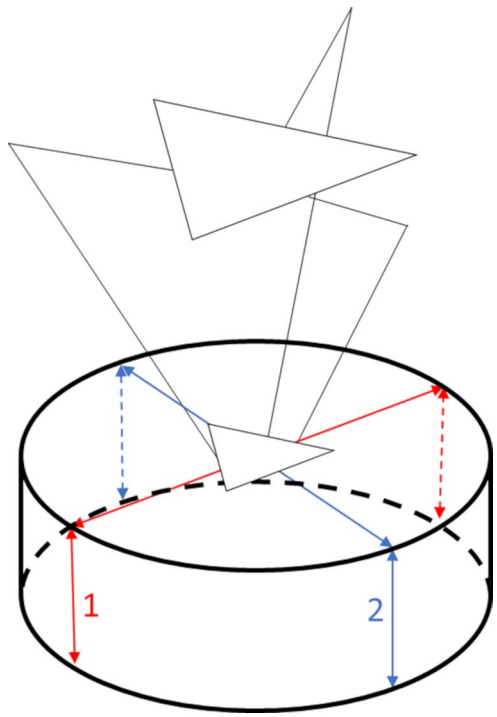


Fig. 11 Trajectory considered for the optimization

($\mathrm{r_{min}}$ has been set equal to 100 mm). Likewise, the lower bound of the base radius, $\mathrm{R_{min}}$, has been chosen equal to 200 mm to allow the assembly of the motors. Then, the minimum allowable value of the active arm $\mathrm{b_{min}}$ has been set at 100 mm, whereas the minimum value of H has been set at $2\mathrm{h}$ to prevent any interference between the links.

In the optimization process, configurations are excluded if they do not allow the end-effector to reach all points within the workspace or, alternatively, if they allow such reachability but fail to avoid singularities. By discretizing the workspace surface uniformly into points, the verification process involves confirming the existence of a valid solution for the inverse kinematic model at each point (via the equations outlined in Sect. 3.1), all while considering the condition to avoid singularity [33, 34]. This assumption holds true as long as the actual workspace lacks internal holes and singular configurations occur on the borders [4, 17]. In the optimizer, the unsuccessful candidates are eliminated by setting the RMS torque value to infinity. A genetic algorithm has been chosen for this optimization task because of its robustness, suitability for multi-parameter objective functions, and its ability to compute the global minimum, thereby overcoming issues associated with local minima [35]. The algorithm is configured to terminate after a maximum number of generations, calculated as the number of variables multiplied by 100. In this study, with five variables, this results in a maximum of 500 generations. The population size for each generation is set to 600 individuals, providing a robust sample for genetic operations such as selection, crossover, and mutation. To prevent unnecessary computations and ensure convergence, the algorithm includes a termination criterion based on stalling generations. Specifically, it would stop if the relative average change in the best fitness value over 50 consecutive generations is less than or equal to a threshold of 1e-4. The crossover fraction is 0.8, meaning that 80% of the population in the subsequent generation is generated through the crossover operation. Additionally, the elite count is determined as 5% of the population size. The optimized parameter set is rounded and utilized in the subsequent step, i.e. the performance validation conducted in RecurDyn environment.

4.2 Multibody simulation

The RecurDyn model utilized in Sect. 3.3 has been updated based on the parameter set derived from the optimization. However, differently from the previous analysis, the current model incorporates the flexibility of the links and the clearance of the ball joints, which has been set at 0.1 mm. Specifically, the links are meshed with 1D beam elements since they are essentially thin, constant-section beams primarily subject to bending. The upper arms are discretized with an

element size of 5 mm, whereas the passive arms are discretized with an element size of 10 mm. These dimensions have been chosen considering the overall dimensions of the links and as an attempt to balance the quality of the result and the required computational time. The mobile platform is modeled as a rigid body due to its small size and greater stiffness compared to the other components. To account for the clearance in the ball joints, these are no longer modeled using the standard joint feature within the RecurDyn library but rather exploiting a 6-Dof force element connector with a custom governing equation. This approach becomes necessary since RecurDyn, like other multibody solvers, treats ball joints as ideal kinematic elements without the capability to include clearances. In the utilized force element, translational forces are configured exclusively (via equations), while rotational contributions are set to zero. In this case, the spherical joint is conceptualized as an outer spherical shell that encloses an inner sphere within it. Clearance between them determines whether they are in contact or not. During motion, if the distance between the center of the inner sphere and the outer spherical shell is less than the specified clearance, no contact force is generated. Conversely, if the distance is equal to or greater than the clearance, an elastic contact force is activated, considering the joint stiffness K. The custom force equation can therefore be expressed as follows:

$$\mathrm{F}_{\mathrm{contact}} = \begin{cases} 0 & if\ DM < c \\ K\eta & if\ DM \geq c \end{cases} \tag{25}$$

where $\eta = \mathrm{DM} - \mathrm{c}$ is the penetration during contact, being DM the total displacement of the internal sphere and c the clearance. A two-dimensional representation of the contact is illustrated in Fig. 12, which can be readily extended to three dimensions. The components of the penetrations can be simply evaluated as:

$$\begin{array}{ll} \eta_{\mathrm{x}} = \left(1 - \frac{\mathrm{c}}{\mathrm{DM}}\right) \mathrm{DX} & \eta_{\mathrm{y}} = \left(1 - \frac{\mathrm{c}}{\mathrm{DM}}\right) \mathrm{DY} \\ \eta_{\mathrm{z}} = \left(1 - \frac{\mathrm{c}}{\mathrm{DM}}\right) \mathrm{DZ} & \end{array} \tag{26}$$

where DX, DY and DZ are the components of the total displacement DM.

In the current implementation, the joint is supposed to be rigid, so the stiffness is set to a high value. Once the model of the manipulator is completed, motion laws are then applied to the active arms to achieve the same trajectories of the end-effector used for the optimization process. The simulation gives as a result the displacement of the end-effector influenced by dynamic forces, flexibility of the links and joints clearance. Also, the stress contourplot is generated to monitor the status of meshed elements (i.e. passive links) during the motion execution.

During the behavioral analysis in RecurDyn, if the module of the position error of the end-effector deviates from the user-specified value e expressed in millimeters (as detailed in Sect. 2), it indicates that the manipulator is excessively compliant. In response to this, the dimensions of the link sections are automatically adjusted, initiating a new iteration of the optimization process described earlier, and this continues until all design requirements are satisfied. The iterative approach is also employed until the maximum stress σ_{max} observed on the flexible links falls below the prescribed limit σ_{lim}, defined as a fraction of the material yield stress. With each new iteration, the optimization parameter bounds are dynamically adjusted based on the calculated solution, aiming to minimize computational time. This adjustment is feasible because the new optimal solution is expected to be in close proximity to the previous one.

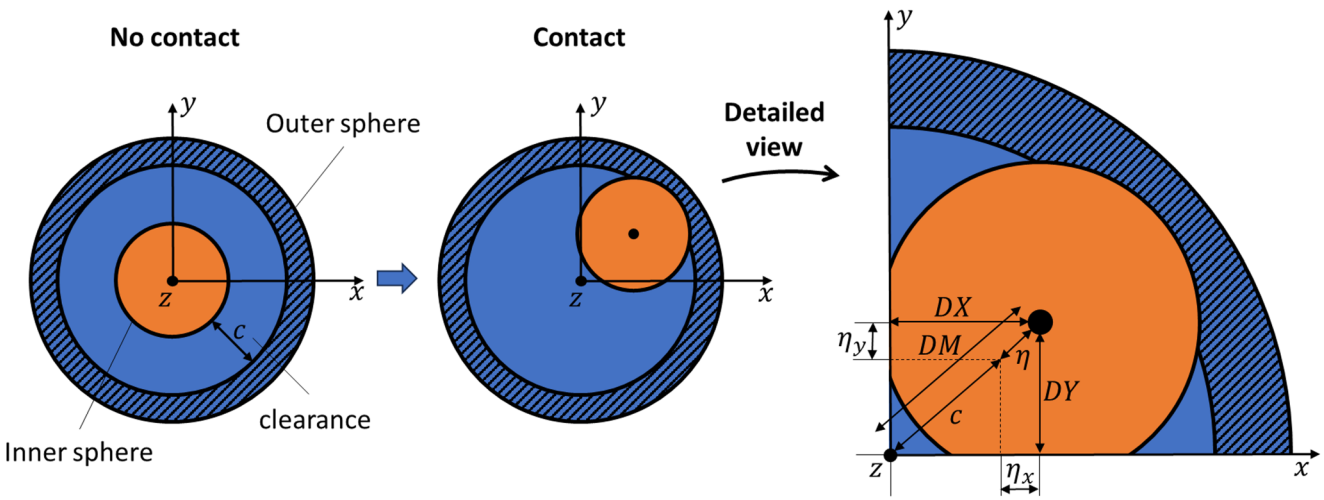


Fig. 12 Evaluation of the clearance in the ball joints

4.3 Motors sizing and performance evaluation

To properly choose the motors, it is crucial to derive values for both peak and rated torque that must be provided during operation [28]. Peak torque values can be extrapolated from the numerical dynamic model, considering the manipulator executing the critical (i.e. most traveling extensive) trajectories. For selecting rated torque, in addition to the RMS torque under dynamic conditions, static considerations must be included. Indeed, the required torques in quasi-static conditions with the maximum payload might exceed the dynamic maximum RMS torque. The evaluation is performed by uniformly discretizing the workspace surfaces into points and considering the manipulator with the end-effector positioned at these points, with the implemented dynamic model [4]. From these considerations, the required rated torque value for the selection of the motors will be the maximum either in static or dynamic conditions.

After selecting the motors, it is possible to evaluate the manipulator's dynamic performance in terms of maximum end-effector acceleration, with the payload becoming a variable. By considering both the capability of the motors in terms of torques and the specified path within the workspace, the performance for each trajectory segment of a pick-and-place task can be evaluated in terms of maximum acceleration. The payload is systematically incremented in 0.1 kg intervals, within the given range of investigation. This evaluation considers a double-S timing law, which, is supposed to be divided in three equal time period, two with constant acceleration and one with constant velocity. Starting with a high acceleration value, the torques required to execute the imposed end-effector motion are computed. If it proves incompatible with the installed motors in terms of peak torque and rated torque, the acceleration is systematically decreased until a feasible value is identified. This iterative process is applied across the designated range of payload investigation. The current framework only considers linear paths, assuming the assessment of delta performance for pick-and-place tasks, although more diversified paths and motion profiles can potentially be included. The implemented Matlab script is illustrated in Fig. 13. The primary outcome of this analysis is the generation of robot performance maps, which are plots illustrating the relationship between acceleration and payload. These maps serve as valuable tools to aid operators in the practical utilization of the manipulator.

5 Case study

The optimization tool described so far has been tested to design a delta manipulator intended for use in the context of an industrial research project. In particular, this manipulator will serve as a testing platform for the development of innovative control algorithms. Therefore, the following user requirements have been considered in accordance with the industrial partner:

- Bounding box $750 \times 750 \times 750$ mm;
- Pick-and-place operation at rate of 40 CPM with payload of 0.1 kg;
- Maximum end-effector accuracy error $e = 1$ mm;
- Maximum payload of 0.6 kg.
- Direct drive motors (i.e. no speed reducers).

The optimization process described in Section 5 has been executed, producing as results the parameters listed in Table 2.

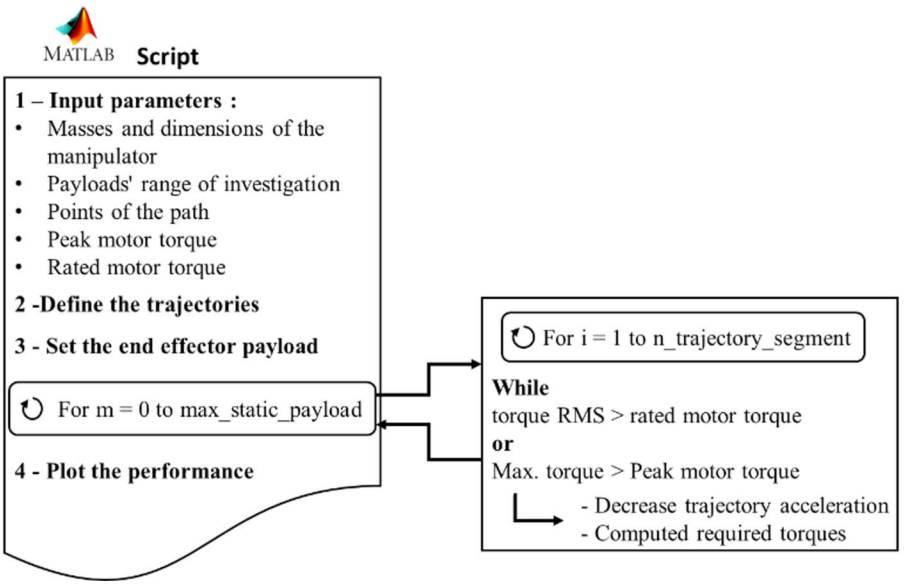


Fig. 13 Performance evaluation routine

Table 2 Optimization results

Parameter	Value	Parameter	Value
d	600 mm	H	572 mm
h	210 mm	s	3 mm
b	243 mm	d_{bi}	26 mm
L	506 mm	d_{be}	30 mm
R	301 mm	d_{Li}	8 mm
δ	1.14 mm	d_{Le}	10 mm
r	100 mm		

The outputs of the final RecurDyn simulation are presented in Fig. 14, where the extrapolated stress and accuracy error functions over time are shown. The results indicate that the maximum Von Mises stress of the active and passive links are 63.4 MPa and 24.2 MPa, respectively, which are significantly lower than the adopted limit, given that the Aluminum yield stress is 240 MPa. Regarding the maximum accuracy error (e) of the platform, calculated as the Euclidean sum of the position deviation from the ideal commanded path, a maximum value of 0.27 mm is observed, which meets the previous specified user requirements.

Static and dynamic analyses allow for determining the maximum torque and maximum RMS torque of the most stressed motor, which are 3220 Nmm and 2070 Nmm, respectively. The maximum torque required in static condition is 1970 Nmm. Based on these values and on the last requirement (i.e. direct drive actuation), it has been decided to choose to mount Beckhoff AM8032 motors, which can deliver a standstill torque of 2.38 Nm and a peak torque of 11.66 Nm. After designing the manipulator and selecting the motors, the performance can be assessed in terms of the maximum acceleration of the end-effector based on the payload. The corner points of the chosen pick-and-place trajectory for the performance evaluation are listed in Table 3.

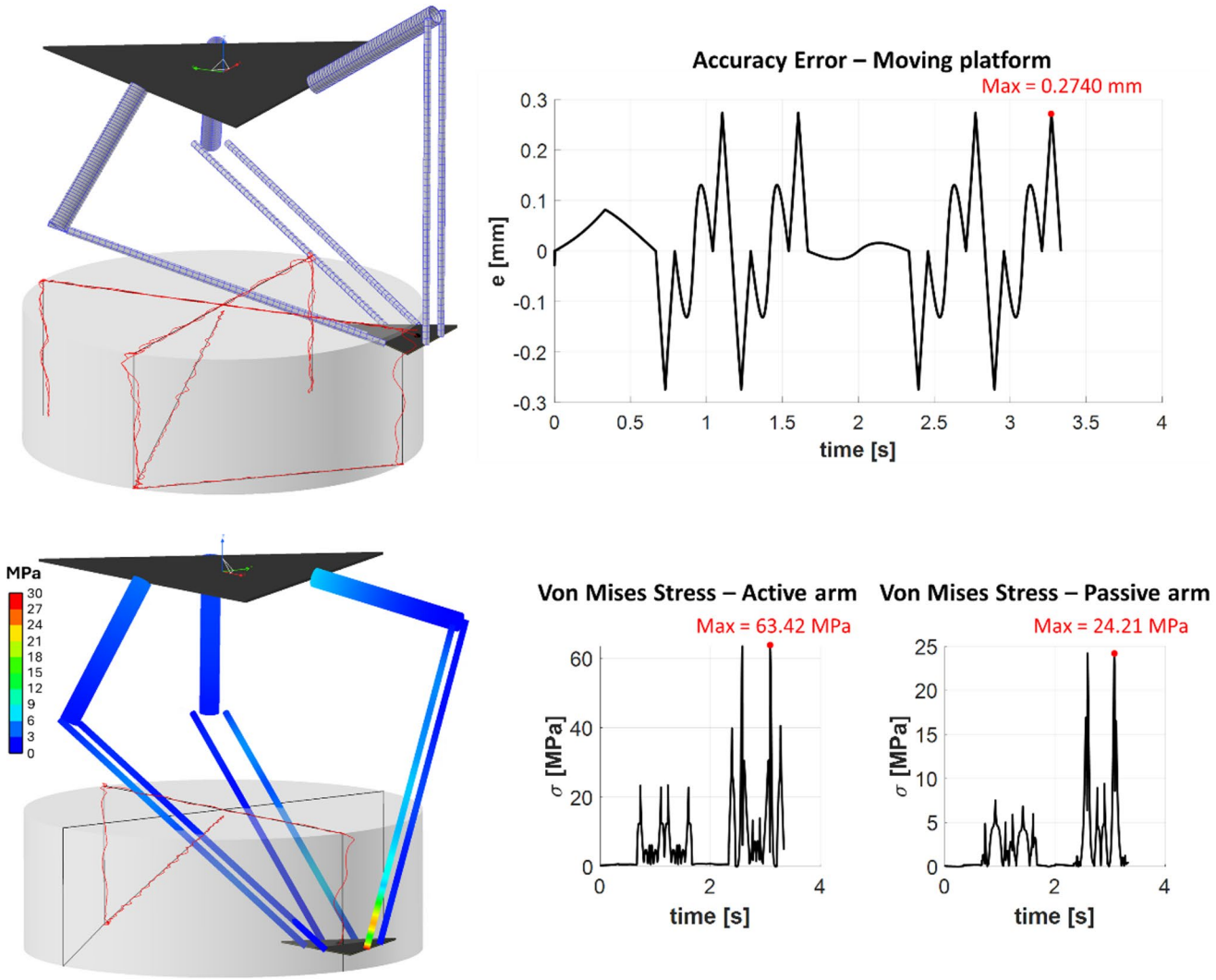


Fig. 14 Output processing of the multibody simulation (scaling factor of two on to the path trace): end-effector position error and Von Mises stresses

Table 3 Corners of the pick-and-place trajectory for performance evaluation

Points	x [mm]	y [mm]	z [mm]
P1	300	0	-572
P2	300	0	-472
P3	-300	0	-472
P4	-300	0	-572

The range of payload investigation has been set from 0 kg to 2 kg. The maximum accelerations for each trajectory segment are shown in Fig. 15. It is evident that increasing the load results in a quasi-hyperbolic decrease in maximum allowable acceleration. Additionally, it is noteworthy that the range of movable loads extends considerably beyond the specified maximum payload in static condition. This is justified by the fact that operating within the workspace allows for a greater load-carrying capacity.

The physical prototype derived from the optimized design (see Table 2) will be actuated via Beckhoff technology, adopting AX8108 drive units, and controlled within the TwinCAT platform. The control framework will be developed in C++, taking advantage of TwinCAT's native real-time integration and direct access to motion control libraries. The implemented strategy will operate in position control mode, where the kinematic models described in Sect. 3.1 will be used to generate the position setpoints transmitted to each drive, with all control loops closed at the drive level every 62.5 µs. Then, the dynamic model presented in Sect. 3.2 will be used to compute the corresponding expected torque profiles, which will be applied in feedforward to enhance tracking accuracy and reduce following error, particularly under high-dynamics conditions. The modular architecture of the TwinCAT-based controller facilitates future extensions, including the integration of additional sensing modules, compensation layers, and advanced control strategies without major reconfiguration of the core framework.

6 Conclusions

This paper presents an integrated approach for the mechanical design optimization of delta robots. A novel design tool is proposed which comprehensively considers key aspects such as kinematics, dynamics, link flexibility, and ball joint clearance. It follows a structured design process that starts with a detailed examination of user requirements, including bounding box specifications, CPM for pick-and-place operations, maximum end-effector accuracy error, maximum static payload, and the objective to minimize costs (e.g. motor size). The optimization process, executed by means of a genetic algorithm from the Matlab library, results in optimal kinematic parameters, link sections, and workspace dimensions. The use of analytical kinematic and dynamic models enables efficient execution of the optimization by significantly reducing computational effort. The optimized design candidate is then virtually prototyped and checked within the flexible multibody tool RecurDyn, considering link flexibility and ball joint clearances. This step supports the assessment of structural integrity and predicts the end-effector accuracy under dynamic conditions. The iterative nature of the process, incorporating user feedback and adjustments, ensures that the final design aligns with performance requirements and cost considerations. After the optimization process, the sizing of the motors is carried out and

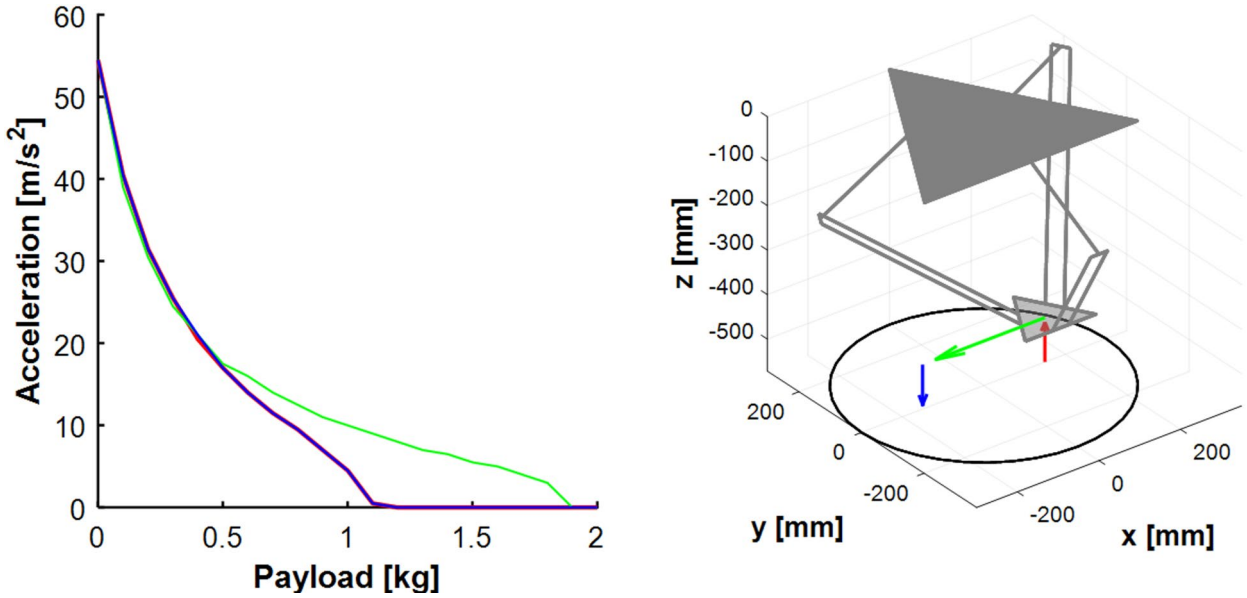


Fig. 15 End-effector acceleration vs. payload for each segment of the pick-and-place path

a method to evaluate the performance of the manipulator in terms of maximum acceleration of the end-effector is proposed. For validation purposes, in the last part of the paper the proposed tool has been successfully applied to design a delta robot in the context of an industrial research project.

The physical prototype corresponding to the optimized design is currently under construction. The servomotors, drives, and cabling have been procured, while the frame and the active and passive links are in fabrication. Meanwhile, the controller development has reached an advanced stage and has already undergone preliminary testing on small laboratory rigs. As a continuation of this research, once the manipulator assembly is completed, the control system will be fully deployed within the Beckhoff TwinCAT environment and tuned to ensure stable and robust closed-loop position control. Following this stage, an extensive experimental campaign will be conducted to assess the manipulator's dynamic performance and validate its mechanical design. During these experiments, the position error will be mapped across the workspace to enable the development of compensation modules addressing key mechanical effects such as link flexibility and joint backlash. These advanced modules will then be integrated into the control framework and invoked during motion trajectory generation to further enhance motion accuracy.

To support further comparisons and future developments, the research material related to this work is made available to the community through an accessible link.

Funding Open access funding provided by Università degli Studi di Modena e Reggio Emilia within the CRUI-CARE Agreement.

Data availability A folder containing the research data and simulation files related to this study is available at the following link https://github.com/XiLab-Robotics/DeltaRobotDesign.git.

Declarations

Competing interest The authors declare that they have no known competing interests that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Benotsmane, R., Dudás, L., Kovács, G.: Survey on new trends of robotic tools in the automotive industry. Lecture Notes in Mechanical Engineering, pp. 443–457. Springer Science and Business Media Deutschland GmbH (2021)
2. Luo, X., Xie, F., Liu, X.J., Xie, Z.: Kinematic calibration of a 5-axis parallel machining robot based on dimensionless error mapping matrix. Robot Comput. Integr. Manuf. **70**. (2021)
3. Rodriguez, E., Alvares, A.J., Jaimes, C.I.R.: Conceptual design and dimensional optimization of the linear delta robot with single legs for additive manufacturing. Proc. Institution Mech. Eng. Part. I: J. Syst. Control Eng. **233**(7), 855–869 (2019)
4. Maya, M., Castillo, E., Lomelí, A., González-Galván, E., Cárdenas, A.: Workspace and Payload-Capacity of a new reconfigurable delta parallel robot regular paper. Int. J. Adv. Robot Syst. **10**. (2013)
5. Kansal, S., Mukherjee, S.: Kinematic and dynamic analysis of a dexterouse multi-fingered delta robot for object catching. Robotica. **40**(8), 2878–2908 (2022)
6. Cretescu, N., Neagoe, M., Saulescu, R.: Dynamic analysis of a delta parallel robot with flexible links and joint clearances. Appl. Sci. (Switzerland), **13**(11). (2023)
7. Condori-Pacori, L., Anchayhua-Arestegui, N.: Modeling and implementation of a 3 degrees of freedom delta robot through Gestalt framework, *2023 9th International Conference on Control, Automation and Robotics, ICCAR 2023*, Institute of Electrical and Electronics Engineers Inc., pp. 13–18. (2023)
8. Angel, L., Viola, J.: Fractional order PID for tracking control of a parallel robotic manipulator type delta. ISA Trans. **79**, 172–188 (2018)
9. Zhu, M., Briot, S., Chriette, A.: Sensor-Based design of a delta parallel robot. Mechatronics, 87. (2022)
10. Zhang, S., Wang, R., Tian, Y., Yao, J., Zhao, Y.: Motion analysis of the Fire-Fighting robot and trajectory correction strategy. Simul. Model. Pract. Theory. **125**. (2023)
11. Schweiger, G., Gomes, C., Engel, G., Hafner, I., Schoeggl, J., Posch, A., Nouidui, T.: An empirical survey on co-simulation: Promising standards, challenges and research needs. Simul. Model. Pract. Theory. **95**, 148–163 (2019)
12. González-García, S., Rodríguez-Arce, J., Loreto-Gómez, G., Montaño-Serrano, V.M.: Teaching forward kinematics in a robotics course using simulations: transfer to a real-world context using LEGO Mindstorms™. Int. J. Interact. Des. Manuf. **14**(3), 773–787 (2020)
13. Gutierrez-Giles, A., Evangelista-Hernandez, L.U., Arteaga, M.A., Cruz-Villar, C.A., Rodriguez-Angeles, A.: A force/motion control approach based on trajectory planning for industrial robots with closed control architecture. IEEE Access. **9**, 80728–80740 (2021)
14. Martínez-Prado, M.A., Rodríguez-Reséndiz, J., Goméz-Loenzo, R.A., Herrera-Ruiz, G., Franco-Gasca, L.A.: An FPGA-based open architecture industrial robot controller. IEEE Access. **6**, 13407–13417 (2018)
15. Bilancia, P., Schmidt, J., Raffaeli, R., Peruzzini, M., Pellicciari, M.: An overview of industrial robots control and programming approaches. Appl. Sci. (Switzerland), **13**(4). (2023)
16. Dastjerdi, A.H., Sheikhi, M.M., Masouleh, M.T.: A complete analytical solution for the dimensional synthesis of 3-DOF delta parallel robot for a prescribed workspace. Mech. Mach. Theory. **153**. (2020)
17. Mahmoodi, M., Tabrizi, M.G., Alipour, K.: A new approach for kinematics-based design of 3-RRR delta robots with a specified workspace, *2015 AI and Robotics, IRANOPEN 2015–5th Conference on Artificial Intelligence and Robotics*, Institute of Electrical and Electronics Engineers Inc. (2015)

18. Malyshev, D., Rybak, L., Carbone, G., Semenenko, T., Nozdracheva, A.: Optimal design of a parallel manipulator for aliquoting of biomaterials considering workspace and singularity zones. Appl. Sci. (Switzerland), **12**(4). (2022)
19. Botello-Aceves, S., Valdez, S.I., Becerra, H.M., Hernandez, E.: Evaluating concurrent design approaches for a delta parallel manipulator. Robotica. **36**(5), 697–714 (2018)
20. Meng, Q., Li, J., Shen, H., Deng, J., Wu, G.: Kinetostatic design and development of a non-fully symmetric parallel delta robot with one structural simplified kinematic linkage. Mech. Based Des. Struct. Mach. **51**(7), 3717–3737 (2023)
21. Zhao, Y.: Dynamic optimum design of a three translational degrees of freedom parallel robot while considering anisotropic property. Robot Comput. Integr. Manuf. **29**(4), 100–112 (2013)
22. Bounab, B.: Multi-objective optimal design based kineto-elastostatic performance for the DELTA parallel mechanism. Robotica. **34**(2), 258–273 (2016)
23. Kermanian, A., Kamali, E., A., and, Taghvaeipour, A.: Dynamic analysis of flexible parallel robots via enhanced co-rotational and rigid finite element formulations. Mech. Mach. Theory. **139**, 144–173 (2019)
24. Cretescu, N.R., Neagoe, M.: Rigid versus flexible link dynamic analysis of a 3DOF delta type parallel manipulator. Appl. Mech. Mater. **762**, 101–106 (2015)
25. Rueda, J.D., Ángel, L.: Structural analysis of a delta-type parallel industrial robot using flexible dynamic of ANSYS 11.0, *IECON Proceedings (Industrial Electronics Conference)*, pp. 2247–2252. (2009)
26. Bilancia, P., Berselli, G., Bruzzone, L., Fanghella, P.: A CAD/CAE integration framework for analyzing and designing spatial compliant mechanisms via pseudo-rigid-body methods. Robot Comput. Integr. Manuf. **56**, 287–302 (2019)
27. Bilancia, P., Baggetta, M., Berselli, G., Bruzzone, L., Fanghella, P.: Design of a bio-inspired contact-aided compliant wrist. Robot Comput. Integr. Manuf. **67**. (2021)
28. Berselli, G., Bilancia, P., Luzi, L.: Project-based learning of advanced CAD/CAE tools in engineering education. Int. J. Interact. Des. Manuf. **14**(3), 1071–1083 (2020)
29. Dahmane, S.A., Azzedine, A., Megueni, A., Slimane, A.: Quantitative and qualitative study of methods for solving the kinematic problem of a planar parallel manipulator based on precision error optimization. Int. J. Interact. Des. Manuf. **13**(2), 567–595 (2019)
30. Zsombor-Murray, P.J.: *An improved approach to the kinematics of Clavel's DELTA Robot*. (2009)
31. IEEE Industrial Electronics Society: *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS 96 : Robotic Intelligence Interacting with Dynamic Worlds, November 4–8, 1996, Senri Life Science Center, Osaka, Japan*, IEEE. (1996)
32. Bilancia, P., Berselli, G.: Conceptual design and virtual prototyping of a wearable upper limb exoskeleton for assisted operations. Int. J. Interact. Des. Manuf. **15**(4), 525–539 (2021)
33. Gharahsofloo, A., Rahmani, A.: *An efficient algorithm for workspace generation of delta robot*. (2015)
34. Pranav, M., Mukilan, A., Sundar Ganesh, C.S.: *A NOVEL DESIGN OF DELTA ROBOT*. (2016)
35. Bilancia, P., Berselli, G.: An overview of procedures and tools for designing nonstandard beam-based compliant mechanisms. CAD Comput. Aided Des. **134**. (2021)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.