International Journal of Dynamics and Control (2024) 12:2467–2479
https://doi.org/10.1007/s40435-023-01377-1

Modeling and analysis of an RUU Delta Robot using SolidWorks and SimMechanics

Hoai Nam Le[1] · **Nhu Thanh Vo**[1]

Received: 31 July 2023 / Revised: 4 December 2023 / Accepted: 22 December 2023 / Published online: 28 January 2024

Abstract

This paper illustrates the methodology for modeling, implementing, and controlling an RUU Delta Robot using computer-aided design and simulation software. Specifically, SolidWorks is employed to construct a CAD model with suitable properties, which is then exported to the MATLAB/SimMechanics environment for generating a multibody system block diagram. A PID controller is implemented to control the robot's position and trajectory movement. The study analyzes moment signal graphs at active joints, comparing them with existing research to validate the accuracy of the results. The investigation reveals that the influence of damping coefficients on the robotic dynamics is negligible, with simulation errors less than 0.02% for different end effect masses. Additionally, the study explores the impact of end effect mass on robotic dynamics and trajectory. Simulation results recommend an optimal operating range with an end effect mass of less than 3 kg, ensuring trajectory errors remain below 2%. As the end effect mass increases, a corresponding fluctuation in the robot's trajectory is observed, leading to longer times to reach the designed trajectory. This study provides valuable insights for practical applications, indicating that the proposed simulation approach is instrumental in assessing the performance of the robot controller before its deployment in an actual prototype.

Keywords SolidWorks · Simulation · MATLAB SimMechanics · RUU Delta Robot · CAD model · PID controller

1 Introduction

The RUU-type Delta Robot, pioneered by Professor Reymond Clavel's team in the early 1980s [1], is a successful parallel robot with various industrial applications, particularly in picking and packaging tasks due to its high-speed performance. The kinematic and dynamic solution of the Delta Robot is complex, making it challenging to use dynamic equations for control simulation [2]. Prior research has attempted to address the control challenges of the Delta Robot. Some methods involved parameter tuning of fixed gain motion controllers [3], while others proposed uncertainty robust designs using interval type-2 fuzzy logic controllers [4]. However, these approaches were applied to dynamic models that might not fully represent the real-world system. In this paper, we took a different approach by utilizing SolidWorks to design a precise 3D model, which was then exported to the MATLAB/SimMechanics environment for control implementation. Previous studies by Sen, et.al, [5] and Ibrahim [6], and Olaya [7] have tackled similar issues but focused on Scara and mobile robots and controlled only the motion of actuators or joints. An analytical solution for the dimensional synthesis of a 3-DOF delta parallel robot was given by Dastjerdi and his team [8]. Other studies indicated the kinematic solution and analytical dynamic modeling of Delta Robot by theory studies and then experiment verified [9, 10]. These approaches need a lot of calculation and time-consuming for building prototypes for experiment testing. Our study has a different approach that applies the simulation process with a 3D design using advanced CAD software that would give a realistic model for experiment verification.

Simulation offers various benefits in the design and development process, providing valuable insights and enhancing the overall product quality [11]. One advantage is cost-efficiency, as engineers can virtually test and optimize designs, reducing the need for physical prototypes and saving on materials and manufacturing costs [12, 13]. Additionally, simulation accelerates the design process, enabling rapid virtual testing compared to physical prototyping, leading to

✉ Nhu Thanh Vo
vnthanh@dut.udn.vn

[1] Department of Mechanical Engineering, The University of Danang – Danang University of Science and Technology, Nguyen Luong Bang 54, Danang, Vietnam

quicker product development [14]. Furthermore, simulations facilitate iterative improvement, allowing designers to refine and enhance designs by modifying parameters and variables, resulting in more robust and efficient final products [15, 16]. Another advantage is optimization, as designers can explore a wide range of design options to find the optimal solution for performance, efficiency, and other criteria [17, 18].

The use of simulation methods in engineering design is well documented in the review article by Jack Collins and his team [11]. The authors extensively explore various simulation techniques applied in robotics, emphasizing the benefits of simulations in testing and validating robot designs before physical prototyping. By leveraging simulation tools and techniques before creating prototypes, engineers can achieve a more efficient design process, reduce development time, and enhance the reliability and quality of the final product. SimMechanics is preferred over Simulink for specific simulations and modeling tasks due to various advantages such as accurate physical modeling, intuitive interface and real-time interaction, advanced structural analysis, automatic hardware-in-the-loop testing and code generation and multi-domain support, fast modeling configuration and integration [19–25]. Indeed, the study by Cretescu, et.al [25] highlights the influence on the dynamic behavior of a Delta Robot by examining the individual and combined effects of clearances and friction in the spherical joints, along with the flexibility of the rod elements, utilizing the Catia and Adams software packages for CAD modeling and simulating its motion on a representative spatial trajectory, reaching the maximum allowed values of speed and acceleration. Utilizing the Euler–Lagrange approach, an inverse dynamic model is developed and validated with real torque data obtained from a model of the Delta parallel robot created by Falezza and his team [26]. However, these studies lack attention to the effect of end effect mass attached to the moving platform of the robot.

This study focuses on modeling and controlling an RUU Delta Robot. The dynamic model of the parallel robot was automatically developed by importing the 3D CAD assembly model from SolidWorks into the MATLAB/Simulink programming environment. This methodology streamlines the dynamic modeling process, making it more convenient. The model was further improved by incorporating sensing and actuating blocks. The first part of the manuscript introduces the mathematical model with the kinematic and dynamic of an RUU Delta Robot. Then, The CAD model is built using SolidWorks software and then exported to the MATLAB/SimMechanics environment for applying a control system. The effect of joint stiffness and damping coefficients on the robotic dynamic is negligible according to [25] and also is verified in this study. In this paper, PID controllers are employed to achieve precise position control for the three joints. The parameters of the PID controllers will be determined to optimize their performance before implementing them in the actual model. The study examines moment signal graphs at active joints, comparing them with existing research to affirm the precision of the analysis and calculation results. Additionally, the study investigates the influence of end effect mass on robotic dynamics and trajectory. Recommendations regarding the selection of the end effect mass are provided based on an analysis of the simulation results. This approach is suitable for evaluating the controller's effectiveness and ensuring accurate position control for the robot's joints.

Table 1 Description of Delta Robot parameters

Symbol	Description	Unit
P_i	Connection point between parallelogram arm and the moving platform	–
s_B	Length of the side of the equilateral triangle of the fixed platform	mm
w_B	Distance from center O to the edge of the fixed platform	mm
u_B	Distance from center O to the vertex of the fixed base	mm
s_P	Length of side of the equilateral triangle of the moving platform	mm
w_P	Distance from center P to the edge of the moving platform	mm
u_P	Distance from center P to the vertex P_i of the moving platform	mm
L	Length of the arm B_iA_i $(i = 1, 2, 3)$	mm
l	Length of each parallelogram arm	mm
h	Width of each parallelogram arm	mm

2 Mathematical modeling

In this study, we apply the model that was constructed in a previous study [2] as shown in Fig. 1. The Delta Robot consists of 3 arms connected to maintain the translational motion of the mobile platform to the fixed platform; each arm includes a revolute joint placed at points B_i and two universal joints placed at points A_i and P_i with $i = \{1, 2, 3\}$ for the parallel mechanism. Coordinate system $\{B\}$ is attached to the fixed platform, and coordinate system $\{P\}$ is attached to the moving platform. The joint variables are $\theta = \{\theta_1, \theta_2, \theta_3\}^T$, and the coordinates of point P in coordinate system $\{B\}$ are represented as ${}^B P_p = [xyz]^T$. The structural parameters of the robot are presented in detail in Table 1.

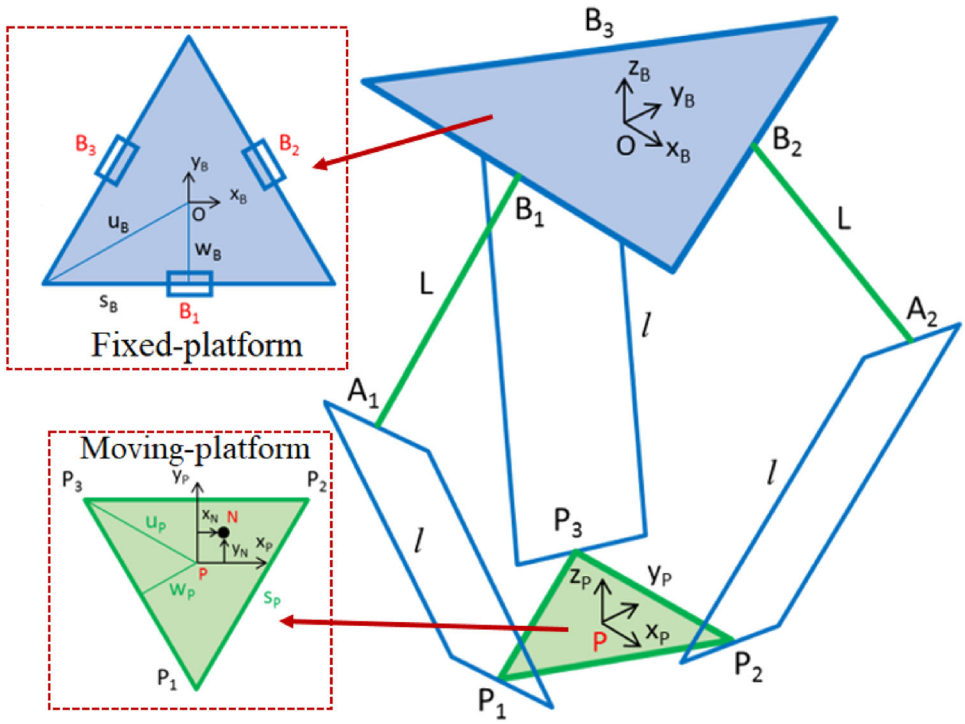

Fig. 1 Geometric parameters of Delta Robot

2.1 Kinematic system

According to [2], three kinematic equations are determined and expressed as shown in Eq. (1):

$$\begin{aligned} f_1 &= 2L(y+a)\cos\theta_1 + 2zL\sin\theta_1 \\ &\quad + x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2 = 0 \\ f_2 &= -L\left(\sqrt{3}(x+b) + y + c\right)\cos\theta_2 + 2zL\sin\theta_2 \\ &\quad + x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2xb + 2yc - l^2 = 0 \\ f_3 &= L\left(\sqrt{3}(x-b) - y - c\right)\cos\theta_3 + 2zL\sin\theta_3 \\ &\quad + x^2 + y^2 + z^2 + b^2 + c^2 + L^2 - 2xb + 2yc - l^2 = 0 \end{aligned} \tag{1}$$

where

$$a = w_B - u_P \quad b = \frac{s_p}{2} - \frac{\sqrt{3}}{2}w_B \quad c = w_p - \frac{1}{2}w_B$$

The kinematic equations are represented in vector form containing generalized coordinates as

$$\boldsymbol{s} = \begin{bmatrix} q \\ x \end{bmatrix} \quad \boldsymbol{q} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \quad \boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2}$$

where **s** is the vector containing the complete set of generalized coordinates,$\boldsymbol{q}$ is the vector containing the minimum set of independent generalized coordinates;$\boldsymbol{x}$ is the vector containing the operational coordinates.

$\boldsymbol{f}(\boldsymbol{s}) = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{x}) = 0, \quad \boldsymbol{f} \in \mathbb{R}^3, \quad \boldsymbol{q} \in \mathbb{R}^3, \quad \boldsymbol{x} \in \mathbb{R}^3$The three constraint equations f_1, f_2, f_3 relate the 6 generalized coordinates, including 3 independent coordinates, expressed in vector form as follows

$$\boldsymbol{f}(\boldsymbol{s}) = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{x}) = 0, \quad \boldsymbol{f} \in \mathbb{R}^3, \quad \boldsymbol{q} \in \mathbb{R}^3, \quad \boldsymbol{x} \in \mathbb{R}^3 \tag{3}$$

Differentiating Eq. (3) with respect to time, we have Eq. (4).

$$\dot{\boldsymbol{f}} = \boldsymbol{J}_q\dot{\boldsymbol{q}} + \boldsymbol{J}_x\dot{\boldsymbol{x}} = 0 \tag{4}$$

where $\boldsymbol{J}_q$,$\boldsymbol{J}_x$ are Jacobian matrixes.

Continuing with the second derivative, we have Eq. (5),

$$\ddot{f} = \dot{\boldsymbol{J}}_q\dot{\boldsymbol{q}} + \boldsymbol{J}_q\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_x\dot{\boldsymbol{x}} + \boldsymbol{J}_x\ddot{\boldsymbol{x}} = 0 \tag{5}$$

The inverse kinematic equation can be solved by transforming it into the form $E_i\cos\theta_i + F_i\sin\theta_i + G_i = 0$ with $i = 1, 2, 3$ [10]. However, this method yields 8 valid solutions, but only one solution is selected. To address this issue, Puglisi and his team proposed a technique for solving this problem using the Newton–Raphson numerical method [27] which is applied in this study.

Equations (4) and (5) yield the velocity and acceleration of the generalized coordinates as shown in Eqs. (6) and (7),

respectively

$$\dot{\boldsymbol{q}} = -\boldsymbol{J}_q^{-1}\boldsymbol{J}_x\dot{\boldsymbol{x}} \tag{6}$$

$$\ddot{\boldsymbol{q}} = -\boldsymbol{J}_q^{-1}(\boldsymbol{J}_q\dot{\boldsymbol{q}} + \boldsymbol{J}_x\dot{\boldsymbol{x}} + \boldsymbol{J}_x\boldsymbol{x}) \tag{7}$$

2.2 Dynamic system

The system consists of 7 rigid bodies (p) with 3 degrees of freedom (f) and 6 redundant generalized coordinates (m). There are 3 independent constraint equations ($r = 3$) linking the bodies ($j = 1,\ldots r$) [2].

$$f_j = f(s, t) = f\left(q_1, q_2, \ldots, q_m, t\right) = 0 \tag{8}$$

The Lagrange equations in multiplier form:

$$\boldsymbol{M}(\boldsymbol{s})\ddot{\boldsymbol{s}} + \boldsymbol{C}(\boldsymbol{s}, \dot{\boldsymbol{s}})\dot{\boldsymbol{s}} + \boldsymbol{g}(\boldsymbol{s}) + \boldsymbol{J}_s^{\mathrm{T}}(\boldsymbol{s})\boldsymbol{\lambda} = \boldsymbol{Q}^{np} \tag{9}$$

where

$\boldsymbol{M}(\boldsymbol{s})$ is the 6×6 matrix of generalized mass;

$\boldsymbol{C}(\boldsymbol{s}, \dot{\boldsymbol{s}})$ is the 6×6 matrix of inertial and Coriolis;

$$\boldsymbol{C}(\boldsymbol{s}, \dot{\boldsymbol{s}}) = \frac{\mathrm{d}\boldsymbol{M}(\boldsymbol{s})}{\mathrm{d}t} - \frac{1}{2}\left(\frac{\partial(\boldsymbol{M}(\boldsymbol{s})\dot{\boldsymbol{s}})}{\partial \boldsymbol{s}}\right)^{\mathrm{T}} \tag{10}$$

$\boldsymbol{g}(\boldsymbol{s})$ is the 6×1 matrix of gravitational effects;

$$\boldsymbol{g}(\boldsymbol{s}) = \left(\frac{\partial \boldsymbol{\Pi}}{\partial \boldsymbol{s}}\right)^{\mathrm{T}} \tag{11}$$

$\boldsymbol{J}_s$ is the 3×6 Jacobian matrix of the complete set of generalized coordinates;

$\boldsymbol{\lambda}$ is the 3×1 vector of Lagrange multipliers;

$\boldsymbol{Q}^{np}$ is the 6×1 vector containing the generalized forces of non-conservative forces.

The fixed coordinate systems $\{B_i\} = B_i x_i y_i z_i$ with $i = 1$, 2, 3 are determined by transforming the original coordinate system through two operations: translating the origin O to B_i and rotating around the z_{B} axis by an angle α_i, where $\alpha_1 = -90°$, $\alpha_2 = 30°$, $\alpha_3 = 150°$ as shown in Fig. 2. The cosine matrix represents the direction of $\{B_i\}$ with respect to $\{B\}$ is $A_Z(\alpha_i)$.

Let C_{1i} be the center of mass of link $B_i A_i$.. The dynamic coordinate system $\{C_{1i}\} = C_{1i}x_{1i}y_{1i}z_{1i}$ s rigidly attached to link $B_i A_i$ so that $B_i A_i$ always lies on the x_{1i} axis. $\{C_{1i}\}$ is determined by transforming the coordinate system $\{B_i\}$ through 2 operations: translating to the origin C_{1i} and rotating around y_i axis by an angle θ_i.

The coordinates of vectors ${}^B\boldsymbol{C}_{1i}$ with $i = 1$, 2, 3 are given by Eq. (12).

$${}^B\boldsymbol{C}_{1i} = {}^B B_1 + A_Z(\alpha_i)^{B1}\boldsymbol{C}_{1i} \tag{12}$$

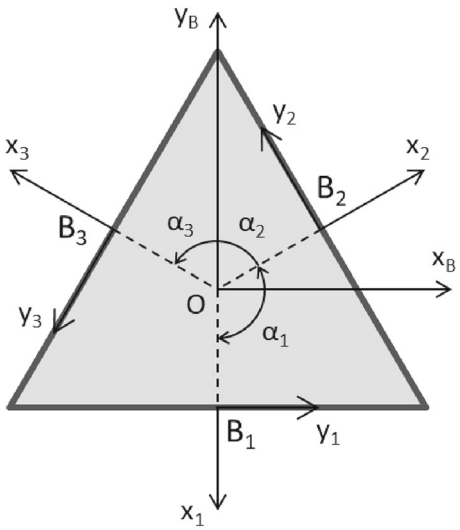

Fig. 2 Configuration parameters of the coordinate systems

Since the mass of parallelogram arms is usually smaller than the remaining links, to simplify the calculation process, the mass of the passive link $A_i P_i$ consolidated at the two joint ends, the mass will be concentrated at $C_{2ai} \equiv A_i$ and $C_{2bi} \equiv P_i$. By consolidating the mass at the two joint ends, the angular velocity and inertial torque of the passive link can be neglected, significantly simplifying the problem..

The coordinates of vectors ${}^B\boldsymbol{C}_{2ai}$ and ${}^B\boldsymbol{C}_{2bi}$ with $i = 1$, 2, 3 are given by

$${}^B\boldsymbol{C}_{2ai} = {}^B B_1 + A_Z(\alpha_i)^{B1}\boldsymbol{C}_{2ai} \tag{13}$$

$${}^B\boldsymbol{C}_{2bi} = {}^B\boldsymbol{P} + {}^P\boldsymbol{P}_i \tag{14}$$

For calculating the matrix of generalized mass $\boldsymbol{M}(\boldsymbol{s})$ in Eq. (9), we have the following relation.

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{s}) &= \sum_{i=1}^{3}\left(\begin{array}{l} m_1\boldsymbol{J}_{T_{1i}}^{\mathrm{T}}\boldsymbol{J}_{T_{1i}} + \frac{1}{2}m_2\left(\boldsymbol{J}_{T_{2ai}}^{\mathrm{T}}\boldsymbol{J}_{TC2ai} + \boldsymbol{J}_{T_{2bi}}^{\mathrm{T}}\boldsymbol{J}_{T_{2bi}}\right) \\ +\boldsymbol{J}_{R_{1i}}^{(1i)T}\boldsymbol{I}_{C1i}^{(1i)}\boldsymbol{J}_{R_{1i}}^{(1i)} \end{array}\right) \\ &\quad + m_{\mathrm{P}}\boldsymbol{J}_{T_{\mathrm{P}}}^{\mathrm{T}}\boldsymbol{J}_{T_{\mathrm{P}}}\end{aligned} \tag{15}$$

where $\boldsymbol{J}_{T_{1i}}, \boldsymbol{J}_{T_{2ai}}, \boldsymbol{J}_{T_{2bi}}, \boldsymbol{J}_{T_{\mathrm{P}}}$ are the Jacobian matrix of translation of the center of mass for the active, passive, and final compliant links in the fixed reference frame. $\boldsymbol{J}_{R_{1i}}^{(1i)}$ is the Jacobian matrix of rotation for the active links when projecting the angular velocity vector $\vec{\omega}_{1i}$ onto the reference frame $\{C_{1i}\}$. m_1, m_2, m_{P} represent the mass of the active, passive, and final compliant links. The inertial moment of inertial of the

active link relative to its center of mass is given by Eq. (16).

$$\boldsymbol{I}_{C1i}^{(1i)} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{16}$$

Calculate the matrices $\boldsymbol{J}_{T_{1i}}$, $\boldsymbol{J}_{T_{2ai}}$, $\boldsymbol{J}_{T_{2bi}}$, $\boldsymbol{J}_{T_{\mathrm{P}}}$, $\boldsymbol{J}_{R_{1i}}^{(1i)}$ and replace them to Eq. (15) to obtain a matrix that has nonzero elements only on the diagonal, as shown in Eqs. (17) and (18), with all other elements being zero.

$$m_{11} = m_{22} = m_{33} = I_y + \frac{L^2}{4}(m_1 + 2m_2) \tag{17}$$

$$m_{44} = m_{55} = m_{66} = \frac{3}{2}m_2 + m_{\mathrm{P}} \tag{18}$$

For calculating the matrix of inertial and Coriolis $\boldsymbol{C}(\boldsymbol{s}, \dot{\boldsymbol{s}})$, we have

$$\boldsymbol{C}(\boldsymbol{s}, \dot{\boldsymbol{s}}) = \frac{\mathrm{d}\boldsymbol{M}(\boldsymbol{s})}{\mathrm{d}t} - \frac{1}{2}\left(\frac{\partial(\boldsymbol{M}(\boldsymbol{s})\dot{\boldsymbol{s}})}{\partial \boldsymbol{s}}\right)^{\mathrm{T}} = 0 \tag{19}$$

For calculating the vector of non-conservative generalized forces $\boldsymbol{Q}^{\boldsymbol{np}}$ with the actuator torque $\vec{\tau}_1$, $\vec{\tau}_2$, $\vec{\tau}_3$ placed at the active joints, we use Eq. (20) as follows:

$$\boldsymbol{Q}^{np} = \begin{bmatrix} \frac{\partial \vec{\omega}_1}{\partial \dot{\boldsymbol{q}}_1} & \cdots & \frac{\partial \vec{\omega}_{\mathrm{p}}}{\partial \dot{\boldsymbol{q}}_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \vec{\omega}_1}{\partial \dot{\boldsymbol{q}}_m} & \cdots & \frac{\partial \vec{\omega}_{\mathrm{p}}}{\partial \dot{\boldsymbol{q}}_m} \end{bmatrix} \begin{bmatrix} \vec{\tau}_1 \\ \vec{\tau}_2 \\ \vec{\tau}_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{20}$$

For calculating the matrix of gravitational effects $\boldsymbol{g}(\boldsymbol{s})$, we have the following relation based on the potential energy of the Delta Robot.

$$\boldsymbol{\Pi} = -\sum_{i=1}^{3}\left[\frac{1}{2}m_1 gL\sin\theta_i + \frac{1}{2}m_2 gL\sin\theta_i\right] + \left(m_{\mathrm{P}} + \frac{3}{2}m_2\right)gz \tag{21}$$

Thus,

$$\boldsymbol{g}(\boldsymbol{s}) = \left(\frac{\partial \boldsymbol{\Pi}}{\partial \boldsymbol{s}}\right)^{\mathrm{T}} = \begin{bmatrix} -\frac{1}{2}(m_1 gL + m_2 gL)\cos\theta_1 \\ -\frac{1}{2}(m_1 gL + m_2 gL)\cos\theta_2 \\ -\frac{1}{2}(m_1 gL + m_2 gL)\cos\theta_3 \\ 0 \\ 0 \\ \left(\frac{3}{2}m_2 + m_{\mathrm{P}}\right)g \end{bmatrix} \tag{22}$$

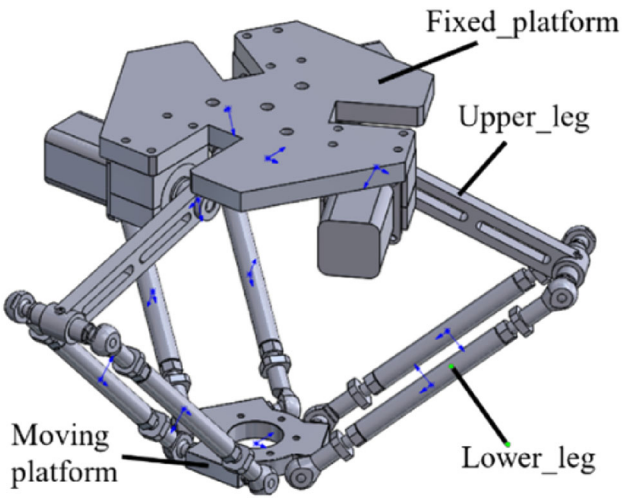

Fig. 3 CAD assembly from SolidWorks

Table 2 Mass of CAD parts

CAD part	Mass (g)	Quantity
Upper leg	382.78	3
Lower leg	339.27	6
Moving platform	757.51	1

The inverse dynamics problem can be solved directly or by transforming it into independent generalized coordinates [10].

3 Materials and methods

3.1 RUU Delta Robot CAD structure

As mentioned in the previous section, the RUU Delta Robot consists of three linked legs connected to universal joints at the base. The 3D design of each leg contains one revolute (R) joint and two universal (U) joints as shown in Fig. 3. It contains one fixed platform, three upper legs, six lower legs, and one moving platform. The parallelogram mechanism consisting of four rods in the three links below ensures reciprocating motion. There are three servo motors for driving the movement of the robot. The mass of each part is shown in Table 2.

3.2 Modeling process

Figure 4 illustrates the process of modeling and controlling the robot using SolidWorks and MATLAB. Firstly, all the CAD parts are created and assembled using SolidWorks. This process begins with the meticulous design of the robotic

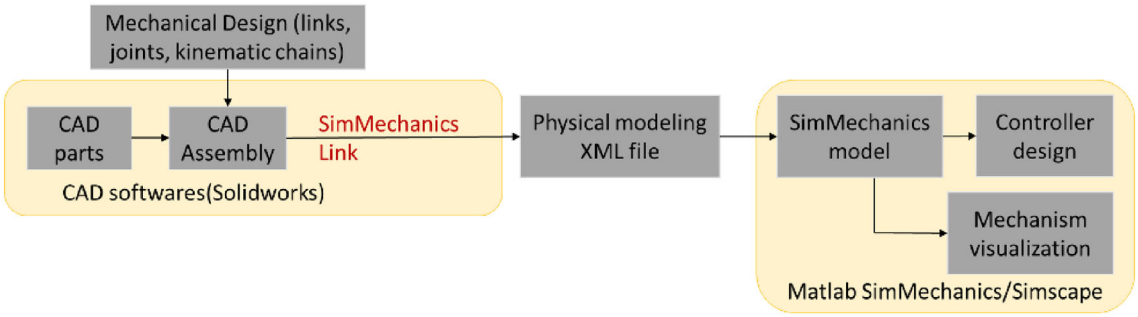

Fig. 4 Modeling and controlling robot process

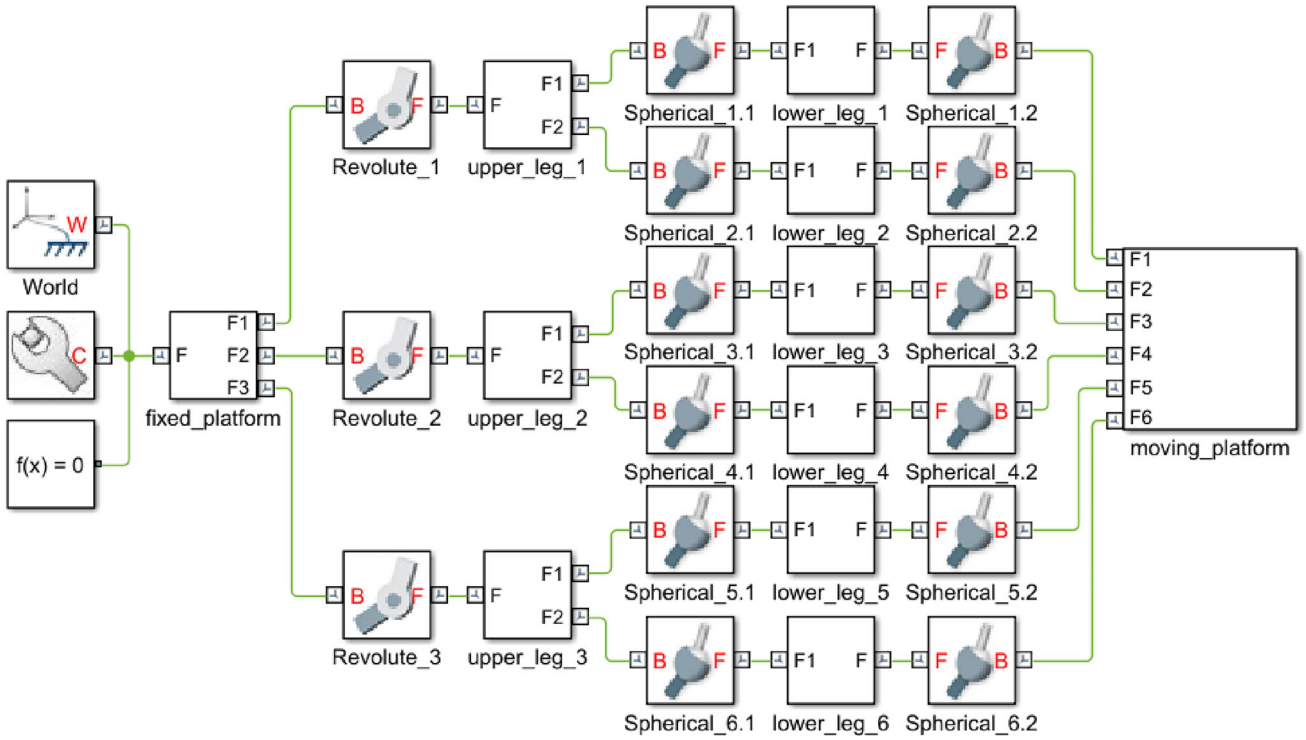

Fig. 5 RUU Delta Robot CAD model in Simulink environment

system within the SolidWorks environment, incorporating detailed mechanical components, joints, and physical properties, for instance, mass of robotic legs, as indicated in Table 2. Next, the physical modeling XML files are extracted for suitable import into the MATLAB Simulink environment. Once imported, actuators, sensors, and control components are incorporated into the SimMechanics environment to construct a comprehensive simulation setup. The 3D visualization model is constructed by connecting appropriate blocks in the SimMechanics toolbox of MATLAB Simulink. Simulation parameters, such as solver settings and simulation duration, need to be properly configured using Simulink and SimMechanics.

In this study, a fixed step size of 0.005 is chosen to ensure uniform output dimensions for all signals. The selected solver is ode14x, which employs a combination of Newton's method and extrapolation from the current value to calculate the model state as an implicit function of the state and the state derivative at the next time step. The final stage involves designing the controller with simulated inputs to visualize and analyze the mechanism. The iterative nature of this methodology allows designers to refine both the mechanical design and control systems based on simulation outcomes, fostering an optimized robotics system.

The CAD assembly is transformed into Simulink blocks as Fig. 5 and visualized as indicated in Fig. 6.

In MATLAB, the model defines a revolute joint as a 'revolute' block and a universal joint as a 'spherical' block. The model incorporates 3 revolute joints that connect the upper legs to the fixed platform, 6 universal joints linking

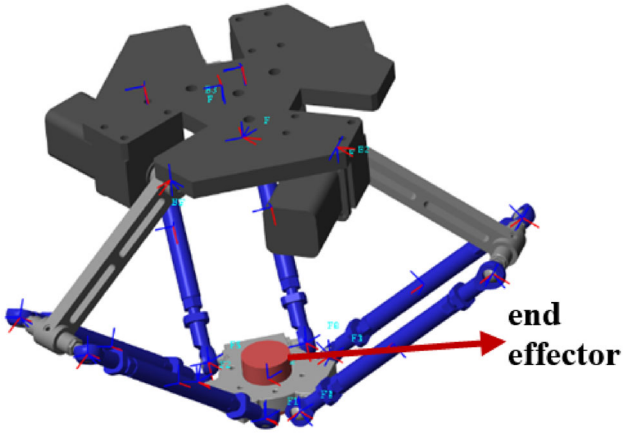

Fig. 6 RUU Delta Robot 3D visualization with end effector

the lower legs to the upper legs, and an additional 6 universal joints connecting the lower legs to the moving platform. A spherical joint, also known as a ball joint, allows rotational movement about multiple axes simultaneously. In contrast, a universal joint provides rotation about two intersecting axes. The decision to choose a spherical joint over a universal joint is influenced by the need for increased flexibility and range of motion in the robotic structure. The spherical joint also allows the end effect or connected link to rotate freely in all directions, providing a more versatile and adaptable robotic design. Each converted CAD block contains several ports, representing different coordinates transformed from the reference coordinate system of the CAD part. These ports establish joint definitions between two points.

As per the findings of the study conducted by Cretescu and his team [25], the impact of joint damping coefficients is deemed negligible. To examine and verify this assertion, we conducted and compared simulation results for two cases: one with joint damping coefficients and the other without. A joint damping coefficient of 0.01 Nms/rad is selected for all joints, and the spring stiffness is set to 0, as its substantial impact on driving torques, as identified in [25], is not the primary focus of this study. The primary focus is on evaluating the influence of the end effector on the dynamic behavior of the robot and considering stiffness may introduce coupling effects that interfere with the testing results.

3.3 PID position controller

The objective of this study is to achieve position control for the RUU Delta Robot. The control system comprises a PID controller and a controlled PWM voltage, with the PWM signal being converted by the H-bridge to a higher voltage signal that dictates the torque applied to the three DC motors [2, 5, 24]. Joint position sensors provide angle values (theta)

Table 3 DC motor parameters

Parameters	Value	Unit
Armature resistance	1	Ohm
Armature inductance	0.005	H
Torque constant	0.05	Nm/A
Rotor inertia	0.0001	kgm^2

for the control system. The PID controller continually calculates the difference between the desired angle (thetaref) and the position sensor value (theta). The control system is implemented in Simulink blocks, as depicted in Fig. 7, resulting in a comprehensive model illustrated in Fig. 8. An end effector block is incorporated to account for load influence. The study utilizes three 24 V DC motors and employs PWM and H-bridge to control their torque. The actuator block and the PID controller are, respectively, depicted in Figs. 9 and 10. Table 3 outlines the electrical and torque characteristics of the DC motors.

4 Results and discussions

The simulation results of the inverse kinematics problem are compared and validated with two pertinent reference documents. One reference, by Quang, pertains to the motor's angular displacement [14], while the other, by Collins, relates to the robot's joint angles [10]. In comparison with the first reference document [14], the robot parameters are assigned values as shown in Table 4.

Coordinate positions over time as indicated in [14] are shown in Eq. (23).

$$\begin{cases} x = 0.3 \times \cos(2\pi t) \\ y = 0.3 \times \sin(2\pi t) \\ z = -0.7 \end{cases} \tag{23}$$

In comparison with the second reference document [10], the robot parameters are assigned values as shown in Table 5.

Coordinate positions over time as indicated in [10] are shown in Eq. (24).

$$\begin{cases} x = 0.5 \times \cos(t) \\ y = 0.5 \times \sin(t) \\ z = -1 + 0.2 \times \sin(2 \times t) \end{cases} \tag{24}$$

The simulation results, compared with the referenced works [10, 14], are presented in Figs. 11 and 12. In Fig. 11a, the angular response of the motor from the reference study is shown, while Fig. 11b displays the simulation results of this study. Similarly, Fig. 12a depicts the angular response of

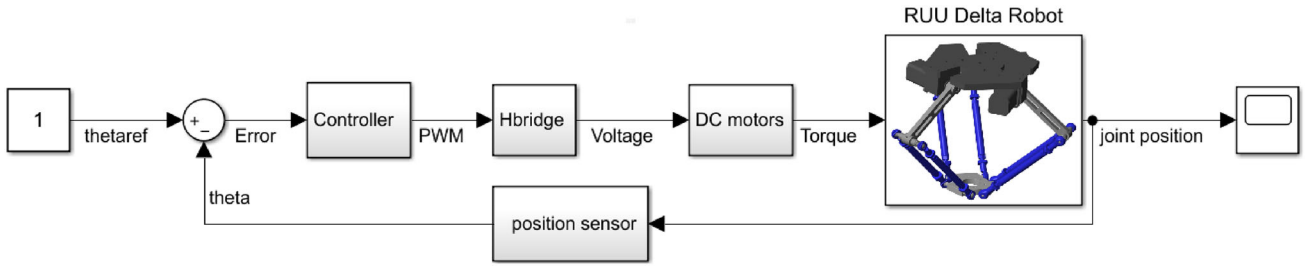

Fig. 7 Position control system of RUU Delta Robot

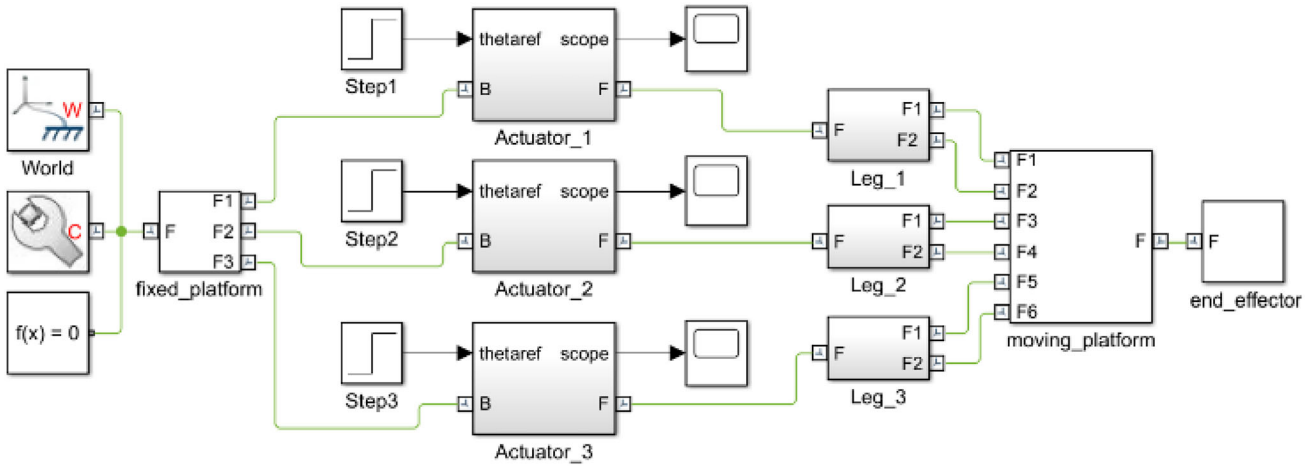

Fig. 8 Modeling RUU Delta Robot with step input

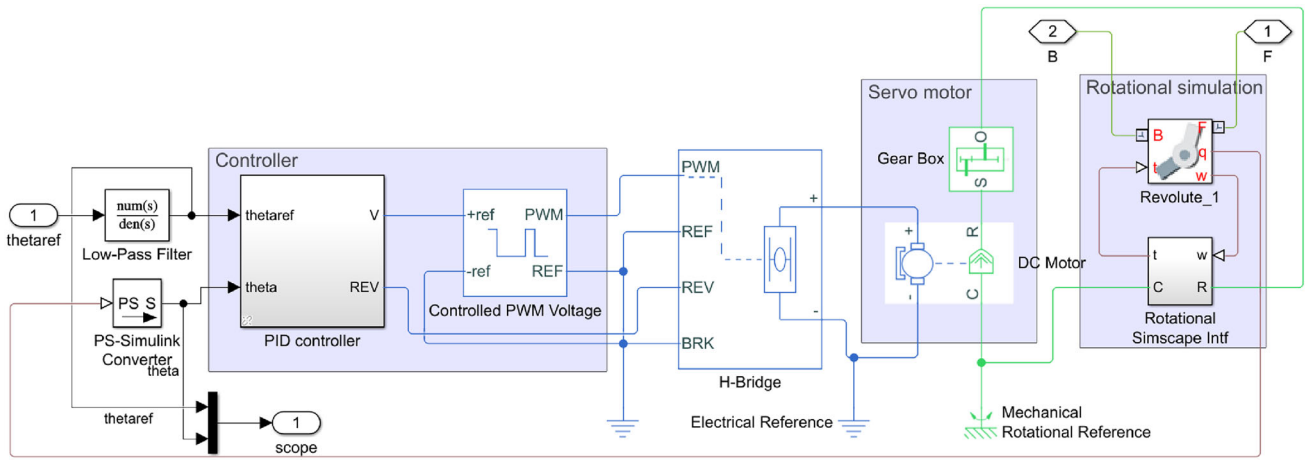

Fig. 9 Actuator block of leg 1

the robot joint from the reference study, and Fig. 12b shows the simulation results of this study. The observed similarities during the test run confirm the stability and reliability of the design and simulation conducted with SimMechanics software. Moreover, this study also includes additional simulation cases involving the mass of the object mounted on the moving platform of the robot, which will be discussed in the following section.

Considering the uniform efficiency of the three DC motors, we focus on one motor's response and use MATLAB's PID tuner tool to determine the appropriate controller parameters. The PID controller parameters (K_p, K_i,K_d, and filter coefficient N) are set at 5, 30, 0.1, and 300, respectively, achieving low overshoot, fast response time, and steady-state stability without oscillation. The simulation encompasses two scenarios: step response and circular trajectory response

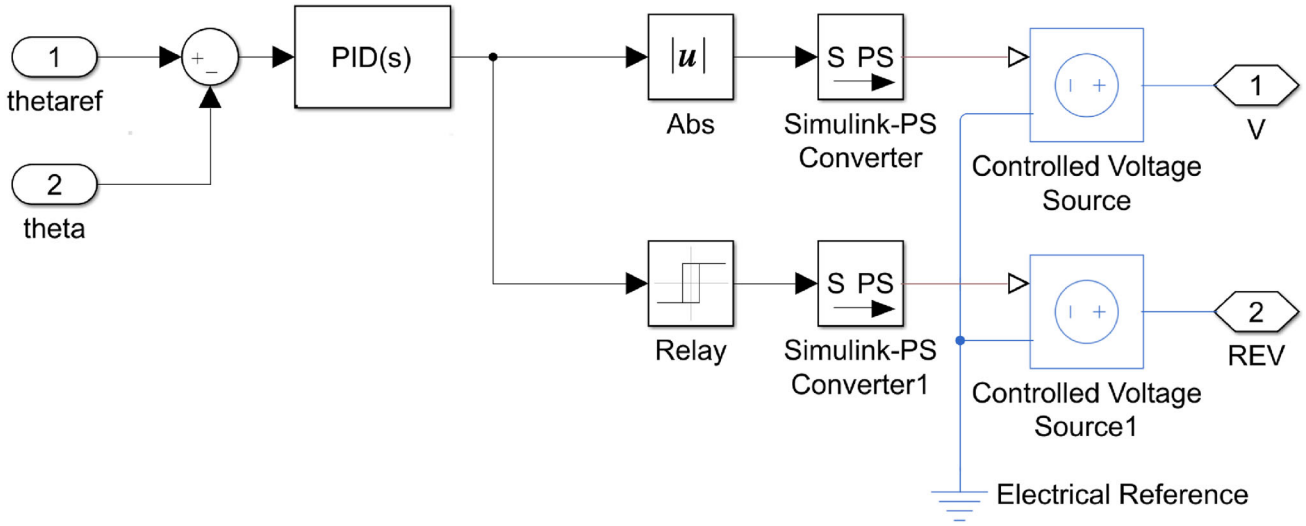

Fig. 10 PID controller block

Table 4 Geometric parameters of the robot based on reference [14]

Unit (m)				Unit (rad)			Unit (kg)		
L_1	L_2	R	R	α_1	α_1	α_1	m_1	m_2	m_3
0.3	0.8	0.26	0.04	0	$2\pi/3$	$4\pi/3$	0.42	2×0.2	0.75

Table 5 Geometric parameters of the robot based on reference [10]

S_b	S_p	L	l	h
0.46072 (m)	0.1645 (m)	0.292 (m)	0.372 (m)	0.131 (m)

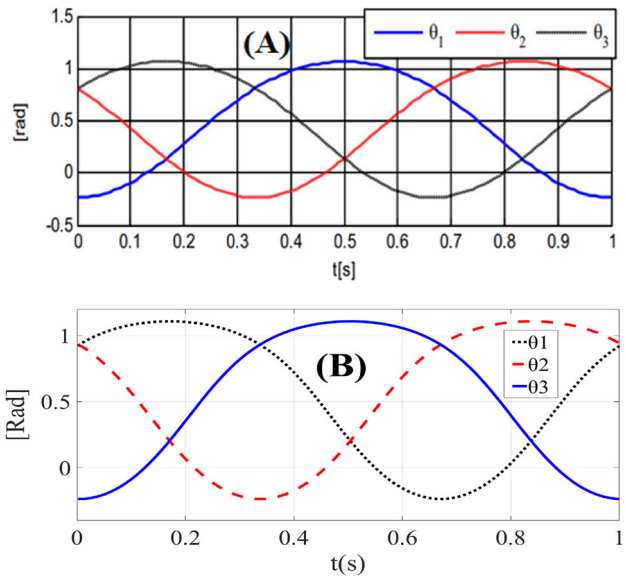

Fig. 11 Comparison of referenced work (**a**) [14] and simulation (**b**)

described in Eq. (25).

$$\begin{cases} x = R\cos\left(\dfrac{2\pi}{5}t - 1.3\pi\right); \\ y = R\sin\left(\dfrac{2\pi}{5}\int t - 1.3\pi\right); \\ z = -294.4106 \end{cases} \tag{25}$$

Figures 13 and 14 display the responses obtained through simulation using the PID controller. In Fig. 13, the graphs represent the motor angle responses to a step input. In Fig. 14, the graphs depict the motion angles of the Delta Robot's motors when starting from the center of the workspace and moving along a circular trajectory. By comparing the simulation "*theta*" request values with the actual output responses from the "*thetaref*" simulation, it is evident that the motors exhibit a fairly accurate response to the desired values.

For the step response, all the motors experience some degree of overshoot, with the first actuator having a minimum overshoot of approximately 6%, and the 2nd and 3rd actuators reaching a maximum overshoot of about 23%. Additionally, the system reaches a steady state within 0.3 s,

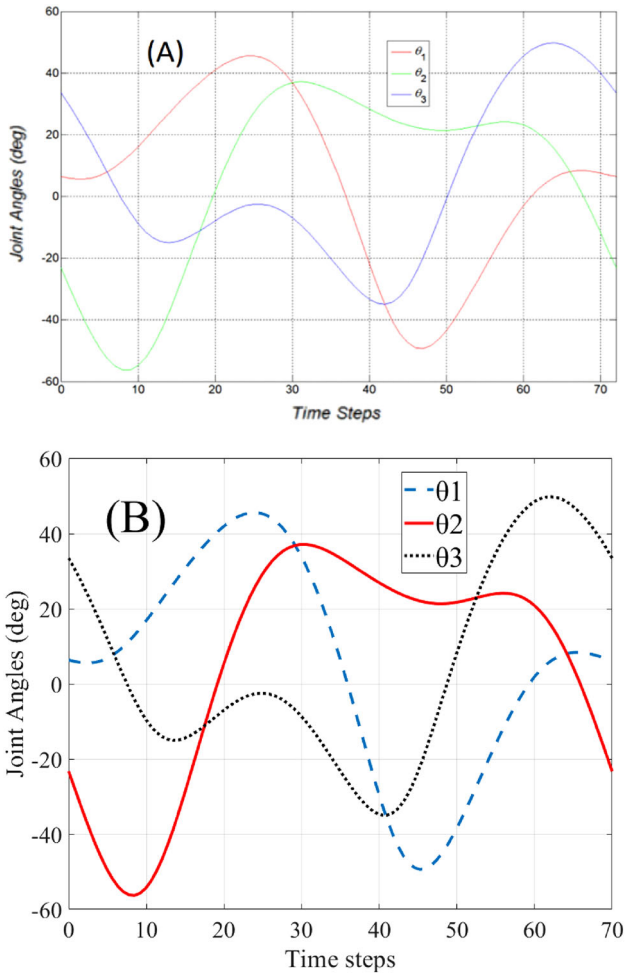

Fig. 12 Comparison of referenced work (**a**) [10] and simulation (**b**)

and during the settling period (after 0.3 s), the rotation angles of the actuators remain stable and closely align with the requirements.

As illustrated in Fig. 14, during the circular trajectory, the actuators show an initial overshoot of around 20%, with the signal in the first second being similar to the step response indicated in Fig. 13. The period from 1 to 7 s represents the time when the robot runs along a circular trajectory in the x,y plane. Notably, the simulation results in Fig. 15 demonstrate that when there is no load on the robot's moving platform, the actuators operate without deviating from the desired signal. This observation is further corroborated by the response trajectory shown in Fig. 15a, where the robot operates without oscillation while following the circular trajectory in the case of no end effector at the moving platform.

The accuracy of the position response may be compromised when the end effect load exceeds a specific limit. To investigate this issue, tests were conducted using various end effect loads: zero, 3 kg, 5 kg, 8 kg, 10 kg, and 15 kg. Figure 15b, c indicates that with a 3 kg and 5 kg load, there is a

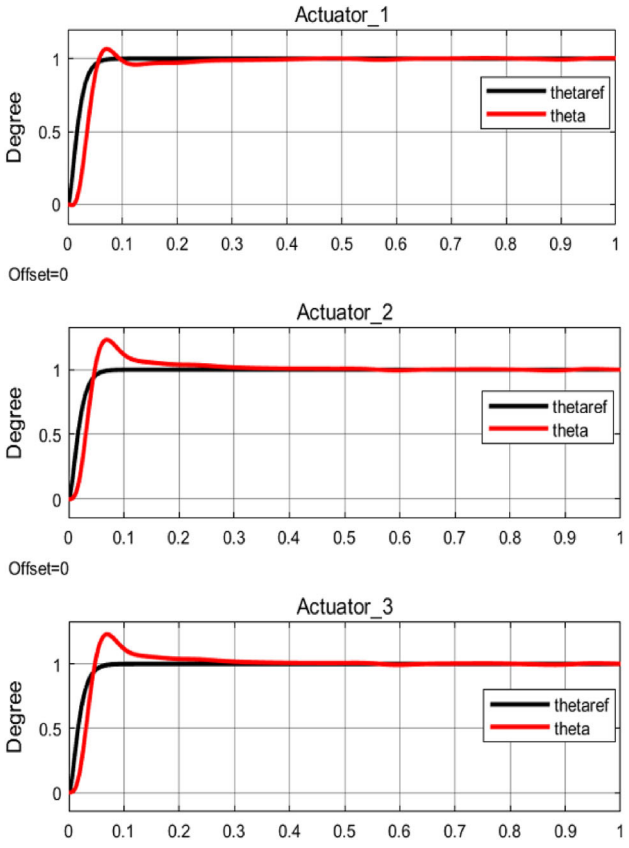

Fig. 13 Simulation result of step responses of three actuators

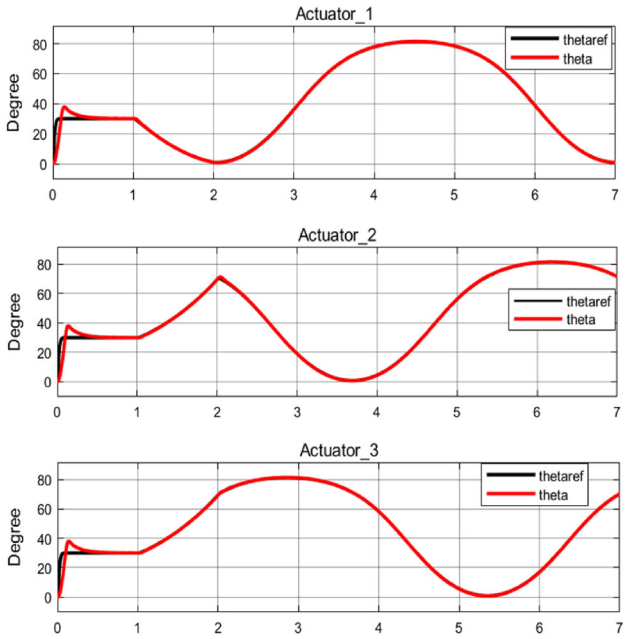

Fig. 14 Simulation result of responses to a circular trajectory

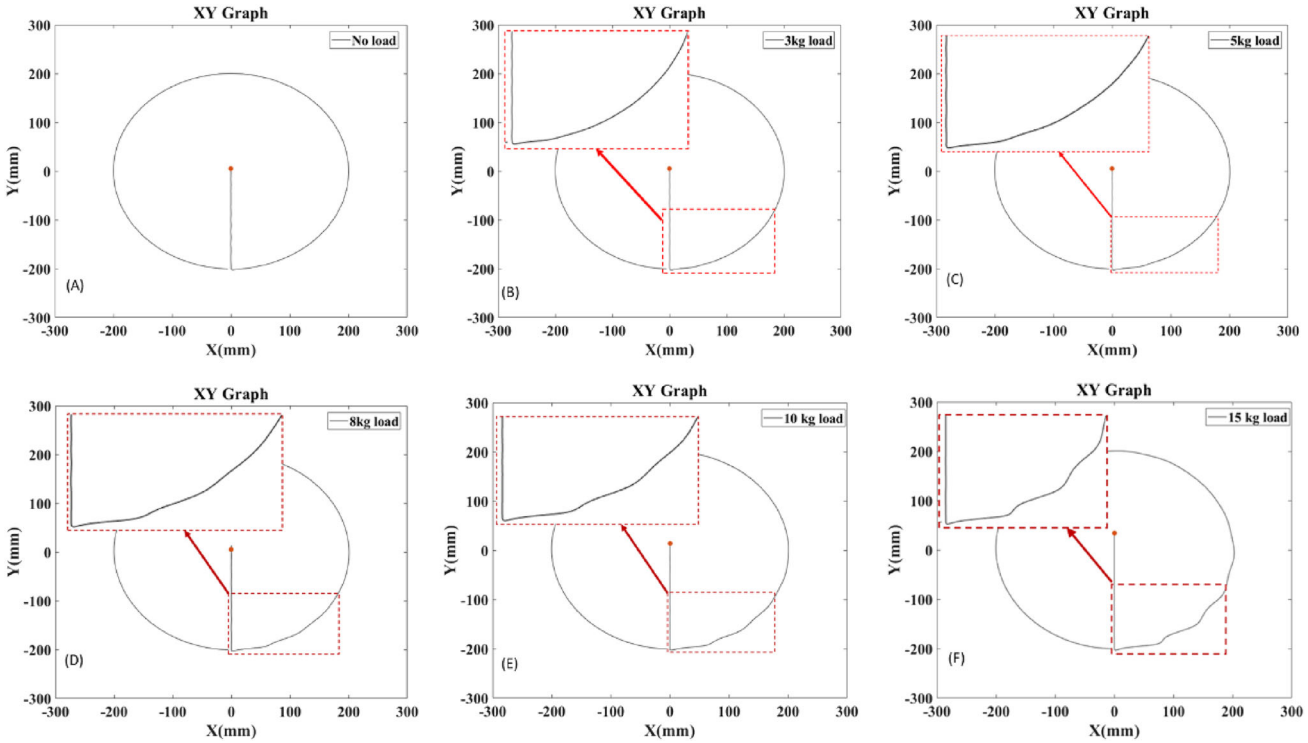

Fig. 15 Influence of end effect load on the position response. **a** No load, **b** 3 kg load, **c** 5 kg load, **d** 8 kg load, **e** 10 kg load, **f** 15 kg load

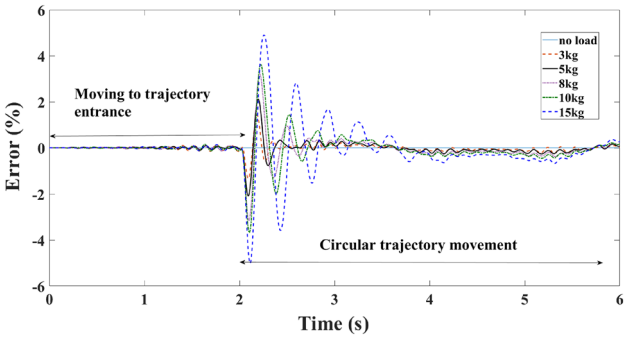

Fig. 16 Errors of robot moving trajectory affected by end effect loads

minor position deviation. However, Fig. 15d, e demonstrates a substantial deviation when the robot arm follows a circular trajectory with an 8 kg or 10 kg load. When the load increases further, such as with a 15 kg load, as shown in Fig. 15f, the trajectory experiences significantly higher fluctuation.

The errors of the robot moving trajectory affected by end effect loads are extracted, as indicated in Fig. 16, including two phases of moving to trajectory entrance and circular trajectory movement. The errors are high at the entrance of the circular trajectory, and the fluctuation is pronounced when the load is over 8 kg. The highest error, over 5%, occurs when the load is 15 kg, while for 10 kg, the error is close to 4%.

Based on these findings, it is recommended for the robot to operate with a lower 3 kg end effect load to mitigate fluctuations and maintain the trajectory error below 2%, as depicted in Fig. 16. The effect of the damping coefficient of robotic joints which is set to be 0.01 Nms/rad is confirmed to be negligible, as the errors are less than 0.02% for all simulation results of 3 kg, 5 kg, 8 kg, and 10 kg, as shown in Fig. 17.

5 Conclusions

This study demonstrates the modeling and control process of the RUU Delta Robot using SolidWorks and the SimMechanics Toolbox. CAD parts are assembled in SolidWorks, and then, Physical modeling XML files are extracted for import into Simulink. The 3D visualization model is created in the SimMechanics toolbox. The PID controller is employed for position control, with the controlled PWM voltage determining the torque applied to three DC motors. The simulation results are compared with relevant reference documents, confirming the stability and reliability of the design and simulation.

Moreover, the study evaluated the performance of the RUU Delta Robot through step response and circular trajectory simulations. The step response revealed varying levels of overshoot in the motors, reaching a steady state within 0.3 s. During the circular trajectory, an initial overshoot of

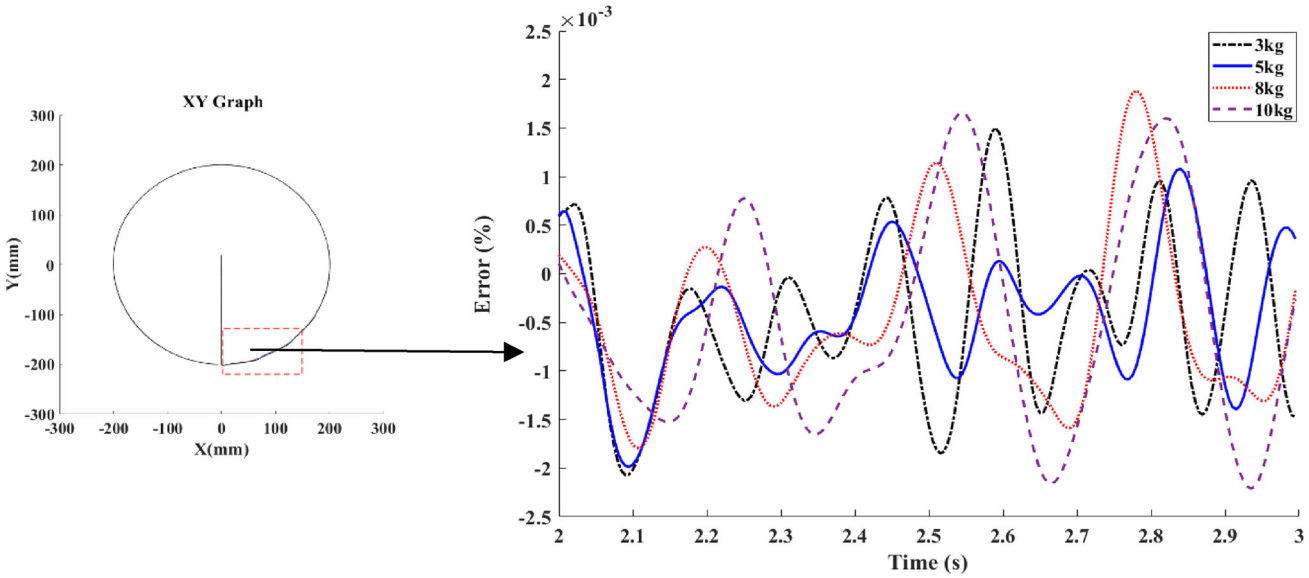

Fig. 17 Errors in the robot's trajectory caused by the damping coefficient of robotic joints

around 20% was observed. The influence of end-effect loads on position response was systematically investigated, indicating that the robot operates without deviation when there is no load. However, substantial position deviations were noted with increasing loads, especially in circular trajectories. The highest error, exceeding 5%, occurred with a 15 kg load. Recommendations include operating the robot with a lower 3 kg end-effect load to minimize fluctuations and maintain trajectory error below 2%. Damping coefficients of robotic joints were confirmed negligible. These findings contribute valuable insights for practical implementation and control optimization of the RUU Delta Robot.

Acknowledgements The authors would like to thank all members of Mechatronics Division, University of Science and Technology-The University of Da Nang, for their support and feedback.

Authors' contributions NTV and HNL contributed equally to this research.

Funding No funding was received on this project.

Availability of data and materials The data and source code used to support the findings of this study are available from the corresponding author upon request.

Declarations

Conflict of interests The authors declare that they have no competing interests.

References

1. Clavel R (1988) DELTA, a fast robot with parallel geometry. In: 18th international symposium on industrial robot, Lausane, pp 91–100
2. Le HN, Le XH (2018) Geometrical design of a RUU type delta robot based on the prescribed workspace. In: 4th international conference on green technology and sustainable development (GTSD), Ho Chi Minh City, Vietnam, pp 359–364. https://doi.org/10.1109/GTSD.2018.8595674
3. Qing Z, Panfeng W, Jiangping M (2015) Controller parameter tuning of delta robot based on servo identification. Chin J Mech Eng 28:267–275. https://doi.org/10.3901/CJME.2014.1117.169
4. Linda O, Manic M (2011) Uncertainty-robust design of interval Type-2 fuzzy logic controller for delta parallel robot. IEEE Trans Industr Inf 7(4):661–670. https://doi.org/10.1109/TII.2011.2166786
5. Sen MA, Bakircloglu V, Kalyoncu M (2017) Modelling and PID control of Scara Robot. In: International conference on engineering technologies (ICENTE), Turkey, pp 1–4
6. Ibrahim BSKK, Zargoun AMA (2014) Modelling and control of SCARA manipulator. Proc Comput Sci 42:106–113
7. Olaya J et al (2017) Analysis of 3 RPS robotic platform motion in SimScape and Matlab GUI environment. Int J Appl Eng ISSNN, Res India Public 12(8):1460–1468
8. Dastjerdi AH, Sheikhi MM, Masouleh MT (2020) A complete analytical solution for the dimensional synthesis of 3-DOF delta parallel robot for a prescribed workspace. Mech Mach Theory 153:103991
9. Asadi F, Heydai A (2020) Analytical dynamic modeling of Delta robot with experimental verification. Proc Inst Mech Eng Part K J Multi-Body Dyn 234:623–663
10. Robert L (2016) The Delta parallel robot: kinematics solutions. Mechanical engineering. Ph.D. thesis, Ohio University, Athens, Ohio
11. Collins J, Chand S, Vanderkop A, Howard D (2021) A review of physics simulators for robotic applications. IEEE Access 9:51416–51431
12. Tran THT, Nguyen DS, Vo NT, Le HN (2020) Design of delta robot arm based on topology optimization and generative design method. In: 5th international conference on green technology and sustainable development (GTSD), Ho Chi Minh City, Vietnam, pp 157–161. https://doi.org/10.1109/GTSD50082.2020.9303083

13. Nguyen HQ et al (2015) Influence of models on computed torque of delta spatial parallel robot. In: Proceeding of the 16th Asian Pacific vibration conference (APVC. 2015), pp 791–798
14. Quang NH, Quyen NV, Hai DT, Hien NN (2020) Dynamic modelling of 3-RUS spatial parallel robot manipulator. Rev Comput Eng Res 7(1):20–26. https://doi.org/10.18488/journal.76.2020.71.20.26
15. Subson S, Maneetham D, Aung MM (2022) Kinematics simulation and experiment for optimum design of a new prototype parallel robot. Int J Eng Trends Technol 70(10):350–362. https://doi.org/10.14445/22315381/IJETT-V70I10P234
16. Hussain S, Jamwal PK, Munir MT (2022) Computer-aided teaching using simmechanics and matlab for project-based learning in a robotics course. Int J Soc Robot 14:85–94. https://doi.org/10.1007/s12369-021-00769-7
17. Damic V, Cohodar M, Kobilica N (2019) Development of dynamic model of robot with parallel structure based on 3D CAD model. Ann DAAAM Proc 30:155–160
18. Carpin S, Lewis M, Wang J, Balakirsky S (2007) USARSim: a robot simulator for research and education. In: Proceedings IEEE international conference on robotics and automation, pp 1400–1405
19. Hugo H, Joan L (2020) The forward and inverse kinematics of a Delta robot, Chapter. In: Proceedings of the advances in computer graphics: 37th computer graphics international conference, CGI 2020, Geneva
20. Chebotar Y et al (2019) Closing the sim-to-real loop: Adapting simulation randomization with real-world experience. Int Conf Robot Autom (ICRA) 2019:8973–8979
21. Aguero C et al (2015) Inside the virtual robotics challenge: simulating real-time robotic disaster response. IEEE Trans Autom Sci Eng 12:494–506
22. Goury O, Duriez C (2018) Fast, generic, and reliable control and simulation of soft robots using model order reduction. IEEE Trans Robot 34:1565–1576
23. Faure F et al (2012) Sofa: a multi-model framework for interactive physical simulation. In: Payan Y (ed) Soft tissue biomechanical modeling for computer-assisted surgery. Springer, Berlin, pp 283–321
24. Le HN, Dang PV, Pham A-D, Vo NT (2020) System identifications of a 2DOF pendulum controlled by QUBE-servo and its unwanted oscillation factors. Arch Mech Eng 67(4):435–450
25. Cretescu N, Neagoe M, Saulescu R (2023) Dynamic analysis of a delta parallel robot with flexible links and joint clearances. Appl Sci 13(11):6693. https://doi.org/10.3390/app13116693
26. Falezza F, Vesentini F, Di Flumeri A, Leopardi L, Fiori G, Mistrorigo G, Muradore R (2022) A novel inverse dynamic model for 3-DoF delta robots. Mechatronics 83:102752
27. Puglisi LJ, Saltaren R, Garcia C, Cardenas P, Moreno H (2017) Implementation of a generic constraint function to solve the direct kinematics of parallel manipulators using Newton–Raphson approach. J Control Eng Appl Inf 19:71–79