ISA Transactions 79 (2018) 172–188

Contents lists available at ScienceDirect

ISA Transactions

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

Research article

Fractional order PID for tracking control of a parallel robotic manipulator type delta

L. Angel, J. Viola*

Universidad Pontificia Bolivariana, Autopista Piedecuesta km7, Bucaramanga, 681004, Colombia

ARTICLE INFO

Article history:
Received 21 June 2017
Received in revised form
9 March 2018
Accepted 20 April 2018
Available online 21 May 2018

Keywords:
Computed torque control
Delta robot
Fractional order PID controller
Parametric identification
Robustness evaluation
SOLIDWORKS/MSC-ADAMS/MATLAB co-simulation

ABSTRACT

This paper presents the tracking control for a robotic manipulator type delta employing fractional order PID controllers with computed torque control strategy. It is contrasted with an integer order PID controller with computed torque control strategy. The mechanical structure, kinematics and dynamic models of the delta robot are descripted. A SOLIDWORKS/MSC-ADAMS/MATLAB cosimulation model of the delta robot is built and employed for the stages of identification, design, and validation of control strategies. Identification of the dynamic model of the robot is performed using the least squares algorithm. A linearized model of the robotic system is obtained employing the computed torque control strategy resulting in a decoupled double integrating system. From the linearized model of the delta robot, fractional order PID and integer order PID controllers are designed, analyzing the dynamical behavior for many evaluation trajectories. Controllers robustness is evaluated against external disturbances employing performance indexes for the joint and spatial error, applied torque in the joints and trajectory tracking. Results show that fractional order PID with the computed torque control strategy has a robust performance and active disturbance rejection when it is applied to parallel robotic manipulators on tracking tasks.

1. Introduction

Robotic manipulators are mechanical structures employed in the industry to perform repetitive tasks in manufacturing processes. These ensure a high production volume with high quality standards [1–3]. Depending on the mechanical structure, industrial robots are classified as serial and parallel manipulators. A serial robotic manipulator has its links joined sequentially. For a parallel robotic manipulator, their links are joined between two platforms, one mobile and another fixed. Particularly for parallel robotic manipulators, designing the control system is a challenge due to its multivariable nature, the presence of non-linear elements and the high interaction between the robot links.

According to [4–6], to control parallel robotic manipulators in regulation tasks, some traditional robotic control strategies are employed as the PD control, the PD control with gravity compensation, the PD control with gravity pre-compensation and the proportional control with velocity feedback. These control strategies use partiality the dynamic model of the robotic system. On the other hand [7–9], show that for tracking tasks the Computed Torque Control (CTC) strategy is employed, which uses all the dynamic model of the robotic manipulator. However, control techniques presented above do not consider for the controller design the presence of external disturbances and parametric uncertainness, which affect the robotic manipulator performance in regulation and tracking tasks.

Robust control techniques as the Quantitative Feedback Theory (QFT) or H infinity are employed for systems affected by the presence of external disturbances and parametric uncertainness [10–17]. These techniques consider the perturbations in the design of the control systems; however, the design and implementation of these techniques are complex.

Besides, the use of fractional order operators allows developing fractional order control strategies, which improve the performance and stability of the system in the presence of external disturbances and parametric uncertainness. Usually, fractional order controllers are employed for process control [18–20]. In the case of manipulator robots, the development of fractional control strategies is poor [21], especially for parallel robots.

* Corresponding author.
E-mail addresses: luis.angel@upb.edu.co (L. Angel), jairo.viola@upb.edu.co, jairoviola92@hotmail.com (J. Viola).

https://doi.org/10.1016/j.isatra.2018.04.010
0019-0578/

This paper proposes the implementation of a tracking control for a parallel robot type delta with three degrees of freedom employing fractional order PID controllers in conjunction with the computed torque control strategy. This control strategy is contrasted with integer order PID controllers in conjunction with the computed torque control strategy against the presence of external disturbances that affect the robotic system operation.

Initially, the kinematic and dynamic models for the delta robot are obtained. Then, a SOLIDWORKS/MSC-ADAMS/MATLAB cosimulation model is built according to the physical dimension of the delta robot prototype placed in the robotics laboratory of the Universidad Pontificia Bolivariana, Seccional Bucaramanga (UPB Bucaramanga).

This cosimulation model is employed for the parametric identification of the dynamic model of the robot and the design of the control strategies. After that, delta robot is decoupled using the computed torque control technique, resulting in a double integrator system for each control loop. From the linearized model for each joint, a fractional order PID (FOPID) and an integer order PID (IOPID) are designed for the tracking control of the delta robot. IOPID and FOPID controllers are validated using the SOLIDWORKS/MSC-ADAMS/MATLAB cosimulation model. Finally, a robustness analysis for the IOPID and FOPID controllers is performed in the presence of external disturbances in the joint applied torque, random noise in the feedback loop and the presence of critical payload on the robotic system.

The main contributions of this paper are the application of fractional order controllers in conjunction with the computed torque control technique for the tracking control of parallel robots and a quantitative robustness analysis in the presence of external disturbances in the robotic system using performance indexes.

This paper is structured as follows. First, kinematic, and dynamic models for the delta robot are obtained. Second, the SOLIDWORKS/MSC-ADAMS/MATLAB cosimulation model is presented. Third, parametric identification of the delta robot is performed using the recursive least squares algorithm. Then, the decoupled robotic system is obtained using the Computed Torque Control strategy. After that, IOPID and FOPID controllers are designed for the tracking control of the delta robot. Next, the validation of IOPID and FOPID controllers is performed employing the SOLIDWORKS/MSC-ADAMS/MATLAB cosimulation model. Then, a quantitative robustness analysis for the IOPID and FOPID controllers in the presence of external disturbances is fulfilled. Finally, result analysis and conclusions are presented.

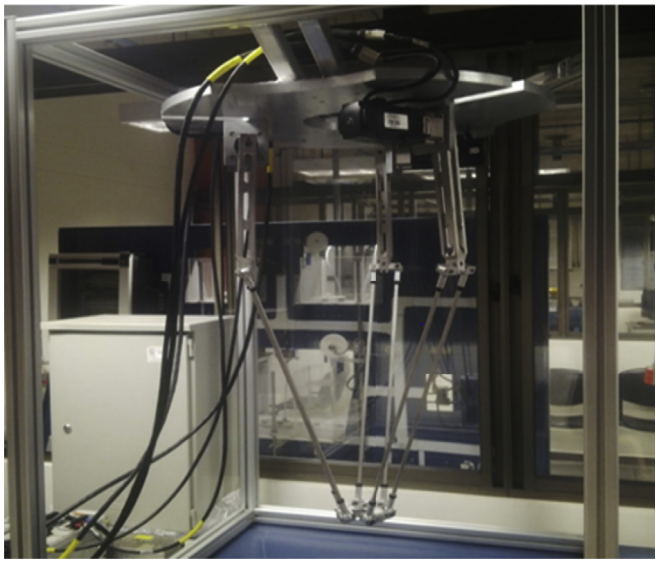

Fig. 1. Delta robot prototype at UPB Bucaramanga.

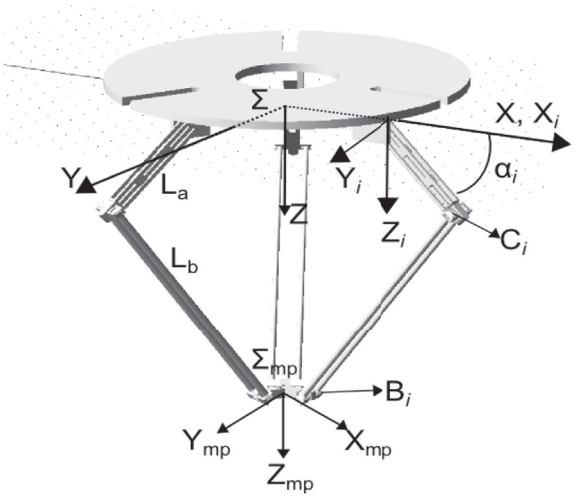

Fig. 2. Geometric parameters for delta robot.

2. Delta robot modelling and identification

A parallel robotic manipulator type delta is a three degree of freedom robot consist of three closed kinematic chains, a fixed platform, and a mobile platform. Due to the actuators of the robot are placed in the fixed platform, delta robot can reach high speeds and accelerations. This robotic structure was proposed by Clavel [22] and is employed in the industry for pick and place tasks, reaching top speeds about 5m/s for payloads up to 8 kg [23]. Fig. 1 shows the delta robot prototype built at Universidad Pontificia Bolivariana, Seccional Bucaramanga. This prototype is an open robotic system that will be employed for the validation of advanced control strategies on robotic parallel manipulators.

2.1. Kinematic model

Kinematic model defines the mathematical relation between the joints positions and the Cartesian position of the mobile platform. Delta robot kinematic model is based in Fig. 2. As can be observed, the robot has a global coordinate reference system $\sum(x, y, z)$ located on the fixed platform, in which z-axis is in the same direction of gravity g.

A second coordinate system $\sum_{mp}(x_{mp}, y_{mp}, z_{mp})$ is established for the position of the mobile platform with respect to $\sum$. Due to the symmetry between the kinematical chains, Fig. 3 describes the i kinematic chain for $i = 1, 2, 3$.

As shown in Fig. 2 and Fig. 3 each kinematic chain has an arm with length L_a and a forearm with length L_b. Over each arm, a coordinate system $\sum_i$ is located at a distance R of $\sum$ and rotated an angle $\phi_i(0°, 120°, 240°)$. α_i is the joint position angle of each arm, R is the fixed platform radius, and r is the mobile platform radius.

2.2. Inverse kinematic model

The inverse kinematic model seeks to find the joints position angles α_i off delta robot for a specific mobile platform position $\sum_{mp}$, which is based on Fig. 2 and Fig. 3 [24]. A first restriction (1)

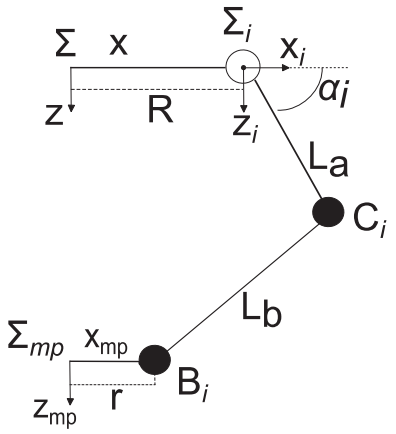

Fig. 3. Delta Robot kinematic chains.

establishes the geometric position of c_i as a sphere centered in B_i with radius L_b.

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = L_b^2 \tag{1}$$

A second restriction (2) is established for the c_i circular trajectory with the center in $\sum_i$ and a L_a radius.

$$(x - R)^2 + z^2 = L_a^2 \tag{2}$$

c_i is located at the intersection between (1) and (2) in the plane (3).

$$y = 0 \tag{3}$$

From (1)–(3), a quadratic equation in x is produced to find the solution of (4).

$$\alpha_i = \sin^{-1}\left(\frac{z}{l_a}\right) \tag{4}$$

From (4), two possible solutions are defined for each joint. However, the manipulator can only reach one solution physically. It comes to the existence of eight possible solutions for the inverse kinematic model for each spatial position. So that, the choice of each joint position is subject to geometric constraints defined in (5) to avoid singularities.

$$\begin{array}{l} \text{If } x - R \geq 0; \ \alpha_i \text{ is calculated from (4)} \\ \text{Else if } x - R < 0; \quad \alpha_i = \pi - \alpha_i \end{array} \tag{5}$$

2.3. Direct kinematic model

For this kinematic model, joint position α_i is established and is to determinate the spatial position of the mobile platform $\sum_{pm}$. Center of the mobile platform is the intersection of three spheres (1) centered in c_i with L_b radius. Two intersection points exist, but only one of them can be physically reached as shown in Ref. [24].

2.4. Analytic dynamic model

Delta robot dynamic analytic model is raised using the Euler-Lagrange formulation, which is based on the energy considerations of the robotic system. Motion equations are written in terms of redundant coordinates. In addition, a set of restrictions is required established by the delta robot kinematics. The Euler-Lagrange formulation is presented in (6).

$$\frac{d}{dt}\left(\frac{dL}{d\dot{q}_j}\right) - \frac{dL}{dq_j} = Q_j + \sum_{i=1}^{k} \lambda_i \frac{df_i}{dq_j} \quad \text{for } j = 1 \text{ to } n \tag{6}$$

where i is the restriction index, k is the number of restrictions, j is the generalized coordinate index, n is the number of generalized coordinates, L corresponds to the Lagrange function, q_j is the generalized coordinate j. λ_i is the Lagrange multipliers, f_i are the kinematic restriction equations and Q_j is the generalized external force defined as $Q_j = \widehat{Q_j} + \tau_i$, where $\widehat{Q_j}$ is the generalized external force applied to the mobile platform and τ_i the applied torque to joint i. To solve (6) and find the motion equations for the robotic system, (6) is divided into two equations set. The first system (7) is defined in terms of Lagrange multipliers as only unknown.

$$\sum_{i=1}^{k} \lambda_i \frac{df_i}{dq_j} = \frac{d}{dt}\left(\frac{dL}{d\dot{q}_j}\right) - \frac{dL}{dq_j} - \widehat{Q}_j \tag{7}$$

From Lagrange multipliers λ_i found in (7), the second system (8) allows estimating the applied torque τ_i by the actuators of the robotic system.

$$Q_j = \frac{d}{dt}\left(\frac{dL}{d\dot{q}_i}\right) - \frac{dL}{dq_j} = \sum_{i=1}^{k} \lambda_i \frac{df_i}{dq_j} \quad \text{for } j = k+1 \text{ to } n \tag{8}$$

For delta robot, Euler-Lagrange formulation uses a six generalized coordinates vector given by (9). Three coordinates belong to the mobile platform spatial position with respect to $\sum$. The last three coordinates are the joint position α_i of the delta robot.

$$q_j = [x, y, z, \alpha_1, \alpha_2, \alpha_3] \text{ with } j = 1 \text{ to } 6 \tag{9}$$

From (9), it is established that Euler-Lagrange formulation (6) requires six unknowns, which correspond to the Lagrange multipliers for $i = 1, 2, 3$ and the joints applied torque Q_j for $j = 4, 5, 6$. Besides, Lagrange formulation requires three kinematic restrictions, which are defined in (1) as x_i, y_i, and z_i and establish as:

$$\begin{array}{l} x_i = (\Delta r + L_a \cos(\alpha_i))\cos(\phi_i) \\ y_i = (\Delta r + L_a \cos(\alpha_i))\sin(\phi_i) \\ z_i = L_a \sin(\alpha_i) \\ \text{for } i = 1, 2, 3 \end{array} \tag{10}$$

where $\Delta r = R - r$. Lagrange function L is defined in (11) as the difference between kinetic energy T and potential energy V.

$$L = T - V \tag{11}$$

For the delta robot, kinetic energy defines as:

$$T = T_c + \sum_{i=1}^{3} T_{ai} + T_{bi} \tag{12}$$

where T_c is the kinetic energy of mobile platform, T_{ai} is the kinetic energy of arm i and T_{bi} is the kinetic energy of forearm i for $i = 1, 2, 3$. If gravity acceleration is in the same direction of the positive z-axis, the potential energy for the delta robot is:

$$V = V_c + \sum_{i=1}^{3} V_{ai} + V_{bi} \tag{13}$$

where V_c is the potential energy of mobile platform, V_{ai} is the potential energy of arm i and V_{bi} is the potential energy of forearm i for $i = 1, 2, 3$. For the equation system (7), unknowns correspond to the Lagrange multipliers for $i = 1, 2, 3$ which can be found from (14)

$$
\begin{aligned}
2\sum_{i=1}^{3}\lambda_i(x + R - r - L_a\cos(\alpha_i))\cos(\phi_i) &= (m_{mp} + 3m_b)\ddot{x} - f_{px}\\
2\sum_{i=1}^{3}\lambda_i(x + R - r - L_a\cos(\alpha_i))\sin(\phi_i) &= (m_{mp} + 3m_b)\ddot{y} - f_{py}\\
2\sum_{i=1}^{3}\lambda_i(z - L_a\cos(\alpha_i)) &= (m_{mp} + 3m_b)\ddot{z} + (m_{mp} + 3m_b)g - f_{pz}
\end{aligned}
\tag{14}
$$

where m_{mp} belongs to the mobile platform mass, m_b is the forearm mass, $[f_{px}, f_{py}, f_{pz}]$ correspond to the external force applied to the mobile platform, $[\ddot{x}, \ddot{y}, \ddot{z}]$ is the mobile platform spatial acceleration and g is the gravity acceleration. Once the Lagrange multipliers in (14) are found, the applied torque τ_i for each joint is estimated for $i = 1, 2, 3$ as shown in (15), where m_a corresponds to the arm mass, and I_m is the delta robot motors inertia.

$$
\begin{aligned}
\tau_1 &= \left(I_m + \frac{1}{3}m_a l_a^2 + m_b l_a^2\right)\ddot{\alpha_1} + \left(\frac{1}{2}m_a + m_b\right)L_a g\cos(\alpha_1)\\
&\quad -2\lambda_1[(x\cos(\phi_1) + y\cos(\phi_1) + R - r)\sin(\alpha_1) - z\cos(\alpha_1)]\\
\tau_2 &= \left(I_m + \frac{1}{3}m_a l_a^2 + m_b l_a^2\right)\ddot{\alpha_2} + \left(\frac{1}{2}m_a + m_b\right)L_a g\cos(\alpha_2)\\
&\quad -2\lambda_2[(x\cos(\phi_2) + y\cos(\phi_2) + R - r)\sin(\alpha_2) - z\cos(\alpha_2)]\\
\tau_3 &= \left(I_m + \frac{1}{3}m_a l_a^2 + m_b l_b^2\right)\ddot{\alpha_3} + \left(\frac{1}{2}m_a + m_b\right)L_a g\cos(a_3)\\
&\quad -2\lambda_3[(x\cos(\phi_3) + y\cos(\phi_3) + R - r)\sin(\alpha_3) - z\cos(\alpha_3)]
\end{aligned}
\tag{15}
$$

2.5. SOLIDWORKS/MSC-ADAMS/MATLAB cosimulation model

A SOLIDWORKS/MSC-ADAMS/MATLAB co-simulation model is built to analyze the dynamic behavior of the delta robot and identify the effects of friction and inertia due to the design of the mechanical structure of the robotic system. Initially, the delta robot components are designed and assembled employing SOLIDWORKS, and then the model is exported to MSC-ADAMS. Fig. 4 shows the delta robot model in MSC-ADAMS. As can be seen, the acceleration of gravity is in the same direction of the positive z-axis. Subsequently, the cosimulation model is exported to MATLAB for the robotic system identification.

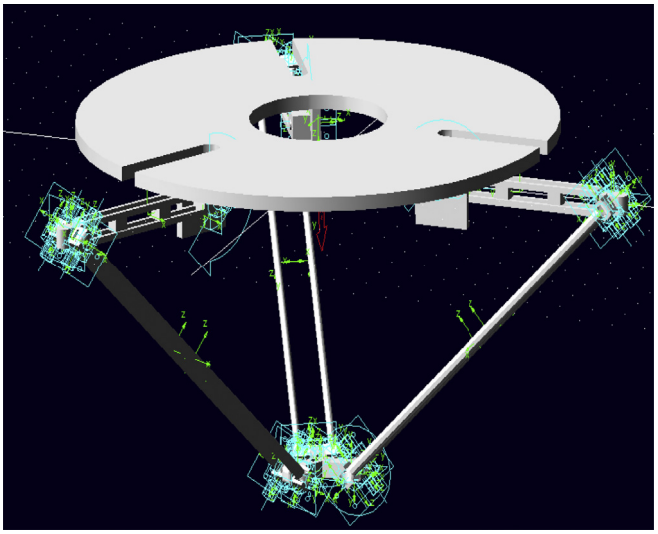

Fig. 4. Delta robot cosimulation model built in SOLIDWORKS/MSC-ADAMS/MATLAB.

2.6. Parametric identification of delta robot analytic dynamic model

2.6.1. Recursive least squares algorithm (RLS)

According to [25], RLS algorithm allows estimating inertia and mass parameters that accompanying the analytical dynamic model of the delta robot, which can be calculated using linear regression matrices. For this, the linear regressor defined in (16) is employed, where $y(k)$ is the vector of output measures, $\psi(k)$ is the observation matrix of known functions and parameters and θ is the vector of parameters to be estimated.

$$
y(k) = \psi(k)^T\,\theta \tag{16}
$$

From (16), the RLS equations are defined (17)–(19). Employing the samples taken from the system, in this case the joint position, velocity, acceleration and applied torque of the robotic system, where $\widehat{\theta}(k)$ is estimated parameters vector, $\psi(k)$ the matrix of observations and parameters, $e(k)$ corresponds to error of each iteration and $P(k)$ is the covariance matrix.

$$
\widehat{\theta}(k) = \widehat{\theta}(k-1) + \frac{P(k-1)\psi(k)e(k)}{1 + \psi(k-1)^T P(k-1)\psi(k)} \tag{17}
$$

$$
P(k) = P(k-1) + \frac{P(k-1)\psi(k)\psi(k)^T e(k)P(k-1)}{1 + \psi(k-1)^T P(k-1)\psi(k)} \tag{18}
$$

$$
e(k) = y(k) - \psi(k)^T\widehat{\theta}(k-1) \tag{19}
$$

2.6.2. Trajectory planner

A trajectory planner establishes the path for each joint to carry the mobile platform from an initial to an ending point through a Cartesian trajectory with an established velocity and acceleration. These trajectories are selected considering the physical restrictions of the motors and certain trajectory quality factors as smoothness or precision [24].

For the delta robot, a sectioned trajectory planner is implemented using a 6-1-6 polynomial. This planner consists of three sections, which are the acceleration section, the constant speed section, and the deceleration section. Certain border conditions join each planner section, generating a continuous and smooth motion profile for the position, velocity, acceleration, and jerk outlines. Border conditions set eight restrictions, two for position, two for velocity, two for acceleration, and two for jerk. Acceleration section uses a sixth order polynomial, executed during the acceleration time of the motor. The constant velocity section employs a first order polynomial, and the deceleration section uses a sixth order polynomial, equal to the deceleration time of the motor.

In this paper, a 6-1-6 planner has been employed in the Cartesian space, which receives an initial cartesian position and a final cartesian position and maps the path between the two points through a series of intermediate spatial points describing a space trajectory as a straight line. For each intermediate point, the corresponding joint position is obtained for each motor of the delta robot using the inverse kinematic model given by (4).

2.6.3. Identification trajectory selection

Trajectory selection is one of the most critical tasks in the identification process because the trajectory must ensure that the robot covers almost of its workspace to observe the dynamic behavior of the robotic system. Many trajectories were implemented employing the trajectory planner presented in section 2.6.2. As an example, Table 1 presents the trajectory T1 used in the identification process and the trajectory T2 employed to validate the estimated analytic dynamic model. Fig. 5 shows the trajectories T1 and T2 in the Cartesian space.

Table 1
Identification and validation trajectories for delta robot.

Identification T1 (mm)	Validation T2 (mm)
Starting point: (0, 0, 360) Waypoints: (0,0,500); (200,0,500) (0,200,400); (0,-200,400) (200,0,500), (0,0,500); Final point (0.0,360)	Starting point: (0, 0, 360) Waypoints:(0,0,500); (0,200,500); (0,0,500); (200,0,500); (0,0,500); Final point (0,0,360)

2.6.4. Analytic dynamic model parametrization

To apply the parametric identification process, the model presented in (15) is expressed in its compact form as shown in (20).

$$\begin{aligned}\tau_1 &= \theta_1\ddot{\alpha_1} - \theta_2 x\sin(\alpha_1) - \theta_3 y\sin(\alpha_1) + \theta_4 z\cos(\alpha_1) \\ &\quad + \theta_5\cos(\alpha_1) + \theta_6\sin(\alpha_1) \\ \tau_2 &= \theta_1\ddot{\alpha_2} - \theta_7 x\sin(\alpha_2) - \theta_8 y\sin(\alpha_2) + \theta_9 z\cos(\alpha_2) \\ &\quad + \theta_5\cos(\alpha_2) + \theta_{10}\sin(\alpha_2) \\ \tau_3 &= \theta_1\ddot{\alpha_3} - \theta_{11} x\sin(\alpha_3) - \theta_{12} y\sin(\alpha_3) + \theta_{13} z\cos(\alpha_3) \\ &\quad + \theta_5\cos(\alpha_3) + \theta_{14}\sin(\alpha_3)\end{aligned} \tag{20}$$

where:

$$\begin{aligned}\theta_1 &= I_m + \frac{1}{3}m_a L_a^2 + m_b L_a^2 \\ \theta_2 &= 2\lambda_1\cos(\phi_1) \\ \theta_3 &= 2\lambda_1 sin(\phi_1) \\ \theta_4 &= 2\lambda_1 \\ \theta_5 &= \left(\frac{1}{2}m_a + m_b\right)L_a g \\ \theta_6 &= 2\lambda_1(r - R) \\ \theta_7 &= 2\lambda_2\cos(\phi_2) \\ \theta_8 &= 2\lambda_2\sin(\phi_2) \\ \theta_9 &= 2\lambda_2 \\ \theta_{10} &= 2\lambda_2(r - R) \\ \theta_{11} &= 2\lambda_3\cos(\phi_3) \\ \theta_{12} &= 2\lambda_3\sin(\phi_3) \\ \theta_{13} &= 2\lambda_3 \\ \theta_{14} &= 2\lambda_3(r - R)\end{aligned}$$

The regression scheme (20) using the regression model presented in (16) is given by (21). This regression model uses as input the data of the spatial and joint accelerations, the spatial position of the mobile platform along with the applied torques at each joint. The information is taken from the MSC-ADAMS/MATLAB cosimulation model during the execution of identification trajectories.

where:

$$\begin{aligned}\psi_{11} &= \ddot{\alpha_1} \\ \psi_{12} &= -x\sin(\alpha_1) \\ \psi_{13} &= -y\sin(\alpha_1) \\ \psi_{14} &= z\cos(\alpha_1) \\ \psi_{15} &= \cos(\alpha_1) \\ \psi_{16} &= \sin(\alpha_1) \\ \psi_{17} &= \psi_{18} = 0 \\ \psi_{19} &= \psi_{110} = 0 \\ \psi_{111} &= \psi_{112} = 0 \\ \psi_{113} &= \psi_{114} = 0 \\ \psi_{21} &= \ddot{\alpha_2} \\ \psi_{25} &= \cos(\alpha_2) \\ \psi_{27} &= -x\sin(\alpha_2) \\ \psi_{28} &= -y\cos(\alpha_2) \\ \psi_{29} &= z\cos(\alpha_2) \\ \psi_{210} &= \sin(\alpha_2) \\ \psi_{22} &= \psi_{23} = 0 \\ \psi_{24} &= \psi_6 = 0 \\ \psi_{211} &= \psi_{212} = 0 \\ \psi_{213} &= \psi_{214} = 0 \\ \psi_{31} &= \ddot{\alpha_3} \\ \psi_{35} &= \cos(\alpha_3) \\ \psi_{311} &= -x\sin(\alpha_3) \\ \psi_{312} &= -y\cos(\alpha_3) \\ \psi_{313} &= z\cos(\alpha_3) \\ \psi_{314} &= \sin(\alpha_3) \\ \psi_{32} &= \psi_{33} = 0 \\ \psi_{34} &= \psi_{36} = 0 \\ \psi_{37} &= \psi_{38} = 0 \\ \psi_{39} &= \psi_{310} = 0\end{aligned}$$

2.6.5. Delta robot parametric identification using the recursive least squares algorithm

From the cosimulation model presented in Fig. 4, the identification trajectory T1 is implemented. After that, the data of the mobile platform spatial position $[x, y, z]$, joint position $[\alpha_1, \alpha_2, \alpha_3]$ is taken. As shown in (18), the analytic dynamic model requires the joint accelerations $[\ddot{\alpha_1}, \ddot{\alpha_2}, \ddot{\alpha_3}]$, which is obtained by applying numerical differentiation to the joint position obtained from the MSC-ADAMS/MATLAB cosimulation model. From the data taken from the cosimulation model, the RLS algorithm defined by (17)–(19) is applied, obtaining the analytical dynamic model parameters for the delta robot, which are presented in Table 2.

To validate the identified parameters of the analytic dynamic model of the delta robot, the analytical dynamic model is built in Simulink using as input the data mentioned above, returning the estimated applied torque τ_i for each joint. This torque is compared with the resulting torque obtained from the MSC-ADAMS/MATLAB cosimulation model. To measure the fit between the applied torque of the estimated analytical dynamic model and the MSC-ADAMS/MATLAB cosimulation model, the NRMSE value is employed as

$$\begin{bmatrix}\tau_1 \\ \tau_2 \\ \tau_3\end{bmatrix} = \begin{bmatrix}\psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} & \psi_{15} & \psi_{16} & \psi_{17} & \psi_{18} & \psi_{19} & \psi_{110} & \psi_{111} & \psi_{112} & \psi_{113} & \psi_{114} \\ \psi_{21} & \psi_{22} & \psi_{23} & \psi_{24} & \psi_{25} & \psi_{26} & \psi_{27} & \psi_{28} & \psi_{29} & \psi_{210} & \psi_{211} & \psi_{212} & \psi_{213} & \psi_{214} \\ \psi_{31} & \psi_{32} & \psi_{33} & \psi_{34} & \psi_{35} & \psi_{36} & \psi_{37} & \psi_{38} & \psi_{39} & \psi_{310} & \psi_{311} & \psi_{312} & \psi_{313} & \psi_{314}\end{bmatrix}\begin{bmatrix}\theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \\ \theta_{10} \\ \theta_{11} \\ \theta_{12} \\ \theta_{13} \\ \theta_{14}\end{bmatrix} \tag{21}$$

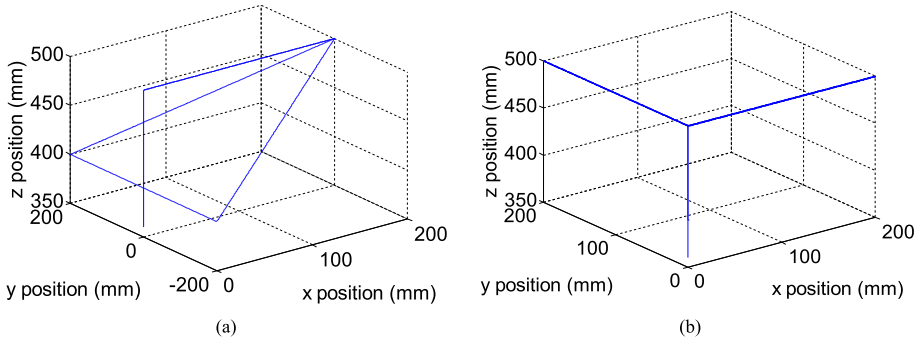

Fig. 5. Identification trajectories (a) T1 and (b) T2.

Table 2
Estimated parameters for the analytic dynamic model of the delta robot.

Parameter	Value	Parameter	Value
θ_1	0.070	θ_8	6×10^{-3}
θ_2	0.023	θ_9	0.006
θ_3	1×10^{-5}	θ_{10}	1
θ_4	0.006	θ_{11}	8×10^{-3}
θ_5	0.253	θ_{12}	6×10^{-3}
θ_6	1.12	θ_{13}	0.006
θ_7	8×10^{-3}	θ_{14}	1

Table 3
NRMSE value for the applied toque using trajectories T1 and T2.

Joint	NRMSE T1	NRMSE T2
α_1	0.879	0.8
α_2	0.884	0.881
α_3	0.883	0.8656

performance index, which is given by (22), where x is, the calculated torque using the estimated analytic dynamic model, x_{ref} is the reference torque, which is taken from MSC-ADAMS/MATLAB cosimulation model and mean(x_{ref}) is the reference torque mean. For (22), if the NRMSE value is closer to one, the estimated model has a good fit. However, if the value is closer or less than zero, the estimated model has a bad fit.

$$\mathrm{NRMSE} = 1 - \left| x - \frac{x_{ref}}{x - \mathrm{mean}\left(x_{ref}\right)} \right|^2 \tag{22}$$

Additionally, delta robot is subject to the validation trajectory T2, calculating its NRMSE value to compare the applied torque calculated by the estimated analytical dynamic model with the applied torque obtained from the cosimulation model for each joint. Table 3 shows the NRMSE value for the applied torque for each joint using the trajectories T1 and T2.

As seen in Table 3, the NRMSE value of the applied torque in the joints of the delta robot is higher than 0.8 using the T1 and T2 trajectories. This value indicates that the estimated parameters for the analytical dynamic model analytical fit the dynamic behavior of the delta robot in the presence of different trajectories with a fit of 80%.

3. Delta robot tracking control strategies

3.1. Computed torque control

Computed Torque Control (CTC) is a control strategy that employs feedback linearization to obtain a linearized and decoupled model of the robotic system. It allows controlling the robotic system using linear control techniques. From Ref. [26], the dynamic model of a robotic system is given by (23).

$$\tau = M(q)\ddot{q} + C(q,\dot{q}) + g(q) \tag{23}$$

where τ is the torque applied at the joints of the robotic system, q, $\dot{q}$, $\ddot{q}$ are the position, velocity, and acceleration of each joint of the manipulator respectively. M(q) is the inertia matrix, $C(q,\dot{q})$ is the Coriolis matrix, and g(q) is the gravity vector. For the delta robot, $M(q)$, $C(q,\dot{q})$ and g(q) are given by (24), where θ_i are presented in Table 2.

$$M(q) = \begin{bmatrix} \theta_1 & 0 & 0 \\ 0 & \theta_1 & 0 \\ 0 & 0 & \theta_1 \end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix} C_{11} & C_{22} & C_{13} & C_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{21} & C_{22} & C_{23} & C_{24} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{31} & C_{32} & C_{33} & C_{34} \end{bmatrix} \tag{24}$$

where:

$C_{11} = -x\theta_2 \sin(\alpha_1)$
$C_{12} = -y\theta_4 \sin(\alpha_1)$
$C_{13} = \theta_4 z\cos(\alpha_1)$
$C_{14} = \theta_6 \sin(\alpha_1)$
$C_{21} = -\theta_7 x\sin(\alpha_2)$
$C_{22} = -\theta_8\, y\sin(\alpha_2)$
$C_{23} = \theta_9 z\cos(\alpha_2)$
$C_{24} = \theta_{10} \sin(\alpha_1)$
$C_{31} = -x\sin(\alpha_1)\theta_{11}$
$C_{32} = -\theta_3 y \sin(\alpha_1)\theta_{12}$
$C_{33} = \theta_{13} z\cos(\alpha_1)$
$C_{34} = \theta_{14}\theta_6 \sin(\alpha_1)$

$$g(q) = \begin{bmatrix} \theta_5 \cos(\alpha_1) \\ \theta_5 \cos(\alpha_2) \\ \theta_5 \cos(\alpha_3) \end{bmatrix}$$

Proposing the linearization law (25), where a is the new system input,

$$\tau = M(q)a + C(q,\dot{q}) + g(q) \tag{25}$$

and assuming an exact model of the dynamic model of the robotic system, (26) is obtained by combining (23) and (25).

$$M(q)a = M(q)\ddot{q} \tag{26}$$

If the inertia matrix $M(q)$ is invertible, (26) results in a decoupled double integrating system presented in (27).

$$\ddot{q} = a \tag{27}$$

3.2. IOPID controller design with the computed torque control strategy

For the IOPID controller design, the control law (27) is proposed for the linearized system (28)

$$a = \ddot{q}_d - k_p\tilde{q} - k_i\int \tilde{q}\,dt - k_d\dot{\tilde{q}} \tag{28}$$

where $\tilde{q}$ is the joint error, $\dot{\tilde{q}}$ is the joint velocity error, and $\ddot{q}_d$ is the desired joint acceleration. Replacing (28) in (27) results in the closed loop equation for the robotic system (29).

$$\ddot{\tilde{q}} + k_p\tilde{q} + k_i\int \tilde{q}\,dt + k_d\dot{\tilde{q}} = 0 \tag{29}$$

Passing (29) to Laplace domain, the closed loop equation for the robotic system is given by (30).

$$s^3 + k_d s^2 + k_p s + k_i = 0 \tag{30}$$

To find the IOPID controllers constants in (30), this equation is compared with the desired characteristic equation (31), which is based on the damping ratio ζ and the natural frequency of the system w_n and represent the desired behavior in closed loop of the robotic system.

$$\left(s^2 + 2\zeta w_n s + w_n^2\right) = 0 \tag{31}$$

Due to (30) is a third order system, (31) must have the same order. Thus, a non-dominant pole β is added to (31) to match (30) and (31) orders, obtaining (32).

$$(s+\beta)\left(s^2 + 2\zeta w_n s + w_n^2\right) = 0 \tag{32}$$

Equating (30) and (32), an expression is found for the IOPID controllers constants given by (33) for each joint.

$$\begin{aligned} k_p &= w_n^2 + 2\beta\zeta w_n \\ k_i &= \beta w_n^2 \\ k_d &= 2\zeta w_n + \beta \end{aligned} \tag{33}$$

Fig. 6 shows the structure for the IOPID controller with the computed torque control strategy for the delta robot.

3.3. FOPID controller design with the computed torque control strategy

According to [27–29] FOPID controller is defined by the integral-differential equation (34).

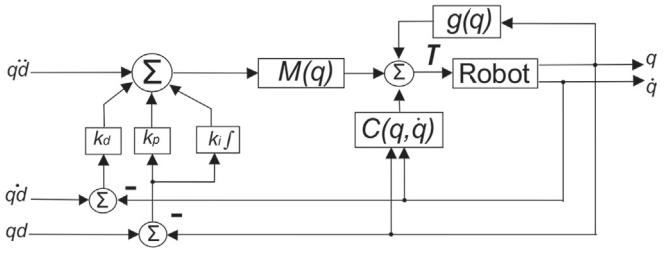

Fig. 6. Control structure for the IOPID controller with computed torque control strategy for the delta robot.

$$u(t) = k_p e(t) + k_i D^{-\lambda} e(t) + k_d D^{\mu} e(t) \tag{34}$$

where k_p is the proportional constant, k_i is the integral constant, k_d is the derivate constant, λ is the non-integer order for the integral term, and μ is the non-integer order for the derivate term. Assuming $e(t) = \tilde{q}$, the new control law for the linearized system (27) with the FOPID controller (34) is given by (35).

$$a = \ddot{q}_d - k_p\tilde{q} - k_i\frac{d^{-\lambda}}{dt}\tilde{q} - k_d\frac{d^{\mu}}{dt}\dot{\tilde{q}} \tag{35}$$

Replacing (35) on (27), the closed loop equation is given by (36).

$$\ddot{\tilde{q}} + k_p\tilde{q} + k_i\frac{d^{-\lambda}}{dt}\tilde{q} + k_d\frac{d^{\mu}}{dt}\dot{\tilde{q}} = 0 \tag{36}$$

Passing (36) to Laplace domain, the closed loop equation of the robotic system is defined by (37).

$$s^2 + k_p + \frac{k_i}{s^{\lambda}} + k_d s^{\mu} = 0 \tag{37}$$

Fig. 7 shows the structure for the FOPID controller with the computed torque control strategy for the delta robot.

From (37), the closed loop equation of the robotic system has fractional order terms (λ,μ) that increase the degrees of freedom the closed loop equation of the system to five. Indeed, this prevents the use of a methodology to find the constants of the FOPID controllers like that presented in Section 3.2. Thus, a solution based on optimization algorithms is proposed to find the FOPID controller constants for each joint, employing equations (38)-(42), which describes the desired closed loop behavior of the robotic system in the frequency domain to reach a robust performance and stability. In equations (38)-(42), C(jw) is the FOPID controller transfer function, P(jw) is the transfer function of the linearized robotic system, *pm* is the phase margin, w_c is the gain crossover frequency, M_s and M_t are the maximum values for the sensitivity and complementary sensitivity.

Phase margin (pm):

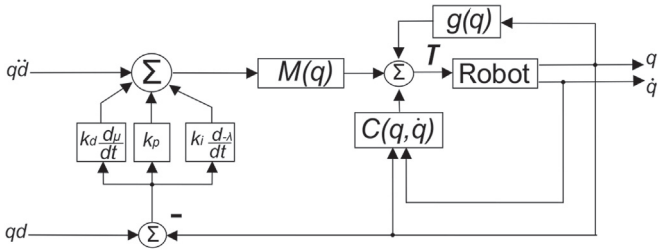

Fig. 7. Control structure for the FOPID controller with computed torque control strategy for the delta robot.

$$\arctan(C(jw_c)P(jw_c)) = -180 + pm \tag{38}$$

Gain crossover frequency (w_c):

$$|C(jw_c)P(jw_c)| = 0 \ \text{dB} \tag{39}$$

Robustness against plant gain variations:

$$\frac{d}{dw}(\arctan(C(jw_c)P(jw_c))) = 0 \tag{40}$$

High frequency noise rejection:

$$\left|\frac{C(jw_c)P(jw_c)}{1 + C(jw_c)P(jw_c)}\right| = M_t \text{dB} \tag{41}$$

External disturbances rejection:

$$\left|\frac{1}{1 + C(jw_c)P(jw_c)}\right| = M_s \ \text{dB} \tag{42}$$

The optimization algorithm to find the FOPID controller constants is presented in Fig. 8, which employs the Matlab FMINCON function that search the constrained minimum of a function of several variables. The Matlab FMINCON function solves problems of the form $min_x\ f(x)$ subject to $C(x) \leq 0, Ceq(x) = 0, x_m \leq x \leq x_M$, where $f(x)$ is the cost function; $C(x)$ and $Ceq(x)$ represent the nonlinear inequalities and equalities, respectively (nonlinear constraints); x is the minimum looked for; x_m and x_M define a set of lower and upper bounds on the design variables x.

In this case, (38) is taken as the cost function to minimize, and (39)–(42) are taken as constraints for the minimization, all of them subjected to the optimization parameters defined within the FMINCON function. The success of this design method depends mainly on the initial conditions considered for the parameters of the controller. As performance index, the integral of square error (ISE) is employed.

The optimization algorithm presented in Fig. 8 is executed as follows. Initially, desired performance specifications and the minimum ISE performance index are defined. Then, the FOPID controller constants are initialized. Next, restrictions (38)–(42) are defined, and the FOPID controller constants are calculated using FMINCON function. Then, step response and frequency response are calculated for the system based on the obtained FOPID controller constants, and it is verified if the performance specifications and the performance index were achieved. If any of them were not reached, FOPID controller constants are recalculated using the FMINCON function until both conditions be reached.

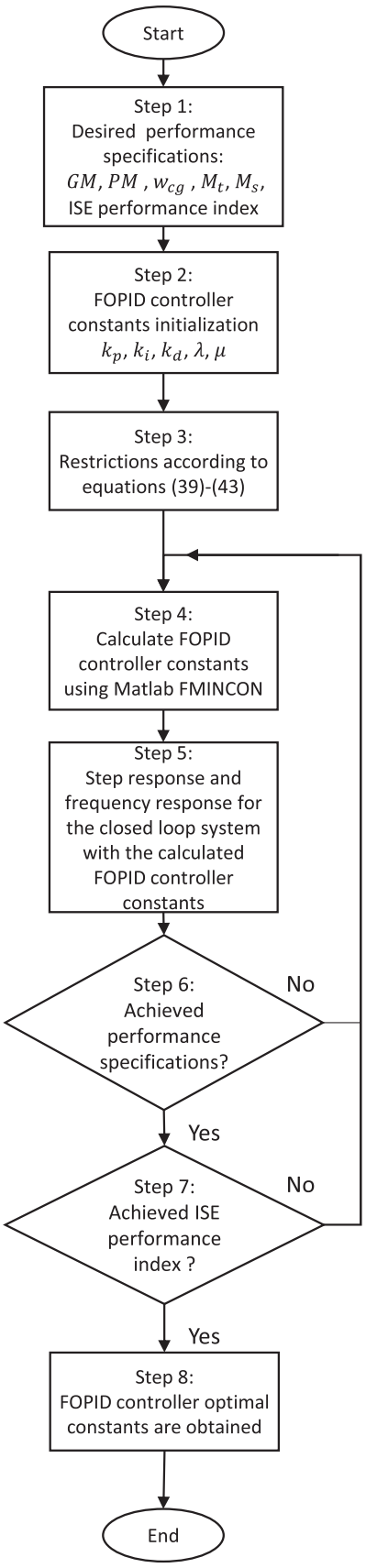

Fig. 8. The main procedure for the optimal FOPID controller design algorithm.

4. Obtained results

This section presents the results for the IOPID and FOPID with the computed torque control strategy employing the proposed methodologies presented in Section 3.2 and Section 3.3. Also, FOPID and IOPID controllers performance is evaluated for tracking tasks in the Cartesian space using as performance indexes the RMSE (Root Mean Square Error) value for the spatial and joint errors, the RMS (Root Mean Square) value for the control action and the individual and global tracking norms N proposed by Ref. [30].

4.1. IOPID and FOPID controllers tuning

Table 4 presents the constants for the IOPID and FOPID controllers, obtained applying the methodologies proposed in Section 3.2 and Section 3.3. In the time domain, the desired closed loop specifications are a damping ration $\zeta = 1$, a settling time of 0.05s, $w_n = 80$ rad/s and $\beta = 0.5$. In the frequency domain, these

Table 4
Obtained constants for the IOPID and FOPID controllers.

Controller	k_p	k_i	k_d	λ	μ
FOPID	2861	10,500	350	1.1	0.88
IOPID	6480	3200	160	1	1

specifications are equal to a phase margin pm = 100°, a gain crossover frequency $w_c = 38.86$ rad/s, $M_s = -20$ dB and $M_t = -20$dB.

4.2. IOPID and FOPID controllers performance

To assess the performance of the IOPID and FOPID controllers with the computed torque control strategy, the robotic system is subjected to tracking tasks. Fig. 9 shows the proposed test trajectories for the performance evaluation of the IOPID and FOPID controllers, which are implemented with the trajectory planner presented in Section 2.6.2. As can be seen, these trajectories allow analyzing the response of the controllers when it goes through different positions of the working space of the robot.

Fig. 10 shows the spatial tracking response of the mobile platform of the robot when the IOPID and FOPID controllers are employed for the trajectories T1, T2, and T3. The red square is an enlargement of the robotic system response at this point to show more in detail its dynamical behavior. As can be observed, IOPID and FOPID controllers exhibit a similar behavior during the execution of desired trajectories.

Fig. 11 shows the spatial error of the mobile platform for the trajectories T1, T2 and T3 using the IOPID and FOPID controllers. As can be observed, the error at x and y coordinates are similar; however, FOPID controller presents a less error in z coordinate at the beginning of each trajectory. This indicates that FOPID controller breaks the initial inertia of the robotic system faster than IOPID controller.

Fig. 12 shows the joint error for α_1, α_2 and α_3 using the IOPID and FOPID controllers. As can be observed, FOPID controller exhibit fewer variations at the beginning of each trajectory. It represents a smooth control over the delta robot joints, resulting in a better performance for tracking tasks.

Fig. 13 shows the applied torque in the delta robot joints α_1, α_2 and α_3 for the IOPID and FOPID controllers. In the case of the IOPID controller, there is a significant variation of the applied torque starting each trajectory. It shows that IOPID controller requires more applied torque to break the initial inertia of the robotic system.

4.3. IOPID and FOPID controllers quantitative performance analysis

To establish quantitatively the performance of IOPID and FOPID controllers with the computed torque control of the delta robot for tracking tasks certain performance indexes are employed. The RMSE value is selected for the spatial and joint errors; the RMS value is employed for the control action and the individual and global tracking norms N are used for the robot tracking performance. The RMSE value is defined in (43).

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(X_{ref}(i) - X(i)\right)^2} \tag{43}$$

where n is the number of samples, $X_{ref}(i)$ is the desired value of the signal and $X(i)$ the obtained value of the SOLIDWORKS/MSC-ADAMS/MATLAB cosimulation model at the instant i. The RMSE value is employed for the spatial and joint errors for the trajectories T1, T2, and T3. The RMS value of a signal is given by (44).

$$\mathrm{RMS} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(X(i)\right)^2} \tag{44}$$

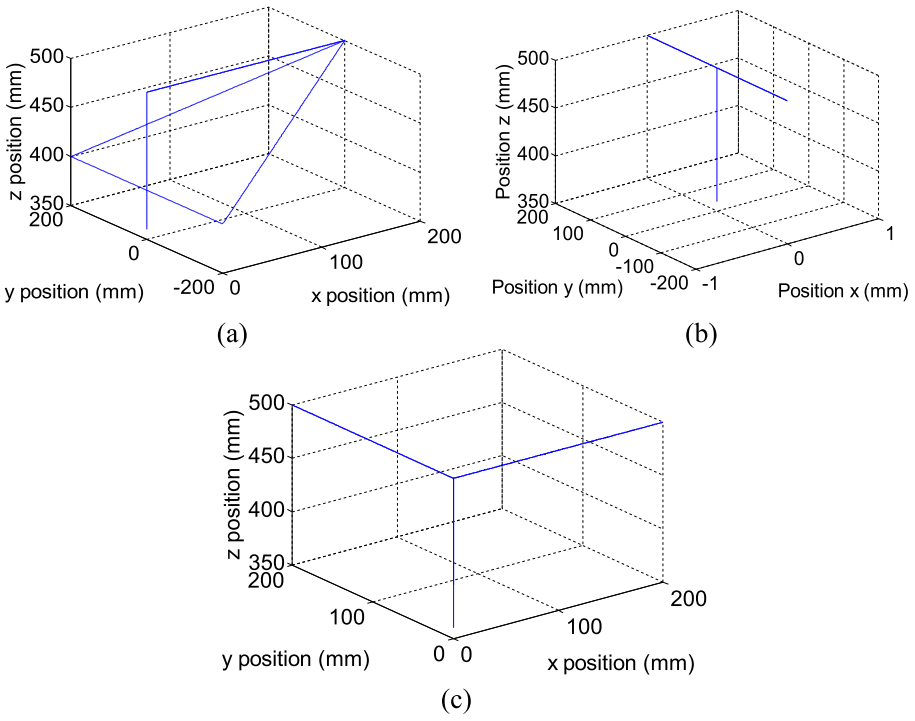

Fig. 9. Test trajectories for the delta robot.

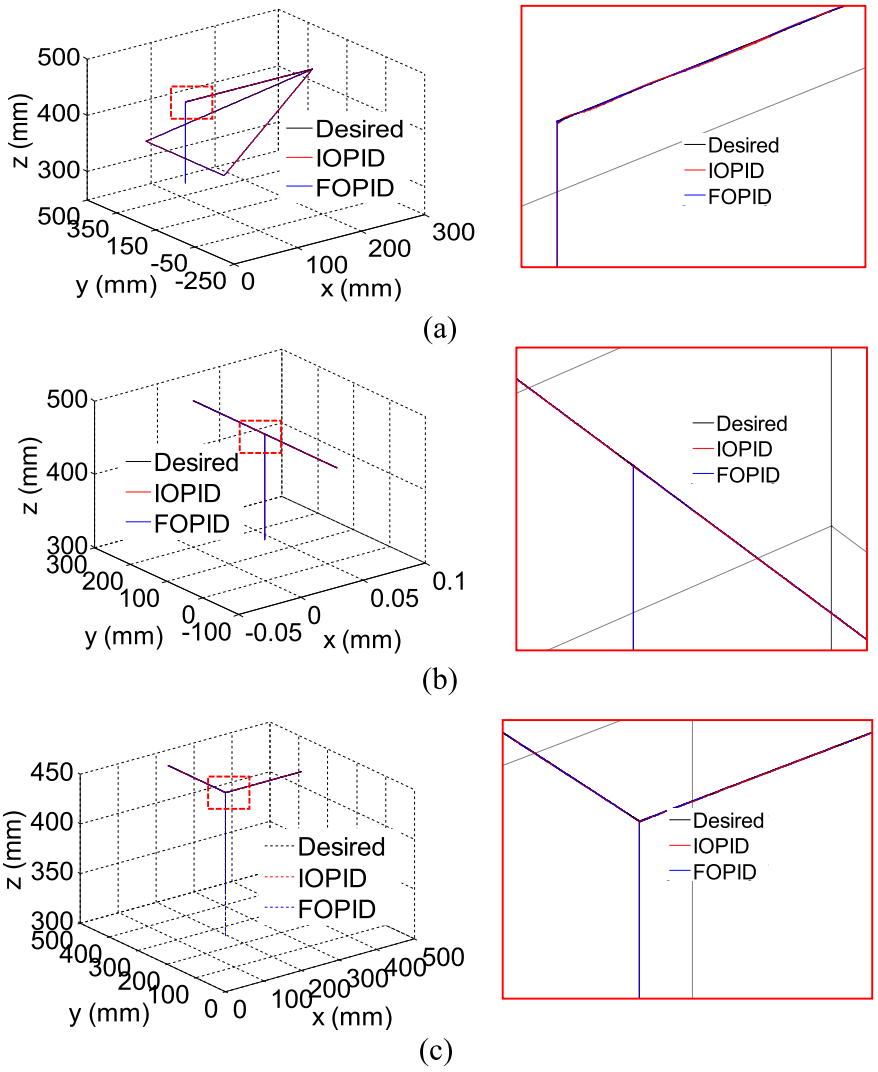

Fig. 10. Tracking response for (a) T1 (b) T2 (c) T3 using the IOPID and FOPID controllers.

where n is the number of samples and $X(i)$ the obtained value of the SOLIDWORKS/MSC-ADAMS/MATLAB cosimulation model at the instant i. The RMS value is employed to measure the mean value of the applied torque in the delta robot joints during the execution of the trajectories T1, T2, and T3. From Ref. [30], the individual tracking norm N employed to evaluate the tracking performance of a joint of the robotic system is expressed by (45).

$$N = \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(\tilde{q}(i)^2 + \dot{\tilde{q}}(i)^2\right)} \tag{45}$$

where n is the number of samples, $\tilde{q}(i)$ is the joint error and $\dot{\tilde{q}}(i)$ the joint velocity error of the delta robot at the instant i.

The global tracking norm N_{total} given by (46) measures the robotic system performance in tracking tasks, considering the performance of each joint of the robotic system from the joint and velocity errors for the k joints of the robotic system.

$$N_{total} = \sqrt{\frac{1}{n}\sum_{k=1}^{k}\sum_{i=0}^{n}\left(\widetilde{q_k}(i)^2 + \dot{\widetilde{q_k}}(i)^2\right)} \tag{46}$$

Table 5 presents the performance indexes defined in (43)–(46) of the IOPID and FOPID controllers with the computed torque control strategy for the trajectories T1, T2 and T3 executed by the delta robot.

As can be observed, for all the trajectories the RMSE value for the spatial error for the mobile platform is similar at x and y coordinates, but for the z coordinate the RMSE value is a 50% less employing FOPID controller than employing IOPID controller. For the joint error, notice that the RMSE value is almost 50% less using the FOPID controller than employing the IOPID controller on each joint for all the trajectories. As for applied torque in the delta robot joints, it can be observed that the FOPID controller requires less applied torque to track the desired trajectories than the IOPID controller. Finally, individual, and global tracking norms N have a small value employing the FOPID controller for the tracking of all trajectories, indicating a better performance for tracking tasks.

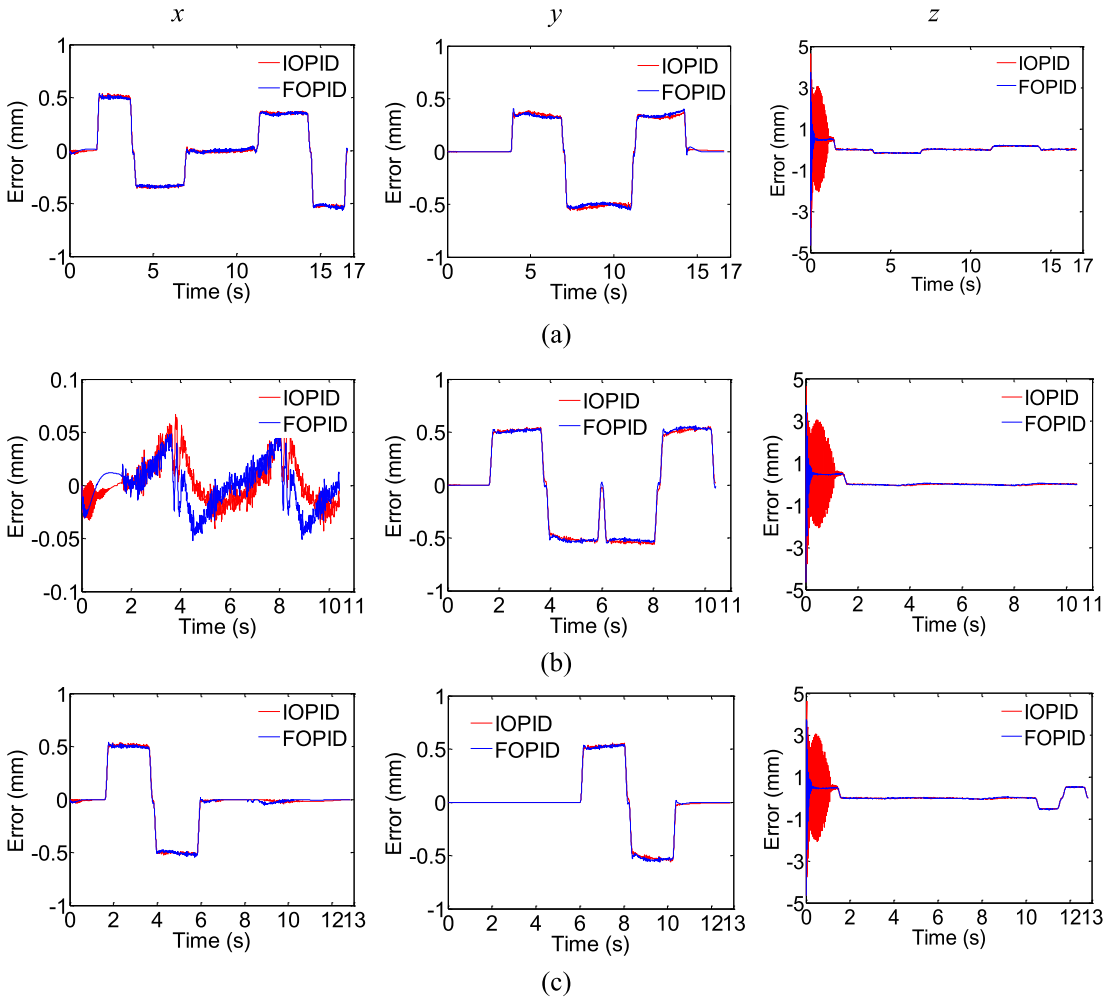

Fig. 11. Spatial error in X, Y, Z for the mobile platform at the tracking trajectories (a) T1 (b) T2 (c) T3 using the IOPID y FOPID controllers.

5. Robustness analysis

Results presented in Section 4 are for the performance of the IOPID and FOPID controllers with the computed torque control strategy for tracking tasks of the delta robot assuming the absence of external disturbances that affect the dynamical behavior of the system. This condition is called nominal operating condition.

Three tests are performed to evaluate the robustness of IOPID and FOPID controllers with the computed torque control strategy proposed in this paper for the tracking control of the delta robot. The first test, an external disturbance is settled in the applied torque of the joints of the robotic system. The second test, a critical payload is set in the mobile platform of the delta robot. The third test, introduce random noise in the feedback loop of the robotic system. Performance indexes (43)–(46) are calculated for each test.

5.1. External disturbance in the applied torque test

For this test, an external disturbance given by (47) is introduced in the applied torque of each joint of the robotic system for 1s which is applied in t = 7s for all trajectories. Fig. 14 shows the IOPID and FOPID controllers spatial tracking response of the mobile platform for the trajectories T1, T2, and T3 against the proposed disturbance. The red square is an enlargement of the robotic system response at this point to show more in detail the dynamical behavior of the robot. As can be observed, the FOPID controller has a better disturbance rejection and a faster response than the IOPID controller.

$$P(t) = 15 \times 10^{6} e^{-2t} \sin(500t) \tag{47}$$

Table 6 presents the performance indexes defined in (43)–(46) for the IOPID and FOPID controllers with the computed torque control strategy for tracking tasks of the delta robot in the external disturbance in applied torque test. As can be observed, for all trajectories, the IOPID and FOPID controllers exhibit a similar RMSE value for the spatial error in the mobile platform for the *x* and y coordinates, while for the z coordinate, the FOPID controller presents a less RMSE value than the IOPID controller. On the other hand, the RMSE value for the joint error decrease of about 40% when the FOPID controller is employed, which keeps the trajectories T1, T2, and T3. This shows that FOPID controller has a better performance against external disturbances in the applied torque of the system. Besides, the RMS values of applied torque for each joint are slightly higher for the FOPID controller than the IOPID controller. Indeed, this represents a higher energy consumption, but it also indicates that the FOPID controller performs an active rejection of the external disturbance evidenced in a less spatial error and a less joint error as mentioned above. Regarding the individual and global tracking norms, the FOPID controller has a

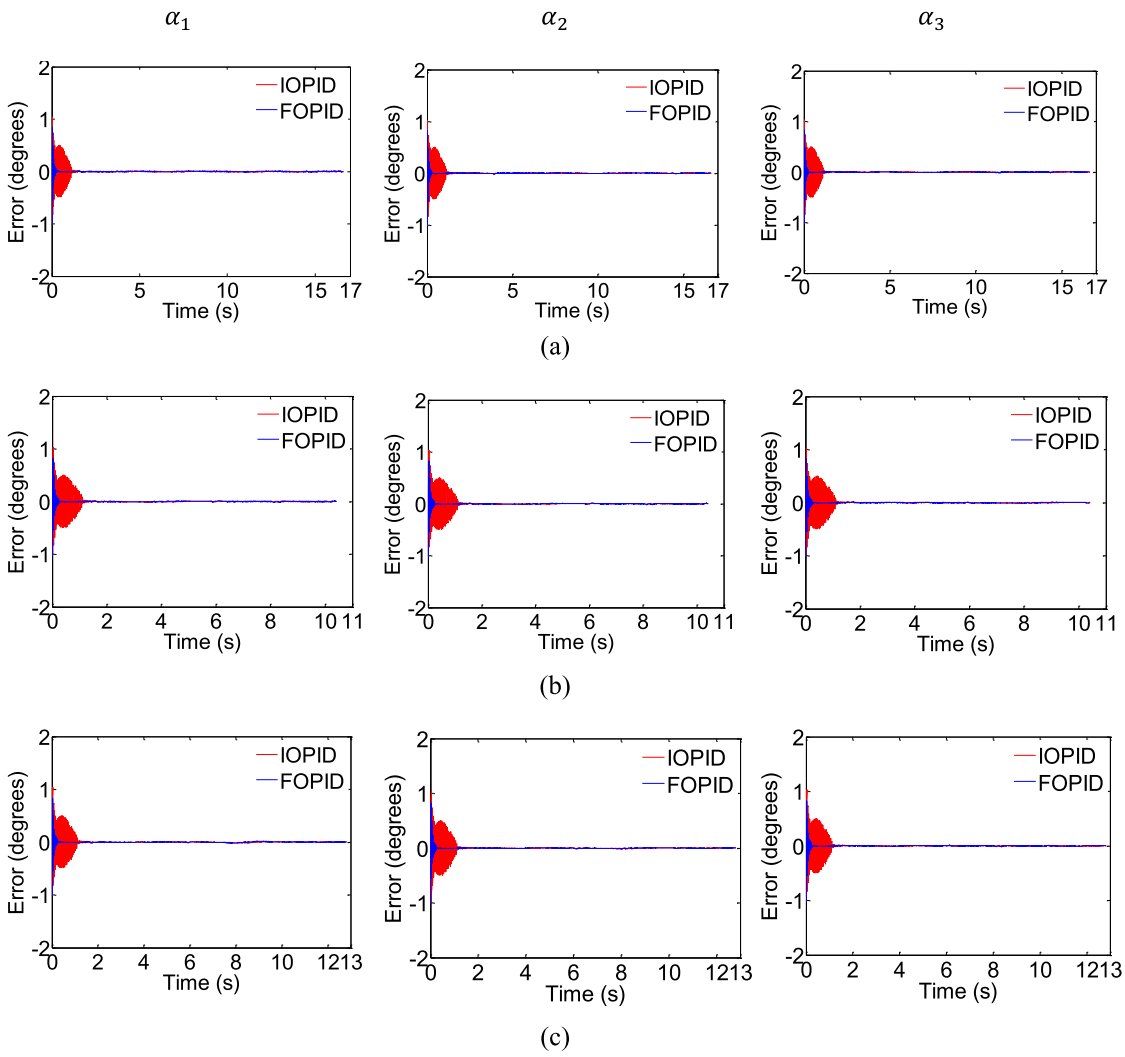

Fig. 12. Joint error at joints α_1, α_2 *and* α_3 for the trajectories (a) T1 (b) T2 (c) T3 employing the IOPID and FOPID controllers.

smaller value for both norms than IOPID controller. This indicates that the FOPID controller has a less joint and velocity errors in the joints of the robotic system, which represents a better tracking performance for the proposed trajectories. Therefore, from the performance indexes presented on Table 6, it can be said that the FOPID controller with the computed torque control strategy presents a robust performance against an external disturbance introduced in the applied torque of the joints of the robotic system regarding the IOPID controller with the computed torque control strategy for tracking tasks.

5.2. Critical payload at mobile platform test

In this test, a 10kg critical payload is placed in the mobile platform of the robotic system to analyze the performance of the delta robot with critical payload on tracking tasks. Fig. 15 shows the IOPID and FOPID controllers spatial tracking response of the mobile platform for the proposed trajectories. The red square is an enlargement of the robotic system response at this point to show more in detail the dynamical behavior of the robot in the presence of critical payload. Notice that this test modifies the inertia and mass parameters of the delta robot regarding the original parameters identified in Section 2.6.5.

As can be observed, the FOPID controller has a smaller spatial error than the IOPID controller. Considering that the delta robot with critical payload change significantly the parameters of the dynamic model, it can be said that the FOPID controller has better features for tracking tasks for the delta robot with payload in the mobile platform. Table 7 presents the performance indexes defined in (43)–(46) for the IOPID and FOPID controllers for the trajectories T1, T2, and T3 against the presence of critical payload in the mobile platform of the robotic system.

On the other hand, the RMSE value for the spatial and joint errors of the delta robot is smaller employing the FOPID controller. It indicates that the FOPID controller has better performance for tracking tasks against the presence of critical payload in the mobile platform. The RMS value for the applied torque in the joints of the delta robot is similar for the IOPID and FOPID controllers. However, the FOPID controller presents a better tracking performance with less spatial and joint errors. Analyzing the individual and global tracking norms for the delta robot, it shows that the FOPID controller exhibit a better tracking performance than the IOPID due to the FOPID presents the smallest norm values. From the performance indexes shown in Table 7, is possible to say that the FOPID controller presents a better tracking response with less spatial and joint errors. So, the FOPID controller with the computed torque control strategy for tracking tasks has a robust performance against

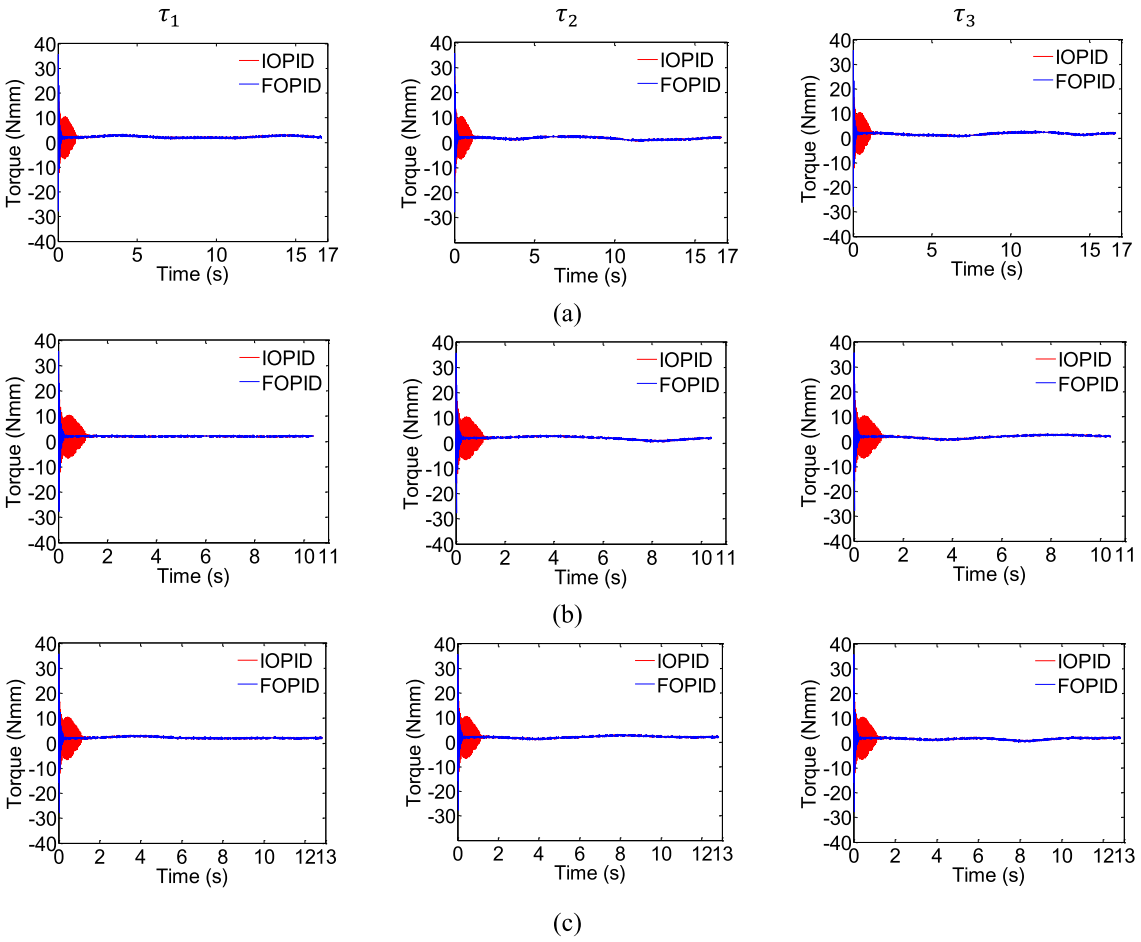

Fig. 13. Applied torque at joints α_1, α_2 and α_3 for the trajectories (a) T1 (b) T2 (c) T3 using the IOPID and FOPID controllers.

Table 5
Obtained performance indexes for the IOPID and FOPID controllers for nominal operating conditions.

Performance index	Signal	Trajectory					
		Trajectory T1		Trajectory T2		Trajectory T3	
		Controller					
		FOPID	IOPID	FOPID	IOPID	FOPID	IOPID
RMSE position error	X	0,323	0325	0,022	0022	0,278	0281
	Y	0,323	0325	0,452	0453	0,291	0290
	Z	0,238	0455	0,273	0561	0,318	0543
RMSE joint error	α_1	0,036	0084	0,046	0107	0,041	0096
	α_2	0,036	0085	0,045	0107	0,041	0097
	α_3	0,036	0085	0,046	0107	0,041	0097
RMS applied torque	α_1	2528	2746	2392	2750	2504	2786
	α_2	2048	2316	2419	2779	2404	2701
	α_3	2051	2320	2422	2782	2108	2442
Individual tracking norm N	α_1	5616	11,491	7086	14,509	6389	13,078
	α_2	5616	11,539	7087	14,569	6389	13,132
	α_3	5618	11,549	7090	14,582	6392	13,143
Global tracking norm N	X, Y, Z	9728	19,964	12,276	25,207	11,068	22,720

the presence of critical payload.

5.3. Random noise in the feedback loop test

This test introduces a random noise in the feedback loop with an amplitude of ±0.5 degrees in each joint of the delta robot to evaluate its effect over the IOPID and FOPID controllers with the computed torque control strategy for tracking tasks. Fig. 16 shows the IOPID and FOPID controllers spatial tracking response of the mobile platform for the trajectories T1, T2 and T3. The red square is an enlargement of the robotic system response at this point to show more in detail the dynamical behavior of the robot in the presence of random noise in the feedback loop. As can be observed, IOPID and FOPID controllers are similarly affected by the random noise in the feedback loop. Table 8 presents the performance indexes defined in (43)–(46) for the IOPID and FOPID controllers for the

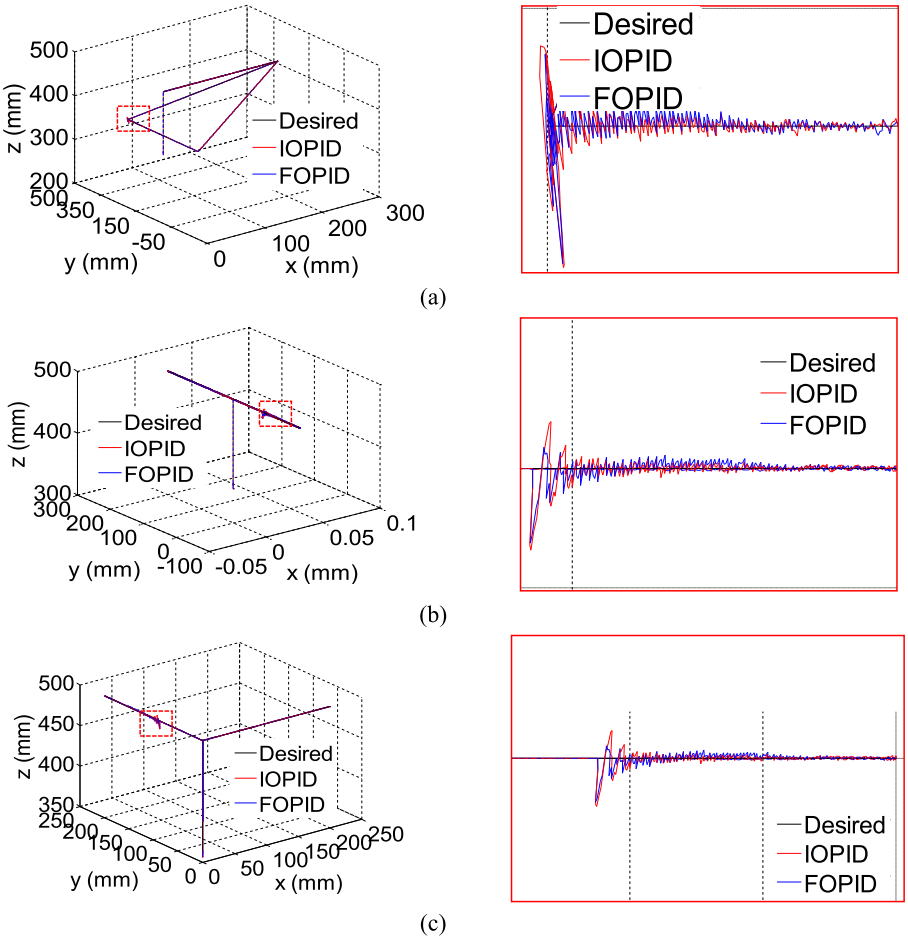

Fig. 14. Tracking performance for (a) T1 (b) T2 (c) T3 employing the IOPID and FOPID Controller against the presence of an external disturbance is settled in the applied torque of the joints of the robotic system.

Table 6
Obtained performance indexes for the IOPID and FOPID controllers for external disturbance in applied torque.

Performance index	Signal	Trajectory					
		Trajectory T1		Trajectory T2		Trajectory T3	
		Controller					
		FOPID	IOPID	FOPID	IOPID	FOPID	IOPID
RMSE position error	X	0,324	0330	0,022	0023	0,278	0281
	Y	0,366	0372	0,458	0468	0,298	0309
	Z	0,366	0576	0,501	0756	0,496	0710
RMSE joint error	α_1	0,072	0115	0,085	0137	0,076	0123
	α_2	0,075	0121	0,084	0136	0,076	0123
	α_3	0,065	0106	0,084	0137	0,075	0123
RMS applied torque	α_1	12,691	11,737	15,539	14,539	14,063	13,165
	α_2	12,721	11,690	15,498	14,517	14,022	13,140
	α_3	12,353	11,538	15,518	14,537	13,954	13,071
Individual tracking norm N	α_1	14,975	18,339	16,093	20,727	14,507	18,680
	α_2	16,075	19,671	15,873	20,576	14,354	18,603
	α_3	12,883	16,445	15,930	20,656	14,290	18,535
Global tracking norm N	X, Y, Z	25,468	31,523	27,653	35,772	24,914	32,227

trajectories T1, T2, and T3 against random noise in the feedback loop of the delta robot. As can be observed, the RMSE value for the spatial error is similar either the IOPID or FOPID controllers. In the case of joint error, the FOPID controller has a smaller RMSE value than the IOPID controller, which shows that the FOPID controller ensures a better joint position despite the presence of random noise in the feedback loop.

Regarding the RMS value of the applied torque at the joints, it is observed that the FOPID controller requires more applied torque at the joints of the delta robot for the active rejection of the random noise in the feedback loop. According to the individual and global tracking norms for the delta robot, the FOPID controller presents a

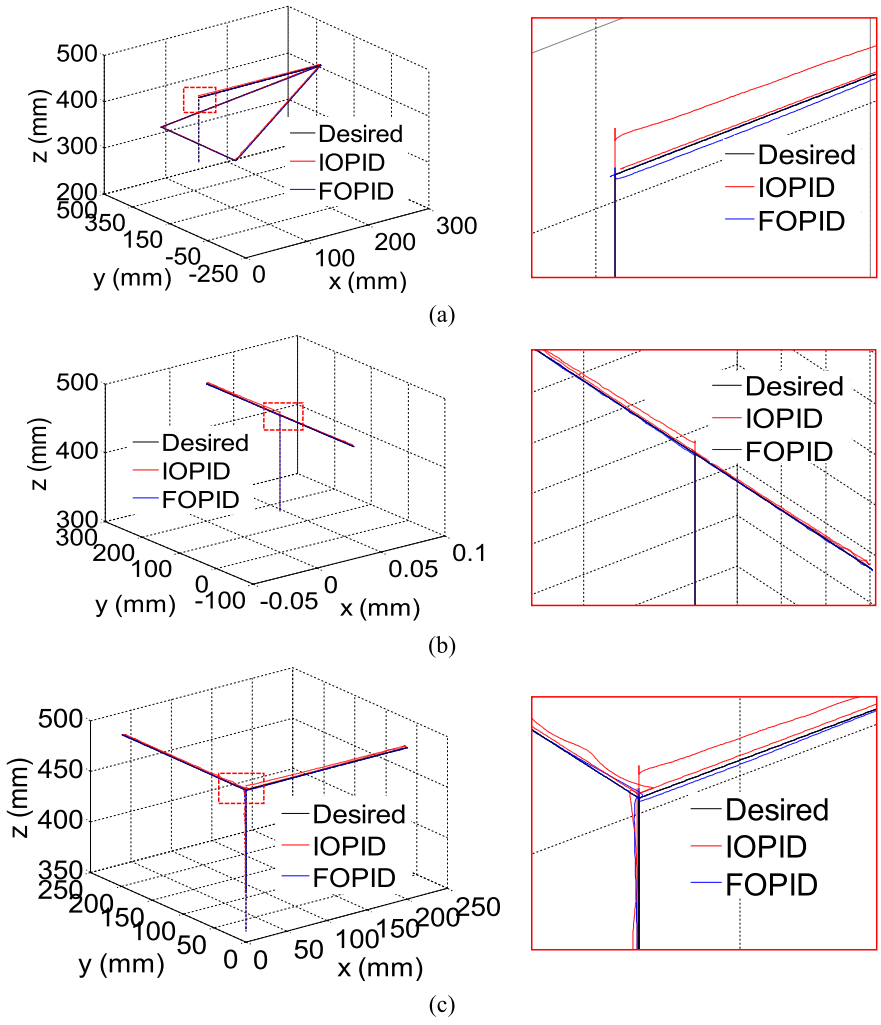

Fig. 15. Tracking performance for (a) T1 (b) T2 (c) T3 employing the IOPID and FOPID controllers in the presence of critical payload.

Table 7
Obtained performance indexes for the IOPID and FOPID controllers in the presence of critical payload.

Performance index	Signal	Trajectory					
		Trajectory T1		Trajectory T2		Trajectory T3	
		Controller					
		FOPID	IOPID	FOPID	IOPID	FOPID	IOPID
RMSE position error	X	1389	1657	0,259	0422	1313	1495
	Y	1313	1939	2275	2922	1449	1729
	Z	1136	1889	1429	2382	1327	2122
RMSE joint error	α_1	0,316	0496	0,310	0410	0,323	0494
	α_2	0,314	0438	0,404	0606	0,341	0449
	α_3	0,315	0428	0,412	0531	0,371	0428
RMS applied torque	α_1	15,076	15,081	12,520	12,509	14,377	14,391
	α_2	10,535	10,535	13,153	13,163	13,409	13,393
	α_3	10,576	10,578	13,178	13,146	10,340	10,334
Individual tracking norm N	α_1	11,711	14,197	11,849	14,858	11,917	14,461
	α_2	11,735	14,976	12,200	15,382	11,935	15,209
	α_3	11,708	14,843	12,216	14,417	11,985	15,283
Global tracking norm N	X, Y, Z	21,98	25,48	21,63	25,87	21,37	25,02

smaller value for both norms. Finally, it can be said that the FOPID controller with the computed torque control strategy for tracking tasks exhibits a better performance against the presence of random noise in the feedback loop than the IOPID controller.

6. Conclusions

In this paper, the design of a control system was presented for tracking tasks in a parallel robotic manipulator type delta of three

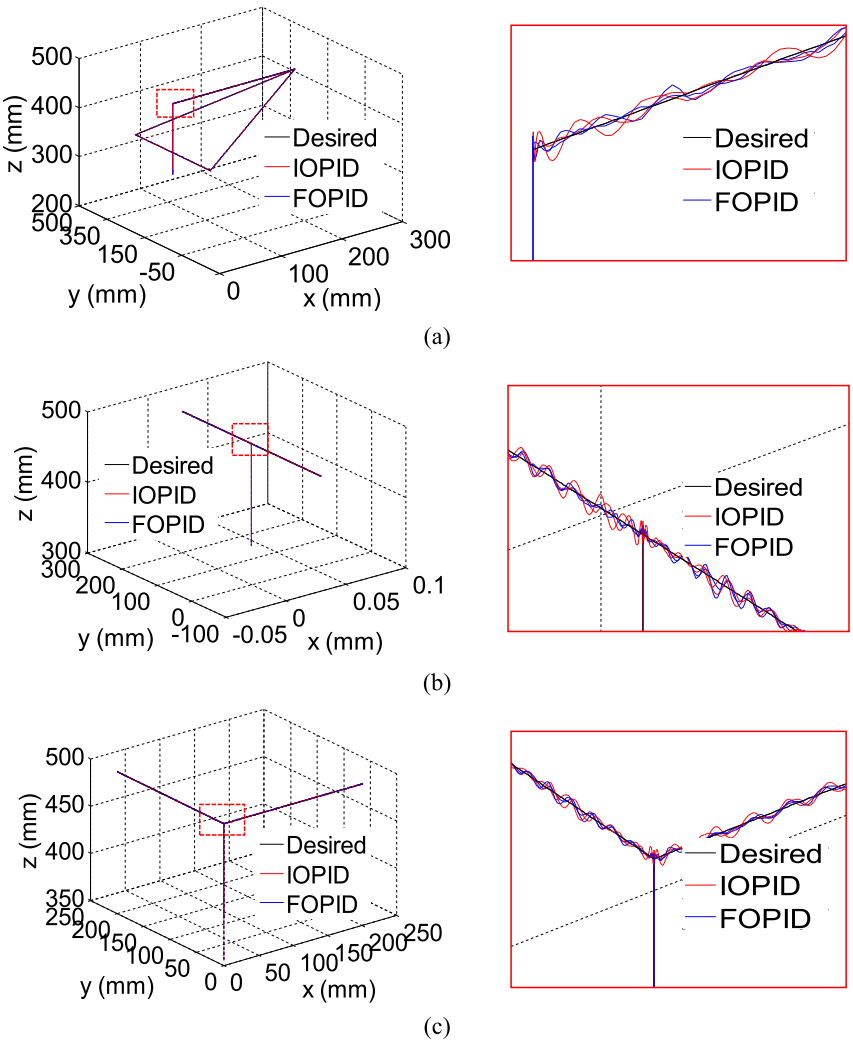

Fig. 16. Tracking performance for (a) T1 (b) T2 (c) T3 employing the IOPID and FOPID controllers against the presence of random noise in the feedback loop of the delta robot.

Table 8
Obtained performance indexes for the IOPID and FOPID controllers against random noise in the feedback loop.

Performance index	Signal	Trajectory					
		Trajectory T1		Trajectory T2		Trajectory T3	
		Controller					
		FOPID	IOPID	FOPID	IOPID	FOPID	IOPID
RMSE position error	X	0,326	0332	0,023	0023	0,280	0282
	Y	0,328	0331	0,454	0455	0,291	0291
	Z	0,314	0424	0,349	0467	0,386	0482
RMSE joint error	α_1	0,053	0078	0,059	0087	0,055	0081
	α_2	0,053	0079	0,058	0086	0,055	0081
	α_3	0,053	0079	0,058	0086	0,055	0081
RMS applied torque	α_1	3504	2828	3384	2678	3469	2779
	α_2	3180	2421	3390	2698	3390	2693
	α_3	3180	2432	3386	2702	3193	2432
Individual tracking norm N	α_1	7822	10,421	8677	11,736	8047	10,872
	α_2	7826	10,560	8601	11,681	8060	10,911
	α_3	7830	10,682	8603	11,698	8072	10,921
Global tracking norm N	X, Y, Z	13,56	18,28	14,94	20,27	13,96	18,88

degrees of freedom using the IOPID and FOPID controllers with the computed torque control technique. A SOLIDWORKS/MSC-ADAMS/MATLAB cosimulation model was employed to identify the dynamic model of the robot and validate the proposed control strategies.

The delta robot was subjected to three tests which were the presence of external disturbance in applied torque, a critical payload at mobile platform and random noise in the feedback loop

to evaluate the robust performance of the IOPID and FOPID controllers with the computed torque control technique in a quantitative way. These tests were performed for three trajectories. As performance indexes were employed the RMSE value for the spatial and joint errors, the RMS value for the applied torque in the delta robot joints and the individual and global tracking norms to measure the overall tracking performance of the proposed control strategies.

From the obtained results of the robustness analysis for the proposed trajectories, it was observed that the FOPID controller presented less spatial and joint errors and smaller tracking norms than the IOPID controller for the performed tests. It indicates that the FOPID controller with the computed torque control strategy has better performance for tracking tasks of the delta robot.

Also, the FOPID controller showed a slightly higher applied torque in the delta robot joints in the presence of random noise in the feedback loop and external disturbance, which indicates that the FOPID controller with the computed torque control strategy has an active disturbance rejection for tracking tasks of the delta robot.

Finally, it can be said that the FOPID controller with the computed torque control strategy presents a robust performance against the presence of external disturbances that affect the operation of the parallel robot type delta in tracking tasks.

As future works, the practical implementation of the FOPID and IOPID controllers with the computed torque control strategy in the delta robot prototype for tracking and positioning tasks is proposed. Another issue to be considered is the study and implementation of an active disturbance rejection control system using the FOPID controllers.

References

[1] Fanuk A. Genkotsu-robot 2 high speed picking and assembly robot FANUC robot M-2 + a operating space. 2013. p. 2–3.
[2] Yoo J, Park IW, To V, Lum JQH, Smith T. Avionics and perching systems of free-flying robots for the International Space Station. In: Systems engineering (ISSE), 2015 IEEE international symposium on; 2015. p. 198–201.
[3] Diftler MA, et al. Robonaut 2 #x2014; Initial activities on-board the ISS. In: Aerospace conference, 2012 IEEE; 2012. p. 1–12.
[4] Mata V, Farhat N. Dynamic parameter identification of parallel robots starting from the measurement of joints position and forces, vol. 32; 2009. p. 119–25. no. 2.
[5] M. Gautier, Dynamic parameter identification of actuation Redondant parallel robots;: Application to the DualV, vol. 1.
[6] Taghirad HD. Parallel robots: mechanics and control. CRC; 2013.
[7] Jang JT, Gong HC, Lyou J. Computed Torque Control of an aerospace craft using nonlinear inverse model and rotation matrix. In: Control, automation and systems (ICCAS), 2015 15th international conference on; 2015. p. 1743–6.
[8] Ha M-T, Kang C-G. Wireless-communicated computed-torque control of a SCARA robot and two-dimensional input shaping for a spherical pendulum. In: Ubiquitous robots and ambient intelligence (URAI), 2015 12th international conference on; 2015. p. 58–62.
[9] Horiguchi Y, Matsubara T, Kidode M. Learning Parametric Inverse Dynamics Models from multiple conditions for fast adaptive computed torque control. In: Robotics and biomimetics (ROBIO), 2011 IEEE international conference on; 2011. p. 2670–5.
[10] Kenmochi M, Avci E, Kawanishi M, Narikiyo T, Kawakami S, Saitou Y. Robust position control of Delta Parallel mechanisms using dynamic model and QFT. In: Industrial electronics (ISIE), 2014 IEEE 23rd international symposium on; 2014. p. 1256–61.
[11] Rigatos G, Siano P. An H-infinity feedback control approach to autonomous robot navigation. In: Industrial electronics society, IECON 2014-40th annual conference of the IEEE; 2014. p. 2689–94.
[12] Wu L, Gao Y, Liu J, Li H. Automatica Event-triggered sliding mode control of stochastic systems via output. Automatica 2017;82:79–92.
[13] Liu J, Vazquez S, Wu L, Marquez A, Gao H, Franquelo LG. Extended state observer-based sliding-mode control for three-phase Power converters. IEEE Trans Ind Electron 2017;64(1):22–31.
[14] Liu J, Luo W, Yang X, Wu L. Robust model-based fault diagnosis for PEM fuel cell air-feed system. IEEE Trans Ind Electron May 2016;63(5):3261–70.
[15] Hybrid controller with observer for the estimation and rejection of disturbances. ISA (Instrum Soc Am) Trans 2016;65:445–55.
[16] A linear active disturbance rejection control for a ball and rigid triangle system. Math Probl Eng 2016;2016:1–11.
[17] Sliding mode control of robotic arms with deadzone. IET Control Theory & Appl 2017;11(8):1214–21.
[18] Rubio JDJ, Ochoa G, Balcazar R, Pacheco J. Disturbance rejection in two mechatronic systems. IEEE Lat. Am. Trans. 2016;14(2).
[19] Rubio JDJ, Pacheco J, Juarez CF, Soriano E. Comparison between two observers. IEEE Lat. Am. Trans. 2016;14(5).
[20] Viola J, Angel L. Factorial design for robustness evaluation of fractional PID controllers. Lat. Am. Trans. IEEE (Revista IEEE Am. Lat. May 2015;13(5):1286–93.
[21] Viola J, Angel L. Identification, control and robustness analysis of a robotic system using fractional control. Lat. Am. Trans. IEEE (Revista IEEE Am. Lat. May 2015;13(5):1294–302.
[22] R. Clavel, Device for the movement and positioning of an element in space. Google Patents, 1990.
[23] ABB. Higher payload IRB 360. 2013.
[24] Angel L, Bermudez J, Munoz O. Dynamic optimization and building of a parallel delta-type robot. In: 2013 IEEE int. Conf. Robot. Biomimetics, ROBIO 2013; 2013. p. 444–9.
[25] Cortés FR. Matlab aplicado a robótica y mecatrónica. first ed. Alfaomega grupo editor; 2012.
[26] Kelly R, Davila VS, Perez JAL. Control of robot manipulators in joint space. Springer London; 2006.
[27] Podlubny I. Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications. Academic Press; 1999.
[28] Oldham KB, Spanier J. The fractional calculus : theory and applications of differentiation and integration to arbitrary order. New York: Academic Press; 1974.
[29] Monje CA, Chen YQ, Vinagre BM, Xue D, Feliu-Batlle V. Fractional-order systems and controls: fundamentals and applications. Springer London; 2010.
[30] Cortés FR. Robotica: control de Robots Manipuladores. Alfaomega grupo editor; 2011.