International Journal of Combinatorial Optimization Problems and Informatics, 13(3), Sep-Dec 2022, 17-33. ISSN: 2007-1558.

www.editada.org

Kinematic, Dynamic and Control of a 3-DOF Parallel Robot (3-PSS)

L.A. Pacheco-Escamilla[1], T. Cortés-Hernández[1], J.A. Meda-Campaña[2] and R. Tapia-Herrera[3]*

[1] Universidad Politécnica de Pachuca
[2] SEPI-ESIME Zacatenco, Instituto Politécnico Nacional, Av. IPN, Col. Lindavista, Ciudad de México, México.
[3] CONACYT-Universidad Tecnológica de Mixteca, Huajuapan de León, Oaxaca, México.
ph.luis05@gmail.com
tonatiuh@upp.edu.mx
jmedac@ipn.mx
rtapiah@conacyt.mx

Abstract. In this paper, the design and simulation of a Fuzzy Control for a 3 Degrees Of Freedom (3-DOF) Delta Parallel Robot with Prismatic actuators are presented. The position of the moving platform and prismatic joints are solved the direct and inverse kinematic analysis. The forces of the actuated prismatic joints are computed with the dynamic analysis using the Lagrangian approach and compared with the obtained forces in simulation; trajectory planning is "point-to-point", with polynomial functions. The proposed controller is a Fuzzy Control System, and the effectiveness has performance comparing the desired trajectory against the obtained via simulation.
Keywords: parallel manipulator, prismatic join, degree of freedom, kinematics, dynamics, trajectory planning, fuzzy control.

Article Info
Received Jun 26, 2022
Accepted Aug 16, 2022

1 Introduction

Robot manipulators are an important part of the industry as they are used for different purposes such as assembling, welding, materials handling, painting, packing, etc. [1]. Generally, robot manipulators are divided into two types, serial robots and parallel robots. Parallel manipulators have many advantages against serial manipulators like high stiffness, high accuracy, low moving inertia, high velocities and links with very small masses. In particular parallel robots are formed by connecting serial kinematic chains to a fixed base and to the moving platform (end-effector) this system is difficult to model due to these relations.

For the kinematic analysis the two problems: forward kinematics, which consists on determining the position and orientation of the end-effector when the joint parameters are known and the second, inverse kinematic, which involves compute the joint trajectories when the coordinates of the end-effector is given, both direct and inverse kinematics are computed generally by geometric approaches that relates the spatial location of the moving platform and the joints see [2], [3] and [4].

On the other hand, the dynamic model is an important issue to formulate strategy of control such as [Wisama, 2017] where the model can obtain by Newton-Euler equations or using the formalism Lagrange-Euler such as [Codourey, 1997]

Control of this type of robots is complex due to the limitations of workspace and highly nonlinear dynamic model adding the singularities that they present. Generally, the most common control requires the dynamic model and presenting limitations for finding controller and access the stability of the system [Zubizarreta, 2018] leading to a complex implementation on a physical model. To avoid this and in order to guaranty the tracking error it is usually consider only active joins, so the movement can be estimated by the kinematic model and the performance can be improve by full model and parameter identification of the

actuator [Ruiz Hidalgo, 2019]. On the other hand (PD or PID) controllers are usually used to industrial robots due to easy implementation [AGUILERA, 2017], however such controllers cannot achieve this performance because parameter variations and disturbances are not considered. In [P. Chiacchio, 1993] the join control scheme is proposed basis on new feedback loop that uses acceleration information; linear feedforward compensation is used for to improve tracking performance.

Intelligent control has proven solve problems of uncertainty is systems and applied to parallel robots as shown an alternative to classical methods such as [Jiangmin, 2018] and [Kim, 1996] where the variation of parameters and disturbances has been considered. Besides, control with adaptive fuzzy sliding mode is used as alternative to consider the dynamic nonlinear model and fuzzy neural network control theory to adjust such controller [Jiangmin, 2018].

Based on the information above the present work presents a kinematics analysis on the basis on geometry and differential kinematics also the dynamic model is presented by the formalism of Lagrange equations and Lagrange multipliers can be use due to the Jacobian is already computed. For control the proposed approach is given by fuzzy controller with three main components: the fuzzification that simply modifies the inputs so that they can be interpreted, the inference mechanism that compares the input value with the rules (in the rule-base), and defuzzification which converts the conclusions reached by the inference mechanism into the inputs to the plant (independent join). Additionally, simulation basis on the Matlab software is carrier out for the system in a simple way.

The work is presented as follows: The first to third sections describes the design of parallel robot, kinematic and dynamic model, respectively; The fourth and fifth section presents the trajectory planning and performance of the proposed control; sixth section present the implementation in Matlab sofware, and finally the conclusion is presented.

2 Prismatic Delta Robot (PDR) description.

The architecture of the PDR is shown in Figure 1. The robot is composed of a fixed base in which there are three vertical prismatic joints. Each prismatic joint has two links that connect it to the moving platform by spherical joints forming a serial kinematic chain of type P(SS). Due to the presence of parallelograms in each kinematic chain, the moving platform can only move along three directions and remains always with constant orientation.

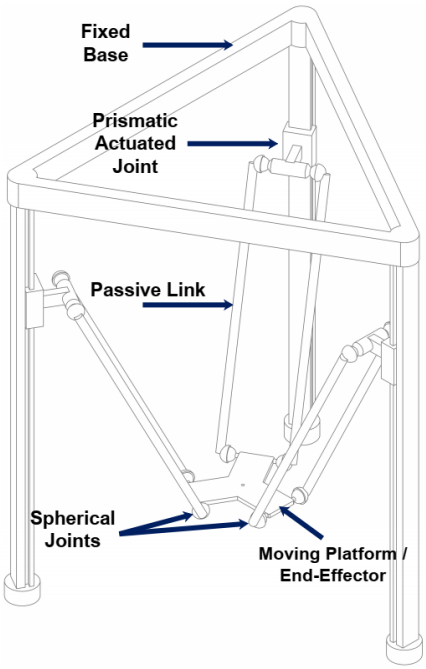


Fig. 1. Parallel Prismatic Delta Robot (PDR).

3 Kinematic and Dynamic Analysis.

a.3 Forward Kinematics.

For direct kinematics a geometric analysis of the mechanical model of the robot is performed, considering that the joints A_1, A_2 and A_3, have displacement along d_1, d_2 and d_3 respectively as shown in Figure 2. The radius of the base (R) and the platform (r) as well as the length of the arms (L) are known.

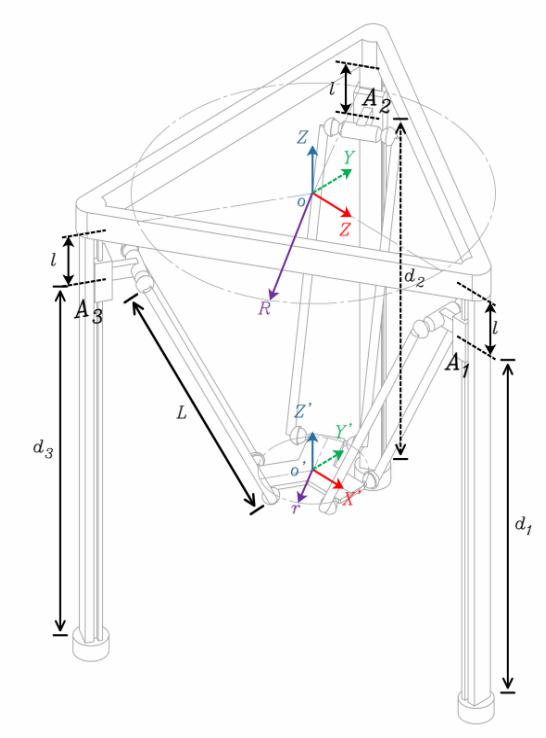


Fig. 2. Isometric view of the PDR and its parameters.

The robot arms rotate around the spherical joints of the prismatic joint A_i, forming three spheres of radius L. Figure 3 shows a side view of the mechanical model, in which only the carriage A_1 is taken into account (for the rest of the equations, a rotation of the frame $O-xyz$ by an angle of $120°$ and $240°$, for the carriages A_2 and A_3 respectively is needed).

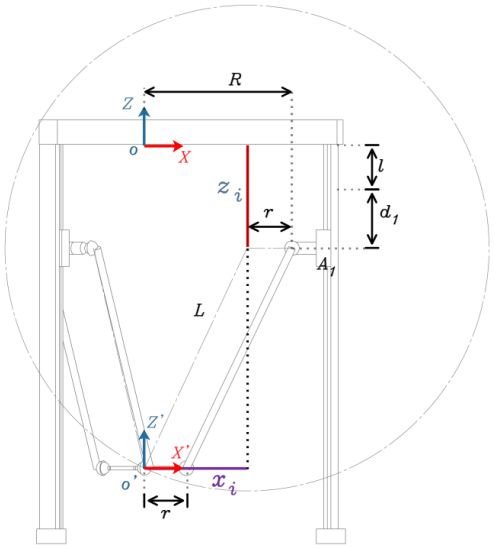
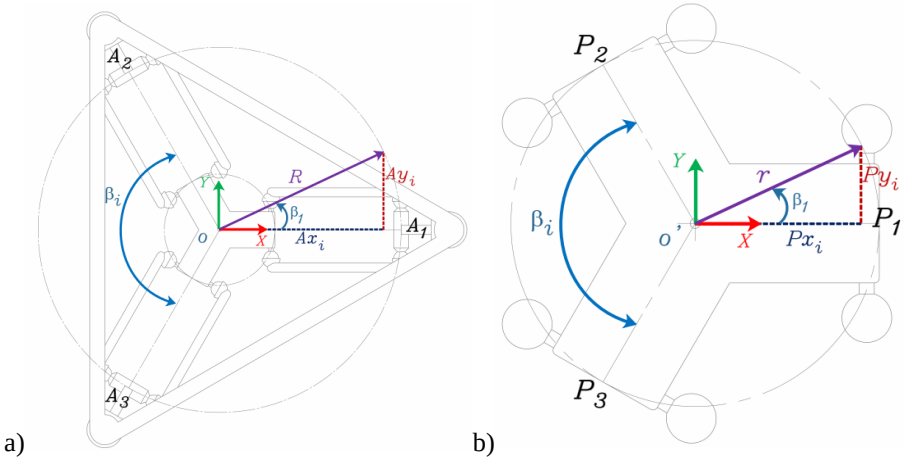


Fig. 3. Side view of the PDR.

In order for the end of the arm L to be tangent to the center of the platform, a distance r is projected in the opposite direction to the axis X, this projection is used for the sphere constraint equations. The equation of a sphere with origin in x_i, y_i, z_i and radius L is as follows:

$$(P_x - x_i)^2 + (P_y - y_i)^2 + (P_z - z_i)^2 = L^2 \qquad i = 1,2,3. \tag{1}$$

Looking at Figure 3, it is concluded that the coordinate z_i is the sum of the distance l with the parameter d_i.
To get the values of x_i and y_i, a top view of both the base and the platform is required, which are shown in Figure 4.



a) b)

Fig. 4. Top view of the PDR *a) Fixed base. B) Moving platform.*

Solving the triangle with the right angle of the Figure 4 (a):

$$A_{x_i} = R\cos\beta_i, \quad A_{y_i} = R\sin\beta_i, \quad A_{z_i} = 0\ . \tag{2}$$

Now if we consider the right angle of the triangle shown in Figure 4 (b):

$$P_{x_i} = r\cos\beta_i, \quad P_{y_i} = r\sin\beta_i, \quad P_{z_i} = l + d_i \ . \tag{3}$$

With the equations (2) and (3), and the Figure 3, the x_i, y_i and z_i values can be obtained, these values represent the origin of the spheres which are formed by the link L,

$$x_i = (R-r)\cos\beta_i, \quad y_i = (R-r)\sin\beta_i, \quad z_i = l + d_i \ , \tag{4}$$

where $\beta_i = (i-1)2\pi/3$.

Equation (1) is developed for every value of $i = 1,2,3,$ as follows:

$$P_x^{\,2} - 2P_x x_1 + x_1^{\,2} + P_y^{\,2} - 2P_y y_1 + y_1^{\,2} + P_z^{\,2} - 2P_z z_1 + z_1^{\,2} = L^2 \ , \tag{5}$$

$$P_x^{\,2} - 2P_x x_2 + x_2^{\,2} + P_y^{\,2} - 2P_y y_2 + y_2^{\,2} + P_z^{\,2} - 2P_z z_2 + z_2^{\,2} = L^2 \ , \tag{6}$$

$$P_x^{\,2} - 2P_x x_3 + x_3^{\,2} + P_y^{\,2} - 2P_y y_3 + y_3^{\,2} + P_z^{\,2} - 2P_z z_3 + z_3^{\,2} = L^2 \ . \tag{7}$$

By subtracting (6) to (5) and (7) to (5), equations (8) and (9) are obtained:

$$P_x(x_2 - x_1) + P_y(y_2 - y_1) + P_z(z_2 - z_1) = \frac{W_2 - W_1}{2} \ , \tag{8}$$

$$P_x(x_3 - x_1) + P_y(y_3 - y_1) + P_z(z_3 - z_1) = \frac{W_3 - W_1}{2} \ , \tag{9}$$

where: $W_1 = x_1^{\,2} + y_1^{\,2} + z_1^{\,2}$, $W_2 = x_2^{\,2} + y_2^{\,2} + z_2^{\,2}$ and $W_3 = x_3^{\,2} + y_3^{\,2} + z_3^{\,2}$.

To obtain P_y in terms of P_z, it is necessary to multiply equation (8) by $\left(\frac{x_3 - x_1}{x_2 - x_1}\right)$, and then subtract (9) to the result,

$$a_{11}P_y + a_{12}P_z = a_{13} \ , \tag{10}$$

where: $a_1 = \left(\frac{x_3 - x_1}{x_2 - x_1}\right)$, $a_{11} = a_1(y_2 - y_1) + y_1 - y_3$ and $a_{13} = \frac{1}{2}[a_1(W_2 - W_1) + W_1 - W_3]$.

Solving (10) for P_y:

$$P_y = b_1 P_z + b_2 \ , \tag{11}$$

where: $b_1 = -\frac{a_{12}}{a_{11}}$ and $b_2 = \frac{a_{13}}{a_{11}}$.

To obtain P_x in P_z terms, it multiplies (9) by $\left(\frac{y_2 - y_1}{y_3 - y_1}\right)$, and then (8) is subtracted to the result:

$$a_{21}P_x + a_{22}P_z = a_{23} \ , \tag{12}$$

where: $a_2 = \left(\frac{y_2 - y_1}{y_3 - y_1} \right)$, $a_{21} = a_2(x_2 - x_1) + x_1 - x_3$, $a_{22} = a_2(z_2 - z_1) + z_1 - z_3$ and $a_{23} = \frac{1}{2}[a_2(W_2 - W_1) + W_1 - W_3]$.

Solving (12) for P_x:

$$P_x = c_1 P_z + c_2 \ , \tag{13}$$

where: $c_1 = -\frac{a_{22}}{a_{21}}$ and $c_2 = \frac{a_{23}}{a_{21}}$.

Now that the values of P_x and P_y are known in P_z terms, to get the P_z value, (11) and (13) are replaced into (5).

$$AP_z^2 + BP_z + C = 0 \ , \tag{14}$$

where: $A = b_1^2 + c_1^2 + 1$, $B = 2[b_1(b_2 - y_1) + c_1(c_2 - x_1) - z_1]$ and $C = (b_2 - y_1)^2 + (c_2 - x_1)^2 + z_1^2 - L^2$.

a.3 **Inverse Kinematics**

To solve the inverse kinematic problem, the same formulation as before applies, so it is explained briefly. The x_i, y_i and z_i values from (4) are used once to replace them in (1):

$$\left[P_x - (R-r)\cos\beta_i\right]^2 + \left[P_y - (R-r)\sin\beta_i\right]^2 + \left[P_z - (l + d_i)\right]^2 = L^2 \ . \tag{15}$$

Then the carriage variables d_1, d_2 and d_3 are:

$$d_1 = \mp\sqrt{L^2 - [P_x - (R-r)]^2 - P_y^2} + P_z - l \ , \tag{16}$$

$$d_2 = \mp\sqrt{L^2 - \left[P_x + \frac{1}{2}(R-r)\right]^2 - \left[P_y - \frac{\sqrt{3}}{2}(R-r)\right]^2} + P_z - l \ , \tag{17}$$

$$d_3 = \mp\sqrt{L^2 - \left[P_x + \frac{1}{2}(R-r)\right]^2 - \left[P_y + \frac{\sqrt{3}}{2}(R-r)\right]^2} + P_z - l \ . \tag{18}$$

a.3 **Jacobian**

The end-effector moves along X, Y, Z axes, so the linear velocities can be obtained. However, the end-effector does not have rotation in any axes, so the angular velocities are considered equal to zero. To perform the Jacobian analysis of the parallel manipulator, the direct kinematics equations must be derived. The Jacobian matrix can be obtained by deriving the constraint equation which is the following:

$$(P_x - x_i)^2 + (P_y - y_i)^2 + (P_z - z_i)^2 = L^2 \ , \tag{19}$$

where: $x_i = (R-r)\cos(\beta_i)$, $y_i = (R-r)\sin(\beta_i)$, $z_i = l + d_i$, $i = 1,2,3$.

The constraint equation is developed as: $P_x^2 + P_y^2 + P_z^2 - 2P_x x_i - 2P_y y_i - 2P_z l - 2P_z d_i + x_i^2 + y_i^2 + l^2 + 2ld_i + d_i^2 = L^2$. Now, deriving with respect to time and solving for $\dot{d}_i$:

$$(P_x - x_i)\dot{P}_x + (P_y - y_i)\dot{P}_y + (P_z - z_i)\dot{P}_z = (P_z - z_i)\dot{d}_i \ . \tag{20}$$

Solving (20) for each $i = 1,2,3$ and writing in matrix form:

$$\begin{bmatrix} P_x - x_1 & P_y - x_1 & P_z - x_1 \\ P_x - x_2 & P_y - x_2 & P_z - x_2 \\ P_x - x_3 & P_y - x_3 & P_z - x_3 \end{bmatrix} \begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = \begin{bmatrix} P_z - z_1 & 0 & 0 \\ 0 & P_z - z_2 & 0 \\ 0 & 0 & P_z - z_3 \end{bmatrix} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} . \tag{21}$$

Thus, the linear velocities for the end-effector are:

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = \begin{bmatrix} \frac{P_x - x_1}{P_z - z_1} & \frac{P_y - y_1}{P_z - z_1} & 1 \\ \frac{P_x - x_2}{P_z - z_2} & \frac{P_y - y_2}{P_z - z_2} & 1 \\ \frac{P_x - x_3}{P_z - z_3} & \frac{P_y - y_3}{P_z - z_3} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} , \tag{22}$$

where: $P_z - z_i = \pm\sqrt{L^2 - (P_x - x_i)^2 - (P_y - y_i)^2} = P_z mz_i$, for $i = 1,2,3$.

From the equation above, the Jacobian that contributes to the end-effector linear velocities is obtained:

$$J = J_p^{-1} J_d = \begin{bmatrix} \frac{P_x - x_1}{Pzmz_1} & \frac{P_y - y_1}{Pzmz_1} & 1 \\ \frac{P_x - x_2}{Pzmz_2} & \frac{P_y - y_2}{Pzmz_2} & 1 \\ \frac{P_x - x_3}{Pzmz_3} & \frac{P_y - y_3}{Pzmz_3} & 1 \end{bmatrix}^{-1} . \tag{23}$$

a.3 **Dynamic model**

To simplify the dynamic analysis, the rotational inertias of the links L have been ignored because of its light weight. The mass of the link was distributed between the carriage and the moving platform. This dynamic model is a representation of the model developed in [3] and [12].

a) *Lagrange formulation:* Lagrange formulation describes the equations of motion (using a set of *generalized coordinates*), because of its use simplifies the dynamics analysis as a direct function of the Lagrange function which is composed of the contribution of kinetic and potential energies of the mechanism. To perform the dynamic analysis of the PDR, the Lagrange equations of the first kind [13] with Lagrange multipliers are used.

The Lagrange function is a state function and is defined as the subtracting between kinetic and potential energies of a system:

$$\mathcal{L} = K - P \ .$$

The Lagrange formulation of the first kind can be written as follows:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j}\right)-\left(\frac{\partial \mathcal{L}}{\partial q_j}\right)=Q_j+\sum_{i=1}^{3}\lambda_i\frac{\partial h_i}{\partial q_j},\quad j=1,\cdots,6. \tag{24}$$

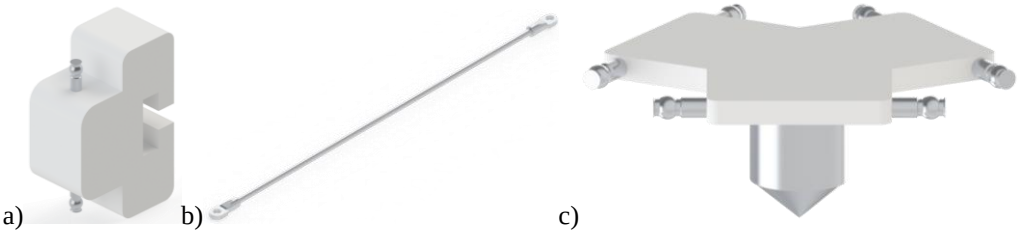
Where q_j are the generalized coordinates. Q_j are the generalized forces, assuming that on the platform are zero: $Q=\begin{bmatrix}Q1 & Q2 & Q3 & 0 & 0 & 0\end{bmatrix}$. h_i are the constraint functions, and λ_i are the Lagrange multipliers.

2) Generalized Coordinates: Generally, enough position and orientation variables of the moving platform corresponding to the number of degrees of freedom of the system should be considered for defining the generalized coordinates. The set of generalized coordinates includes independent and dependent variables (so it is a redundant set of coordinates). The number of independent generalized coordinates is equal to the number of degrees of freedom of the system, while the rest of variables (dependent variables) indicates the position of the moving platform by the restriction equations previously obtained in kinematics.

$$q_j=\begin{bmatrix}z_1 & z_2 & z_3 & P_x & P_y & P_z\end{bmatrix}^T. \tag{25}$$

3) Parallel manipulator components: Consider that in the whole system are:

- 3 Prismatic joints m_c (see Figure 5 a).
- 6 Links m_L (see Figure 5 b).
- Moving Platform m_{MP} (see Figure 5 b).



a) b) c)

Fig. 5. Parallel manipulator components. *A) Prismatic joint. B) Link L. C) Moving platform.*

4) Kinetic energy: The kinetic energy of the system K is a non-negative scalar function of the joint coordinates and their derivatives. The kinetic energy is an additive function so it should be computed for every component of the parallel robot. The total kinetic energy is defined by:

$$K=K_{MP}+\sum_{i=1}^{3}K_{c_i}+\sum_{j=1}^{6}K_{L_j}. \tag{26}$$

Adding the contributions and simplifying (26):

$$K=\frac{1}{2}(3m_L+m_{MP})(\dot{P}_x^2+\dot{P}_y^2+\dot{P}_z^2)+\frac{1}{2}(3m_L+m_c)(\dot{d}_1^2+\dot{d}_2^2+\dot{d}_3^2). \tag{27}$$

5) Potential energy: The only thing that generates potential energy is gravity. Like kinetic energy, it is an additive function and is given by:

$$P = P_{MP} + \sum_{i=1}^{3} P_{c_i} + \sum_{j=1}^{6} P_{L_j} \quad . \tag{28}$$

Adding the contributions and simplifying (28):

$$P = -g(3m_L + m_{MP})(P_z) - g(m_L + m_{MP})(z_1 + z_2 + z_3) \quad . \tag{29}$$

6) Lagrange Function: Once the kinetic and potential energy equations were obtained, the Lagrange function can be expressed in generalized coordinates terms.

$$\mathcal{L} = \frac{1}{2}(3m_L + m_{MP})(\dot{P}_x^2 + \dot{P}_y^2 + \dot{P}_z^2) + \frac{1}{2}(3m_L + m_c)(\dot{d}_1^2 + \dot{d}_2^2 + \dot{d}_3^2) + g(3m_L + m_{MP})(P_z) + g(m_L + m_{MP})(z_1 + z_2 + z_3) \quad . \tag{30}$$

7) Constraint Equations: The constraint equations are used to link the prismatic joints with the moving platform. The constraint equations were obtained in the kinematic analysis:

$$h_i = (P_x - x_i)^2 + (P_y - y_i)^2 + (P_z - z_i)^2 - L^2 = 0 \quad . \tag{31}$$

This constraint equation h_i ($i = 1,2,3$) links all generalized coordinates both dependents and independents.

8) First kind equations of Lagrange:

- Lagrange equation for the generalized coordinate P_x:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{P}_x}\right) - \frac{\partial \mathcal{L}}{\partial P_x} = \sum_{i=1}^{3} \lambda_i \cdot \frac{\partial h_i}{\partial P_x} \quad . \tag{32}$$

Considering that $m_1 = (3m_L + m_{MP})$. Obtaining and adding the contributions:

$$(m_1)\ddot{P}_x = 2\lambda_1\left[P_x - (R-r)\cos\beta_1\right] + 2\lambda_2\left[P_x - (R-r)\cos\beta_2\right] + 2\lambda_3\left[P_x - (R-r)\cos\beta_3\right] \quad . \tag{33}$$

- Lagrange equation for the generalized coordinate P_y:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{P}_y}\right) - \frac{\partial \mathcal{L}}{\partial P_y} = \sum_{i=1}^{3} \lambda_i \cdot \frac{\partial h_i}{\partial P_y} \quad , \tag{34}$$

$$(m_1)\ddot{P}_y = 2\lambda_1\left[P_y - (R-r)\sin\beta_1\right] + 2\lambda_2\left[P_y - (R-r)\sin\beta_2\right] + 2\lambda_3\left[P_y - (R-r)\sin\beta_3\right] \quad . \tag{35}$$

- Lagrange equation for the generalized coordinate P_z:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{P}_z}\right) - \frac{\partial \mathcal{L}}{\partial P_z} = \sum_{i=1}^{3} \lambda_i \cdot \frac{\partial h_i}{\partial P_z} \quad , \tag{36}$$

$$(m_1)(\ddot{P}_z - g) = 2\lambda_1\left(P_z - z_1\right) + 2\lambda_2\left(P_z - z_2\right) + 2\lambda_3\left(P_z - z_3\right) \quad . \tag{37}$$

9) Lagrange multipliers: The Lagrange multipliers can be obtained by simplifying (33), (35), and (37). Rewritten those equations in matrix form the Lagrange multipliers are:

$$\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\end{bmatrix} = \frac{1}{2}(m_1)(J_p{}^T)^{-1}\left(\begin{bmatrix}\ddot{P}_x\\ \ddot{P}_y\\ \ddot{P}_z\end{bmatrix} - g\begin{bmatrix}0\\0\\1\end{bmatrix}\right) . \tag{38}$$

10) Dynamic equations: The dynamic equations can be obtained solving (24), then using the Lagrange function (30) with respect to the generalized coordinates:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{z}_i}\right) - \left(\frac{\partial \mathcal{L}}{\partial z_i}\right) = Q_i + \sum_{i=1}^{3}\lambda_i \frac{\partial h_i}{\partial z_i} . \tag{39}$$

Getting the contributions from above equation $\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{z}_i}\right) = (3m_L + m_c)\ddot{z}_i$, $\left(\frac{\partial \mathcal{L}}{\partial z_i}\right) = g(m_L + m_c)$, $\frac{\partial h_i}{\partial z_i} = -2(P_z - z_i)$, then the dynamic equations are:

$$Q_1 = (m_1)\ddot{z}_1 - (m_2)g + 2\lambda_1(P_z - z_1) , \tag{40}$$

$$Q_2 = (m_1)\ddot{z}_2 - (m_2)g + 2\lambda_1(P_z - z_2) , \tag{41}$$

$$Q_3 = (m_1)\ddot{z}_3 - (m_2)g + 2\lambda_1(P_z - z_3) , \tag{42}$$

where: $m_2 = m_L + m_c$.

4 Trajectory Planning

Trajectory planning with a fifth-order polynomial allow to know the position, velocity and acceleration of the prismatic actuated joints by adding two more conditions (α_0 and α_f), so it has two more coefficients to compute.

The equations for the position, velocity and acceleration are:

$$q(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 , \tag{43}$$

$$\dot{q}(t) = 5a_5t^4 + 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1 , \tag{44}$$

$$\ddot{q}(t) = 20a_5t^3 + 12a_4t^2 + 6a_3t + 2a_2 . \tag{45}$$

From the above equations, six coefficients are obtained that can be calculated by evaluating the equations with the initial and final conditions of time, position, velocity and acceleration, so six equations can be obtained for the initial conditions:

$$\begin{aligned} q(t_o) &= a_5t_o^5 + a_4t_o^4 + a_3t_o^3 + a_2t_o^2 + a_1t_o + a_0 ,\\ \dot{q}(t_o) &= 5a_5t_o^4 + 4a_4t_o^3 + 3a_3t_o^2 + 2a_2t_o + a_1 ,\\ \ddot{q}(t_o) &= 20a_5t_o^3 + 12a_4t_o^2 + 6a_3t_o + 2a_2 . \end{aligned} \tag{46}$$

For the final conditions:

$$q(t_f) = a_5 t_f^5 + a_4 t_f^4 + a_3 t_f^3 + a_2 t_f^2 + a_1 t_f + a_0 \ ,$$

$$\dot{q}(t_f) = 5a_5 t_f^4 + 4a_4 t_f^3 + 3a_3 t_f^2 + 2a_2 t_f + a_1 \ , \tag{47}$$

$$\ddot{q}(t_f) = 20a_5 t_f^3 + 12a_4 t_f^2 + 6a_3 t_f + 2a_2 \ .$$

Equations (46) and (47) can be written in matrix form as follows:

$$\begin{bmatrix} q_0 \\ \dot{q}_0 \\ \ddot{q}_0 \\ q_f \\ \dot{q}_f \\ \ddot{q}_f \end{bmatrix} = \begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0 & 1 \\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0 & 1 & 0 \\ 20t_0^3 & 12t_0^2 & 6t_0 & 2 & 0 & 0 \\ t_f^5 & t_f^4 & t_f^3 & t_f^2 & t_f & 1 \\ 5t_f^4 & 4t_f^3 & 3t_f^2 & 2t_f & 1 & 0 \\ 20t_f^3 & 12t_f^2 & 6t_f & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} . \tag{48}$$

The above equation has the form $Ax = B$ and can be rewritten as $x = A^{-1}B$ to get the coefficients values. In Table 1 is depicted the transformation to operational space to joint space by kinematics method and the points that robot has to reach.

Table 1. Via points and joint space motions

	Operational space (cm)			Joint space (cm)		
Point	x	y	z	d_1	d_2	d_3
0	0	0	-35.57	0	0	0
1	0	0	-38.97	-3.39	-3.39	-3.39
2	13.64	0	-38.96	0	-12.06	-12.06
3	0	13.64	-38.96	-7.29	-0.87	-16.66
4	-13.64	0	-38.96	-18.39	-3.39	-3.39
5	0	-13.64	-38.96	-7.29	-16.66	-0.87
6	13.64	0	-67	-28.04	-40.11	-40.11
7	0	13.64	-67	-35.33	-28.91	-44.71
8	-13.64	0	-67	-46.89	-31.43	-31.43
9	0	-13.64	-67	-35.33	-44.71	-28.91
10	0	0	-35.57	0	0	0

5 Fuzzy Control for the PDR

a.3 Fuzzification

The universe of discourse of the controller input in this case the position error is defined by the rank $[-47 \quad 47]cm$. The input variable will be evaluated with five fuzzy sets and their membership functions for each joint. The linguistic terms for this fuzzy control design are: Negative Large Input (NLI), Negative Small Input (NSI), Positive Small Input (PSI), Input Zero (IZ), Positive Small Input (PSI), Positive Large Input (PLI), Negative Large Output (NLO), Negative Small Output (NSO), Output Zero (OZ), Positive Small Output (PSO) and Positive Large Output (PLO).

A) Input membership functions for the first and second joints:

- NLI, trapezoidal function: $[-47 \quad -47 \quad -2 \quad 1]$.
- NSI, triangular function: $[-2 \quad -1 \quad 0]$.
- IZ, triangular function: $[-0.2 \quad 0 \quad 0.2]$
- PSI, triangular function: $[0 \quad 1 \quad 2]$.
- PLI, trapezoidal function: $[1 \quad 2 \quad 47 \quad 47]$.

b) Input membership functions for the third joint:

- NLI, trapezoidal function: $[-47 \quad -47 \quad -2 \quad 1]$.
- NSI, triangular function: $[-2 \quad -1 \quad 0]$.
- IZ, triangular function: $[-0.3 \quad 0 \quad 0.3]$.
- PSI, triangular function: $[0 \quad 1 \quad 2]$.
- PLI, trapezoidal function: $[1 \quad 2 \quad 47 \quad 47]$.

The Figures 6 and 7 show the membership functions of the prismatic joints. The rank of the Figures has been minimized for a better interpretation of the membership functions, but the real rank is, as mentioned previously, $[-47 \quad 47]cm$.

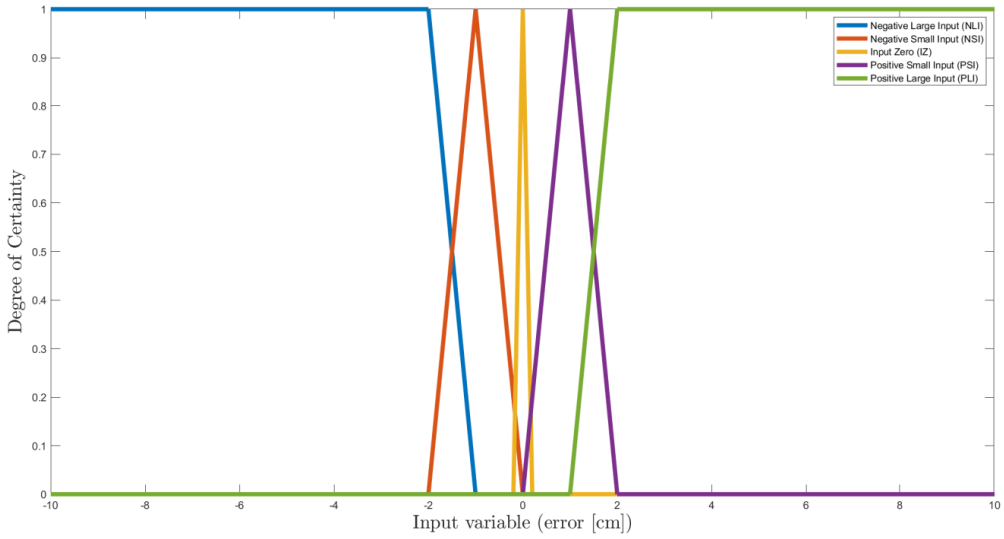


Fig. 6. Fuzzy sets for the error position, first and second joints.

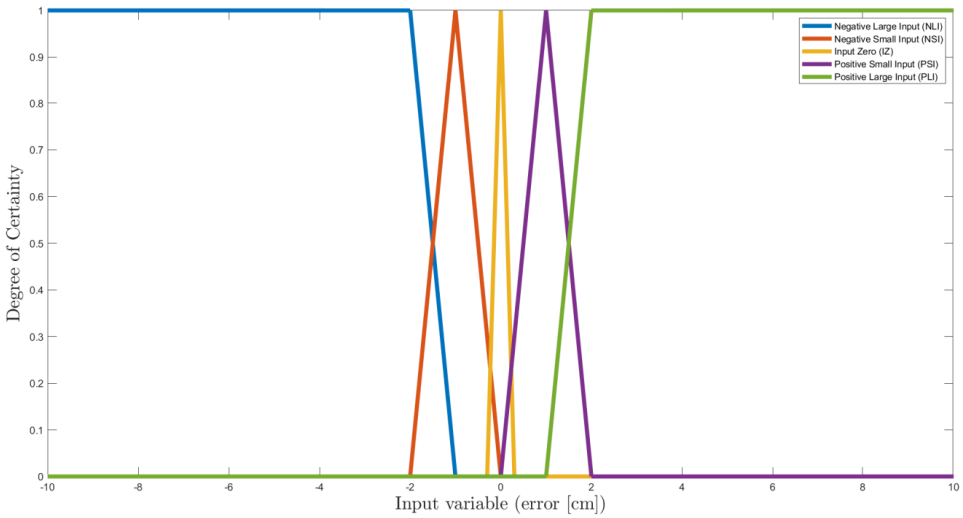


Fig. 7. Fuzzy sets for the error position, third joint.

a.3 **Inference mechanism**

The inference mechanism must evaluate the input value (position error) once fuzzified in order to move the joint to the desired position. The inference mechanism uses the following rule-base to perform its task.

a) Rule-base:

- IF *Negative Large Input*, THEN *Negative Large Output*
- IF *Negative Small Input*, THEN *Negative Small Output*
- IF *Input Zero* THEN *Output Zero*
- IF *Positive Small Input*, THEN *Positive Small Output*
- IF *Positive Large Input*, THEN *Positive Large Output*

a.3 **Defuzzification**

The output variable will be evaluated with five fuzzy sets and their membership functions for each joint. The universe of discourse of the controller output is defined by the rank $[-47 \quad 47]cm$. The Figure 8 shows the membership functions for the three controller outputs of the prismatic joints.

a) Output membership functions for the three joints:

- NLO, triangular function: $[-47 \quad -47 \quad -46]$.
- NSO, triangular function: $[-24.5 \quad -23.5 \quad -22.5]$.
- OZ, triangular function: $[-0.5 \quad 0 \quad 0.5]$.
- PSO, triangular function: $[22.5 \quad 23.5 \quad 24.5]$.
- PLO, triangular function: $[46 \quad 47 \quad 47]$.

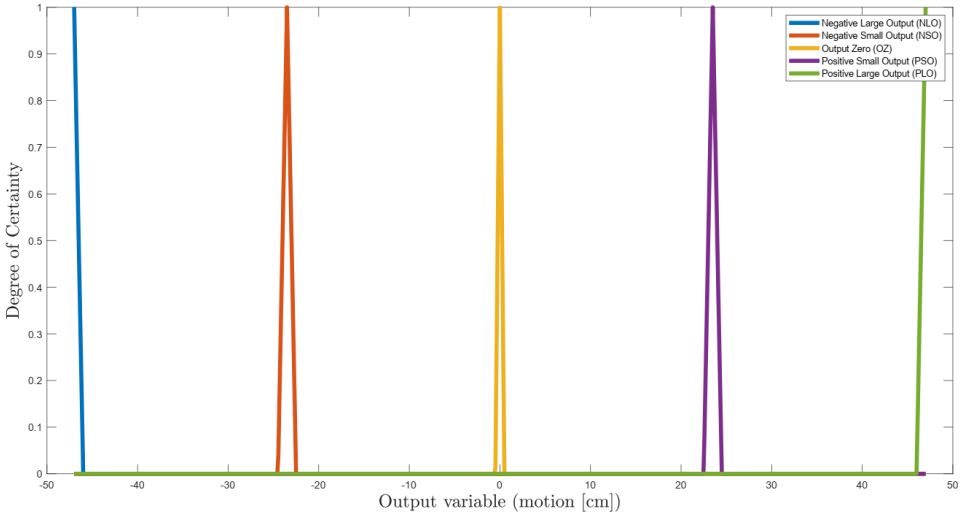


Fig. 8. Fuzzy sets of the controller outputs.

6 Results

This section presents the performance of the proposed controller to a plant (joint) under following control scheme depicted in Figure 9. The simulation was carried out in Matlab and Simulik 2018b. The Figure 10 shows the tracking signal for the prismatic joint 1 and Figures 11 and 12 for the two extra joins. One can observe the proposed controller is adequate.
On the other hand, the tracking error for the joints are depicted in Figure 13 the performance of error can be modifying by adjust membership functions.

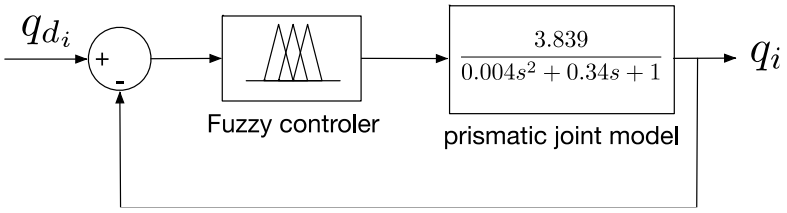


Fig. 9. Comparison between the desired trajectory and the trajectory obtained via simulation for the prismatic joint 1.

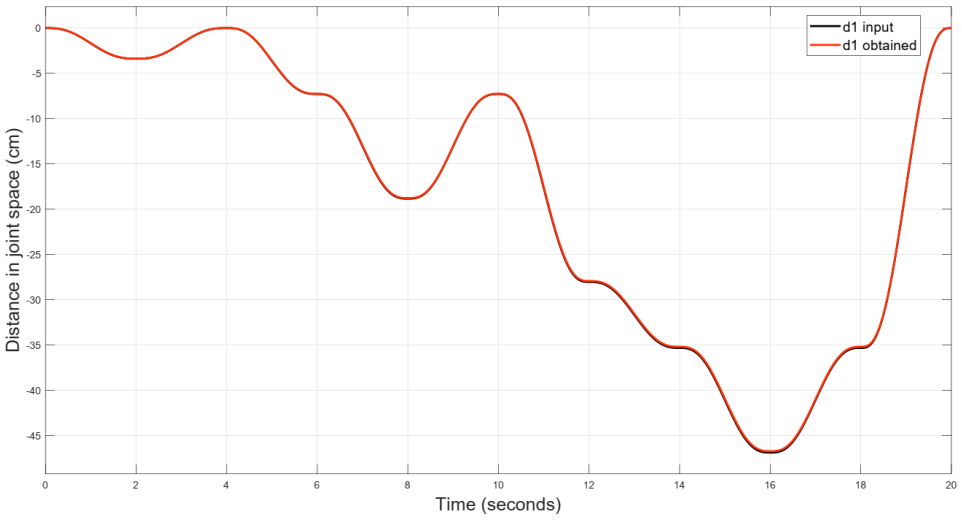


Fig. 10. Comparison between the desired trajectory and the trajectory obtained via simulation for the prismatic joint 1.

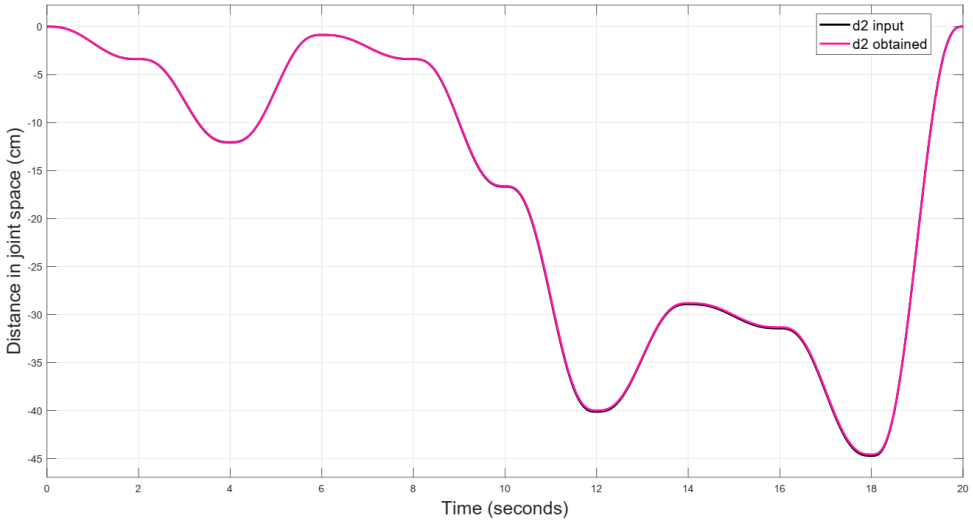


Fig. 11. Comparison between the desired trajectory and the trajectory obtained via simulation for the prismatic joint 2.

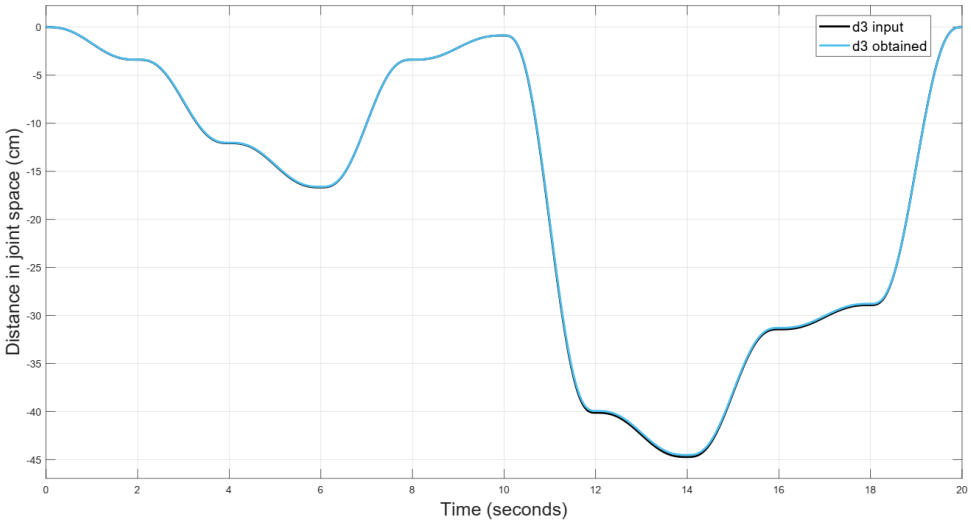


Fig. 12. Comparison between the desired trajectory and the trajectory obtained via simulation for the prismatic joint 3.

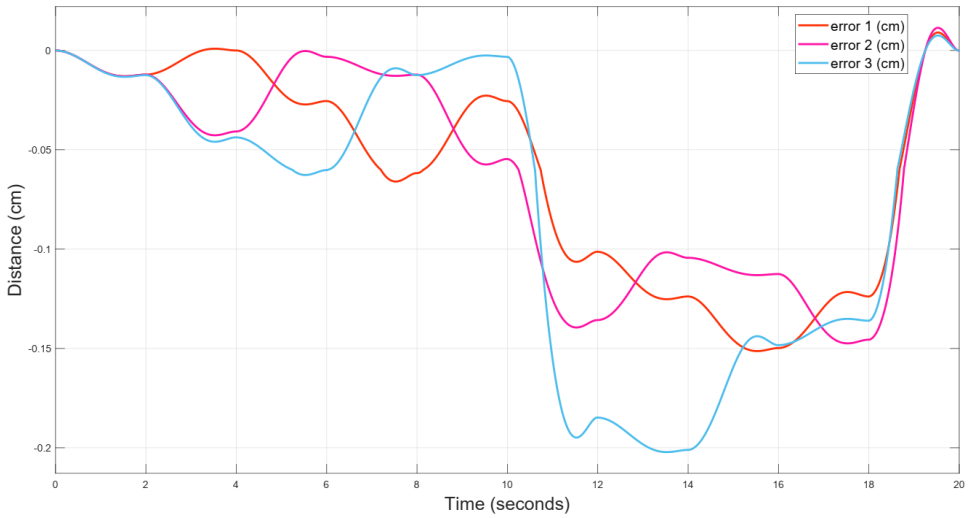


Fig. 13. Error tracking of prismatic joints $(d_1,\ d_2,$ and $d_3)$ implementing the fuzzy controller.

In order to show the trajectory in the operational space, Figure 13 shows the trajectory of the final effector in the proposed robot.

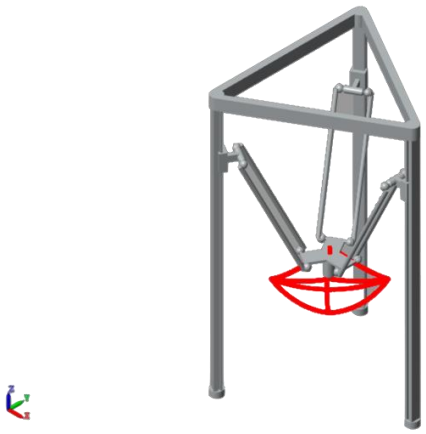


Fig. 13. Final effector path trajectory in operational space, according to Table 1.

7 Conclusions

Generally, a manipulator can perform tasks that requires smooth, accurate and fast movements. Prismatic Delta robot is a good option to perform these tasks because it can be able to reach high operation speeds and high accelerations. However, a controller is required to allow smooth movements in the robot joints. In the first part of this work the kinematics of the parallel manipulator were analyzed, and although the way in which the calculations were obtained was not by conventional methods (as is the case with serial robots), in general, it is not difficult to analyze the robotic model by making use of its geometric properties. However, for simulation purposes, it is necessary to know the exact measurements and positions of the components of the manipulator to avoid offset errors in the trajectory.

The dynamic analysis was carried out through the Lagrange formalism and the use of the Lagrange multipliers, because the Jacobian calculation is available. In the second part of the work the trajectory was designed for the Delta Robot through the most convenient method for the trajectory interpolation for this manipulator, and later the corresponding calculations were made that would allow simulating the trajectory.

Finally, the design of the fuzzy control was made that allows to control the smoothness of the trajectory maintaining a very small margin of error (position error). Although it is not necessary to know the system plant to design and implement a fuzzy

control. However, it is a complicated task tuning this type of controller. With this work we hoped to encourage the development of experimental platforms that include this type of controller.

References

1. Bonev. I. Delta parallel robot the story of success. *ParallelMIC*, 2001.
2. J.-P. Merlet et al. Advances in Robot Kinematics 2016. Springer International Publishing, 2018.
3. V. Borelli. Kinematic and dynamic analysis of a machine for additive manufacturing. *POLITECNICO DI TORINO*, 2018.
4. F. Z. B. Quafae Hamdoun, Larbi El Bakkali. Analysis and optimum kinematic design of a parallel robot. Procedia Engineering, 2017.
5. O. I. Wisama KHALIL. General solution for the dynamic modeling of parallel robots. Journal of Intelligent and Robotic Systems, 2007.
6. E. B. A. Codourey. A body-oriented method for finding a linear form of the dynamic equation of fully parallel robots. Proceedings of International Conference on Robotics and Automation, 1917.
7. I. E. Zubizarreta A, Larrea M. Real time direct kinematic problem computation of the 3prs robot using neural networks. Neurocomputing, 2018.
8. A. A. P. J. C.-O. NC Ruiz-Hidalgo, A Blanco Ortega and W. M. A. Rosado. Dynamic analysis and control of a three revolute prismatic spherical parallel robot by algebraic parameters identification. International Journal of Advanced Robotic Systems, 2019.
9. Aguilera-Jiménez Miguel Angel J, PAUL-Christopher et al. Diseño y control de un sistema generador de electricidad de disco-stirling. Revista del Diseño Innovativo, 2017.
10. L. S. P. Chiacchio, F. Pierrot and B. Siciliano. Robust design of independent joint controllers with experimentation on a high-speed parallel robot. IEEE Transactions on Industrial Electronics, 40(4):393– 403, 1993.
11. Q. L. Jiangmin Xu, Qi Wang. Parallel robot with fuzzy neural network sliding mode control. Advances in Mechanical Engineering, 2018.
12. F. Colomboand L. Lentini. Prismatic delta robot: A Lagrangian approach. In International Conference on Robotics in Alpe-Adria Danube Region, pages 315–324. Springer, 2020.
13. H. D. Taghirad. Parallel robots: mechanics and control. CRC press, 2013.