

Original Article

Kinematics Simulation and Experiment for Optimum Design of a New Prototype Parallel Robot

Surin Subson¹, Dechrit Maneetham², Myo Min Aung³

^{1,2,3}*Mechatronics Engineering, Rajamangala University of Technology Thanyaburi, Thailand*

²*Correspond Author: dechrit_m@rmutt.ac.th*

Received: 11 July 2022

Revised: 10 October 2022

Accepted: 18 October 2022

Published: 23 October 2022

Abstract - Technological progress allows for the more efficient production of more and better goods and services, bringing economic and social benefits. The labor shortage in production highlights the issues that can be helped by automated technologies, one of the reasons the industrial sector sees more opportunities to adopt robots. Parallel robots are prevalent industrial robots because they are ideal for packing lines; their high flexibility and adaptability can be used to arrange workpieces of different sizes, colors, or shapes. For the design or selection of robots to suit the working area conditions, it is imperative to study the use of robotic arms in an industrial setting. The most important issue is controlling robotic arms to get the job done accurately. To obtain accurate location coordinates, two methods of robot analysis are applied: forward kinematics and reverse kinematics. In this research, the theory of inverse kinematics analysis is applied to parallel robots to test the coordinate position of the robotic arm in robots' movement and compare the computational results. The simulation results and the actual test results are evaluated to determine the accuracy of the robot's working area. It gives a clear idea of how much the robot can move. To plan the path of movement and calculate the forearm position, forward kinematics is studied, and compare the results of the simulation using the MATLAB simulation program, the operation of the robot is analyzed and checked for acceptable errors to make the robot functional based on the results of the inverse kinematics simulations compared with the actual experiments, the joint angles steady-state error of the prototype robot was in the range of θ_1 , θ_2 , and θ_3 between 0° to 4° degrees.

Keywords - Parallel robot, MATLAB Simulink, Trajectory Tracking, Kinematic model, Workspace Modeling.

1. Introduction

In both small and large industries, parallel robots are used for pick and place jobs and packaging where the robot arm can move quickly to do the tasks [7,8,17,18]. Today, parallel robots are developed with much better technology. For example, Delta X has produced Open-source robots: Model Delta X1, Delta X2, and Delta XS, etc. [17]. Delta X robots are complete desktop robots with modular kits and can be used with various end effectors for various applications; it provides increased speed and flexibility and is much cheaper than industrial robots with similar features. It is suitable for robot researchers and uses in industrial plants.

The parallel robot was first developed by Raymond Clavel [1,3,4,6,8], who studied and prototyped a parallel robot in 1980. The base of the parallel robot is mounted above the work area, and all actuators are placed on it. From the base, three central arm joints are extended. These forearms are connected to a small triangular platform. Calling the input link will move the platform. Triangle along the X, Y, or Z direction, excitation can be done with the linear or rotary actuator with or without reduction (direct drive).

Parallel robots are designed to be stable and work properly within a given space. It is a difficult system because it is non-linear, and all robotic forearm joints are connected in a closed system. However, the solutions are infinite. In designing the robot, it is important to include the functions that support the robot's movement that work with the workspace the production process needs. This paper presents the parallel robot design based on mathematical calculation and uses MATLAB program to simulate [14-16]. The data reflecting the robot's movement is collected through the C++ Arduino terminal to control and display the robot's movement, which helps to check the robot's behaviour compared to the robot motion simulation results.

2. Parallel Robot Modelling

The parallel robot is a robot with a degree of freedom greater than a degree of manipulation, the smallest number of degrees of freedom used in the position and orientation of the robot's end-effector. For example, a robot that moves in a plane will have three degrees of control: the distance along the x and y axes is used to tell the position, and the angle of rotation around the axis z is used to give directions.



2.1. Design of Robot Manipulator

In designing the parallel robot, the work begins with an analysis of the original structure of the robot parallel and the data from the robot's operation, which is used to create the project. Then the structure of the robot was designed using the SOLIDWORK program. After purchasing the equipment comprising various components, the components are tested to see if any parts need to be improved to work according to the designed steps. The physical structure of the parallel robot can be seen in Fig. 1, Fig. 2, and Fig. 3.

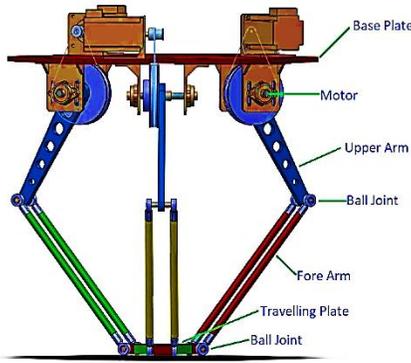


Fig. 1 Physical of 3 DOF parallel robot

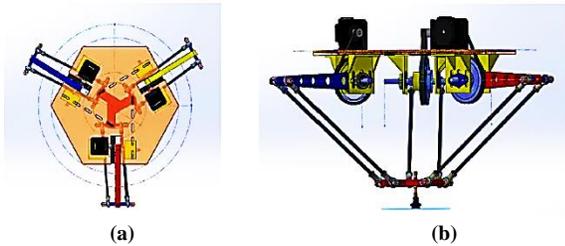


Fig. 2 Parallel Robot structure (a) Top View (b) assembly Section

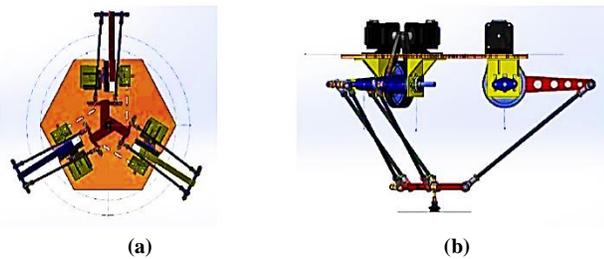


Fig. 3 Parallel Robot structure (a) Bottom View (b) Side View

2.2. Parallel Robot Parameters

When designing a Parallel robot, one of the essentials is the parameters of the various parts used to build the parallel robot. These parameters are very detrimental when not available [2,5,6,8,9]. Accurate presentation with parameter locations can be observed in Fig.4.

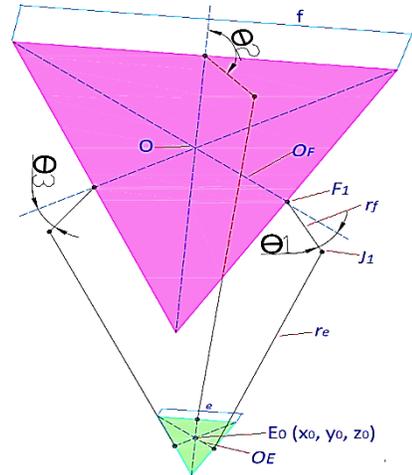


Fig. 4 Parallel robot mechanism diagram.

Determination of important parameters can be done as follows: parameter f , which is the value measured by the lengths of the sides of the base triangle. Parameter e , is the value measured by the side lengths of the lower triangle. The rf parameter, the value measured by the length of the upper arm, is measured from the centre connected to the actuator unit. For the definition of parallel robot parameters, see Table 1.

Table 1. Geometrical Parameter of the parallel robot

Parameter Name	Symbol
Length of upper joint	rf
Length of lower joint	re
Side of the fixed triangle	f
Side of the end effector triangle	e
Distance between the center of the base and the center of the motor	OF
Distance between center of travelling plate and ball joint	OE

2.3. Control System Design

The control system uses a microcontroller board Arduino R3 model. It uses the C++ IDE (C++ Integrated Development Environment) program for processing data and issuing conditional control commands to the driver of the motor. The microcontroller board No.1 receives a signal from camera vision (Machine Vision) via the NI Vision LABVIEW program.

For data acquisition and image processing, the control signals go to microcontroller boards no. 2, 3, and 4 to control stepper motors that communicate through the serial communication system.

Table 2. Geometrical Dimensions of the parallel robot

Parameter Name	Symbol	Value (mm)
Length of the upper link	r_f	200
Length of the lower link	r_e	500
Side of the fixed triangle	f	554.26
Side of the end effector triangle	e	259.81
Distance between the center of the base and the center of the motor	OF	160
Distance between center of travelling plate and ball joint	OE	75

A parallel robot is designed by a total of 3 stepper motors with rotation sensors that detect the motors' rotation position and send the feedback signal to microcontroller board No.1 to present the signal result. An operation control system uses a computer to communicate with the microcontroller boards and control the parallel robot's operation. Each device has to communicate and send information to each other, as shown in Fig. 5.

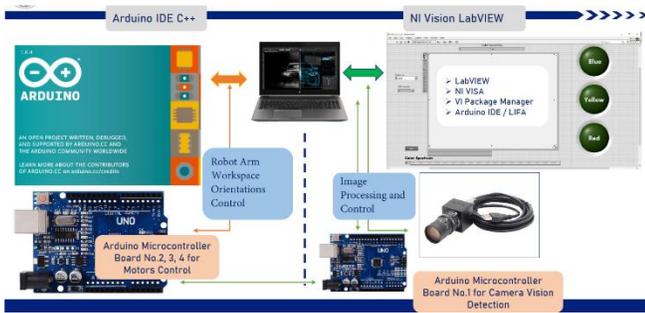


Fig. 5 Interface of all microcontroller boards with camera vision system and Control Program

2.3.1. Control System Work Methods

The microcontroller board No.1 receives a signal from the push button switch to start the system first. After that, the conveyor system will start to transport the workpiece through the camera vision system that will detect the colour of the workpiece in which the red workpiece is the target piece. After that, the control camera vision system sends a detection signal to microcontroller boards No.2, 3, and 4 to control the stepper motors driver, to control all motors to rotate and guide the robot arm in a certain direction according to predetermined conditions. The rotation angle sensor of each motor signals the position of the upper arm angle of the robot to microcontroller board No. 1, which shows the position of the robot's upper arm angle by looking at the serial monitor program page from the program Arduino C++ IDE. The resulting signal can be compared with the kinematic simulation results obtained by the MATLAB Simulation program [2], [14-16]. For further data analysis. The Parallel Robot control system schematic is shown in Fig. 6, and Fig.7 demonstrates the control process workflow.

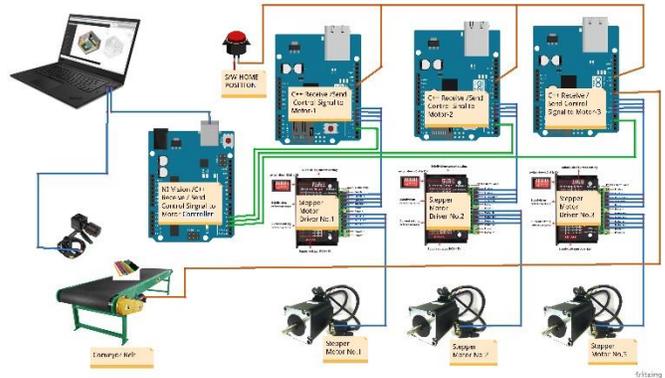


Fig. 6 Parallel Robot control system schematic

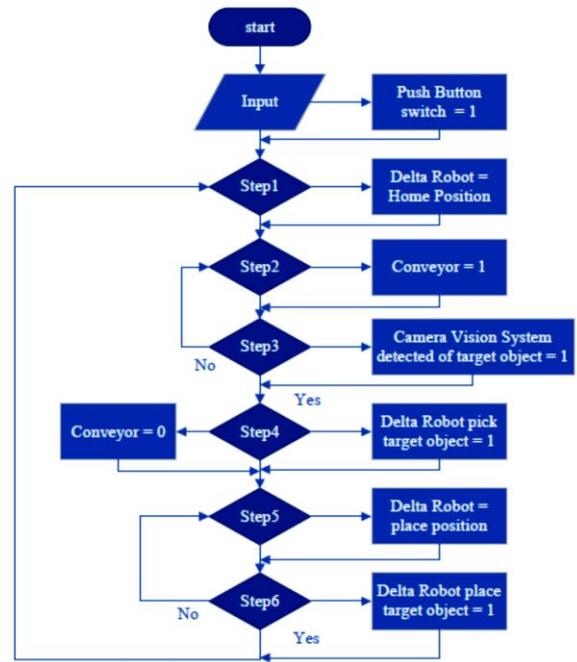


Fig. 7 Interfacings of Parallel robot system demonstrates control process workflow

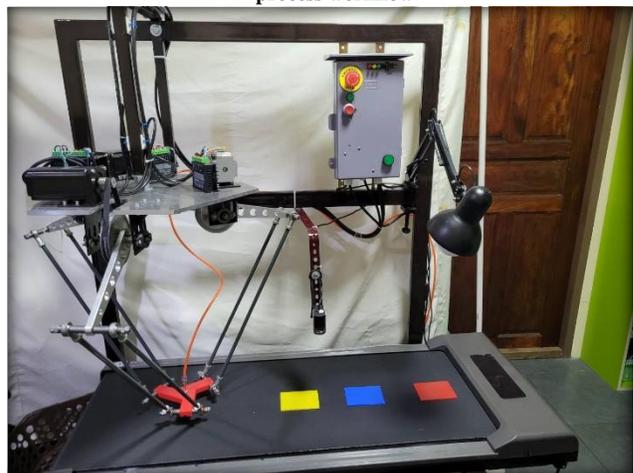


Fig. 8 Parallel robot mechanism interfacing all three motors with the microcontroller, camera vision, and conveyor system.

2.4. Parallel Robot Kinematics

Kinematic analysis of the Parallel robot is used to find the correlation equation to control the motion of the Parallel robot [2-13]. Robot kinematics describes the relationship between the position of the robot arm (End Effector) and the arrangement of the robot's joints and arms through mathematical equations regardless of the force acting on the link and dynamics. The results from the analytical equation can come out in 2 forms depending on the usage; Forward Kinematic, which provides an answer to the position of the robot's tip when the position of the joint angles is known, and inverse kinematic, which provides an answer to the position of the joint angles when the position of the end of the robot is known. And since Parallel robots are industrial robots with a connecting rod structure on all three axes arranged in parallel. It makes the kinematic analysis different from a robot with a serial junction structure. The analytical method in this research will be analyzed by geometric method.

2.4.1. Parallel Robot Forward Kinematic

Forward kinematics in the case of knowing common angles given that the three corners are θ_1 , θ_2 and θ_3 respectively [2-13], with the center position of the moving plate (X, Y, Z) must be found. First, it is necessary to determine the robot's parameters to calculate the forward kinetics [3] of a 3-axis parallel robot. To use for calculations in mathematical equations, then determine the length of the top base plate is f ; the length of the moving plate is e ; the length of the upper arm is rf ; t is moving the center of a circle from points J_1, J_2 and J_3 to points J_1', J_2' and J_3' in 3D view. The length of the lower arm is re . The origin point of the three-sphere is expressed as

$$(x - x_j) + (y - Y_j) + (z - Z_j) = re^2 \rightarrow j = 1,2,3$$

$$OF_1 = OF_2 = OF_3 = \frac{f}{2} \tan(30^\circ) = \frac{f}{2\sqrt{3}} \tag{1}$$

$$J_1J_1' = J_2J_2' = J_3J_3' = \frac{e}{2} \tan(30^\circ) = \frac{e}{2\sqrt{3}} \tag{2}$$

$$F_1J_1 = rfcos(\theta_1) \tag{3}$$

$$F_2J_2 = rfcos(\theta_2) \tag{4}$$

$$F_3J_3 = rfcos(\theta_3) \tag{5}$$

$$J_1' = \begin{bmatrix} 0 \\ \frac{-(f-e)}{2\sqrt{3}} - rf \cos(\theta_1) \\ -rf \sin(\theta_1) \end{bmatrix} = (x_1, y_1, z_1) \tag{6}$$

$$J_2' = \begin{bmatrix} \left[\frac{f-e}{2\sqrt{3}} + rf \cos(\theta_2) \right] \cos(30^\circ) \\ \left[\frac{f-e}{2\sqrt{3}} + rf \cos(\theta_2) \right] \cos(30^\circ) \\ -rf \sin(\theta_2) \end{bmatrix} = (x_2, y_2, z_2) \tag{7}$$

$$J_3' = \begin{bmatrix} \left[\frac{f-e}{2\sqrt{3}} + rf \cos(\theta_3) \right] \cos(30^\circ) \\ \left[\frac{f-e}{2\sqrt{3}} + rf \cos(\theta_3) \right] \cos(30^\circ) \\ -rf \sin(\theta_3) \end{bmatrix} = (x_3, y_3, z_3) \tag{8}$$

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = re^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = re^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = re^2 \end{cases} \tag{9}$$

$$\begin{cases} x^2 + y^2 + z^2 - 2y_1y - 2z_1z = re^2 - y_1^2 - z_1^2 & [1] \\ x^2 + y^2 + z^2 - 2x_2x - 2y_2y - 2z_2z = re^2 - x_2^2 - y_2^2 - z_2^2 & [2] \\ x^2 + y^2 + z^2 - 2x_3x - 2y_3y - 2z_3z = re^2 - x_3^2 - y_3^2 - z_3^2 & [3] \end{cases} \tag{10}$$

$$w_i = x_i^2 + y_i^2 + z_i^2 \tag{11}$$

$$\begin{cases} x_2x + (y_1 - y_2)y + (z_1 - z_2)z = \frac{w_1 - w_2}{2} \rightarrow [4] = [1] - [2] \\ x_3x + (y_1 - y_3)y + (z_1 - z_3)z = \frac{w_1 - w_3}{2} \rightarrow [5] = [1] - [3] \\ (x_2 - x_3)x + (y_2 - y_3)y + (z_2 - z_3)z = \frac{w_3 - w_2}{2} \rightarrow [6] = [2] - [3] \end{cases} \tag{12}$$

From [4] – [5]

$$x = a_1z + b_1 \tag{7}$$

$$a_1 = \frac{1}{d} [(z_2 - z_1)(y_3 - y_1) - (z_3 - z_2)(y_2 - y_1)] \tag{14}$$

$$b_1 = -\frac{1}{2d} [(w_2 - w_1)(y_3 - y_1) - (w_3 - w_2)(y_2 - y_1)] \tag{15}$$

$$d = (y_2 - y_1)x_3 - (y_3 - y_1)x_2 \tag{16}$$

$$y = a_2z + b_2 \tag{8}$$

$$a_2 = \frac{-1}{d} [(z_2 - z_1)(x_3) - (z_3 - z_1)(x_2)] \tag{18}$$

$$b_2 = -\frac{1}{2d} [(w_2 - w_1)(x_3) - (w_3 - w_2)(x_2)] \tag{19}$$

Solving the last Equation and calculating for x_0 and y_0 from Equations [7] and [8]

Solving Equation Substituting for [7] and [8] in [1]

$$(a_1^2 + a_2^2)z^2 + 2(a_1 + a_2(b_2 - y) - z_1)z + (b_1^2 + (b_2 - y_1)^2 + z_1^2 - re^2) = 0 \tag{20}$$

2.4.2. Parallel Robot Forward Kinematic

Inverse kinematics, in case the location of the end-effector (X, Y, Z) is known [2-13], needs to find the common angles θ_1 , θ_2 and θ_3 respectively. Calculating the inverse kinetics of a 3-axis parallel robot with a delta is necessary. First, it is necessary to determine the parameters of the robot

to be used for the calculations in the mathematical equations. Then determine the side length of the upper triangle is f , the side length of the lower triangle is e , the upper arm length is rf , and the lower arm length is re .

$$E = (x_0, y_0, z_0) \tag{21}$$

$$EE_1 = \frac{e}{2} \tan(30^\circ) = \frac{e}{2\sqrt{3}} \tag{22}$$

$$E_1E'_1 = x_0 \begin{cases} E_1 = (x_0, y_0 - \frac{e}{2\sqrt{3}}, z_0) \\ E'_1 = (0, y_0 - \frac{e}{2\sqrt{3}}, z_0) \end{cases} \tag{23}$$

$$F_1 = x_0, \frac{f}{2\sqrt{3}}, 0 \tag{24}$$

$$E'_1J_1 = \sqrt{(E_1J_1)^2 - (E_1E'_1)^2} = \sqrt{re^2 - x_0^2} \tag{25}$$

$$\begin{cases} (y_{J1} - y_{F1})^2 + (z_{J1} - z_{F1})^2 = rf^2 \\ (y_{J1} - y_{E'_1})^2 + (z_{J1} - z_{E'_1})^2 = re^2 - x_0^2 \end{cases} \tag{26}$$

$$\begin{cases} (y_{J1} - \frac{f}{2\sqrt{3}})^2 + (z_{J1})^2 = rf^2 \\ (y_{J1} - y_0 + \frac{e}{2\sqrt{3}})^2 + (z_{J1} - z_0)^2 = re^2 - x_0^2 \end{cases} \tag{27}$$

$$J_1 = -(0, y_{J1}, z_{J1}) = f(rf, re, f, e, x_0, y_0, z_0) \tag{28}$$

$$\theta_1 = \tan^{-1}\left(\frac{z_{J1}}{y_{F1} - y_{J1}}\right) \tag{29}$$

$$\theta_2 y \theta_3 \rightarrow E_3(x_0, y_0, z_0) \rightarrow E_{0'}(x_0, y_0', z_0') \tag{30}$$

$$\begin{cases} X'_0 = (X_0 \cos(\pm 120^\circ)) + (Y_0 \sin(\pm 120^\circ)) \\ Y'_0 = (X_0 \sin(\pm 120^\circ)) + (Y_0 \cos(\pm 120^\circ)) \\ Z'_0 = Z_0 \end{cases} \tag{31}$$

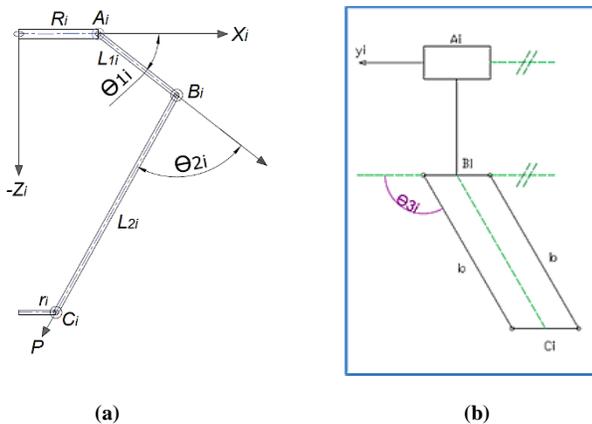


Fig. 9 The projection of the parallel robot kinematic

Parallel robots move on the $x_i z_i$ plane [2,5,8,9,10,11] and

help determine kinetic variables. Fig.9 (b) shows the end view. It is the point where the limb is attached to the platform and corresponds to the position of the actuator P is the center of the moving platform. Two convenient Cartesian coordinate frames, xyz , and $x_i y_i z_i$ are defined in such a way that the xy -plane and the $x_i y_i$ plane are identical and correspond to the plane of the mounting platform. The z and z_i axes are perpendicular to the upper plane. So it's the same. The angle between the x -axis, i.e., Ox , and the x_i -axis, i.e., Ox_i , OA_i , or $A_i x_i$ is ϕ_i . OA_i is always parallel to the PC_i , due to the pivot joint at A_i , and $A_i B_i$ moves in the $x_i z_i$ -plane, where B_i is a passive spherical joint. Thus, $B_i C_i$ can have components along all three axes: $A_i x_i$, $A_i y_i$, and $A_i z_i$. θ_{1i} is the angle between $A_i B_i$ and $A_i x_i$; θ_{2i} is the angle between $A_i B_i$ and the line formed by the intersection of the $x_i z_i$ plane, and the parallelogram at θ_{3i} is the angle between $B_i C_i$ and $A_i y_i$. As a result, $b \cos \theta_{3i}$ it is a component of $B_i C_i$ parallel to $A_i y_i$ and $b \sin \theta_{3i}$ is a component of $B_i C_i$ in the $x_i z_i$ -plane.

The most relevant loop for the desired Jacobian analysis should be selected, let $\vec{\theta}$ be the vector composed of the common excitatory variable and \vec{p} is the position vector of the travelling plate. Then

$$\vec{\theta} = \theta_{1i} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix}, \vec{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{32}$$

The Jacobean matrix is obtained by differentiating the appropriate loop-closing equation and rearranging the results in the dependent form (32).

$$J_\theta = \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix}, \dot{\mathbf{p}} = \begin{bmatrix} \dot{p}_x = v_x \\ \dot{p}_y = v_y \\ \dot{p}_z = v_z \end{bmatrix} \tag{33}$$

Where x , y , and z of the velocity of point P on the platform moving in the frame x, y, z have v_x, v_y , and v_z components. And the equations are derived from the loop $OA_i B_i C_i P$. The corresponding closure equation in the $x_i y_i z_i$ frame is The equation corresponding to the $x_i y_i z_i$ plan is

$$\vec{OP} + \vec{PC}_i = \vec{OA}_i + \vec{A_i B_i} + \vec{B_i C_i} \tag{34}$$

The corresponding transformation matrix is written as:

$$\begin{bmatrix} p_x \cos \phi_i - p_y \sin \phi_i \\ p_x \cos \phi_i - p_y \cos \phi_i \\ p_z \end{bmatrix} = \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} + a \begin{bmatrix} \cos \theta_{1i} \\ 0 \\ \sin \theta_{1i} \end{bmatrix} + b \begin{bmatrix} \sin \theta_{3i} \cos(\theta_{2i} + \theta_{1i}) \\ \cos \theta_{3i} \\ \sin \theta_{3i} \sin(\theta_{2i} + \theta_{1i}) \end{bmatrix} \tag{35}$$

Table 3. Parallel robot configuration parameter for workspace simulation

Parameter Name	Symbol	Value (mm)
Length of the upper link	rf	200
Length of the lower link	re	500
Distance between the center of the base and the center of the motor	OF	160
Distance between center of travelling plate and ball joint	OE	75

3. Results and Discussion

3.1. Workspace Analysis Simulation Base on MATLAB

Robot kinematic simulations are done to determine the limitations of robot motion and to determine the suitable working area for the prototype of the parallel robot using MATLAB Simulink [14-16]; it is necessary to know the length of the main components of the parallel robot: the distance between the centre of the base and the centre of the motor (OF), the length of upper link (rf), the length of the lower link (re), and the distance between the centre of traveling plate and ball joint (OE). As shown in Table 3. and Fig.10, there are main component parameters that relate to the movement of the robot arm.

In this section, the simulation using inverse kinematics is performed in four scenarios to prove the limitation of the prototype parallel robot's motion.

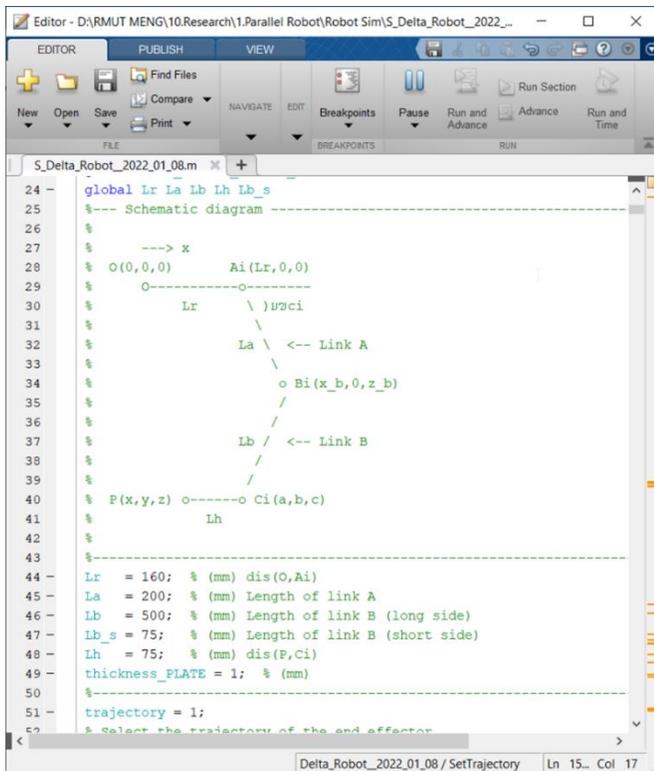


Fig. 10 Parallel-Robot Workspace trajectory tracking simulation parameter for $[OF, rf, re, OE] = [160, 200, 500, 75]$ (mm)

3.1.1. Scenario 1, inverse kinematics simulation for $z1$

The sliding of the robot is simulated to find the limitation of the robot's space of motion at the lowest point. $z1$ by determining the distance of the robotic arm's end in the $z1$ axis to a distance greater than the maximum distance of the robot's kinematic calculation [$z1 = -200, z2 = 150, R = 250$] by setting [$z1 = -210, z2 = 150, R = 250$].

From Fig.12, the 3D model shows that when setting $z1 = -210$ mm, which is a value greater than the calculated kinematic value = -200 mm, and when using the RUN command, it lets the MATLAB Simulation program run. The movement of the robot arm is simulated to move in the $z1$ axis from the -411 mm position to the -615 mm position. The 3D Model program shows the message "Out of workspace!" comparing the positions of the graphs in Fig.13 from the start of the z -axis, the value = -411 mm. Thus, it can be seen that the distance traveled from the start point of the z -axis to the statement of the problem point is $z1 = -204$ mm. the error position is shown in table 4.

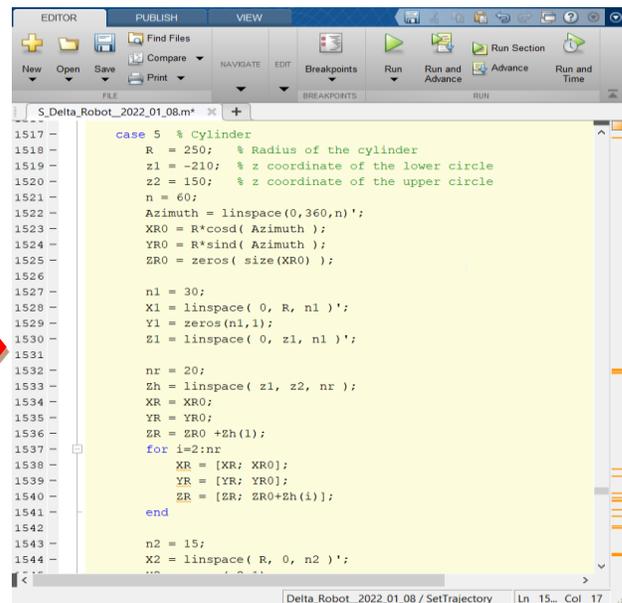


Fig. 11 The Parallel-Robot parameter for simulation and analyzed limit lower-level workspace for $[z1, z2, R] = [-210,150,250]$ (mm)

Table 4. Scenario 1, the limit workspace simulation result

Parameter Name	Statement of problem point	
	End-Effector Distance from HP	Position (mm)
$x,$	241.5	241.5
$y,$	0	0
$z1,$	-615	204
Parameter Name	Joint Angles (degrees)	
θ_1		34.8°
θ_2		95.8°
θ_3		95.8°

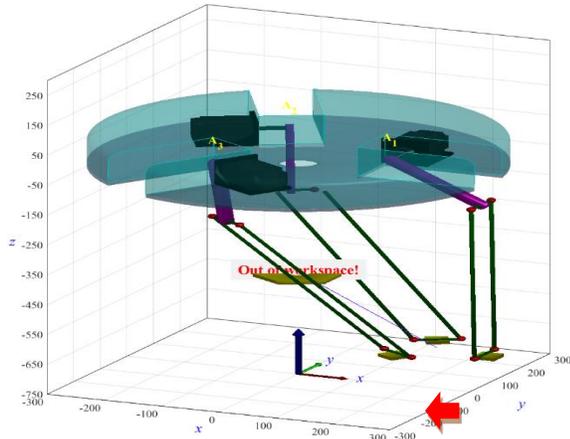


Fig. 12 3D Model simulation of Maximize lower-level limit lower-level workspace trajectory tracking for $[z_1, z_2, R] = [-210, 150, 250]$ (mm)

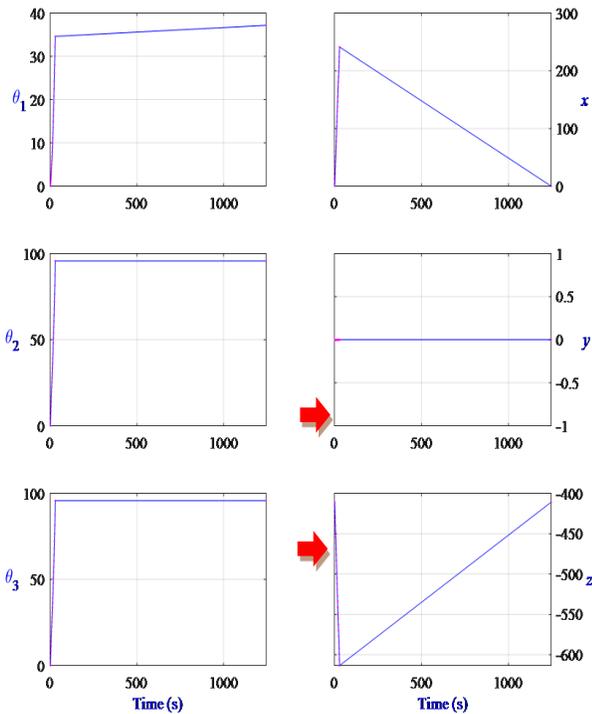


Fig. 13 Maximize lower-level workspace trajectory tracking graphs for $[z_1, z_2, R] = [-210, 150, 250]$ (mm)

3.1.2. Scenario 2, inverse kinematics simulation for z-2

The sliding of the robot is simulated to find the limitation of the robot's space of motion at the upper point z_2 by determining the distance of the robotic arm's end in the z_2 axis to a distance greater than the maximum distance of the robot's kinematic calculation $[z_1 = -200, z_2 = 150, R = 250]$ by setting $[z_1 = -200, z_2 = 160, R = 250,]$

From Fig.15, The 3D Model, it can be seen that when setting $z_2 = 160$ mm, which is a value greater than the

calculated kinematic value = 150 mm, and when using the RUN command, it lets the MATLAB Simulation program run. The movement of the robot arm is simulated to move in the z_2 axis to the -252 mm position. The 3D Model program shows the message "Out of workspace!" comparing the positions of the graphs in Fig.16 from the start of the z -axis, the value = -411mm. Thus, it can be seen that the distance traveled from the start point of the z -axis to the statement of the problem point is $z_2 = 159$ mm. the error position is shown in table 5.

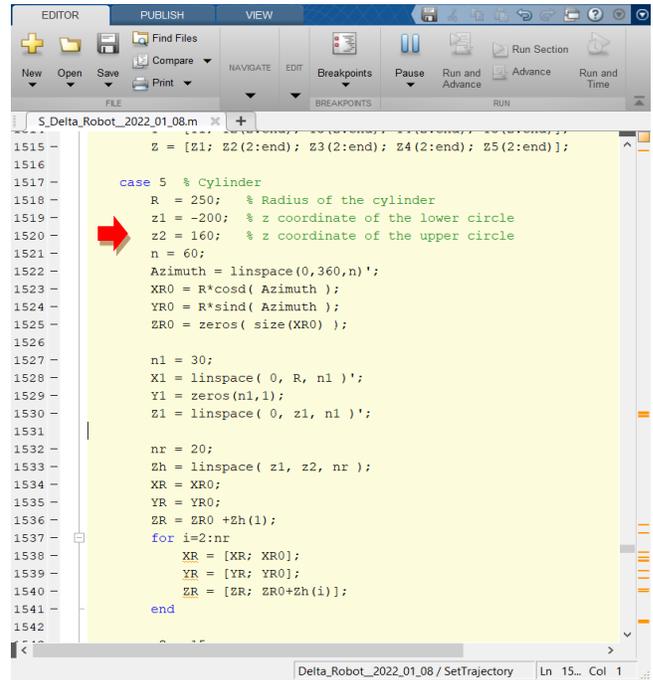


Fig. 14 The Parallel-Robot parameter for simulation and analyzed limit Upper-level workspace for $[z_1, z_2, R] = [-200, 160, 250]$ (mm)

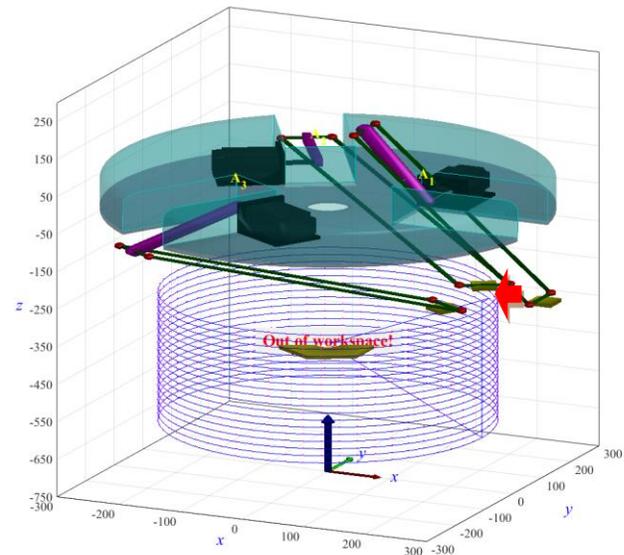


Fig. 15 3D Model simulation of Maximize the Upper-level limit workspace trajectory tracking for $[z_1, z_2, R] = [-200, 160, 250]$ (mm)

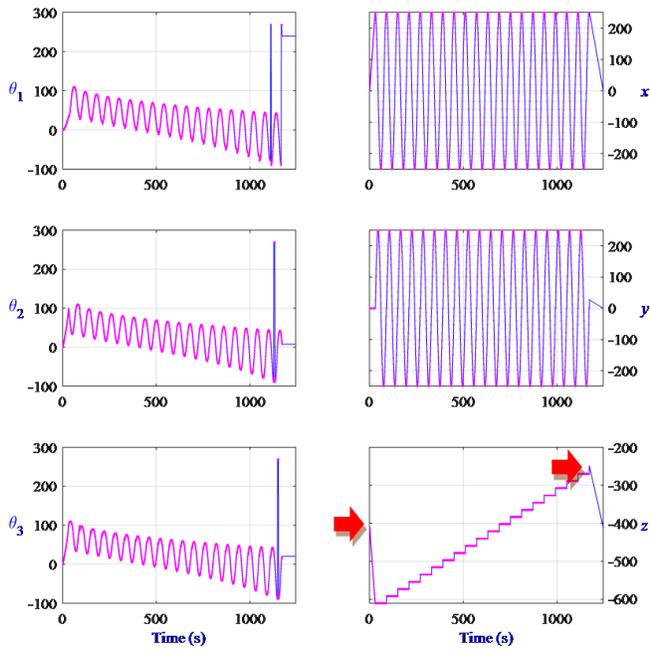


Fig. 16 Maximize the Upper-level workspace trajectory tracking graphs for $[z1, z2, R] = [-200, 160, 250]$ (mm)

Table 5. Scenario 2, the limit workspace simulation result

Parameter Name	Statement of problem point	
	End-Effector Position	Distance from HP (mm)
$x,$	250	250
$y,$	27	27
$z2,$	-252	159
Parameter Name	Joint Angles (degrees)	
θ_1	-90°	
θ_2	8°	
θ_3	20°	

3.1.3. Scenario 3, inverse kinematics simulation for R (x-axis and y-axis)

The sliding of the robot is simulated to find the limitation of the robot's space of motion at the radius of the cylinder point, R (limit Radius of the x-axis and y-axis workspace) by determining the distance of the robotic arm's end in the x-axis to a distance greater than the maximum distance of the robot's kinematic calculation $[z1 = -200, z2 = 150, R = 250]$ by setting $[z1 = -200, z2 = 150, R = 260]$

```

case 5 % Cylinder
R = 260; % Radius of the cylinder
z1 = -200; % z coordinate of the lower circle
z2 = 150; % z coordinate of the upper circle
n = 60;
Azimuth = linspace(0,360,n)';
XR0 = R*cosd( Azimuth );
YR0 = R*sind( Azimuth );
ZR0 = zeros( size(XR0) );

n1 = 30;
X1 = linspace( 0, R, n1 )';
Y1 = zeros(n1,1);
Z1 = linspace( 0, z1, n1 )';

nr = 20;
Zh = linspace( z1, z2, nr );
XR = XR0;
YR = YR0;
ZR = ZR0 +Zh(1);
for i=2:nr
    XR = [XR; XR0];
    YR = [YR; YR0];
    ZR = [ZR; ZR0+Zh(i)];
end

n2 = 15;
X2 = linspace( R, 0, n2 )';

```

Fig. 17 The Parallel-Robot parameter for simulation and analyzed limit radius of the x-axis and y-axis workspace for $[z1, z2, R] = [-200,160,250]$ (mm)

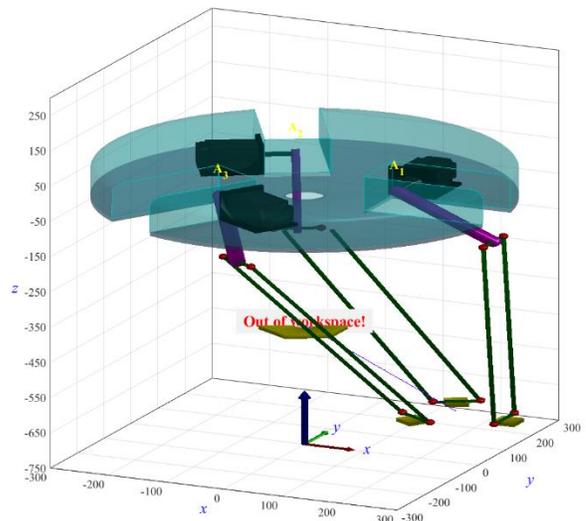


Fig. 18 3D Model simulation of Maximize the limit radius of the x-axis, and y-axis workspace trajectory tracking for $[z1, z2, R] = [-200, 160, 250]$ (mm)

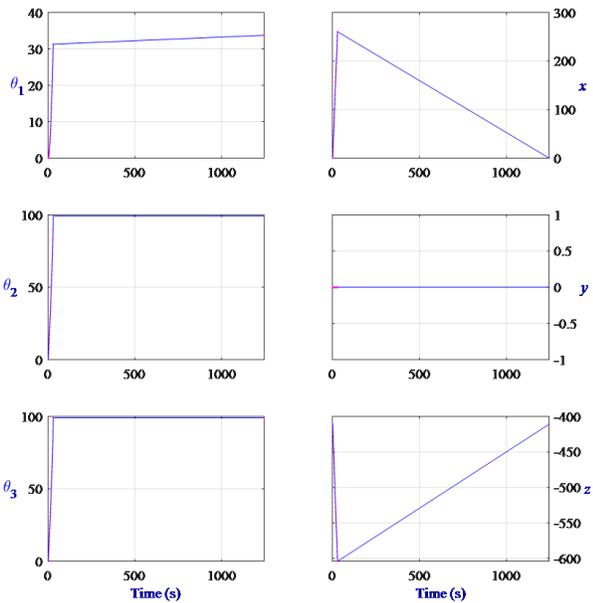


Fig. 19 Maximize the radius of x-axis, and y-axis workspace trajectory tracking graphs for [z1, z2, R] = [-200,160,250] (mm)

From Fig.18, The 3D Model, it can be seen that when setting R = 260 mm, which is a value greater than the calculated kinematic value = 250 mm, and when using the RUN command, it lets the MATLAB Simulation program run. Simulate the movement of the robot arm to move in the x-axis to the = 250 mm position. The 3D Model program shows the message “Out of workspace!” comparing the positions of the graphs in Fig.16 from the start of the x-axis, the value = 250 mm. Thus, it can be seen that the radius distance traveled from the start point of the x-axis to the statement of problem point $x_1 = 250$ mm. the error position is shown in table 6.

Table 6. Scenario 1, the limit workspace simulation result

Parameter Name	Statement of problem point	
	End-Effector Position	Distance from HP (mm)
x,	250	250
y,	0	0
z,	-615	204
Parameter Name	Joint Angles (degrees)	
θ_1	31.3°	
θ_2	94°	
θ_3	94°	

3.1.4. Scenario 4, inverse kinematics simulation for the x-axis, y-axis, and z-axis

The sliding of the robot is simulated to find the limitation of the robot's space of motion at the x-axis, y-axis, z-axis which is same as the maximum distance of the robot's

kinematic calculation [z1 = -200, z2 = 150, R = 250] by setting [z1 = -200, z2 = 150, R = 250].

```

EDITOR PUBLISH VIEW
+ Find Files
+ Compare
+ Print
+ Breakpoints
+ Run
+ Run and Advance
+ Advance
+ Run and Time

S_Delta_Robot_2022_01_08.m
% Cylinder
R = 250; % Radius of the cylinder
z1 = -200; % z coordinate of the lower circle
z2 = 150; % z coordinate of the upper circle
n = 60;
Azimuth = linspace(0,360,n)';
XR0 = R*cosd(Azimuth);
YR0 = R*sind(Azimuth);
ZR0 = zeros(size(XR0));
n1 = 30;
X1 = linspace(0, R, n1)';
Y1 = zeros(n1,1);
Z1 = linspace(0, z1, n1)';
nr = 20;
Zh = linspace(z1, z2, nr);
XR = XR0;
YR = YR0;
ZR = ZR0 + Zh(1);
for i=2:nr
    XR = [XR; XR0];
    YR = [YR; YR0];
    ZR = [ZR; ZR0+Zh(i)];
end
n2 = 15;
X2 = linspace(R, 0, n2)';

```

Fig. 20 The Parallel-Robot parameter for simulation workspace limit of x-axis, y-axis and z-axis workspace for [z1, z2, R] = [-200, 150, 250](mm)

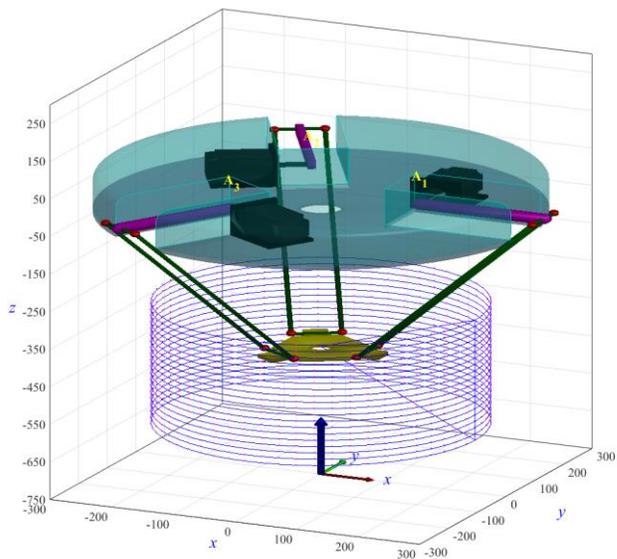


Fig. 21 The 3D Model workspace of simulation of Parallel robot for [z1, z2, R] = [-200, 150, 250] (mm)

From the 3D Model in Fig. 21 and the trajectory tracking graphs in Fig.22; it can be seen that when setting z1=-200mm, z2=150mm, R = 250 mm, which is a value same as the calculated kinematic value, and when using the RUN command, it let MATLAB Simulation program run.

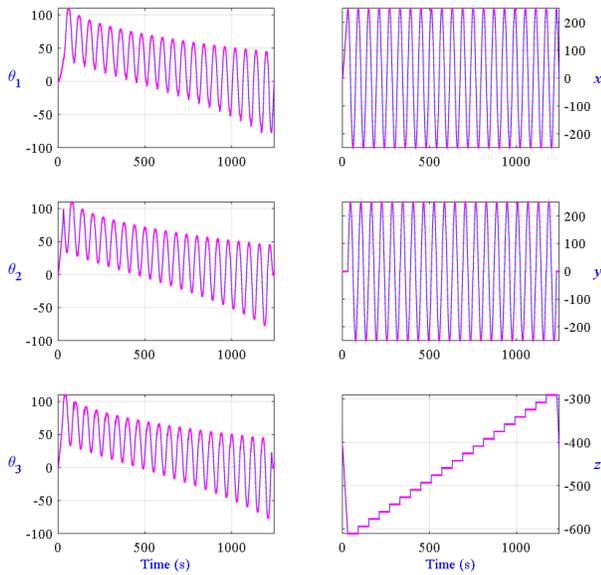


Fig. 22 The radius of x-axis, y-axis and z-axis workspace trajectory tracking graphs for $[z1, z2, R] = [-200,150,250]$ (mm)

The movement of the robot arm is simulated to move in the process that can work until the end of the process. From the trajectory tracking graphs in Fig.22, the parallel robot kinematic simulation based on MATLAB Simulink makes it possible to know the form of space constraints of the prototype parallel robot in all the workspace dimensions and summarize the values according to Table 7.

Table 7. The Parallel robot prototype limit workspace simulation result

Parameter Name	Value (mm) from Home position	
	Min.	Max.
$x,$	-250	250
$y,$	-250	250
$z, (z1, z2)$	-200	150
Parameter Name	Joint Angles (degrees)	
	Robotic Arm orientation	
θ_1	110°	-85°
θ_2	110°	-85°
θ_3	110°	-85°

3.2. Prototype Parallel Robot workspace and Trajectory Tracking Experiment

Using an experimental parallel robot prototype, it will test for results of the position of the robot arm joint angles movement by using the principle of inverse kinematics to know the movement characteristics of the robotic arm in each movement position by a joint angles measured sensor mounted on the three motor axis. It measures the joint angles position of the motors ($\theta_1, \theta_2,$ and θ_3) and signals the joint angles position of motors measured to the microcontroller. The result of the motor angle position signal obtained can be viewed from the serial monitor program page of the program

C++ ID. The program page can compare the $\theta_1, \theta_2,$ and θ_3 using the signal received from the sensor with the joint angles value of the motor, which is defined according to the experimental conditions in graph form. In this experiment, the signal data obtained from the serial monitor map is graphed in the Microsoft Excel program to see clear signal patterns and the experimental procedure according to Fig. 23.

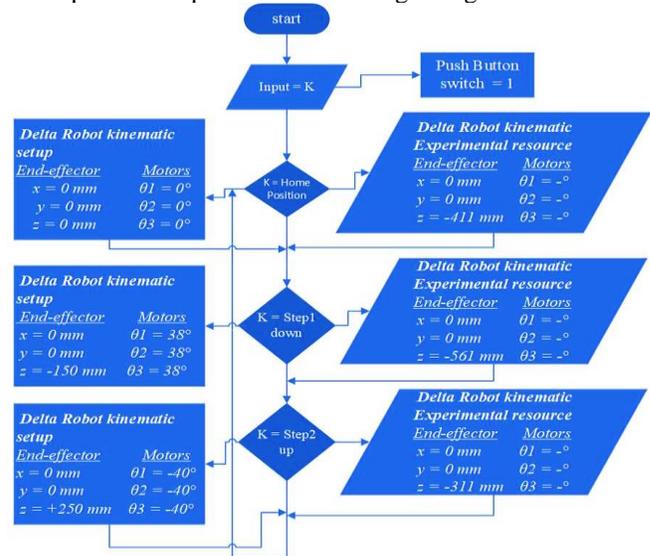


Fig. 23 Workflow step of Parallel Robot kinematic experimental process

The experiment is done by a prototyped parallel robot using principles of inverse kinematics to see the efficiency and results of the joint angles motion pattern of the prototype parallel robot arm according to the experimental procedure shown in Figure. 23. After pressing the push button switch, the number of robot joint angles positions are set in 3 steps in 1 loop. The total number of experiments will be 5 rounds, as shown in Table. 6, and the position of the robot arm and the motor angle are defined for each step which was obtained from the kinetic simulation by the MATLAB Simulink program in Fig.24 and Fig.25.

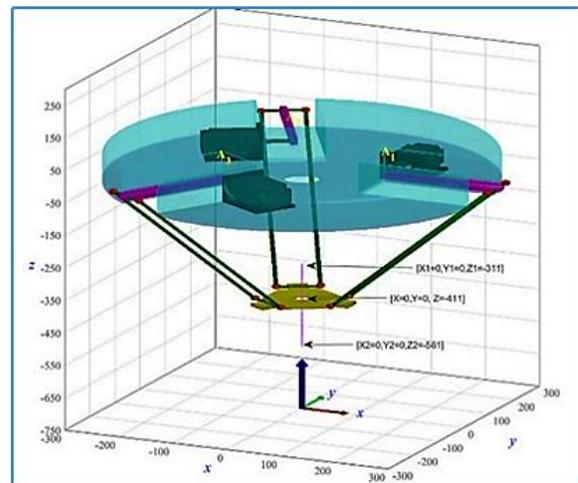


Fig. 24 The 3D Model simulation results from inverse kinematics for $[x, y, z] = [0, 0, -411], [0, 0, -561], [0, 0, -311]$.

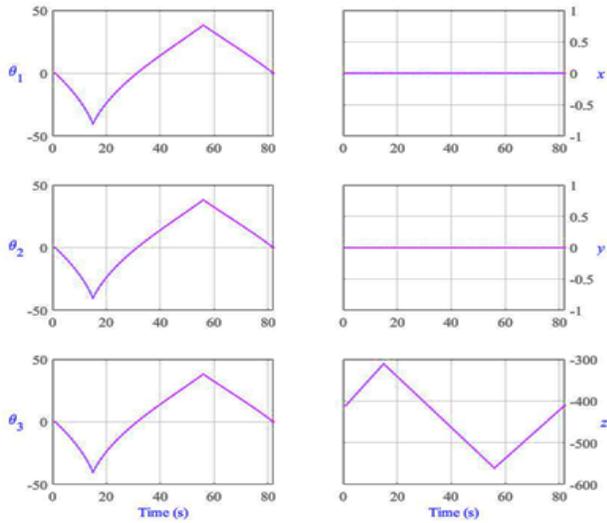


Fig. 25 Parallel Trajectory Tracking results from inverse kinematics simulation for [x, y, z] = [0, 0, -411], [0, 0, -561], [0, 0, -311]

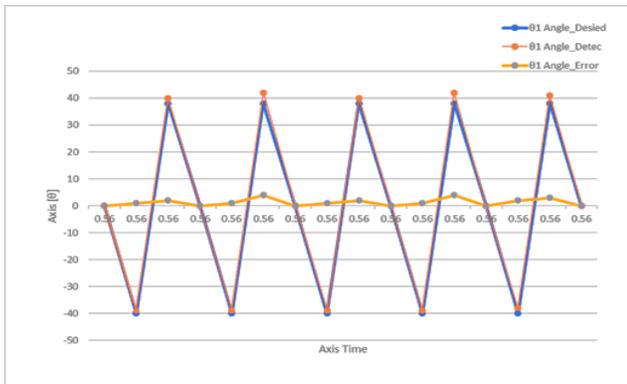


Fig. 26 Trajectory Tracking experiment results of 01

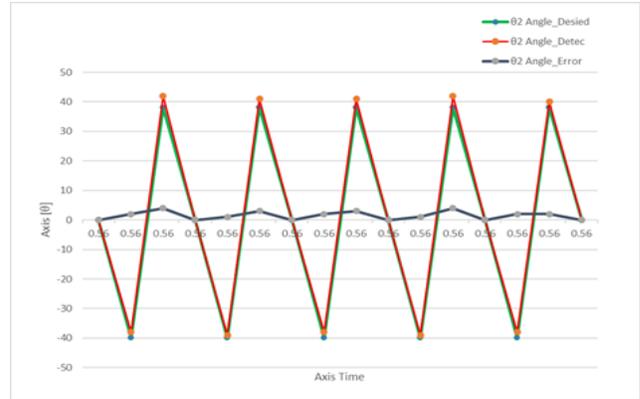


Fig. 27 Trajectory Tracking experiment results of 02

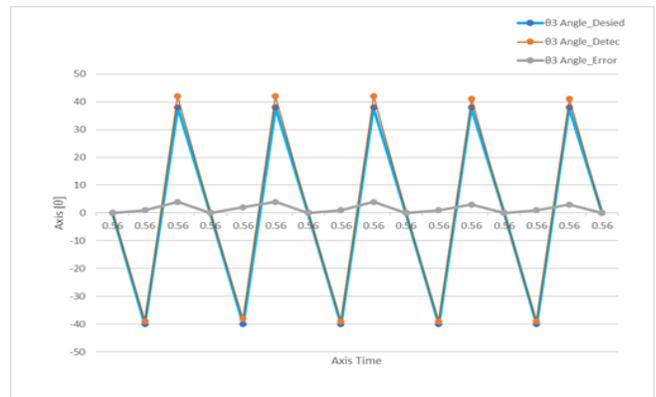


Fig. 28 Trajectory Tracking experiment results of 03

From Fig. 24 and Fig.25, The simulation result shows that the movement of the robot using the developed program from the MATLAB/Simulink deriving the working area of the parallel robot; $x = 0$ mm, $y = 0$ mm, $z = -150$ to 100 mm, and $\theta_1 = 38^\circ$ to $3-40^\circ$, $\theta_2 = -38^\circ$ to -40° , $\theta_3 = -38^\circ$ to -40° .

Table 8. Experiment condition and results for the Parallel robot workspace analysis

Experiment Step	⁰¹ Angle_Desied	⁰² Angle_Desied	⁰³ Angle_Desied	⁰¹ Angle_Detec	⁰² Angle_Detec	⁰³ Angle_Detec	⁰¹ Angle_Error	⁰² Angle_Error	⁰³ Angle_Error
Home Position	0	0	0	0	0	0	0	0	0
Step 1	-40	-40	-40	-39	-38	-39	1	2	1
Step 2	38	38	38	40	42	42	2	4	4
Step 3	0	0	0	0	0	0	0	0	0
Step 4	-40	-40	-40	-39	-39	-38	1	1	2
Step 5	38	38	38	42	41	42	4	3	4
Step 6	0	0	0	0	0	0	0	0	0
Step 7	-40	-40	-40	-39	-38	-39	1	2	1
Step 8	38	38	38	40	41	42	2	3	4
Step 9	0	0	0	0	0	0	0	0	0
Step 10	-40	-40	-40	-39	-39	-39	1	1	1
Step 11	38	38	38	42	42	41	4	4	3
Step 12	0	0	0	0	0	0	0	0	0
Step 13	-40	-40	-40	-38	-38	-39	2	2	1
Step 14	38	38	38	41	40	41	3	2	3
Home Position	0	0	0	0	0	0	0	0	0

Based on the end-effector position manipulation results, the position discrepancy arises because the robot has multiple joints moving while controlled. And the control system does not consider the inertia of the robot. Controlling the inertia of a robot is a challenge. The robot's inertia constantly changes when the end effector is placed in different positions.

The result from 3 steps experiment with the inverse kinematic method; is the position of the robot's arm joint angle θ_1 , θ_2 , and θ_3 ; the signal is received from the angle measured sensor. From the graphs in Fig. 26, Fig.27, Fig. 28, and Table 4, the error that occurred in each step exhibits the highest value at 4° .

4. Conclusion

In designing a robot prototype, selecting a robot suitable for the type of work area is necessary. Using kinematic calculations against kinematic simulations and the actual experimental results with the parallel prototype robot in Table 8 is slightly inaccurate at some intervals due to various latent

factors. However, the actual results obtained were consistent with the target experiment results obtained from the kinematic simulations. It helps us understand and know how the robot works in every dimension of the joint angle movement in each situation. And can also use that method to develop different types of robots. It is to prove the limitations in the design and selection of robots or to use them to develop existing robots to make them truly function at their best. The next step is to develop the robots to potentially be more efficient by installing additional devices such as a camera for machine vision and sensors for robot traveling plate positioning and actuation. The data acquisition and control of the robot movement are being evaluated to implement the necessary applications for parallel robots.

Acknowledgments

This research was conducted under and was supported by the Mechatronics and Robotics Engineering Department, Faculty of Technical Education, Rajamangala University of Technology Thanyaburi, Pathum Thani, Thailand.

References

- [1] Staicu St. & Carp-Ciocardia D. C, "Dynamic Analysis of Clavel'S Delta Parallel Robot," *Proceedings of the 2003 IEEE International Conference on Robotics & Automation Taipei*, Taiwan, 2003.
- [2] Ohlsson A., "Modeling and Control of a Delta-3 Robot," LUTFD2/TFRT—5834—SE Master Thesis, Lund University, Department of Automatic Control, Sweden, 2009.
- [3] Zsombor-Murray P.J., "Descriptive Geometric Kinematic Analysis of Clavel's "Delta" Robot," McGill University Department of Mechanical Engineering Centre for Intelligent Machines Rm. 454, 817 Sherbrooke St. W.Montr' eal (Qu' ebec) Canada, H3A 2K6, 2004.
- [4] Saeed Rahimi, "Design and Practical Implementation of a Neural Network Self-Tuned Inverse Dynamic Controller for a 3-Dof Delta Parallel Robot Based on Arc Length Function for Smooth Trajectory Tracking," *Mechatronics*, vol. 84, pp. 102772, 2022.
- [5] Hamdoun O., Bakkali L.E. & Baghli F.Z, "Analysis and Optimum Kinematic Design of a Parallel Robot," *10th International Conference Interdisciplinarity in Engineering, INTER-ENG*, 2016.
- [6] Falezza F, "A Novel Inverse Dynamic Model for 3-Dof Delta Robots," *Mechatronics*, vol. 83, pp. 102752, 2022.
- [7] M Lo'pez, Centro de Ingenier' a and Desarrollo Industrial, Quere' taro, Me' xico, "Delta Robot: Inverse, Direct, and Intermediate Jacobians," *Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science 1989-1996*, 2005.
- [8] Robert L, Williams II, Ph.D, "The Delta Parallel Robot: Kinematics Solutions," *Mechanical Engineering*, Ohio University, 2016.
- [9] Hugo Hadfield, Joan Lasenby, "The Forward and Inverse Kinematics of a Delta Robot," *Chapter*, 2020. Doi:10.1007/978-3-030-61864-3_38
- [10] Yun-Joo Nam & Myeong-Kwan Park, "Kinematics and optimization of 2-DOF Parallel Manipulator with Revolute Actuators and a Passive Leg," *Journal of Mechanical Science and Technology*, Article number: 828, vol. 20, 2006.
- [11] C. Gosselin, J. Angeles, "The Optimum Kinematic Design of a Planar Three-Degree-of-Freedom Parallel Manipulator," *Journal of Mechanisms, Transmissions, and Automation*, vol. 110, no. 1, pp. 35-41, 1988.
- [12] Cheng Liua, Guohua Caoa, and Yongyin Qub, "Safety Analysis via Forward Kinematics of Delta Parallel Robot using Machine Learning," *Safety Science*, vol. 117, pp. 243-249, 2019.
- [13] M. Shehata, M. Elshami, Q. Bai, X. Zhao, "Parameter Estimation for Multibody System Dynamic Model of Delta Robot from Experimental Data," *IFAC Papers OnLine*, vol. 54-14, pp. 72-77, 2021
- [14] Ma Li, Dexue Bi, Zhipeng Xiao, "Mechanism Simulation and Experiment of 3-DOF Parallel Robot Based on MATLAB," *Proceedings of the 2015 International Power, Electronics and Materials Engineering Conference*, 2015.
- [15] Swaraj Zodeya, Sharad K. Pradhanb, "Matlab Toolbox for Kinematic Analysis and Simulation of Dexterous Robotic Grippers," *12th Global Congress on Manufacturing and Management, GCMM 2014, Procedia Engineering*, vol. 97, pp. 1886-1895, 2014.
- [16] Carmelo Mineo, Stephen Gareth Pierce, Pascual Ian Nicholson, Ian Cooper, "Robotic path Planning for Non-Destructive Testing – A Custom MATLAB Toolbox," *Approach, Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 1-12, 2016.

- [17] Delta X1 Specifications. [Online]. Available: <https://www.deltarobot.com/p/specifications.html>
- [18] Delta Robots from B&R Specifications. [Online]. Available: <https://www.br-automation.com/en/products/machine-centric-robotics/robotics-portfolio/maximum-performance-with-delta-robots/>
- [19] Dinh Tho Long, To Van Binh, Roan Van Hoa, Le Van Anh, Nguyen Van Toan, "Robotic Arm Simulation by using Matlab and Robotics Toolbox for Industry Application," *SSRG International Journal of Electronics and Communication Engineering*, vol. 7, no. 10, pp. 1-4, 2020. Crossref, <https://doi.org/10.14445/23488549/IJECE-V7I10P101>
- [20] Indrazno Siradjuddin, Aang Junaidi, Ratna Ika Putri, Erfan Rohadi, Supriatna Adhisuwignjo, "Kinematics and Control A Three Wheeled Omnidirectional Mobile Robot," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 6, no. 12, pp. 1-6, 2019. Crossref, <https://doi.org/10.14445/23488379/IJEEE-V6I12P101>
- [21] D.Deepak, S.Pathmasharma, "Design and Fabrication of Kinematic Robotic Walker with Left and Right Motion with Camera," *SSRG International Journal of Mechanical Engineering*, vol. 4, no. 4, pp. 53-56, 2017. Crossref, <https://doi.org/10.14445/23488360/IJME-V4I4P112>
- [22] Ngoc Trung Dang and Xuan Thuan Nguyen, "A Control Method for SMMS Teleoperation System with Affection of Disturbance," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 5, no. 11, pp. 1-8, 2018. Crossref, <https://doi.org/10.14445/23488379/IJEEE-V5I11P101>
- [23] Roan Van Hoa, Nguyen Duc Dien, Lai Khac Lai, "Research and Apply Deep Reinforcement Learning Technology to Control Mobile Robot," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 8, no. 4, pp. 30-35, 2021. Crossref, <https://doi.org/10.14445/23488379/IJEEE-V8I4P106>
- [24] Wen-Kung Tseng, Hou-Yu Chen, "The study of tracking control for autonomous vehicle," *SSRG International Journal of Mechanical Engineering*, vol. 7, no. 11, pp. 57-62, 2020. Crossref, <https://doi.org/10.14445/23488360/IJME-V7I11P108>
- [25] Ngo Phuong Thanh, Nguyen Nam Trung, "The Fan and Plate Angular Position Control Techniques and Applications," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 8, no. 6, pp. 38-42, 2021. Crossref, <https://doi.org/10.14445/23488379/IJEEE-V8I6P106>