ICMIEE20-195

Design, Simulation and Workspace Analysis of a 5 DOF Robotic Arm

Faisal Ahmed Shanta[*], *Md. Helal-An-Nahiyan*
Department of Mechanical Engineering, Khulna University of Engineering & Technology, Khulna-9203, BANGLADESH

ABSTRACT
This paper presents the design, simulation, and workspace analysis of a 5 DOF multi-purpose robotic manipulator. This manipulator is specially developed for a Mars Rover and the tasks to be executed by using this manipulator are screw tightening, pressing a key on the keyboard, weight carrying, etc. To solve the kinematics of this manipulator, Denavit Hartenberg (DH) method is used; and to obtain the desired position and orientation of end-effector, inverse kinematics model is used. After solving the equations generated from the Forward Kinematics model, the Inverse Kinematics model is implemented, which provides the rotation and orientation of each link of the manipulator. The physical structure of the manipulator is constructed with steel and aluminum plates of different thicknesses. The position and orientation of the end effector is controlled by using a microcontroller based on the parameters computed from the kinematic models. Motion of the links is operated by using DC gear motors and linear actuators. The microcontroller can be controlled remotely by the user, thus the manipulator. The kinematic models are simulated by Python programming language, which simulates the position and orientation of the end effector in the 3D coordinate system. From the simulation results, the workspace volume, minimum and maximum distance of the end effector from the base point are calculated. All the kinematics model and simulation process presented in this work is also applicable for a similar type of manipulator having different types of variables.

Keywords: 5 DOF, Manipulator Kinematics, Robotic Simulation, Workspace Analysis.

1. Introduction

Nowadays, University-based Rover competitions like URC (University rover challenge), ERC (European Rover Challenge), etc. have been very popular among young students. In these competitions, students perform their designed rover in a demo environment of Mars on the earth's surface. The actual Mars Rover has to perform several tasks in the Martian environment. Some of the tasks are conducted by the manipulator of the rover. For example, rock sample collecting, drilling, pick and place, etc. According to the competition criteria, students from all over the world compete over tasks that are performed by the real rover on Mars. So different rovers are designed and fabricated by the students to perform the tasks in the competition.

To fulfill the tasks of the competitions, a multi DOF (Degree of freedom) manipulator is very important. As the rover will be maneuvered on non-uniform surfaces in a very challenging environment. So, for a better result, a 5 DOF manipulator is designed to conduct several tasks, for example, screw tightening, carrying dead weight, pick and place a small-sized object, pressing a key of a computer keyboard, opening and closing a drawer, holding a screwdriver, etc.

A multipurpose end effector is proposed in this paper having the capability of independent rotation. In [1] Gosselin et al. proposed a parallel mechanism for 360-degree rotation of the end effector using a slip ring for continuous power transmission. An almost similar type of mechanism is used here where a high torque stepper motor is coupled with the end effector for the independent rotation. The three jaws of the end effector will help to hold objects of different sizes and shapes. Thus, the rotating and gripping features are combined in the end effector. Three linear actuators are used to rotate the links, and one of them is hinged at the base frame.

In case of autonomous and semi-autonomous tasks, the precise controlling of the manipulator by implementing kinematics models provide a better result. The kinematic model of the manipulator focuses on the motion of each link without considering the force causing the motion [2]. The complexity of a manipulator increases with the increase of DOF, each DOF indicates an independent variable required to specify the location of one link with respect to the previous link [3]. Higher DOF manipulator comes up with more complex kinematics. In that case, the Forward Kinematics model is developed by using the DH parameters. The orientation and position of the end effector are computed by multiplying all the transformation matrix formed by each joint parameter. Then, by solving the equations of the matrix, the value of each variable can be found, which is called Inverse Kinematics. There are two types of solutions for the inverse kinematics of a manipulator which are closed from solution and numerical solution. As there are successive planner joints in this designed manipulator, a closed-form method is used [4]. In some cases, the geometrical solution is also used in this paper.

The kinematic models presented in this paper can be used to design and analyze other multi DOF manipulators. In case of inverse kinematics modeling, the position and orientation of the end effector are provided by the user or by any feedback system. Then, the inverse kinematics took place to obtain the link angle of each link of the manipulator. Once the equation is solved, the controlling system finds out the value of the variables of all respective links at the respective coordinates.

The kinematics models are simulated and verified using Python Programming Language. The joint angles

* Corresponding author. Tel.: +88-01910510673
E-mail addresses: faisal.me16@gmail.com

for a coordinate are found from the equation of Inverse Kinematics Modeling. On the other hand, to obtain a desired position by the end effector, the joints angles are given in Forward Kinematics modeling. The Python Program provides visual graphics of the end effector positions by using these models. Therefore, the design and development of a manipulator come up with so many constraints, variables, limitations, and complexity. Designing a manipulator is a combination of mechanical design, mathematical modeling, kinematics and dynamics, control system, and so on.

The link length and DOF combinedly provide a certain workspace, which defines all the possible coordinate positions where the end effector can reach [5]. The large volume of workspace allows the manipulator to work in different orientations and positions in the assigned coordinates. Here, the workspace is simulated by Python Programming Language in an IDE (Integrated Development Environment).

2. Description and Kinematic Modeling of the manipulator

2.1 Structural Description

The robotic manipulator is designed with 5 revolute joints, where 3 of them are planner revolute joints, one revolute joint is in the basement, and another revolute joint is at the end effector. The planner joints are regulated with 3 linear actuators and the other 2 joints are actuated by two gear motors. The Computer-Aided Modeling (CAD) of the manipulator is done by SolidWorks 2019 software which is shown in Fig.1.

The rigid structure between two consecutive joints is known as a link. The last member of the link is called End Effector. In this manipulator, the end effector is capable of rotating along its axis. Forward Kinematics makes a relation between the constraints and variables for a certain rotation and position of the end effector from the base point to the end effector. Assume that, three-planner links have the length of *l, m, n,* respectively. All the link lengths of this manipulator are constant. Joint angles are represented by $\theta_1,\ \theta_2,\ \theta_3,\ \theta_4,\ \theta_5$, respectively. Linear actuators guide the links to rotate within a range of angles for each joint, which is denoted as Range of Motion (ROM). The structural and geometrical features of the manipulator are shown in Table 1.

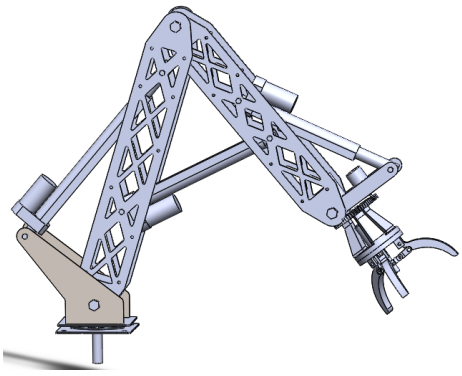

Fig.1 CAD model of the manipulator.

Table 1 Structural and geometric features of the manipulator.

Feature	Description
Load Capacity	5Kg
Weight	20Kg (Approximately)
Range of motion (ROM)	$\theta_1 = 30° \; to \; 150°$ $\theta_2 = 22.6° \; to \; 100°$ $\theta_3 = -88.8° \; to \; -130.8°$ $\theta_4 = 0° \; to \; 360°$ $\theta_5 = 31.8° \; to \; -34.9°$

2.2 Forward Kinematics Modeling

All the five joints of the designed 5 DOF robotic manipulator are revolute joints. Where the first revolute joint let the whole manipulator to rotate with respect to the base frame and the last revolute joint let the end effector to rotate along its axis. The other three joints are planner revolute joints.

There are two common methods for kinematics modeling such as Denavit-Hartenberg (DH) parameter and the successive screw displacement method. These methods are suitable for multi-linked manipulators [6]. Geometric methods are also used in research fields as it is simple to solve [7]. Here, DH parameters are used as it is convenient for its simplicity and diversity. Moreover, this method is much more expedient for multi-linked manipulators in terms of calculation complexity. The link length of the manipulator is shown in Table 2.

Four parameters are considered in the DH method. These are: link twist angle (α), link length (a), link offset (d) and joint angle (θ), respectively. Where rotation about the *X*-axis is represented by α_i, transition along *X*-axis by a_i, transition along *Z*-axis by d_i, and rotation about the *Z*-axis by θi, respectively [8]. The kinematic model of the manipulator with coordinate frames, link, and joint parameters is represented by Fig.2. According to the kinematic model, the values of the DH parameters are given in Table 3.

Table 2: Link Length.

Symbol	*l*	*m*	*n*
Link Length (mm)	60	50	20

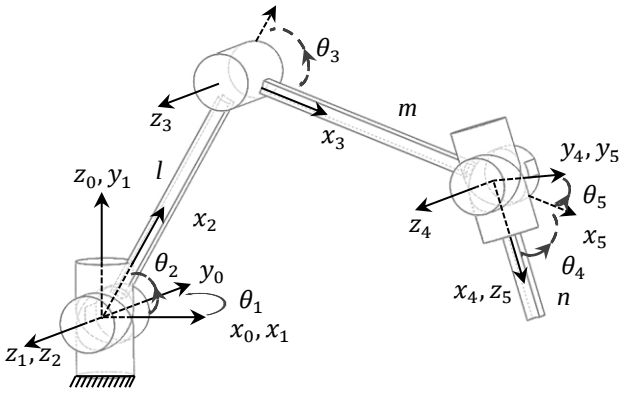

Fig.2 Kinematic model and coordinate distribution.

Table 3 DH parameters of the manipulator.

Link i	*Link Twist* $\boldsymbol{\alpha_{i-1}}$	*Link Length* $\boldsymbol{a_{i-1}}$	*Link Offset* $\boldsymbol{d_i}$	*Joint Angle* $\boldsymbol{\theta_i}$
1	90°	*0*	*0*	θ_1
2	*0*	*l*	*0*	θ_2
3	*0*	*m*	*0*	θ_3
4	−90°	*0*	*0*	θ_4
5	*0*	*n*	*0*	θ_5

From each link to another link corresponding transformation matrices are,

$$T_0^1 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_1^2 = \begin{bmatrix} c_2 & -s_2 & 0 & lc_2 \\ s_2 & c_2 & 0 & ls_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_2^3 = \begin{bmatrix} c_3 & -s_3 & 0 & mc_3 \\ s_3 & c_3 & 0 & ms_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_3^4 = \begin{bmatrix} 0 & s_4 & c_4 & 0 \\ 0 & c_4 & s_4 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_4^5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Multiplying all the matrices, the final transformation matrix is found to be,

$$T_0^5 = \begin{bmatrix} -s_5c_1s_{234}-s_1c_5 & -c_5c_1s_{234}+s_1s_5 & c_1c_{234}+s_1 & P_x \\ -s_5s_1s_{234}+c_1c_5 & -c_5s_1s_{234}-c_1s_5 & s_1c_{234}-c_1 & P_y \\ s_5c_{234} & c_5c_{234} & s_{234} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

where,

$P_x = nc_1(c_{234} + tan_1) + mc_1c_{23} + lc_1c_2$ (i)

$P_y = ns_1(c_{234} - cot_1) + ms_1c_{23} + ls_1c_2$ (ii)

$P_z = ns_{234} + ms_{23} + ls_2$ (iii)

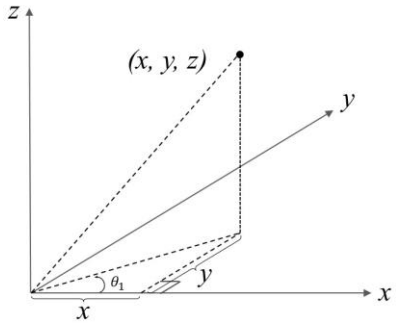

Fig.3 Coordinate projection in x, y, z frame.

2.3 Inverse Kinematics Modeling

By solving Eq. (6) from the given coordinate position and orientation of the end effector, the inverse kinematics model can be formed. However, the equation can be simpler if the number of variables is reduced. Noticeably, the 3R planner links rotate about the base frame, so, the rotation of the end effector with respect to the base frame can be determined by geometric calculation. Here, the rotation of the Link-1 with respect to the base frame (X_0, Y_0, Z_0) is indicated by θ_1. To find out the value of θ_1, the geometry is shown in Fig.3.

$$\theta_1 = tan^{-1}\left(\frac{y}{x}\right) \quad (7)$$

Therefore, from Eq. (7), the value of θ_1 can be found. To find out the value of the remaining variables, the transformation matrix T_1^5 is solved.

$$T_1^5 = \begin{bmatrix} -s_5s_{234} & -c_5s_{234} & c_{234} & x \\ s_5c_{234} & c_5c_{234} & s_{234} & y \\ -c_5 & s_5 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

where,

$x = nc_{234} + mc_{23} + lc_2$ (iv)

$y = ns_{234} + ms_{23} + lc_2$ (v)

$z = 0$

By solving Eq. (iv) and Eq. (v),

$$c_3 = \left(\frac{(x-p)^2+(y-q)^2 - l^2-m^2}{2lm}\right) \quad (9)$$

where,

$p = nc_{234}$ (vi)

$q = ns_{234}$ (vii)

$$s_3 = \pm\sqrt{1 - {c_3}^2} \quad (10)$$

$$\theta_3 = \pm tan^{-1}(s_3, c_3) \quad (11)$$

$$\theta_2 = sin^{-1}\left(\frac{(y-q)(l+mc_3) - s_3(xm-pm)}{(x-p)^2+(y-q)^2}\right) \quad (12)$$

$$\theta_2 = \pm tan^{-1}(s_2, c_2) \quad (13)$$

$$\theta_4 = \theta_{234} - \theta_{23} \quad (14)$$

From Eq. (13), Eq. (11), and Eq. (14), the value of $\theta_2, \theta_3, \theta_4$ can be obtained. θ_5 represents only the rotation of the end effector along its axis. So, the, value of θ_5 is arbitrary user input.

3. Simulating and analysis of the Workspace

There are two types of workspace, Reachable Workspace and Dexterous Workspace [9]. In this paper, the reachable workspace of the manipulator is discussed.

3.1 Workspace Simulation

The workspace has been simulated by Python Programming Language using different libraries such as NumPy, Matplotlib, etc. The workspace is modeled by using forward kinematic. The position and orientation of the end effector with respect to the base frame is found by the transformation matrix T_0^5. Then, by implementing the limiting values of the variables, the position of the end effector is plotted as red dots in a 3D graph to visualize properly. For example, considering all the joints angle to be $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = (45°, 65°, -70°, 30°, 0°)$, then the transformation matrix (T_0^5) becomes,

$$T_0^5 = \begin{bmatrix} -0.707 & -0.299 & 0.641 & 65.968 \\ 0.707 & -0.299 & 0.641 & 65.968 \\ 0 & 0.906 & 0.423 & 58.473 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The corresponding Python 3D plot for the given angle is illustrated in Fig.4.

Again, for the joints angle $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = (45°, 30°, -30°, -30°, 0°)$. The corresponding Python 3D plot is illustrated in Fig.5.

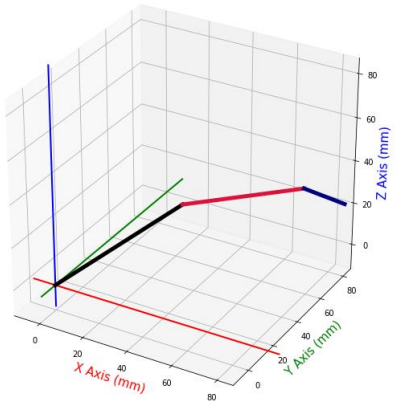

Fig.4 3D plot in Python for joint angles $(45°, 65°, -70°, 30°, 0°)$

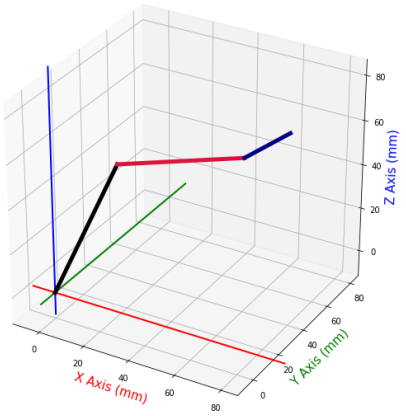

Fig.5 3D plot in Python for joint angles $(45°, 30°, -30°, -30°, 0°)$.

$$T_0^5 = \begin{bmatrix} -0.707 & 0.354 & 0.612 & 84.345 \\ 0.707 & 0.354 & 0.612 & 84.345 \\ 0 & 0.866 & -0.5 & 20 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2 Workspace Analysis

The workspace is a very important parameter to describe the ability and realize the capability of a manipulator [10]. The workspace is like a controlled volume where the end effector can reach and conduct its work. Based on the Range of Motion (ROM) the workspace of a manipulator can be determined. By putting the value of the length of each link, the joint angle in the equations the workspace is determined. The corresponding Python 3D plot of the workspace is illustrated in Fig.6. The workspace in the XZ plane and XY plane are shown in Fig.7 and Fig.8, respectively.

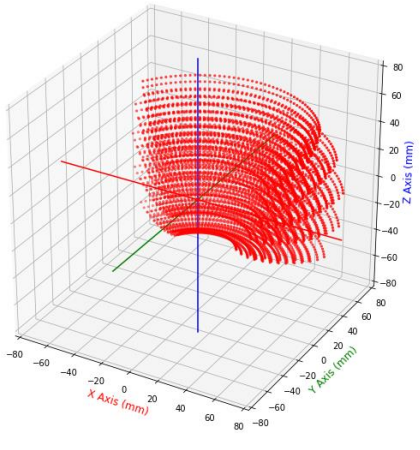

Fig.6 Isometric view of the workspace.

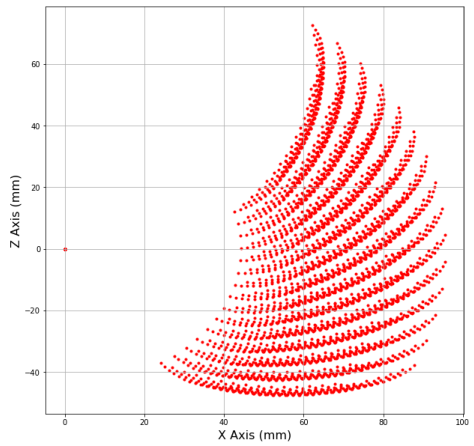

Fig.7 ZX plane plot of workspace.

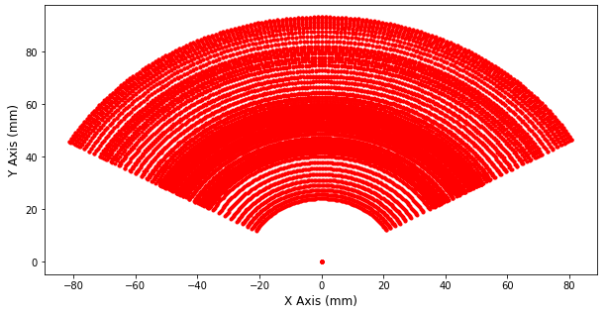

Fig.8 XY plane plot of workspace.

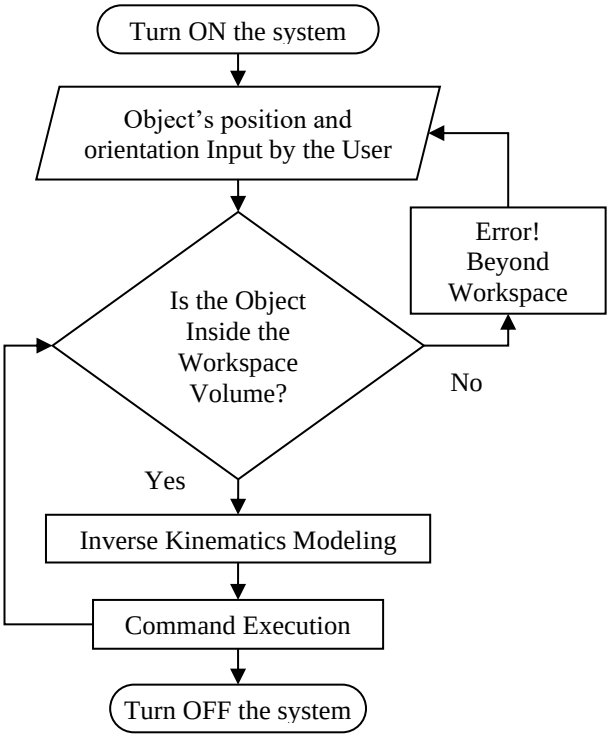

Fig.9 Flow chart of the inverse kinematics model development and application.

4. Inverse Kinematics Model Implementation and Controlling Mechanism

At first, the object and its position are detected by the user, and then, an RC communication module is used to transfer the data to the controller of the manipulator. After that, the algorithm checks out whether the object's position is inside or outside of the workspace. If the object is outside of the workspace, the control system will send a notation to the controller and request for a valid command. Otherwise, the command will be executed, and the control system will make an inverse kinematics model based on the desired orientation and position of the end effector. From Eq. (11) and Eq. (13), it is found that θ_2 and θ_3 have both positive and negative values. In this case, the algorithm will choose either the positive or negative value of these angles based on the actual movement of the manipulator links. Finally, the control system will execute the command. The detailed flow chart is shown in Fig.9.

The movement of the manipulator is taking place by linear actuators and gear motors which are controlled by a microcontroller. The movement mechanism of Link-2 of the manipulator by using a linear actuator is demonstrated in Fig.10.

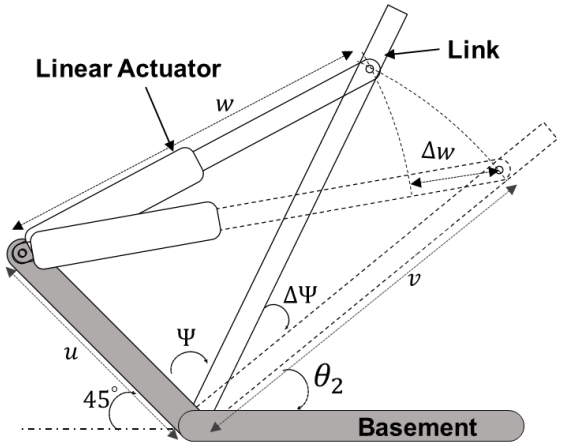

Fig.10 Angular displacement mechanism.

Here,
u = basement length.
v = the constant length from link joint to actuator joint.
w = length of the actuator at any instant.
Δw = change in length of the actuator.
Ψ =link angle for any position of the actuator.
$\Delta\Psi$ = change in angle due to change in actuator length
θ_2= Joint angle.

From Fig. 10,

$$\Delta\Psi = \frac{3\pi}{4} - \Psi - \theta_2 \quad (15)$$

From the cosine rule of the triangle,

$$\cos(\Psi + \Delta\Psi) = \frac{u^2 + v^2 - (w + \Delta w)^2}{2uv} \quad (16)$$

Or, $$w + \Delta w = \pm\sqrt{u^2 + v^2 - 2uv.\cos(\Psi + \Delta\Psi)} \quad (17)$$

Here, only the positive value of ($w + \Delta w$) is acceptable, so the Eq. (17) becomes,

$$\Delta w = \sqrt{u^2 + v^2 - 2uv.\cos(\Psi + \Delta\Psi)} - w \qquad (18)$$

Inverse kinematics model provides the value of θ_2. So, from Eq. (15), the value of $\Delta\Psi$ is found for the particular value of θ_2, and from Eq. (18), the amount of the change in actuator length is computed for the change in angle $\Delta\Psi$. That is how the linear actuator is used to change the link angle of the manipulator.

Whenever the controlling system receives a user input of a coordinate, the algorithm validates whether it is approachable or not. If approachable, then the inverse kinematics model is generated and implemented on the manipulator by the controlling system. Besides, there is always a home position that can be achieved when the manipulator is at rest.

5. Results and Discussion

From the simulation, the coordinates obtained by the end effector are shown in Fig. 5, Fig. 6, and Fig. 7, respectively. For a certain position and orientation of the end effector, the inverse kinematics model gives all the joint angles. Later, the data found from the inverse kinematics model is put into the forward kinematics model for verification.

The analysis of the workspace ensures the movement of the end effector at any certain orientation inside the working volume. The arbitrary value of θ_5 indicates the independent orientation of the end effector by the user. So, the operations like key rotating, screw rotating can be performed.

The simulation results show that the approximate volume and cross-sectional area along the ZX plane of the workspace are found to be 0.683 m^3 and 0.468 m^2, respectively. The minimum and maximum distance from the base point that can be achieved by the end effector are 44.3 cm and 95.2 cm respectively.

6. Conclusion

A robotic manipulator of 5 DOF has been presented in this paper. This manipulator is designed to accomplish different purposes, for example, load carrying, screw tightening, etc. The designed kinematics modeling will help this manipulator to conduct these tasks. Moreover, the controlling mechanism and control system is designed and programmed in such a way, so that this manipulator runs as smoothly as possible.

The Forward Kinematics and Inverse Kinematics modeling developed and described in this paper can be used for any manipulator having the same type of variables and structures. The precise control of each link will provide more accuracy to the movement of the end effector.

References:

[1] Gosselin, C., Isaksson, M., Marlow, K., & Laliberté, T., Workspace and sensitivity analysis of a novel nonredundant parallel SCARA robot featuring infinite tool rotation. *IEEE Robotics and Automation Letters*, *1*(2), 776-783, 2016.

[2] Iqbal, J., Islam, R. U., Khan, H., Modeling and Analysis of a 6 DOF Robotic Arm Manipulator, *Canadian Journal on Electrical and Electronics Engineering*, vol. 3(6), pp. 300-306, 2012.

[3] Briot, S., Khalil, W., Dynamics of Parallel Robots-From Rigid Bodies to Flexible Elements, *Springer*, 2015.

[4] Pieper, D. L., The kinematics of manipulators under computer control, *Proc.2nd Int. Congress for the Theory of Machines and Mechanisms*, vol. 2, pp. 159-168, 1968.

[5] Bonev, I. A., Ryu, J., A new approach to orientation workspace analysis of 6-DOF parallel manipulators, *Mechanism and Machine Theory*, vol. 36, no. 1, pp. 15-28, 2001.

[6] Denavit, J., Hartenberg R. S., A kinematic notation for lower-pair mechanisms based upon matrices, *Journal of Applied Mechanics*, vol. 77(2), pp. 215-221, 1955.

[7] Clothier, K. E., Shang, Y., A geometric approach for robotic arm kinematics with hardware design, electrical design and implementation, *Journal of Robotics*, 2010.

[8] Xu, D., Calderon, C. A. A., Gan, J. Q., Hu, H., Tan, M., An analysis of the inverse kinematics for a 5-DOF manipulator, *International Journal of Automation and Computing*, vol. 2, no. 2, pp. 114-124, 2005.

[9] Kumar, A., Waldron, K.J., The Workspace of a Mechanical Manipulator, *ASME J. Mech. Design*, vol. 103, pp. 665-672, 1981.

[10] Wang, L. C. T., Hsieh, J. H., Extreme reaches and reachable workspace analysis of general parallel robotic manipulators, *Journal of Robotic Systems*, vol. 15, no. 3, pp. 145-159, 1998.