

PAPER • OPEN ACCESS

Kinematic Modelling and Motion Mapping of Robotic Arms

To cite this article: Mangesh Saraf *et al* 2021 *J. Phys.: Conf. Ser.* **1969** 012002

View the [article online](#) for updates and enhancements.

You may also like

- [DE-based Algorithm for Solving the Inverse Kinematics on a Robotic Arm Manipulators](#)
B=UI Tam, TAO Linh, Trung Nguyen et al.
- [Inverse kinematic solver based on machine learning sequential procedure for robotic applications](#)
Francesco Aggogeri, Nicola Pellegrini, Claudio Taesi et al.
- [Neural Network Based Inverse Kinematic Solution of a 5 DOF Manipulator for Industrial Application](#)
S. Aravindhakshan, Sudhanv Apte and S M Akash



The advertisement features a dark blue background on the left with white and orange text, and a photograph of a woman at a podium on the right. The woman is smiling and looking towards the camera. The text on the left includes the ECS logo, the society's name, the meeting details, and promotional messages.

ECS The Electrochemical Society
Advancing solid state & electrochemical science & technology

243rd Meeting with SOFC-XVIII
Boston, MA • May 28 – June 2, 2023

Early registration discounts end **April 24!**
Accelerate scientific discovery!

Learn More & Register

Kinematic Modelling and Motion Mapping of Robotic Arms

Mangesh Saraf¹, A Agarwal¹, A Chaudhary¹ and A Ganthale¹

¹ Department of Mechanical Engineering, MIT-WPU, Pune, Maharashtra, 411038, India

Abstract: Robotic systems are known for their precision and accuracy. To achieve a desired set of operating parameters the system must have a good control system. Robotic arms can vary in size shape and purpose of use. With increasing DOF (Degree of Freedom) the complexity of the system increases and hence a DAQ system is required to check various parameters of the process. This paper discusses the implementation of kinematic and inverse kinematic modelling of a 6 Degree of Freedom (DOF) hydraulic arm in MATLAB environment. The forward kinematic model is based on Denavit-Hartenberg parameters. To obtain a given set of position and orientation of end effector, inverse kinematics gives the required set of joint angles. Potentiometers mounted on arm joint are used as the prime feedback elements for extracting the joint angles. All six joints of the arm are revolute joints. The values of potentiometer obtained are compared to the desired input values. Proportional direction control valve is the main controlling element of the hydraulic circuit. The forward and inverse kinematics is validated and implemented in MATLAB robotics toolbox (developed by Peter Croke). A GUI is also used to ease the process of retrieving and feeding the information in the system. The results of forward and inverse kinematics from MATLAB are verified by the physical model discussed in this paper.

Keywords: Kinematics; Inverse Kinematics; D-H (Denavit Hartenberg) parameters; Kinematic Modelling; Motion Planning; MATLAB GUI

1. Introduction

Robotic arms are one of the most used types of robotic systems in today's world. There are several types of robots, and they can be classified based on factors like the type of joints, number of degrees of freedom, actuation mechanism, etc. Robotic arms can have either revolute joints or prismatic joints. Some can also have a combination of both these types. Some of the most common types of robotics arms are:

1. Cartesian Robot
2. SCARA Robot
3. Articulated Robot
4. Cylindrical Robot
5. Polar Robot
6. Delta Robot

The actuation systems for a robot determines the responses and other characteristics of a robot. The table below compares various characteristics which are desired in a robot.



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Table 1: Comparison between various types of actuators.

Characteristic	Hydraulic	Pneumatic	Belt Driven	Electronic actuators
Response Time	Slow	Fast	Moderate	Fast
Load Capacity	High	Low	Moderate	Moderate
Complexity of construction	High	High	Moderate	Low
Cost	Very High	High	Moderate	Moderate

It can be inferred from the above table; hydraulics actuated robots have a remarkably high load bearing capacity which is an important characteristic for many industrial and explorational applications. These systems, however, lack in the response time which is a huge disadvantage. To overcome this hurdle faced, a PID control loop can be implemented to reduce the errors dynamically, thus reducing the response time drastically, resulting in a fast system.

Kinematic modelling deals with the robot's motion irrespective of the forces that act on it. It involves study of its geometry and its positions in space. It is used to study relations between end effector and base along with intermediate links. Kinematics is divided into two sections. First, forward kinematics or direct kinematics which gives the cartesian position of end effector for a given set of fully defined joint parameters. second, inverse kinematics (IK)- which involves calculating joint angles, position for a given end effector position and orientation. IK is more complicated and requires more constraints compared to forward kinematics.

2. Literature review

P Corke [8] has developed an extensive toolbox for robotic arms in the MATLAB software. This toolbox has many features including functions to develop transformation matrices, ability to develop trajectories, etc. The toolbox can be used to model custom robots with the DH parameters as an input.

Gupta E. [7] have researched about the different software which can be used for the simulation and modelling of robotic systems. They have compared the most feasible options based on various parameters. In this paper, they have concluded that MATLAB would be the most suitable software for kinematic modelling of robots due to factors like huge host of inbuilt toolboxes and a strong developer community.

The modelling of a hydraulic circuit for multiple DOF robotic arm has been discussed by Stanislav and Krejsa [11] in their research paper. The authors have used the MATLAB Sim Hydraulics to design the circuit and simulate their arm and observe the position-time plots for various arms.

Hanan A.R. Akkar and Ahlam Najim A-Amir [12] have discussed the use of LabVIEW software for modelling and analysis of robotic arms. LabVIEW has been used to design a Graphical User Interface for the study of different parameters of a robotic arm.

Modelling of robotic arms has been hugely supplemented by the work of Denavit J. & R.S. Hartenberg [3]. They have developed a set of six parameters which can be used to design and model any robot with multiple degree of freedoms with either revolute or prismatic joints.

3. Plant Setup

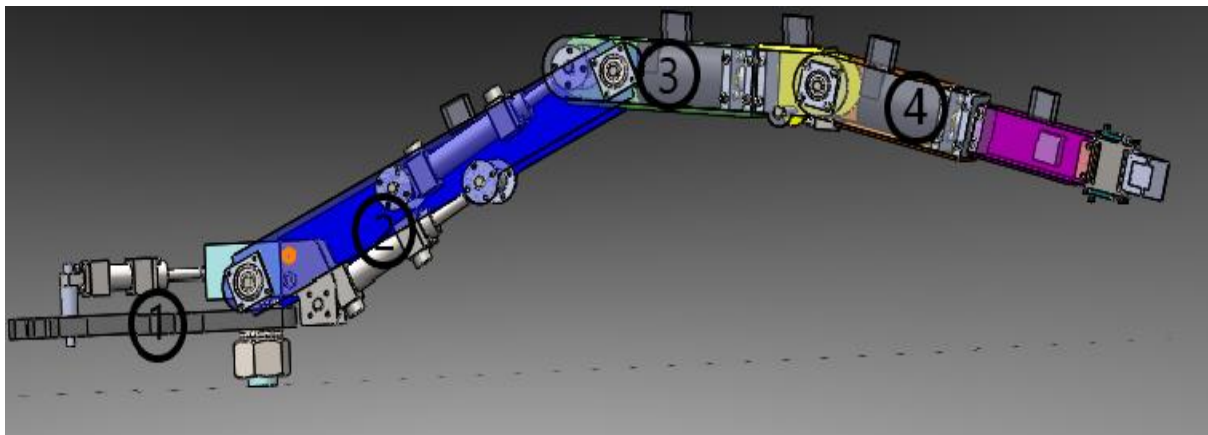
The robotic platform used is a 6 DOF robotic arm manipulator. It is a serial manipulator having all joints as revolute. The arm geometrical configuration is made up of Waist (Body), Shoulder, Elbow and Wrist in correspondence with the human arm joints. Each of these joints except the wrist has a single DOF.

Wrist has 3 DOF (Yaw, Pitch, Roll), thus making the end effector more flexible in terms of operation. The arm is fully actuated with each DOF achieved by a Proportional Derivative Control Valves (PDCVs) with all PDCVs connected to PID. The end-effector is a two-state gripper. The built-in mechanical safety limits restrict the joint motion in case somethings in the algorithm goes wrong.

Table 2: Salient features of the robotic arm.

Feature	Description
Position Precision	$\pm 1.0\text{mm}$ (approx.)
Operating Movement Speed	20 mm/sec
Hydraulic Pressure	Actuators (70 bar) Gripper (140 bar)
Pump Capacity	80 lpm
Overall Weight	30 Kgs
Load Capacity	15kgs
Hydraulic Hose Diameter	3/8"

The final model of our robot was designed on Solid works software after considering the required specifications. The CAD model is shown below.



1 – Waist

2 – Shoulder

3 – Elbow

4 – Wrist

Figure 1: CAD Rendering of Robotic Arm.

4. Kinematic Modelling

Forward kinematics is simple to implement compared to inverse kinematics where some form of numerical method or algorithm is needed to get the desired result. The output of forward kinematics is singular and unique while inverse kinematics can have multiple outputs based upon the constrains.

4.1. Forward Kinematic Model

The study of Kinematic Problem of a robot can be carried out by different methods. The method used in this paper is based on Denavit - Hartenberg (DH) parameters [3]. This Method is systematic in nature and application of method is easy for modelling serial manipulators. DH method is used to develop the kinematic model of the robot because of its versatility and acceptability for modelling of any number of joints and links of an arm irrespective of its complexity.

The kinematic diagram for our robot considering the DH parameters (from Table 3) has been shown below where the axes of rotation for each joint have been considered as the Z axis.

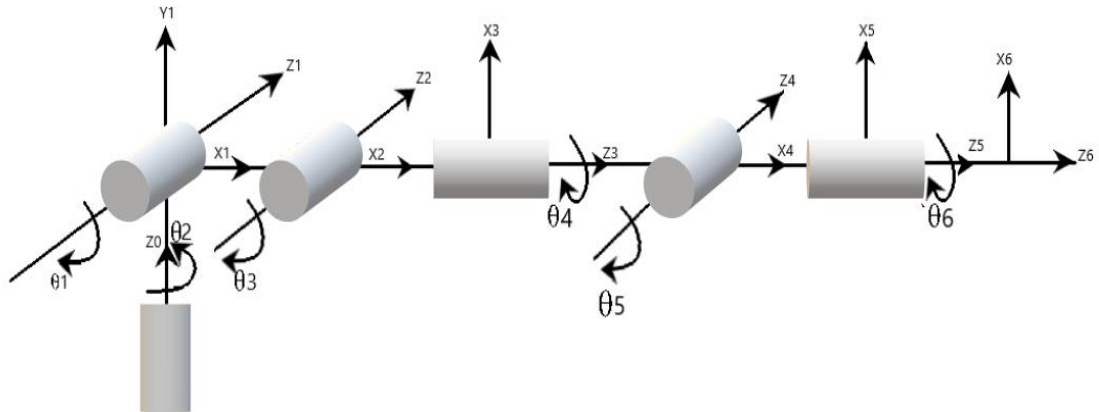


Figure 2: Coordinate frame assignment.

DH works with quadruple $\{\alpha_{i-1}, a_{i-1}, d, \theta_i\}$ [3] represents twist angle, link length, link offset and joint angle, respectively.

Table 3: DH Parameters

Theta	d (mm)	Alpha (deg)	a (mm)
θ_1	37	$\pi/2$	0
θ_2	0	0	500
θ_3	0	$\pi/2$	0
θ_4	250	$-\pi/2$	0
θ_5	0	$\pi/2$	0
θ_6	400	0	0

Using the general form of transformation matrix for each link is given by [4]:

$$T = \left[\begin{array}{ccc|c} R & & & T \\ \hline 0 & 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{ccc|c} \cos \theta & -\sin \theta \cos \alpha & \sin \theta \sin \alpha & r \cos \theta \\ \sin \theta & \cos \theta \cos \alpha & -\cos \theta \sin \alpha & r \sin \theta \\ 0 & \sin \alpha & \cos \alpha & d \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

Figure 3: General Form of Transformation Matrix

The Calculation of these individual matrices is direct. Computing each of the link transformation matrix using above equation, as follows

A1 =

$$\begin{bmatrix} C1, & 0, & S1, & 0 \\ S1, & 0, & -C1, & 0 \\ 0, & 1, & 0, & 37/1000 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

A2 =

$$\begin{bmatrix} C2, & -S2, & 0, & C2/2 \\ S2, & C2, & 0, & S2/2 \\ 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

A3 =

$$\begin{bmatrix} C3, & 0, & S3, & 0 \\ S3, & 0, & -C3, & 0 \\ 0, & 1, & 0, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

A4 =

$$\begin{bmatrix} C4, & 0, & -S4, & 0 \\ S4, & 0, & C4, & 0 \\ 0, & -1, & 0, & 1/4 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

A5 =

$$\begin{bmatrix} C5, & 0, & S5, & 0 \\ S5, & 0, & -C5, & 0 \\ 0, & 1, & 0, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

A6 =

$$\begin{bmatrix} C6, & -S6, & 0, & 0 \\ S6, & C6, & 0, & 0 \\ 0, & 0, & 1, & 2/5 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

Multiplying all the matrices together, starting with the first joint all the way up to the end effector. The final T vector will contain the position of the end effector. The R matrix will contain the orientation of the end effector.

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$r_{11} = C6*(C5*(S1*S4 + C4*(C1*C2*C3 - C1*S2*S3)) - S5*(C1*C2*S3 + C1*C3*S2)) - S6*(S4*(C1*C2*C3 - C1*S2*S3) - C4*S1)$$

$$r_{12} = -S6*(C5*(S1*S4 + C4*(C1*C2*C3 - C1*S2*S3)) - S5*(C1*C2*S3 + C1*C3*S2)) - C6*(S4*(C1*C2*C3 - C1*S2*S3) - C4*S1)$$

$$r_{13} = C5*(C1*C2*S3 + C1*C3*S2) + S5*(S1*S4 + C4*(C1*C2*C3 - C1*S2*S3))$$

$$p_x = (2*C5*(C1*C2*S3 + C1*C3*S2))/5 + (2*S5*(S1*S4 + C4*(C1*C2*C3 - C1*S2*S3)))/5 + (C1*C2)/2 + (C1*C2*S3)/4 + (C1*C3*S2)/4$$

$$r_{21} = C6*(C5*(C4*(C2*C3*S1 - S1*S2*S3) - C1*S4) - S5*(C2*S1*S3 + C3*S1*S2)) - S6*(S4*(C2*C3*S1 - S1*S2*S3) + C1*C4)$$

$$r_{22} = -S6*(C5*(C4*(C2*C3*S1 - S1*S2*S3) - C1*S4) - S5*(C2*S1*S3 + C3*S1*S2)) - C6*(S4*(C2*C3*S1 - S1*S2*S3) + C1*C4)$$

$$r_{23} = C5*(C2*S1*S3 + C3*S1*S2) + S5*(C4*(C2*C3*S1 - S1*S2*S3) - C1*S4)$$

$$p_y = (2*C5*(C2*S1*S3 + C3*S1*S2))/5 + (2*S5*(C4*(C2*C3*S1 - S1*S2*S3) - C1*S4))/5 + (C2*S1)/2 + (C2*S1*S3)/4 + (C3*S1*S2)/4$$

$$r_{31} = -C6*(S5*(S2*S3 - C2*C3) - C4*C5*(C2*S3 + C3*S2)) - S4*S6*(C2*S3 + C3*S2)$$

$$r_{32} = S6*(S5*(S2*S3 - C2*C3) - C4*C5*(C2*S3 + C3*S2)) - C6*S4*(C2*S3 + C3*S2)$$

$$r_{33} = C5*(S2*S3 - C2*C3) + C4*S5*(C2*S3 + C3*S2)$$

$$p_z = S2/2 + (S2*S3)/4 - (C2*C3)/4 + (2*C5*(S2*S3 - C2*C3))/5 + (2*C4*S5*(C2*S3 + C3*S2))/5 + 37/1000$$

4.2. Inverse Kinematic Model

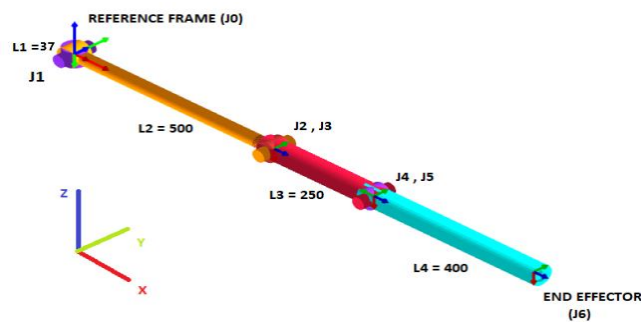


Figure 4: Frame Assignment for Inverse Kinematics Calculations

Inverse kinematics involves the calculation of joint angles with the end effector coordinates as input. This is useful in applications where we know the final position which is to be reached with respect to a reference frame. The inverse kinematic gives the joints values i.e. $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$, where theta is the angle to rotate around the Z axis. Multiple solution can be obtained for one end effector position, though not all solutions are feasible, the solution that provides a smooth operation and minimum joint motion is said to be best solution. Optimal Solution to supply a robust operation. Software such as MATLAB directly provides the optimal solution.

In this paper, both analytical method and software approached is mentioned. This method verifies makes certain for any object in the vicinity of the arms, the model provides correct joint angles.

Inverse Kinematic calculations for 1st three joints:

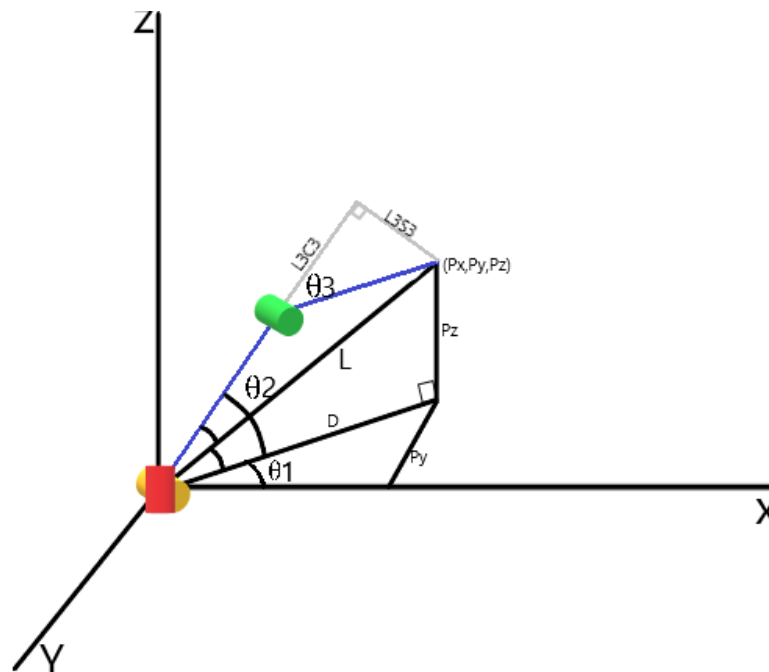


Figure 5: Line diagram for inverse kinematic calculations

The equations for inverse kinematics of the first three joints are as follows:

$$\Theta_1 = \arctan (P_y/P_x) \quad -(1)$$

$$\Theta_3 = \arctan ((1-M^2)^{1/2}/M) \quad -(2)$$

Where, $M = (L^2 - L_2^2 - L_3^2) / 2L_2L_3$

$$\Theta_2 = \arctan (L_3 * \sin \Theta_3 * D + P_z (L_3 \cos \Theta_3 + L_2) / L_3 * \cos \Theta_3 + L_2 * D - L_3 * \sin \Theta_3 * P_z) \quad -(3)$$

Where, $D = (P_x^2 + P_y^2)^{1/2}$

5. Verification of kinematic model

The verification of the model is done by the physical model and cross checking the results with analytical calculations. A proposed analytical method is given in inverse modelling. MATLAB is used to make the calculation part simpler. Verification is done based on required accuracy of 1mm for forward kinematics and 1 degree for inverse kinematics.

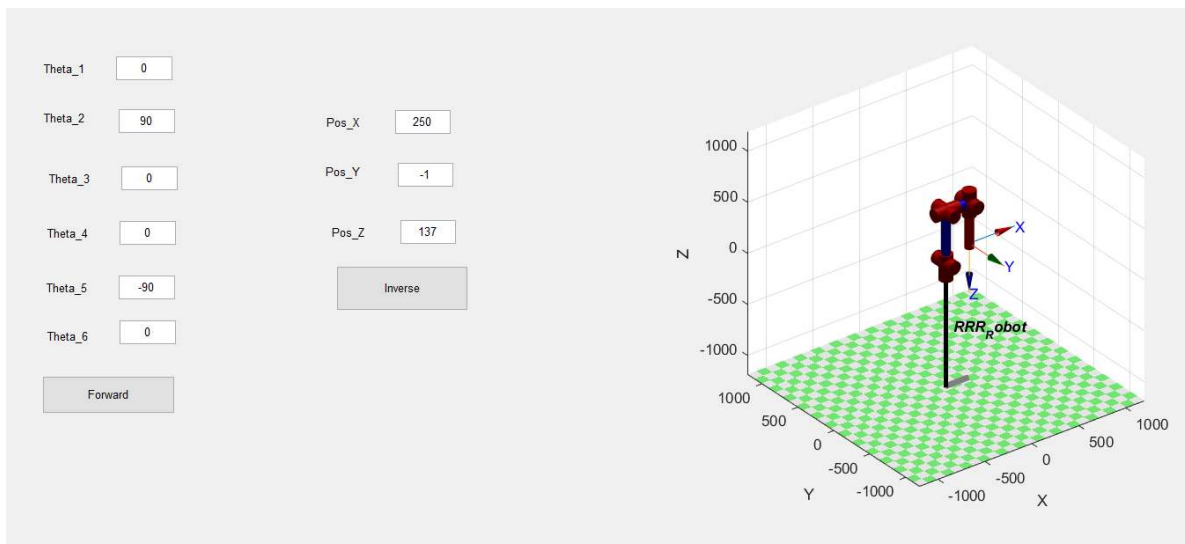
5.1. Verification of forward kinematic model

The analytical calculations for forward kinematics of the robot have been verified on MATLAB using the robotics toolbox. The analytical calculation is given in section 2.1.

The same inputs were given in the MATLAB environment and the position of robot was plotted after running the MATLAB script. The code also gives the end effector positions as an output. These coordinates are compared with the ones obtained from analytical calculations and have been compared in the table below.

Table 4. Comparison between the results from analytical and MATLAB results

Input angles	Θ_i (deg)	End effector Position (mm)	MATLAB Output (mm)	Experimental Output (mm)
θ_1	0	X	250	250
θ_2	90	Y	-1	0
θ_3	0	Z	137	137
θ_4	0			
θ_5	-90			
θ_6	0			

**Figure 6:** MATLAB plot of forward kinematics

5.2. Verification of Inverse Kinematic Model

The verification of inverse kinematic model involves the measurement of joint angles for a particular end effector position using a set of potentiometers mounted on the joint axis. Potentiometers are sensors which convert the rotational motion into a change of resistance. This change of resistance can be converted into an electrical signal and this feedback would be used to perform the inverse kinematic calculation analytically. The accuracy of potentiometer readings is 0.1 deg. A Data Acquisition system is used to collect data from 6 potentiometers and plot the results to a GUI.

We have used the robotics toolbox mentioned earlier to calculate the joint angles from a desired end effector position. These values have been further verified by analytical calculations and then compared with the physical potentiometer feedbacks mentioned above. The results from analytical calculations are compared with the actual end effector position.

The table below summarizes the joint angles obtained for the end effector coordinates (500, -100,100).

Table 5: Verification of inverse kinematics with physical model

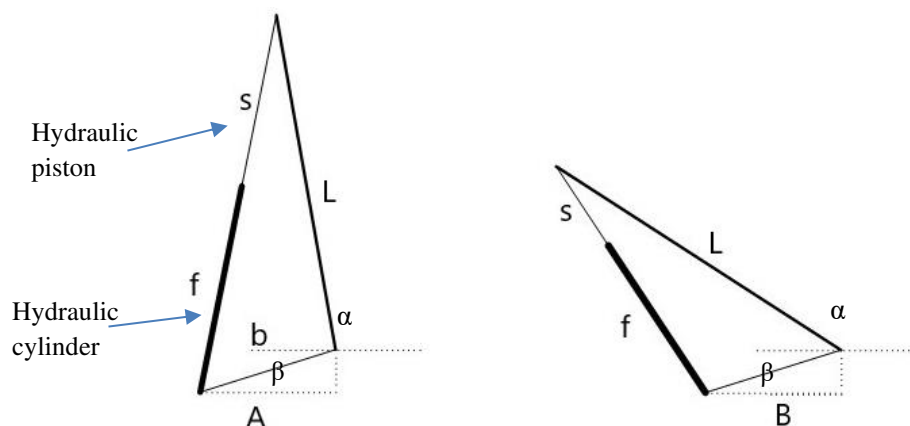
Joint Number	Parameter	Potentiometer	MATLAB
1	θ_1	-11.6	-12
2	θ_2	79.30	80
3	θ_3	-9.80	-10
4	θ_4	0	0
F5	θ_5	-44.2	-44
6	θ_6	0.1	0

6. MATLAB GUI for verification

S Mahalingam. [6] has discussed the method and algorithm for the development of a simple Graphic User Interface (GUI) which can be used for performing the forward and inverse kinematics calculations. We have modified and developed a GUI for simulation of our robot. We have re-developed by considering the DH parameters for our robot. This GUI will be developed further in the coming months to make it a more polished interface. We incorporated the real time angle feedback from the potentiometer feedback system.

7. Converting joint angles into equivalent liner displacement of hydraulic cylinders

There are various methods to transfer joint angles into linear displacement of hydraulic cylinders based on geometry, differential calculus. This paper discusses the use of cosine rule to transfer required angle into required cylinder extension position. As the length of every link is fixed there are only two variables that are present into the formed triangle. The length of cylinder (fixed length+ stroke length) and required angle α (required angle). Implantation of cosine rule can be done easy in primitive controllers while the implantation of counterpart differentiation equations requires higher computing power.

**Figure 7.** Line diagram of hydraulic cylinder and joint angle triangle**Fig A-** cylinder in extended position**Fig B-** cylinder in intermediate position

α - required link angle with previous link

L- link length

- s- Stroke length of piston
- f- Fixed length of cylinder
- b- fixed distance between link and cylinder

According to Cosine Rule: -

$$s + f = \sqrt{L^2 + b^2 - 2Lb \cos(180 - \alpha + \beta)} \quad (4)$$

Thus

$$s = \sqrt{L^2 + b^2 + 2Lb \cos(\beta - \alpha)} - f \quad (5)$$

The equation (5) gives the required stroke length of cylinder to achieve the desired angle of rotation. The parameters in the equation can be changed to suit various types of actuators of different types of robots. Dynamic modelling based upon above formula is also possible.

8. Conclusion

In various sections of this paper, we have verified and calculated the kinematic models for our specific robot with the help of various tools including the robotics toolbox. We have also verified the physical results further using a feedback system involving potentiometers. As it can be noted from the result tables, the errors between real world physical readings and the simulation results are within acceptable limits and can be used for further control of the robotic arm.

Kinematic modelling of a robot is a key step in designing of a robotic system and in this paper, we have covered Forward and Inverse Kinematics. Both methods have different application areas, but these can primarily be used to cross verify the results of each other. This paper also covers the geometric modelling approach to calculate the end effector positions. This can be programmed further in MATLAB to make the process more fluid. The scope of this project can be extended for more applications like a PID controlled robotic arm with multiple degrees of freedom. These quick response systems can also be utilized for industrial applications like pick and place mechanisms for sorting. Further applications can also include the integration of these arms with sophisticated systems like an underwater remotely operated vehicle.

References

1. Iqbal J., Islam R. and Khan H., 2012. *Modeling and Analysis of a 6 DOF Robotic Arm*. (Canadian Journal on Electrical and Electronics Engineering) 3(6)
2. Deshpande V., George P. M., 2014 *Kinematic Modelling and Analysis Of 5 DOF Robotic Arm*. (International Journal of Robotics Research and Development (IJRRD))4(2) 17-24
3. Denavit J. & R.S. Hartenberg 1955. *A kinematic Notation for Lower- Pair Mechanism Based on Matrices*. (ASME Journal of Applied Mechanics, 215-221.)
4. R K Mittal, J Nagrath. 2005. *"Robotics and Control"*, (Tata McGraw-Hill.)
5. S. B. Niku. 2001. *Introduction to Robotics: Analysis, Systems, Applications*. (Prentice Hall)
6. Mahalingam S.V 2018, *A GUI Based Kinematic Model Development OF 6 DOF Manipulator Using MATLAB*. (International Journal of Engineering Development and Research, 6(4).)
7. Gupta E, Somani S.K. 2019, *Design, Analysis and Kinematics of Robotic Manipulator: A Literature Review*. (International Journal of Scientific Research and Review7(5))
8. Cubero S., 2008 *An inverse kinematics method for controlling all types of serial-link robot arms*. (Mechatronics and Machine Vision in Practice, Springer-Verlag, Berlin), pp. 217-232.
9. *Robotics toolbox for MATLAB by Peter Corke (Version9)/MATLAB INC*

10. Peter Corke 2011 “*Robotics, vision and control: Fundamental algorithms in MATLAB*”, (Springer)
11. Stanislav v., Krejsa J. 2009 *Hydraulic arm modeling via MATLAB Sim hydraulics*. (Engineering MECHANICS) Vol. 16, No. 4, p. 287–296
12. Akkar H.R., Amir A. 2016. *Kinematics Analysis and Modeling of 6 Degree of Freedom Robotic Arm from DFROBOT on LabVIEW*. (Research Journal of Applied Sciences, Engineering and Technology)

Acknowledgment: We would like to thank all those who have actively or indirectly contributed toward the research of this project. We would like to express our gratitude towards Dr. Vishwanath world Peace University for providing all the needed facilities.