# FINAL PROJECT REPORT

**on**

**‘Motion Planning of Cooperative Robots’**

**for**

**Introduction to Robot Modelling**

**ENPM 662**

**by-**

**Himanshu Singhal**

**115891944**

Contents

## ABSTRACT

This project presents the aspects of motion planning of Cooperative Robotic arm that is designed using CAD software for the purpose of industrial application where one arm grasps and orients the object (Industrial PCB) while other performs desired action on it (Scanning or cleaning). Designing is followed by mathematical modeling and kinematics using MATLAB. A smooth trajectory is generated using Peter Corke software and the model is validated by comparing the values obtained by trajectory and kinematics.

## INTRODUCTION

The Robotic arm used in this project is a 5 DOF manipulator with 5 revolute joints and 0 prismatic joints because the linear motion is not required in the desired application. The manipulator has a fixed base of reference and Joint 1 is attached to this base frame. Joint 1 rotates 360 degrees about the joint z-axis as shown in the figure. Joint 2, Joint 3 and Joint 4 align itself with the parallel axis of rotation and coinciding x-axes at zero configuration. There are no mechanical constraints in the model, however, due to the design structure, joint 2, joint 3 and joint 4 do not undergo a full rotation.

The main idea behind this project was to design a system of manipulators that can undergo both static and dynamic tasks eliminating the need of a human. These collaborative robots called Cobots work synchronously with each other to obtain the desired task. The suitable area here would be an industrial sector and can be extended to medical purposes. In the industrial sector, cleaning and scanning of heavy PCBs, holding and tightening of nuts and bolts, wrapping and unwrapping of goods are a few good examples where Cobots could be used. In the medical sector (specifically during surgeries) activities like simultaneous cutting and soaking the blood or shifting of patients between beds are great examples of Cobots application.

## MOTIVATION

Both my parents are doctors and I have had opportunities to sneak into their workspace and get a glimpse of their work. After analyzing multiple instances either in real life or reel life, I observed that ‘shifting of patients from the stretcher to the operation bed’ is a strenuous task that often requires much more manpower and effort than normal. Also, I got an opportunity to work in a manufacturing plant where cleaning of huge PCBs (specifically) was a strenuous task. An operator holds and tilts it while other simultaneously clean it, which is neither safe for the PCB nor for the operators. These cooperative manipulators work like the human hands where one hand performs a static action while other performs a dynamic one.

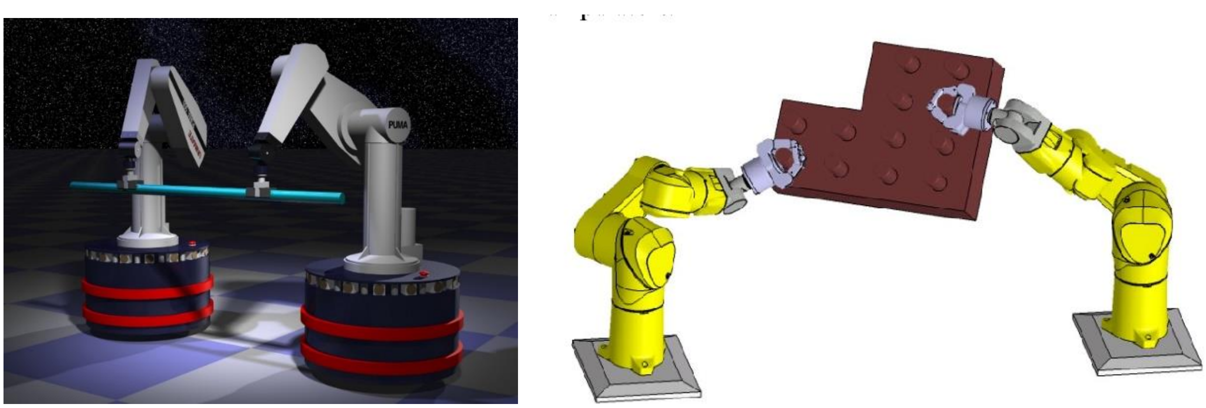


**Fig.1: Basic structure of Cobots**

**ASSUMPTIONS**

The following assumptions are made while designing and modeling the manipulators:

1. Joint 1 and Joint 5 completely rotate 360 degrees while Joint 2, 3 and 4 can rotate almost 360 degrees
   *Justification* - Since Joint 1 and Joint 5 are both attached to only 1 link, their motion is unrestricted unlike the other 3 joints, hence they can undergo a full rotation

2. There are no obstacles in the workspace i.e. the environment is known and safe to work
   *Justification* - Since the robot is not equipped with any camera or other vision-based interferences, it cannot detect and avoid any obstacle (unless pre-planned), hence the workspace is obstacle free

3. The object to be moved, picked, rotated or acted upon is rigid with known dimensions and weight
   *Justification* - The maximum dimensions of the object decide the minimum safe distance between the two robots, and if the object is on unknown dimensions or not rigid, then that distance may not be fixed, and the robots may not function as desired

4. All the joints rotate smoothly producing a jerk-free trajectory
   *Justification* – The joints in real-world use servo motor which produces smooth motion. If the joints produce a jerky trajectory, the vibrations can damage the object (PCB) and fail the task

5. The start and end position and orientation of the end effectors lies within the shared workspace
   *Justification* – There exists a common workspace between the two robots and the endpoints of each specifically lies in that shared workspace

6. The object is kept within the achievable zone of both the robots
   *Justification* – The object is within the reach of the robots so that both the robots can interact with it

7. Self-collision and cross collision are ignored while simulating the trajectory
   *Justification* – For simplicity, self and cross-collision are ignored in this project

**KINEMATICS**

The branch of mechanics concerned with the motion of objects without reference to the forces that cause the motion is called Kinematics. In robotics, we deal with two types of kinematics, forward and inverse. In forward kinematics, the input is the joint angles (variables) while the output is the corresponding position of the end effector. In Inverse kinematics, the input is the position and orientation of the end effector while the output is the corresponding joint angles (variables).

For forward kinematics, the joint variables are the known angles of rotations while the unknown coordinates are the end effector locations i.e. the edge of PCB.
For Inverse Kinematics, the known coordinates are the end effector locations at the edge of the PCB while the unknown variables are the joint angles that lead to the known position and orientation. There can be multiple solutions to this problem and they are all acceptable if they fall in the specified angle range.

The important point here is that since the object is rigid, the dimensions are known and will not change with time or under the application of force, therefore, the maximum dimensions of the object contributes to the distance between the 2 manipulators or end effectors frames to avoid any inter collision (ideally).

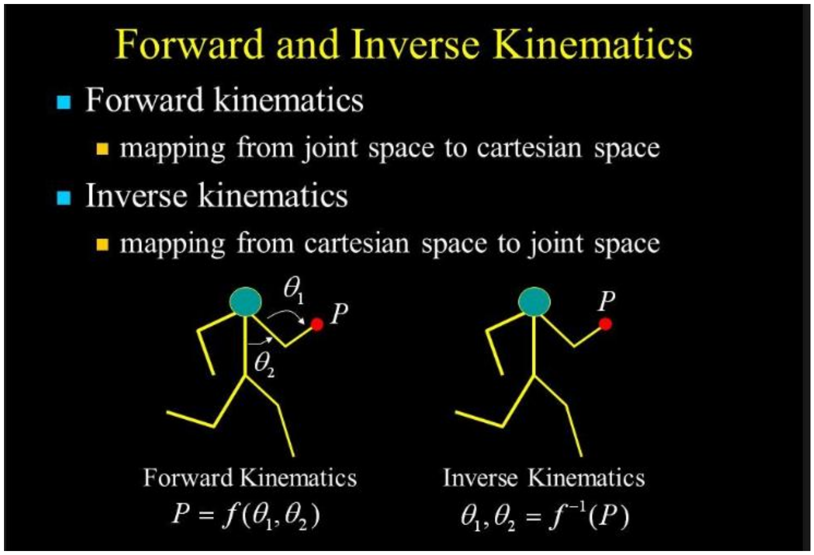


**Fig.2: Kinematics mapping**

**Forward Kinematics (FK) - DH Parameter**

A reference frame is assigned to each link, and homogeneous transformations matrices are used to describe the relative position/orientation of these frames. The reference frames are assigned according to a convention, and therefore the number of parameters needed to describe the pose of each link, and consequently of the robot, is minimized.

- **DH Table for the manipulator**

**Forward Kinematics for Manipulator 1 (Right)**

| Link | θ | d | a | α |
|---|---|---|---|---|
| 1 | $\theta_1^*$ | $d_1$ | 0 | $90^{\circ}$ |
| 2 | $\theta_2^*$ | 0 | $l_2$ | 0 |
| 3 | $\theta_3^*$ | 0 | $l_3$ | 0 |
| 4 | $\theta_4^* + 90$ | 0 | 0 | 90 |
| 5 | $\theta_5^*$ | $d_5$ | 0 | 0 |

**Forward Kinematics for Manipulator 1 (Left)**

| Link | θ | d | a | α |
|---|---|---|---|---|
| 1 | $\theta_1^*$ | $d_1$ | 0 | $-90^{\circ}$ |
| 2 | $\theta_2^*$ | 0 | $l_2$ | 0 |
| 3 | $\theta_3^*$ | 0 | $l_3$ | 0 |
| 4 | $\theta_4^* - 90$ | 0 | 0 | $-90^{\circ}$ |
| 5 | $\theta_5^*$ | $d_5$ | 0 | 0 |

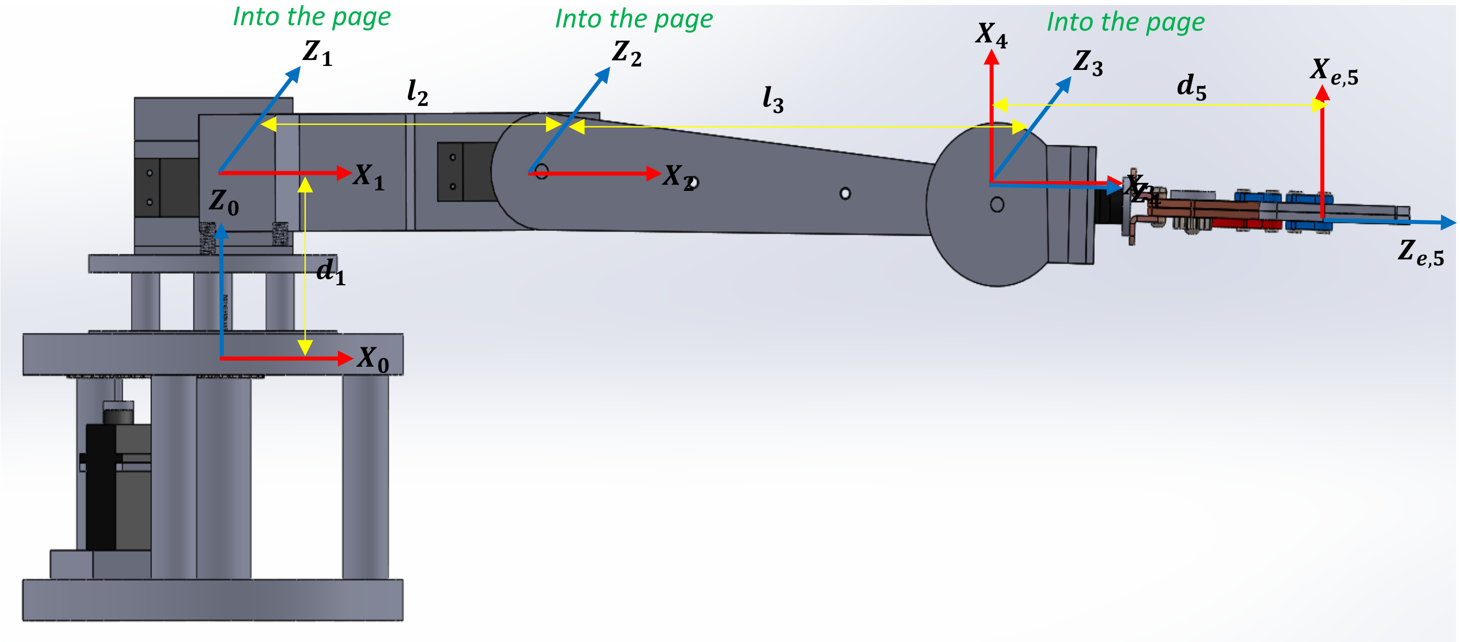


**Fig.3: Manipulator 1 (Left) in the zero configuration**

Following are the notations for the manipulator frame:

a. $Z_i$ is the axis of rotation of Joint $i$
b. $l_i$ is the distance between the $Z_{i-1}$ and $Z_i$ measured along $X_i$
c. $d_i$ is the distance between the $X_{i-1}$ and $X_i$ measured along $Z_{i-1}$

From DH table for the <u>left manipulator</u>, we can find the homogeneous transformation matrix that relates the end effector frame to the base frame and is given by $TL_5$ =

$$\begin{bmatrix} C_1 * S_{234} * C_5 + S_1 * S_5 & -C_1 * S_{234} * S_5 + S_1 * C_5 & C_1 * C_{234} & C_1 * (d_5 * C_{234} + l_3 * C_{23} + l_2 * C_2) \\ S_1 * S_{234} * C_5 - C_1 * S_5 & -S_1 * C_{234} * S_5 - C_1 * C_5 & S_1 * C_{234} & S_1 * (d_5 * C_{234} + l_3 * C_{23} + l_2 * C_2) \\ C_{234} * C_5 & -C_{234} * S_5 & -S_{234} & d_1 - l_2 * S_2 - l_3 * S_{23} - d_5 * S_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where $S_{ijk} = \sin(q_i + q_j + q_k)$ and $C_{ijk} = \cos(q_i + q_j + q_k)$

From DH table for the <u>right manipulator</u>, we can find the homogeneous transformation matrix that relates the end effector frame to the base frame and is given by $TR_5$ =

$$\begin{bmatrix} S_1 * S_5 - C_1 * S_{234} * C_5 & C_1 * S_{234} * S_5 + S_1 * C_5 & -C_1 * C_{234} & C_1 * (-d_5 * C_{234} + l_3 * C_{23} + l_2 * C_2) \\ -S_1 * S_{234} * C_5 - C_1 * S_5 & S_1 * C_{234} * S_5 - C_1 * C_5 & -S_1 * C_{234} & S_1 * (-d_5 * C_{234} + l_3 * C_{23} + l_2 * C_2) \\ C_{234} * C_5 & -C_{234} * S_5 & S_{234} & d_1 + l_2 * S_2 + l_3 * S_{23} - d_5 * S_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

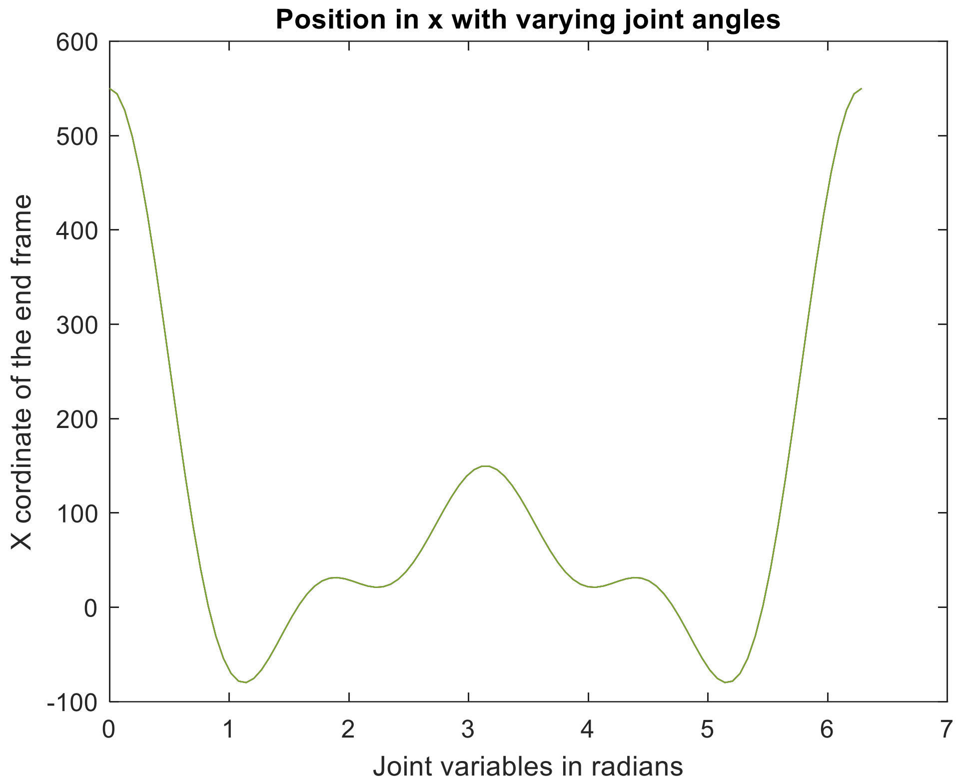


**Fig. 4: Variation of X with Joint Angles**

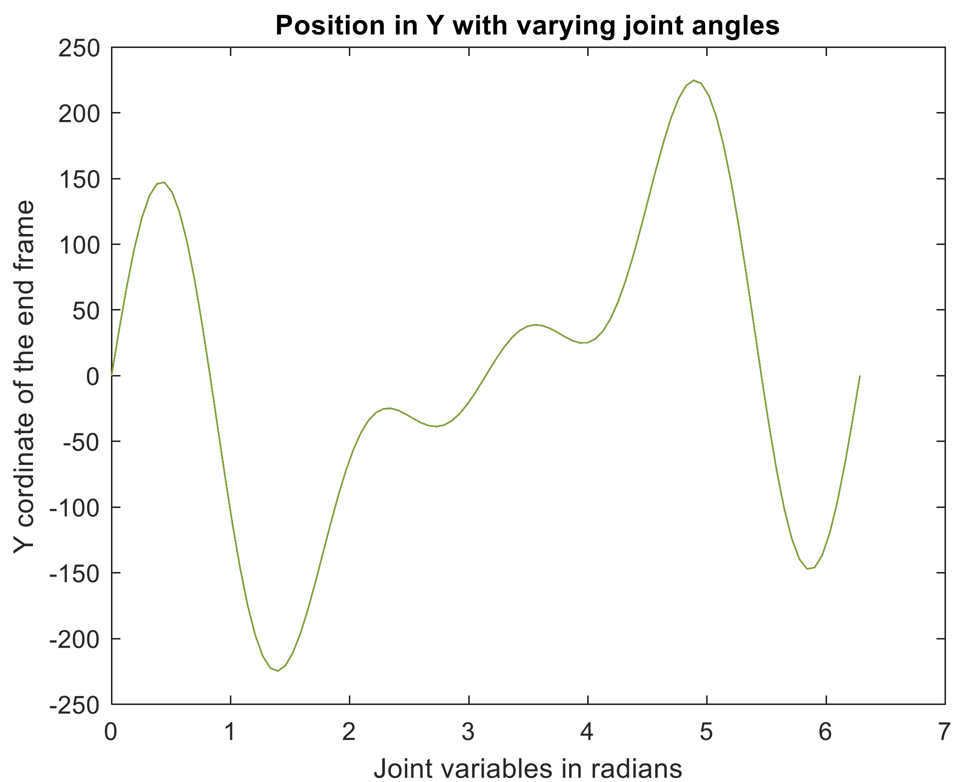


**Fig. 5: Variation of Y with Joint Angles**

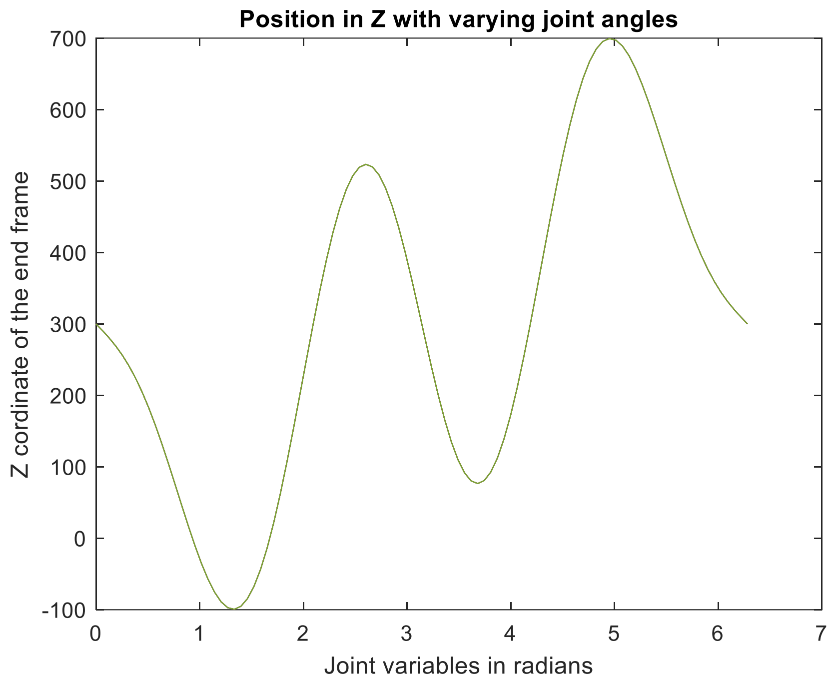


**Fig. 6: Variation of Z with Joint Angles**

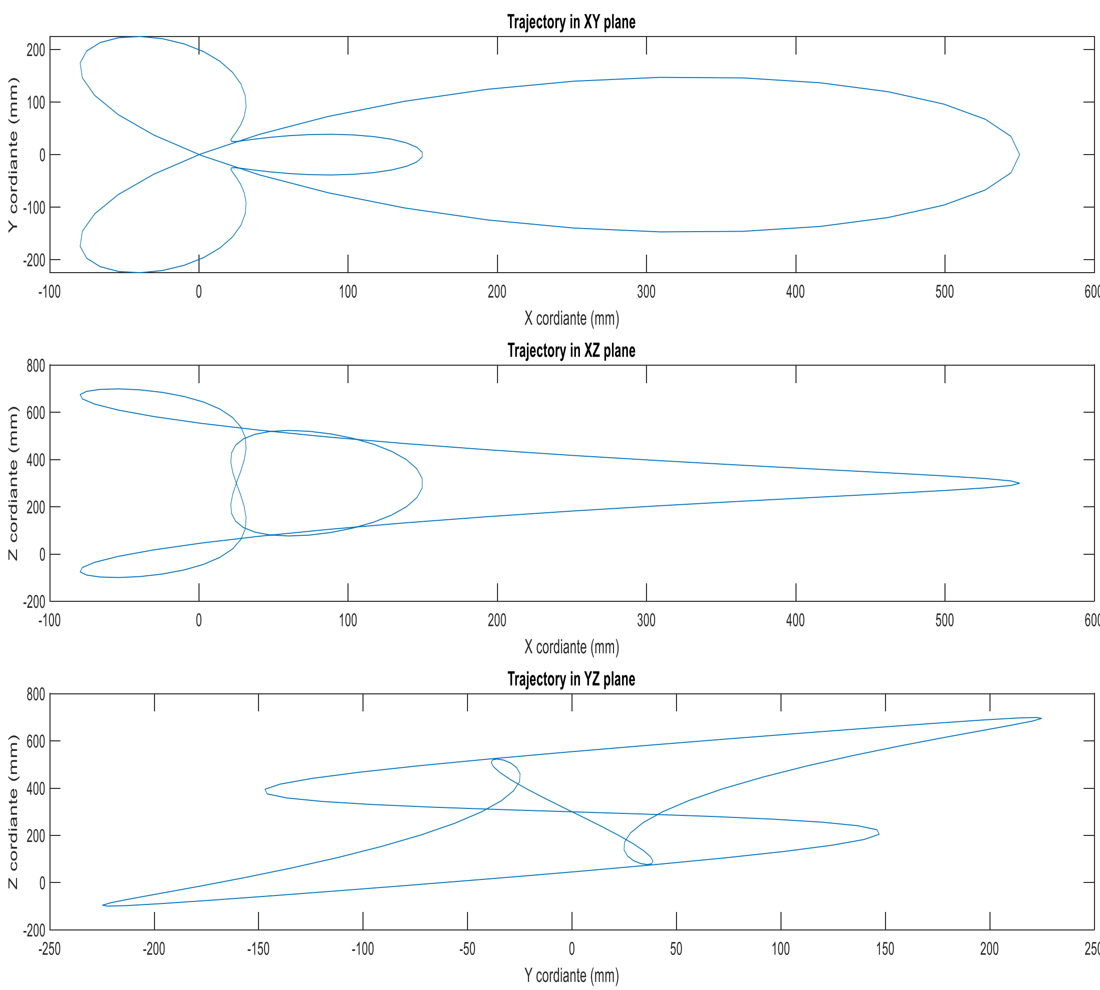


**Fig. 7: Trajectory Generation**

**Inverse Kinematics (IK)**

Inverse kinematics makes use of the kinematics equations based on the geometry to determine the joint parameters that provide a desired position and orientation of the robot's end-effectors. The unknown joint parameters here are $\theta_1^*$ , $\theta_2^*$ , $\theta_3^*$ , $\theta_4^*$ and $\theta_5^*$ while the known parameters are the end effector coordinates $X, Y, Z$ and the orientation R.

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} & X \\ R_{21} & R_{22} & R_{23} & Y \\ R_{31} & R_{32} & R_{33} & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The unknown parameters can be calculated algebraically by solving the above matrix

$$X = C_1 * (d_5 * C_{234} + l_3 * C_{23} + l_2 * C_2)$$
$$Y = S_1 * (d_5 * C_{234} + l_3 * C_{23} + l_2 * C_2)$$
$$Z = d_1 - l_2 * S_2 - l_3 * S_{23} - d_5 * S_{234}$$

Using the end effector coordinates, joint variables for the left manipulator can be found as

$$\theta_1^* = atan2(Y, X) \; or \; \theta_1^* = atan2(-Y, -X)$$

$$\theta_2^* = atan2(m * (l_2 + l_3 * C_3) - n * l_3 * S_3 \, , m * l_3 * S_3 + n * (l_2 + l_3 * C_3))$$

$$\theta_3^* = \mathrm{acos}(\frac{m^2 + n^2 - l_2^2 - l_3^2)}{2 l_2 l_3}$$

*where* $m = d_1 - d_5 * S_{234} - Z$ *and* $n = X * C_1 + Y * S_1 - d_5 * C_{234}$

$$\theta_4^* = \theta_{234}^* - \theta_2^* - \theta_3^*$$

$$\theta_5^* = atan2(R_{32}, -R_{31})$$

**SCOPE OF THE PROJECT**

This project is divided into 4 Milestones as follows:

1. <u>Designing the CAD model</u>
   Due to easy unavailability of a CAD model of a 5DOF robot arm, A model of the same was designed using CAD software (Solid Works) and was replicated for another robot. Since I was unaware of this software and was using it for the first time, a huge amount of time and phase of this project was used to design the CAD model. After multiple failures, a complete CAD model was designed with assumed dimensions and assumptions.

2. <u>Modeling the CAD model (MATLAB)</u>
   Even after the CAD model was successfully designed, it was not ready for exporting or importing to the MATLAB due to constraints error. A new fully defined model was sketched in SOLIDWORKS which again although was complete and successfully imported to MATLAB, was not able to give the desired results

due to constraints. Due to the lack of availability of time, I resorted to using stick figures for modeling and simulation. Peter Corke was used as a last resort to display some figures and simulation.

3. Simulation(MATLAB)
   Due to multiple failures with importing CAD model, Peter Corke was extensively used to simulate the model. All the mathematical parameters and equations were converted to a form understood by robotics toolbox by Peter Corke.

4. Validation
   The model was validated by comparing the values obtained by the trajectory simulation in Peter Corke and mathematical model using kinematics. Accuracy was found to validate the model and the equations.

**APPROACH**

The approach of this project has been very simple and focused from the beginning, however, due to multiple failures and constraints, time allocation for each milestone varied greatly. The desired approach was to design a model in CAD software, import it to the MATLAB and simulate it using Simulink. Validation was to be done either by a reference paper or by geometry.

However, this approach turned out to be time-consuming and an impasse was reached after an instant. So, a new approach was chosen to complete this project. It involved designing a stick figure using RVC toolbox, simulate it generating its trajectory and validate it by comparing the desired and obtained values for the model accuracy.

**SIMULATION**

A simulation is an imitation of the operation of a real-world process or system (Motion of cooperative robots). The act of simulating something first requires that a model is developed representing the key parameters and the functions of the selected system. The model represents the system itself whereas the simulation represents the operation of the system over time.

Although all the algorithm and codes were written in MATLAB to obtain the plots and solutions to equations, Simulation was done using RVC (Robotic Vision & Control) toolbox developed by Peter Coke. The main purpose of the simulation was to compute a visible and synchronous trajectory of the Cobots with a virtual object (not shown). As assumed, there will be no obstacles in the workspace and trajectory will be computed for different positions using the kinematics model.

For simulation, all the variables and parameters were converted and written in a form readable by the toolbox. Some conversion factors were applied, and some assumption was made to obtain the desired result and successful simulation was obtained.

One of the important conversion factors that were applied is

- $\theta_4^* - 90$ from the DH table was written as $-pi/2$ where all the other joint variables were written as $0$
- All the lengths and translation in the DH table were in $mm$ while writing the kinematics equation but they were represented as just units in RVC toolbox.
- Self and cross-collision were completed ignore while simulating the model

Converting joint variables from Mathematical Model to RVC toolbox

| End Effector position (X Y Z) | Joint Angles in Kinematics (q1, q2, q3, q4, q5) | Equivalent Joint angles in Peter Corke |
|---|---|---|
| (550, 0, 300) | (0, 0, 0, 0, 0, 0) | (0, 0, 0, -pi/2, 0) |
| (0, -350, 100) | (Pi/2 ,pi/2, pi/2, 0 ,pi/2) | (Pi/2, pi/2, pi/2, -pi/2, pi/2) |
| (318, 0, 38) | (Pi, pi/2, pi/3, pi/4, pi/6) | (Pi, pi/2, pi/3, -pi/4, pi/6) |
| (47.5, 0. 547.5) | (0, -pi/4, -3pi/4, -3pi/4, 0) | (0, -pi/4, -3pi/4, pi/4, 0) |

The simulated results can be of two types (but only one is presented here):

- A smooth parallel trajectory showing both the manipulators synchronously moving the same object.
- A smooth sequential trajectory showing both the manipulators picking the respective required object/tool from their individual workspace and performing the desired task in the shared workspace. Since modeling of a gripper is out of the scope of this project, Picking and performing is shown by simply being able to reach a specified point where that object is either placed or needs to be acted upon.

Sequential Trajectory is presented in this project i.e. first robot one generates a trajectory from its home point to a goal point followed by the second robot generating a new trajectory from its home point to the same goal point. There can be multiple initial configurations with these models but only three were considered and one was used to simulate the trajectory.

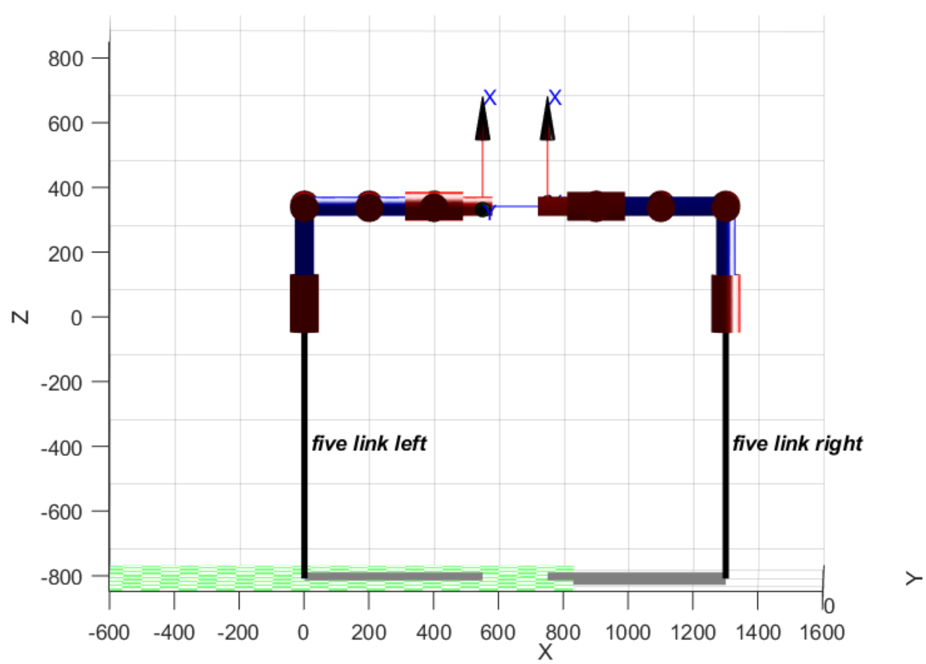


**Fig. 8 : Initial configuration 1 (used)**

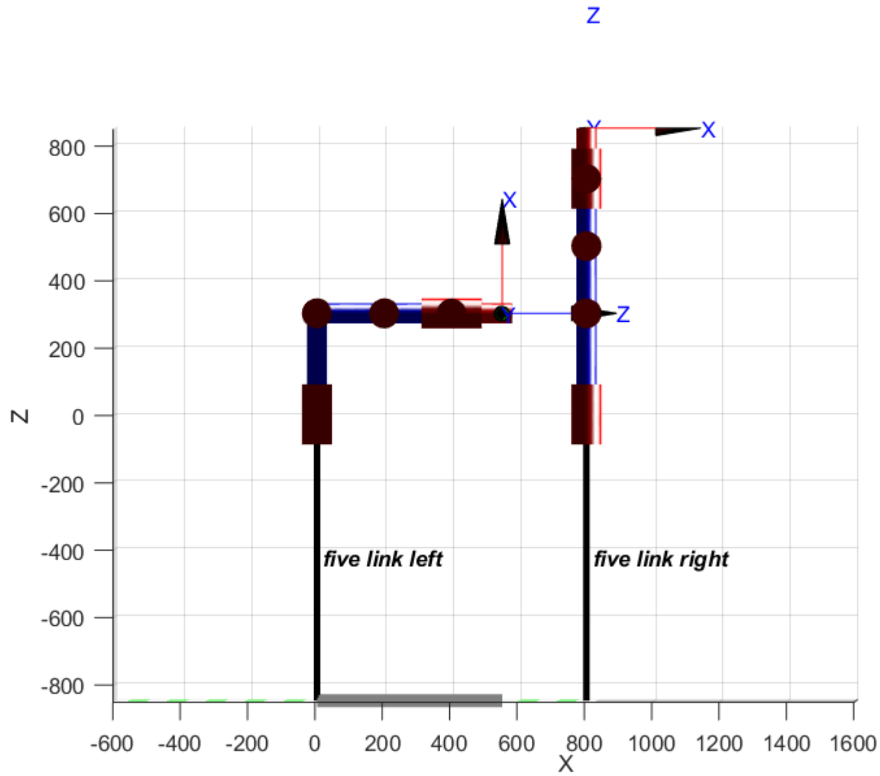


**Fig. 9 : Initial Configuration 2**

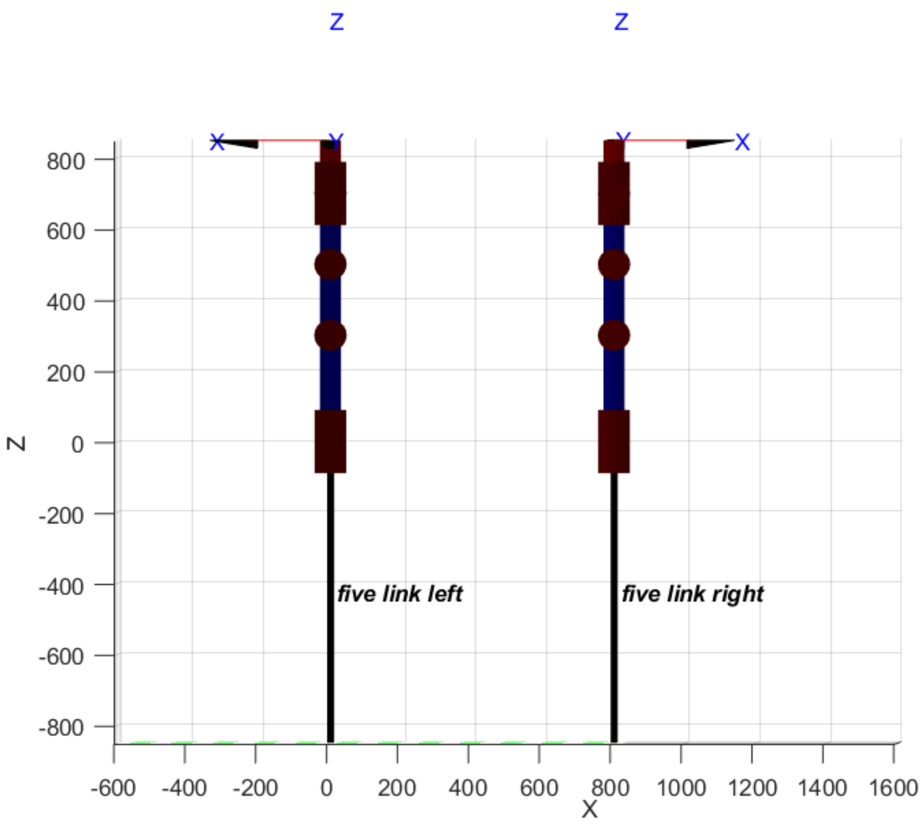


**Fig. 10 : Initial configuration 3**

Simulation Process

Simulation is the most exciting and important part of any system modeling because it gives a better idea of what the process looks like. Below are the steps that were followed to simulate the cooperative Robots-

- The initial and Final configuration is given to both the robots while masking the unnecessary details
- Initially, the robot is in its previous configuration. As the simulation runs, first the robot returns to its specified home position and then proceeds to the goal position. For the same process, the left robot moves first followed by the right robot.
- The simulation runs until both the robots have reached its final specified configuration
- The joint angles that were taken to obtain a trajectory can be obtained easily.

Simulation Results

A simulation was done for a workspace that is common to both the arms. Due to this constraint, the possibility of possible configuration reduces drastically. Also, since only integers are considered for this project, the possible choices are reduced to a small number.

Nonetheless, numerous pairs exist where a successful trajectory can be obtained but for the scope of this project, only a few were tested, and a successful trajectory was obtained for the following Test Data –

| **Left Robot configuration (X,Y,Z)** | | **Right Robot Configuration (X,Y,Z)** | |
|---|---|---|---|
| Initial | Final | Initial | Final |
| (550,0,300) | (350,0,300) | (250,200,300) | (350,200,300) |
| (550,0,300) | (400,0,450) | (250,200,300) | (400,200,450) |
| (550,0,300) | (200,300,550) | (250,200,300) | (450,300,550) |
| (550,0,300) | (300,200,400) | (250,200,300) | (300,200,400) |
| (550,0,300) | (150,0,0) | (250,200,300) | (350,200,0) |

Note* - The (1D, 2D, 3D)object dimension is fixed such that one robot grasps it from one end and the other acts upon it from the other end. If grasping is not required, then both the robots can hold the object from opposite surfaces at the same point or hold the object from the same surface at different points.

**MODEL VALIDATION**

Validation of any system is as important as designing it. Without validating, no system can be implemented in the real world. Validation in a simple sense means “Did I build what I said I would?”. The process of validation aims at testing whether a right system was built and requires realization at the system/model level usually in the implementation phase.

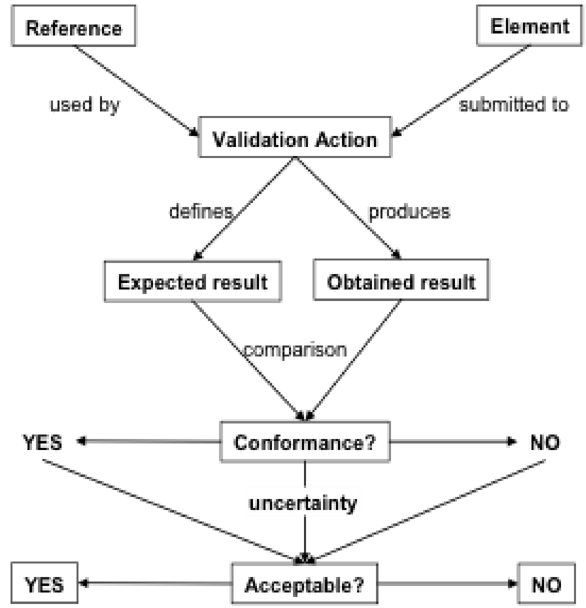


**Fig. 11: Validation Process Flow**

The model (only one robot) was validated as follows-

- The initial and final position of the end effector was assumed, and the trajectory was plotted
- The results were substituted in the FK model and the end effector position was recorded
- Accuracy between the two results was used as a factor to validate the model

**Validation for the Left Manipulator**

| **Joint Angles** | **End frame position (X Y Z)** | | **Accuracy** |
|---|---|---|---|
| [q1, q2, q3, q4, q5] | Forward Kinematics | Peter Corke (PC) | Avg. accuracy of X Y Z |
| [0, 0, 0, -pi/2, 0] | (550 0 300) | (550 0 300) | 100% |
| [0, -0.08pi, -0.05pi, -0.03pi, 0] | (449 0 326) | (450 0 300) | 95% |
| [0.18pi, 0.18pi, -0.27pi, -0.9pi, 0.81pi] | (285 206 394) | (300 200 400) | 97% |
| [-pi/2, -0.66pi, 0.708pi, 0.96pi , -pi/2] | (0 -97 595) | (0 -100 600) | 98% |
| [0.25pi, 0.27pi, 0.21pi, 0.52pi, 0.75pi] | (109 109 100) | (100 100 100) | 92% |
| [0, 0.43pi, 0.48pi, -0.42pi, 0] | (-298 0 53) | (-300 0 50) | 96% |
| [-0.8pi, 0.09pi, -0.27pi, -1.3pi, -0.2pi ] | (-294 -200 -500) | (-300 -200 500) | 98% |
| [0.31pi, -0.2pi, 0.23pi, 0.97pi, 0.68pi] | (202 298 548) | (200 300 550) | 99% |
| [pi/2, -0.047pi, 0.544pi, 1.55pi, pi/2] | (0 348 348) | (0 350 350) | 99% |
| [0, -pi, 0, -pi/2, 0] | (-550 0 300) | (-550 0 300) | 100% |
| | | **Model Accuracy** | **97.4%** |

Note* - All the positions are rounded to the nearest integer

## CONCLUSIONS

A CAD model was designed successfully to display the basic structure of the desired robot. A mathematical model was successfully accomplished and validated using the RVC toolbox. We were successfully able to simulate the Cobots and motion analysis was done to achieve the desired task.

## FUTURE SCOPE

This area of research is not much used or explored in the real world. However, the area of improvement remains the same as any other single manipulator, but for this specific project, the future scope will be-

i. Include obstacles in the workspace and plan for a collision-free trajectory
ii. Install onboard sensors and cameras allowing the robot to grasp the object from an unknown environment.
iii. The object to be moved may not be rigid, therefore dynamic modeling becomes a big area of research.

## REFERENCES


1. V.N. Iliukhin, K.B. Mitkovskii, D.A. Bizyanova, A.A. Akopyan, “The modeling of Inverse Kinematics for 5 DOF manipulator”, Elsevier Ltd, 2017
2. V. Deshpande, P.M George, “Kinematic Modelling and Analysis of 5DOF Robotic Arm”, International Journal of Robotics, Vol. 4, Issue 2, April 2014