Modelling, Simulation and Control of SCARA(RRP) Manipulator Arm

Drives, Controls and Modelling Laboratory (MTE 3161)

MINI PROJECT REPORT

submitted by

Bhavya Doshi –210929184

Shantanu B. – 210929106

Abhishek Biswas - 210929276

DEPARTMENT OF MECHATRONICS

MANIPAL INSTITUTE OF TECHNOLOGY MANIPAL - 576104

Nov, 2021

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

ABSTRACT

Modern companies rely heavily on robotic manipulator arms, and maximising their effectiveness requires an understanding of their dynamics and control systems. Considering this, the goal of this research is to use innovative instruments and methods to improve the RRP manipulator arm's performance and add to the current discussion over the use of control systems in industrial automation.

With the increasing use of robots in modern industrial production, it is of great significance to study the robotic arm as a main tool in this field. This paper investigates the modelling and simulation of a robotic arm for various applications. Using CAD modelling software such as SolidWorks,a SCARA CAD model is developed. Kinematics and dynamics model of the multiple degrees of freedom robot based on MATLAB and Simscape Multibody platform are established. The forward kinematics of the multiple degrees of freedom manipulator is analysed, and the joint angle is obtained by using the inverse kinematics and then the dynamics solution of the manipulator is derived. Also to get accurate results a PID controller is used to fine tune . Simulation trials and results are presented and discussed. This work provides a potential basis for the realization of robotic grinding and polishing in industrial field, which is of great significance for improving manufacturing efficiency, ensuring product quality, and reducing labor intensity of workers.

CONTENTS

List of Tables

LIST OF FIGURES

ABBREVIATIONS

SCARA- selective Compliance Assembly Robot Arm

RRP – Revolute Revolute Prismatic

IK – Inverse Kinematics

FK – Forward Kinematics

OEMS – Original Equipment Manufacturer

BLDC – Brushless DC

FOC- Field Oriented Control

DTC- Direct Torque Control

PID – Proportional Integral Derivative

CHAPTER 1

INTRODUCTION

Robotics has revolutionized industries by introducing automated solutions that streamline processes, enhance efficiency, and ensure precision in various operations. Within this landscape, manipulators stand as pivotal tools, offering a spectrum of possibilities in industrial settings. These versatile robotic arms have redefined manufacturing capabilities, allowing for tasks ranging from material handling to intricate assembly with unparalleled accuracy. Integrating seamlessly into production lines, manipulators operate tirelessly, contributing to heightened productivity while accommodating rapid reprogramming for evolving production needs. Moreover, their potential extends to collaborative applications, augmenting warehouse automation by efficiently handling tasks like sorting, palletizing, and material movement. Furthermore, by incorporating sensors and vision systems, manipulators not only bolster safety in hazardous environments but also present a pathway to integrate seamlessly with existing systems. However, despite their immense potential, challenges persist, particularly concerning cost-efficiency, complex control mechanisms, and the need for skilled operators. These challenges set the stage for innovative approaches, such as exploring alternative motor types like DC motors and simulating their performance, aiming to achieve comparable precision and control at a reduced cost, potentially revolutionizing the landscape of industrial manipulators.

1.1 Potential of Manipulators in Industrial robotics

Manipulators play a crucial role in industrial robotics, offering a high degree of flexibility and precision in performing a variety of tasks. Their potential in industrial settings is vast and continues to grow, contributing to increased efficiency, productivity, and safety. Manipulators are versatile robotic arms capable of executing a wide range of tasks. These tasks include material handling, assembly, welding, painting, and other repetitive or intricate operations in manufacturing processes.

These robotic arms can work continuously without fatigue, increasing the productivity of the manufacturing, giving out high precision and accuracy which is crucial for industry where assembly and delicate material handling is required. The added advantage it gives is that it can be reprogrammed easily allowing quick adoption go changes in the path and production requirement, this is essential for small-batch production and it can be integrate with collaborative robots enhancing thier potential in warehouse automation, with common application like loading and unloading materials, palletizing, and sorting.

Manipulator can be integrated with sensors and vision systems which can be used to enhance the safety, which makes it favourable to use it in hazardous enviroment and poses less risk to human life

1.2 Overview of SCARA(RRP Manipulator Arm)

The RRP manipulator arm configuration is a type of robotic arm with a specific arrangement of joints and links. In the RRP configuration, "R" stands for a revolute joint, and "P" stands for a prismatic joint. This configuration is characterized by the arrangement of these joint types in the sequence of joints along the robotic arm.The RRP Manipulator arm consist of 2 Revolute joints and 1 prismatic joint. The revolute joint allows rotational motion around a specific axis. In the RRP configuration, one or more revolute joints are typically present, enabling the robotic arm to rotate or articulate at specific points, one or more prismatic joints are typically included, providing the robotic arm with the ability to extend or retract along a straight path. The prismatic joint allows linear motion along a specific axis. An End Effector is connected to the Prismatic joint in this configuration. The end effector can typically be of any type. There is different type of end effector used in the industry according to the type of application such as grippers, welding arm,cutting tools, magnets etc.The arrangement of these joints, coupled with interconnected links forming the structure of the robotic arm, determines the reach, mobility, and overall functionality of the RRP manipulator.

1.3 Objective and Challenges

In this Project we Aim to model, simulate, and control the RRP arm that is been designed in SolidWorks achieving the Industry Standard. There is a rise of Industrial Manipulators in the Indian Market, increasing OEMs are trying to collaborate this kind of robot in the supply chain, but it is expensive, and it is difficult with not great efficiency and control. It is expensive because BLDC motors are used for actuation of the joints, while they are highly accurate, they require complex control and are expensive.

There are too many challenges that are faced while using Manipulator, since existing models uses BLDC motor, it has Complex control which requires complex algorithms like FOC and DTC, moreover it also requires skilled workers to operate it. Programming manipulators for specific tasks can be intricate. The complexity increases when dealing with tasks that require a high degree of precision, adaptability to variable conditions, or collaboration with other robots or human operators. Since MSME still work with DC controls it will be a task for them to change it to BLDC controls. Hence we aim to use DC motor and simulate and tune the motors on the manipulator such that it can give a precise control, accuracy, and position, such that these simulation can help to compare and gain results of DC motor usage in Manipulator.

1.4 Project work Schedule

1.4.1: Modelling

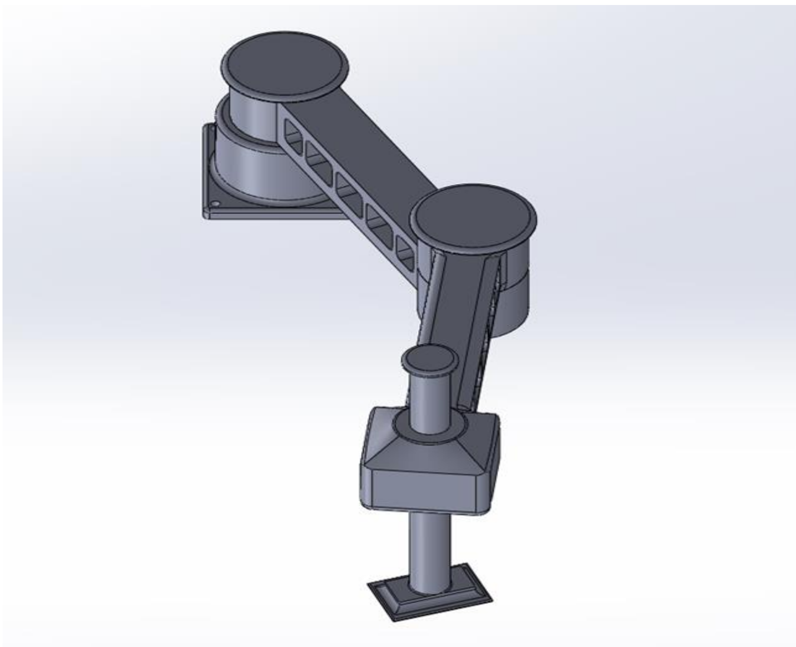

Fig 1 CAD Model of SCARA

Using SolidWorks, a 3D modelling system renowned for its accuracy, the RRP manipulator arm was painstakingly designed as part of the project's approach. The arm has three links including two revolute joints and one prismatic joint. The swivel is capable of 360 degree rotation while the revolute between the first and second link can move –160 to 140.. This is the soft limit for the second link as it is constrained by the first link. The prismatic joint has a stroke length of 140 mm. The workspace of the arm is a cylinder with a diameter of 500 mm. All the link specifications are modelled and take reference from industrial scara arms.

1.4.2. Importing Model in Matlab

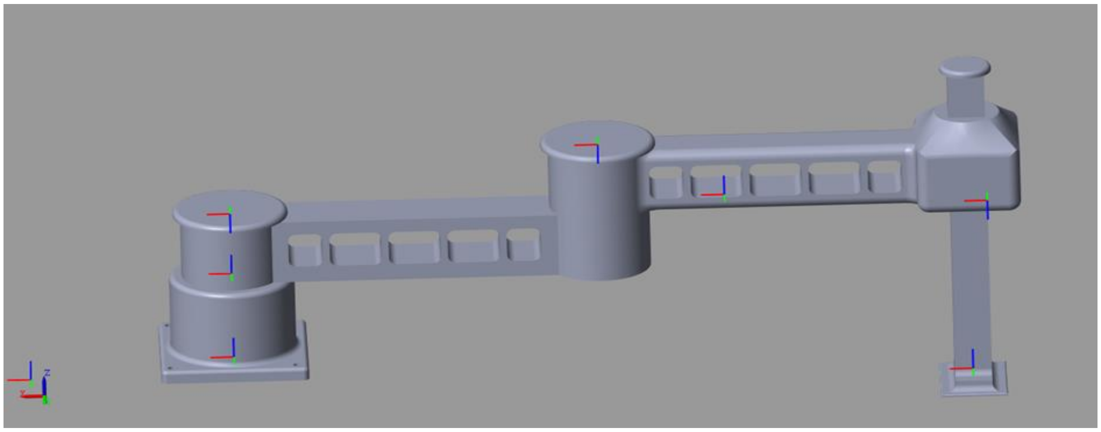

Fig 2: Coordinate System

In the development of the CAD model in Solidworks Software, a plugin is to be added to MATLAB which is Simscape Multibody Link Plugin which links the Solidworks Model to MATLAB, then we proceed to export the CAD model to MATLAB by transforming it into a XML file and feeding the data into Simscape.

1.4.3 Adding Other subsystems and function blocks

After the import of the Robot model from SolidWorks to MATLAB it was time to add other function blocks to complete the control system of the Model. We added a block of Inverse Kinematics, using a function block. The function block had the equation of Inverse Kinematics while another function block was created for Forward Kinematics. The input coordinates would be passed through the Inverse Kinematics Block which would processed and angle values would be given in the output which would be passed to the robot model through the PID block(which was yet to be tuned) and the angle output from the robot model was to given to the forward kinematics block which would be visualized in the X-Y Graph.

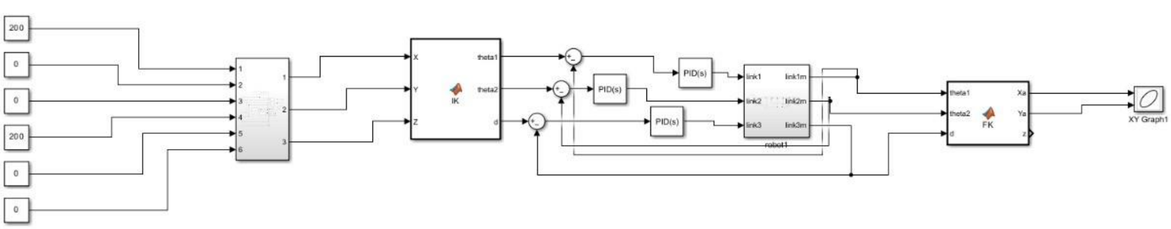

Fig 3. RRP Manipulator Arm Model

1.4.4 Simulating and PID tuning

We now had to tune the parameters accordingly to get a perfect output. First we tuned the Kp value of the firstnt and then other joints. After reaching to a steady state we added Integral and Derivative accordingly as we required. The PID tuning is based on Trial and error method itself. While it gave some reliable results which had tolerance value in y direction in order of 10^-12 order. We achieved a good result.

1.4.5 Selecting the Motor and importing in the motor

We decided to model our arm controls using PMDC servo motors as its control is simple and understandable with enough resources to take guide from. To make a more robust model of the motor we made the motor circuit from scratch including all the physical parameters like shaft inertia, gear ratio, and the motor circuitry itself. This approach gives us a far more precise control over our simulation rather than using the DC machine block in Matlab. We have modelled our motor based on IG-45 which is a robust industrial grade PMDC motor that suits our project.

1.4.6 Tuning and Analysis

The joint error which is the difference between desired and observed trajectory is fed to the PID controller, after this, the signal goes to the motor after converting it to a imulink signal. This signal/current controls the actuation of the motor. Manual tuning of PID controller is the most widely used and accepted method. Rather than blindly changing Kp, Ki and Kd values there are some procedures to follow to make it easier to reach the optimum values which will be explained in the methodology section. After tuning the controllers the simulation result gave an optimum tolerance of 10, to further improve the tolerances it seems our approach has reached its limit and we have to apply new and sophisticated methods to make it more realizable.

CHAPTER 2

MATERIALS AND METHODS

2.1 Software and Packages used

1.Matlab

2.Simulink

3.Simscape

4.Simscape Multibody

5.Solidworks

2.2 Modelling in Solidworks:

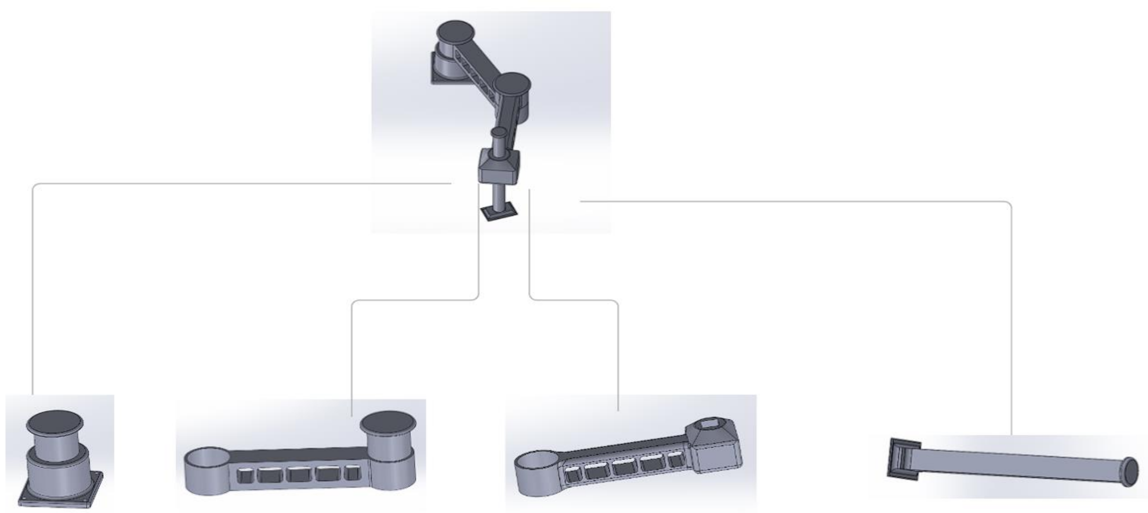

Fig 4. Links of the CAD Model

PARTS	Tools used in solidworks
Swivel Base	Boss-Extrude, Fillet
Link 1	Boss-Extrude, Cut-Extrude Fillet
Link 2	Boss-Extrude, Cut-Extrude Fillet, Loft
Link 3	Boss-Extrude, Fillet

Table 1: Links of the CAD Model

2.3 Simulation and Tuning

2.3.1 Exporting in Matlab

After the Modelling of the Manipulator was done, we need to export it in MATLAB in SimScape Multibody Toolbox of Simscape Library.

In the matlab workspace type the command smlink_linksw this connects the matlab workspace to solidworks which allows the plugin to import the cad file as a simscape model. To import the converted xml file to simulink type the command-

Smimport (‘scara.xml’)

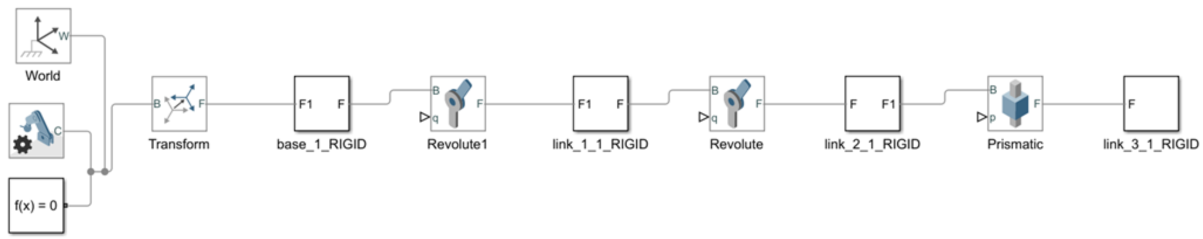

Fig 5. Simscape Model

Now we can continue on with adding functional blocks to it.

2.3.2 Adding Inverse and Forward Kinematics Block, Coordinate Generator

```
function [Xa,Ya,z] = FK(theta1,theta2,d)

l1 = 250;
l2 = 250;

Xa = l1*cos(theta1) + l2*cos(theta2 + theta1);
Ya = l1*sin(theta1) + l2*sin(theta2 + theta1);
z = d;
```

Fig 6. Forward Kinematics Function block

```
function [theta1,theta2,d] = IK(X,Y,Z)

l1 = 250; % length of first arm
l2 = 250; % length of second arm
if(Z<-140)
    Z = -100;
end
c2 = (X.^2 + Y.^2 - l1^2 - l2^2)/(2*l1*l2);
s2 = sqrt(1 - c2.^2);
theta2 = atan2(s2,c2);
theta1 = atan2(Y,X)-atan2(l2*sin(theta2),l1+l2*cos(theta2));
d = Z ;
%data1 = [X(:) Y(:) THETA1(:)]; % create x-y-theta1 dataset
%data2 = [X(:) Y(:) THETA2(:)]; % create x-y-theta2 dataset
```

Fig 7. Inverse Kinematics Function block

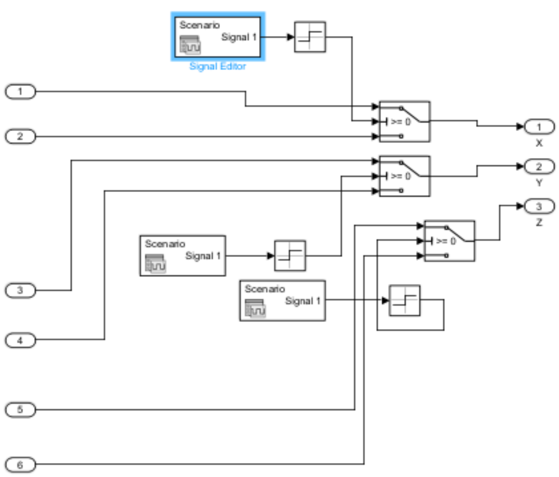

Fig 8. Coordinate Generator Subsystem

The IK and FK function blocks are created using function input from the library Browser, while the equations were inserted in the function block. The Coordinate Generator Subsystem is created to send desired coordinates in a particular time. It is done by using signal editor blocks and Switches. After a particular trigger time period the signal changes this leads to switching of coordinates and hence the manipulator travels to different Coordinates.

2.3.3 PID Tuning

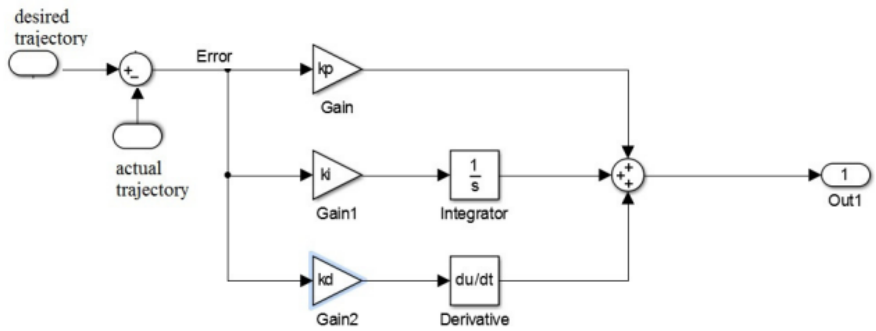

Fig 9. PID Algorithm

After adding the block PID blocks were added in between the IK Block and Robot model Subsystem. We had to tune the model. The method we chose was to first tweak the P value parameter in the different PID blocks and attaint a steady state error. To decrease the error we start to tweak the I values but there are high chances that it will lead to formation of high oscillations . These are reduced by the introduction and tuning of D paramter of PID tuner.

The whole result is visualized in the XY graph as the OUTPUT, to visually check how much is it tuned.

At first we only simulated it in moving in x direction to check how much errror we get and then tuned it. This was done similarly for Y axis and then we gave it a circular trajectory. After tuning it the values obtaines were :

Controllers	Kp	Ki	Kd	Filter coefficient
PID 1	5	5	0.2	100
PID 2	80	80	1	100
PID 3	0.001	0.001	0.001	100

Table 2: PID Values before adding Motors

2.3.4 Overview and initial analysis

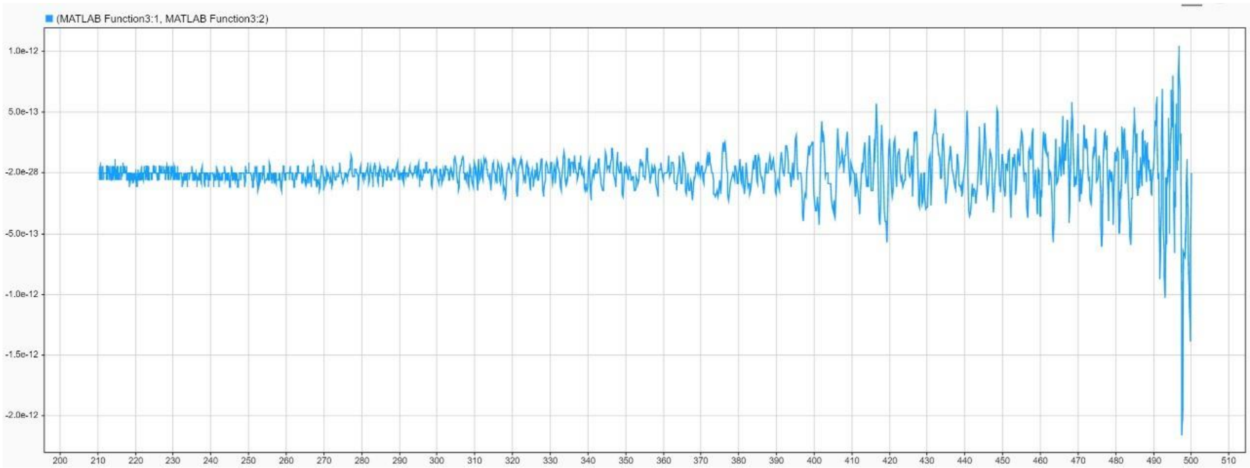

Fig 10. Results of simulating the RRP Model in X direction

Note: the tolerance values are in order of Magnitude of 10^-12 which are extremely good results, another thing to note here is that this is X-Y graph and hence, it starts with x coordinate of 500 to 200 units.

Similarly we tuned the Y coordinates and gave a circular trajectory to the robot,

The results were following:

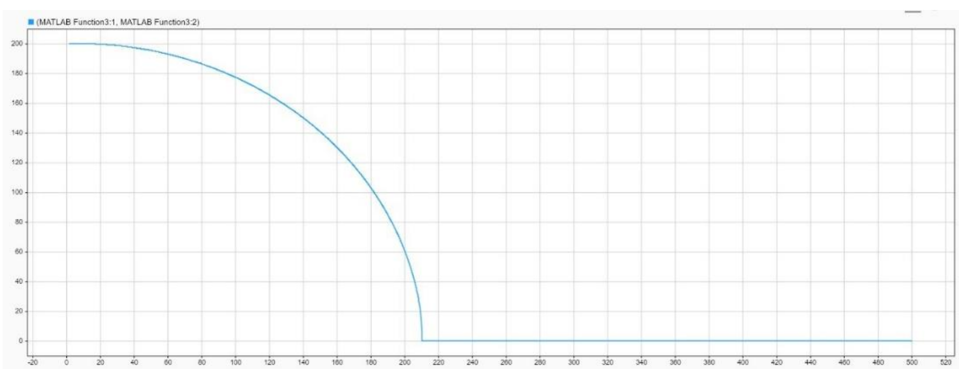

Fig 11. Ideal result of simulating the RRP Model when it moves (500,0) -> (200,0) -> (0,200)

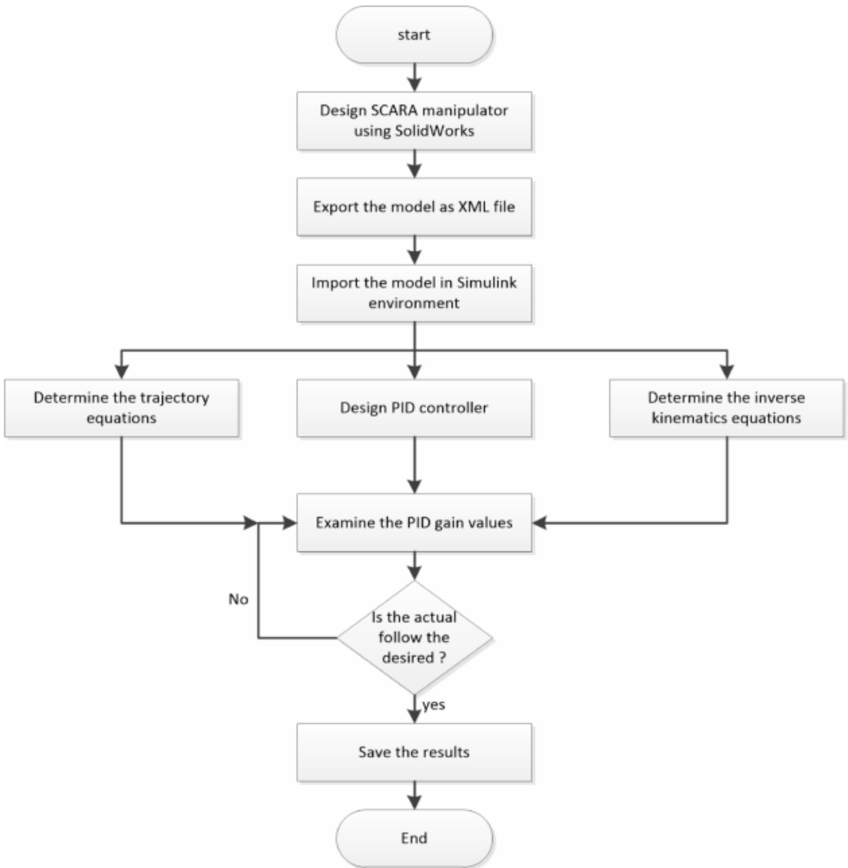

Fig 12. Overall working of our model

2.4 Selecting Motor and simulation

For our project we have used IG-45 motor which has a gear reduction of 125:1 and a no load speed of 7000 rpm. It is an industrial grade planetary PMDC geared motor known for its robustness and efficiency. It is cost effective and robust and has great accuracy and control while its motor drivers are also easily available and comparatively lighter. In the simscape model of the scara arm we have to choose position and velocity sensing while changing the torque to provided by input as it will now be given by the motors.

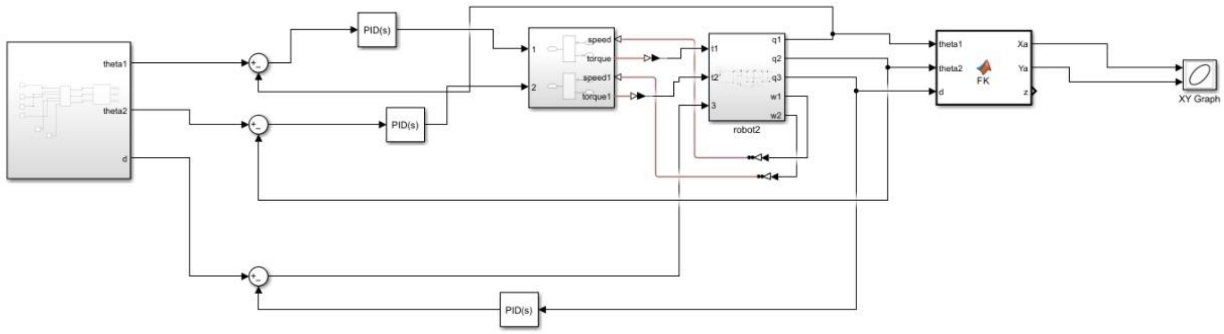

Fig 13. Final model with DC motors

We designed the circuit of the DC motor from scratch rather than using the DC machine block. This gives us far more precise control over the physical parameters of motor, shaft and gears. It also ensures that it simulates real life scenarios as close as possible.

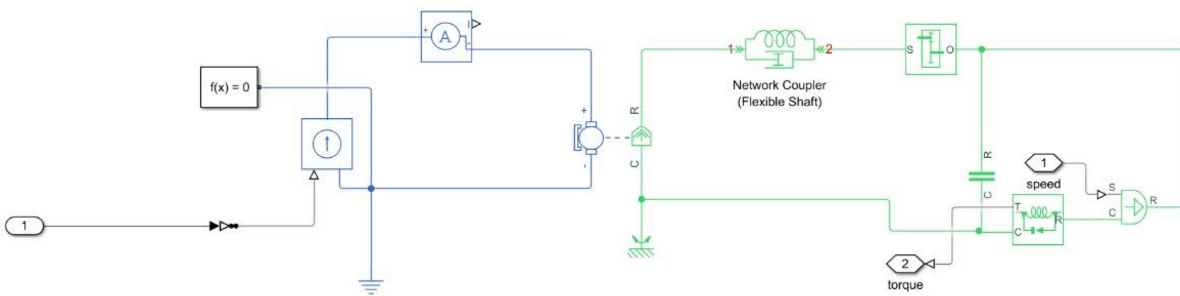

Fig 14. Circuit of DC Motor

The joints positional errors is fed to the PID controllers . After the tuning is done it is then fed to the current source in the circuit which then controls the output torque. The positional error is converted to a current source via simulink-P5 converter.

PID tuning was done meticulously to ensure optimum trajectory in the output. First we started with Ki and Kd as zero. We varied values of Kp until oscillations start to appear. Then we took half the value of this Kp then varied Ki in small magnitudes. The final PID values obtained were:

Controllers	Kp	Ki	Kd	Filter coefficient
PID 1	0.099	0.19	0.2	300
PID 2	0.1	0.4	1	300

PID 3	0	0	0	100

Table 3: PID Values after adding the Motors

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Results

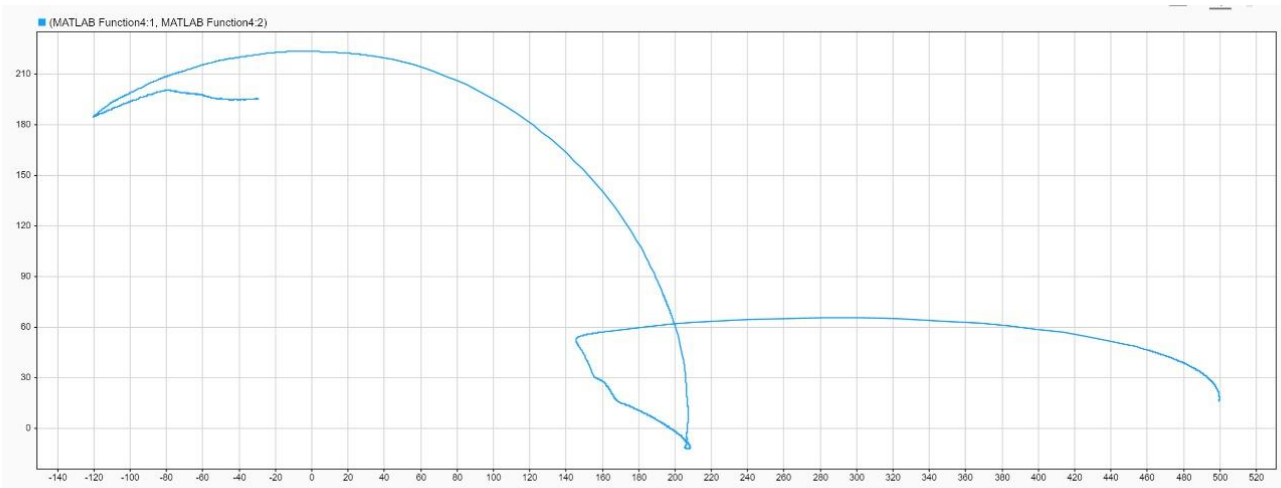

Fig 15. Results of simulating the RRP Model when it moves (500,0) -> (200,0) -> (0,200)

This was the final result after adding the motor. The graph is much similar to the graph we tuned it without adding the Motor.
We have a error of 15 units in Y-axis while 40 Units in X-axis.

3.2 Discussion

We can infer from the graph and the errors that the Y-axis has been accurately simulated while X-axis needs to be work on. This means that the manipulator on X-coordinate had a higher oscillation before it could be stablized. Precisely tuning the X-coordinate can help the graph look much better and give out good results. But it can be concluded that if more advance control systems are used these can be used in Industry scenrio.

It can be also seen that changing the gear ratio actually affects the position of the manipulator arm. If the Gear ratio is increased the speed will get reduced and better control and results can be seen.

CONCLUSION

Precise motor control is essential to robotics because of the complicated interaction between motors and controllers, as demonstrated by the difficulties simulated by error amplification. Robots are essential components of many different kinds of companies and sectors, and their ability to do tasks quickly and precisely depends greatly on their motor control systems.

Within this framework, the application of advanced control techniques such as fuzzy logic control presents itself as a viable path towards improving DC machine performance. These improvements are essential for fulfilling the changing needs of robotic systems and guaranteeing their efficacy in practical uses.

Simulations in MATLAB are critical for studying and developing motor control techniques. Before implementing these simulations in real-world systems, engineers can repeatedly optimize motor responses and reduce mistakes by using them as a controlled environment for evaluating and optimizing control algorithms. Developing more reliable and effective robotic systems is made possible by the capacity to simulate intricate relationships between motors and controls using MATLAB simulations.

In order to obtain a more thorough comprehension, additional examination may entail adjusting the voltages and currents provided to the motor, more closely emulating actual situations. This investigation of various operating environments would offer priceless insights into how the motor behaves under various circumstances.

Furthermore, broadening the scope of the research to include motors other than brushed DC motors, including stepper or servo motors, could potentially yield more definitive findings. This comparative analysis would help us comprehend motor control systems from a more comprehensive perspective by illuminating the distinctive qualities and applicability of different motor types.

REFERENCES

1. Donald E. Knuth (1986) *The TEX Book*, Addison-Wesley Professional.

2. Leslie Lamport (1994) *LATEX: a document preparation system*, Addison Wesley, Mas- sachusetts, 2nd ed.

3. Das, Taylan & Dülger, Lale. (2005). Mathematical modelling, simulation, and experimental verification of a scara robot. Simulation Modelling Practice and Theory. 13. 257-271. 10.1016/j.simpat.2004.11.004.

4. J. Liu and Q. Luo, "Modeling and Simulation of Robotic Arm in MATLAB for Industrial Applications," 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 2019, pp. 346-349, doi: 10.1109/IHMSC.2019.00086.

5. https://www.youtube.com/watch?v=s3o5zRs6Ddc

6. https://www.youtube.com/watch?v=pDiwAA1cnb0&t=2790s&pp=ygUgbW9kZWxsaW5nIG9mIHNjYXJhIGFybSBpbiBtYXRsYWI%3D

7. https://nrsyed.com/2018/01/21/how-to-read-a-dc-motor-datasheet/

8. https://robokits.download/downloads/RMCS-1212_V2.pdf