

**MCTE 4352**

**ROBOTICS**

**PROJECT**

**SECTION 1**

**SEM 2 || SESSION 20/21**

|  |  |
| --- | --- |
| **NAME** | **MATRIC NO.** |
| **ANAS HAFIFI BIN ‘ARIFIN** | **1726805** |

|  |  |
| --- | --- |
| **DATE OF SUBMISSION**  **10 JUNE 2021** | **TOTAL MARKS** |

1. **INTRODUCTION**

Modern day robotics are evolving as time goes on. Its design and function ability are increasing to a point that it can almost imitate the action of a real human being. The industries also not left behind in utilizing the robotics in their production.

For this project, we are going to design a simple 3 Degree of Freedom planar robot. It can be a success if the robot is able to write the name ‘ANAS’ in the font of Brush Script MT on an A4 paper. The calculation of the theta that we calculate via normal calculation should be the same as the one in the RTB toolbox.

**2.0 OBJECTIVES**

1. To simulate 3DOF Robot by using Matlab Add Ons, RTB Toolbox
2. To compare path created by the RTB Toolbox in Matlab and line graph created by the Excel
3. To verify the values of theta 1 & theta 2 using the inverse kinematics in Excel and the one generated by the MATLAB
4. To simulate motion of 3D planar Robot into the last name Anas in the font of Brush Script MT

**3.0 PROCEDURE**

1. First step is to get the x and y coordinates of the name ‘ANAS’.
2. The image of the word ‘ANAS’ is then put into a software that will give the coordinates in X and Y coordinates
3. The coordinates are inserted into EXCEL and plotted into a graph to view the final sketch.
4. A DH representation table of the robot is done.
5. Use manual pen and paper base calculation with the X and Y coordinates to find the theta1 and theta2 using the inverse kinematics formula.
6. Coding for the RTB toolbox is done using the same coordinates.
7. Final output of the system is observed.
8. The value of theta1 and theta 2 is made into comparison.

**4.0 RESULTS**

****

Figure 1: Intended last name

The image file of the demanded last name is generated using Microsoft word with the Brush Script MT font.

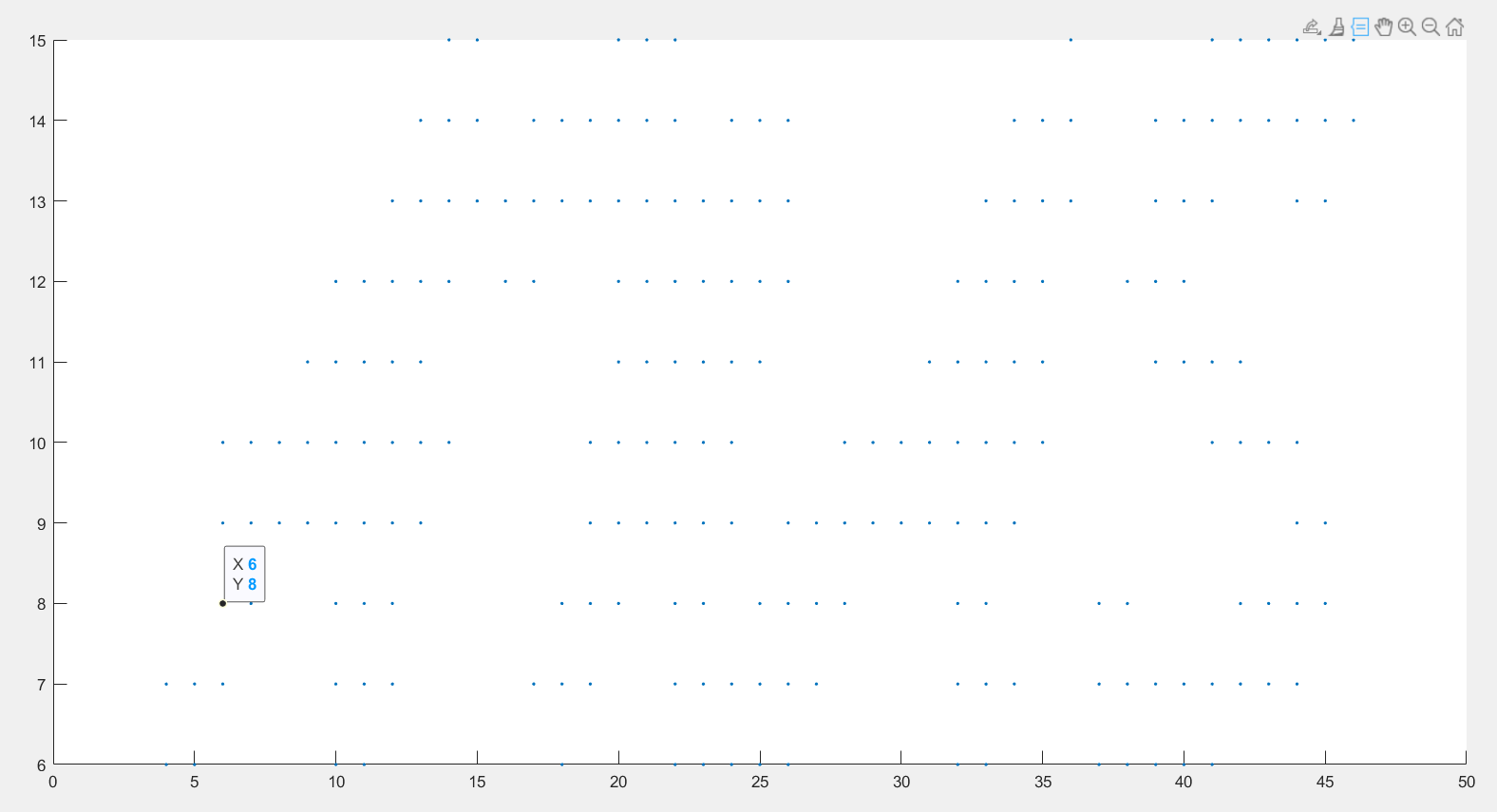


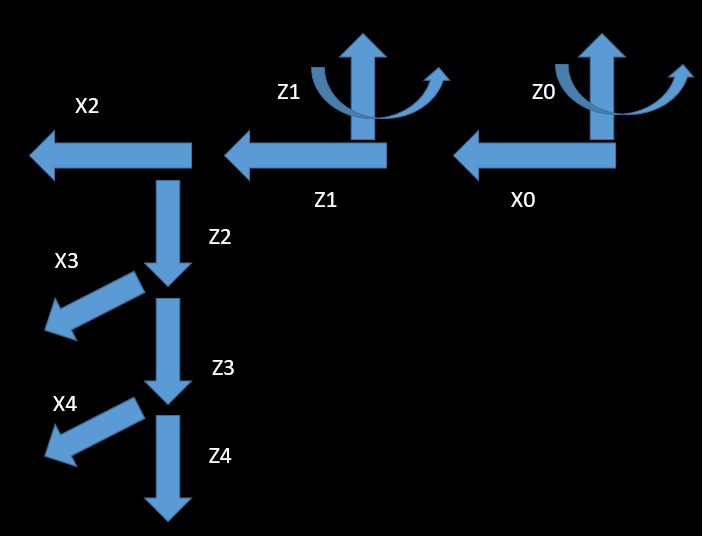
Figure 2: Plotting using software

An open source software is used to convert the image file into plots in X and Y coordinates. However, the X and Y coordinates are given in a huge amount, thus manual selection is done on the coordinates for each letter.

**Generated Motion Path Graph.**

Figure 1 Plot of Last Name, 2D Planar

The coordinates is then verified using simple Microsoft Excel to generate the plotted graph.



**Figure 1: DH ASSIGNMENT**

**DH PARAMETERS TABLE**

**LET A1 = 80, A2=80, D1=20, D3=2.5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Joint | a | d | 𝛼 | 𝛳 |
| 0-1 | 80 | 20 | 0 | 𝛳1 |
| 1-2 | 80 | 0 | 180 | 𝛳2 |
| 2-3 | 0 | 2.5 | 0 | -90 |
| 3-4 | 0 | 0 | 0 | 0 |

**EQUATION TO CALCULATE THETA1 AND THETA 2**

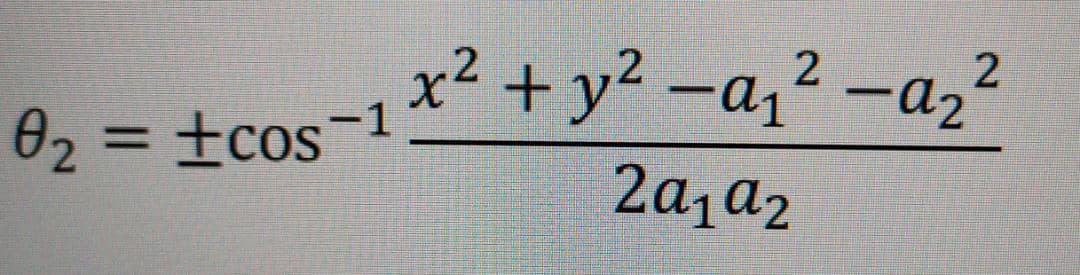


Figure 2: Formula to calculate theta2

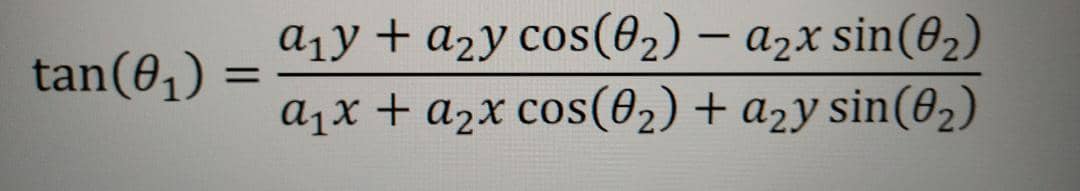


Figure 3: Formula to calculate theta1

Inverse kinematics to find theta1 and theta 2 using Excel

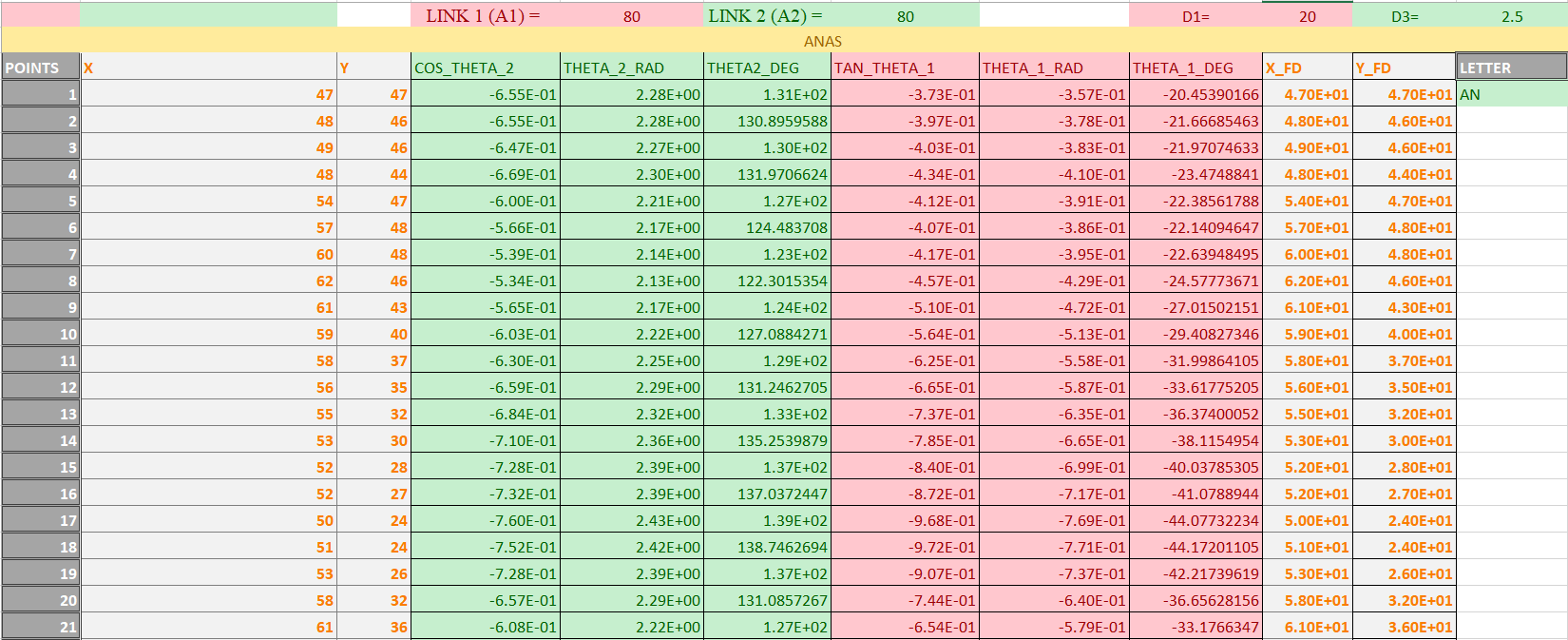
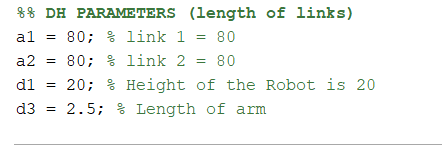


Figure 4: Pen and paper based using excel

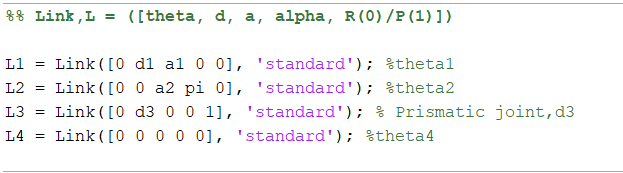
A manual analytics using Microsoft Excel to find the value of theta1 and theta2. Inverse kinematics and forward kinematics is done for each coordinates to verify the calculation.

CODING AND SIMULATION

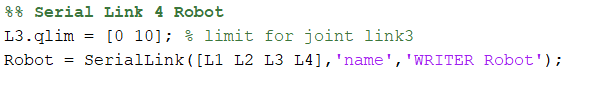
CODING MATLAB



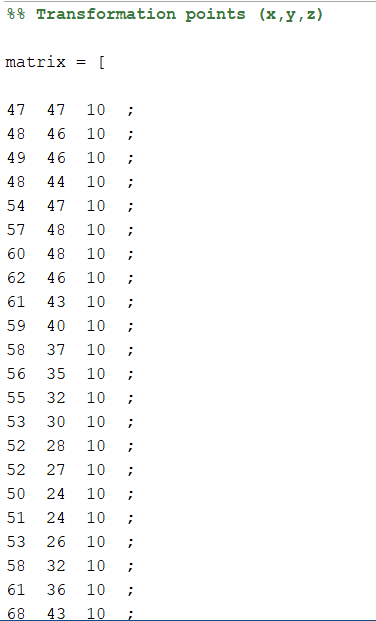
At the start of the coding, we need to initialize the DH parameters.



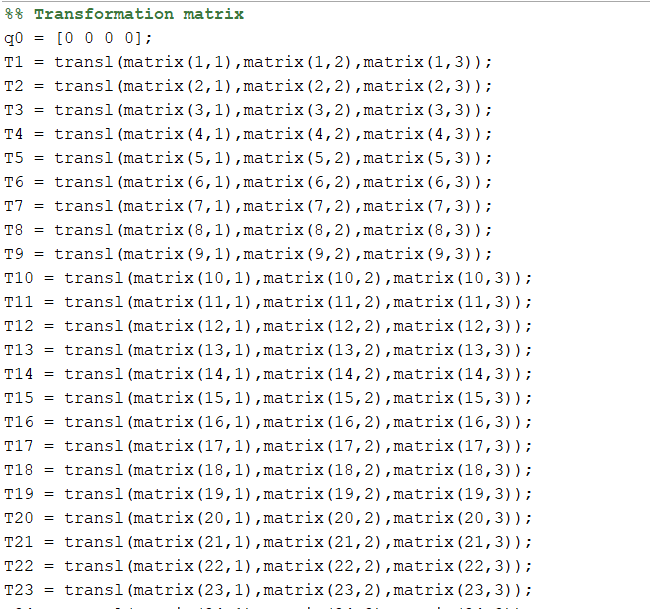
Then, we need to assign the links accordingly with the theta, d, a, alpha and finally whether the robot is revolute or prismatic joint. For this case we are going for a prismatic joint for link 3.



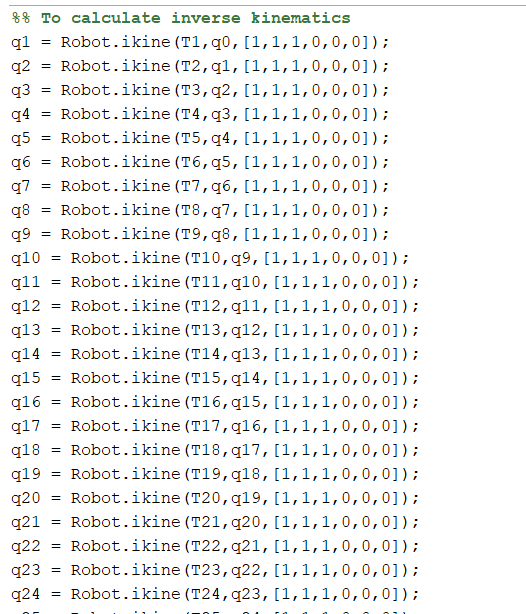
Next, we need to limit the joint of link 3 since we set it to be prismatic joint. We also need to use serial link function to set the links to the robot and name it.



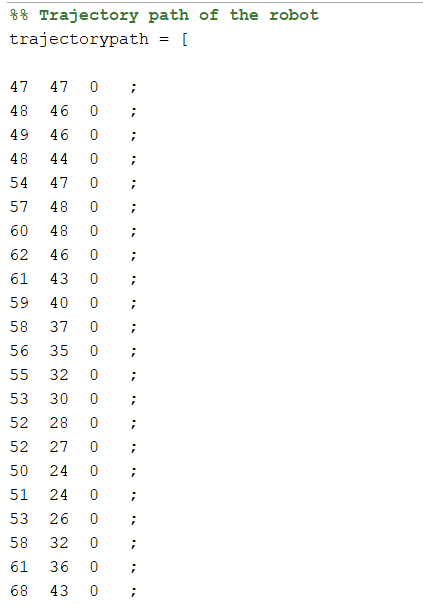
Moving on, we need to assign the coordinates that we have found in the matrix.

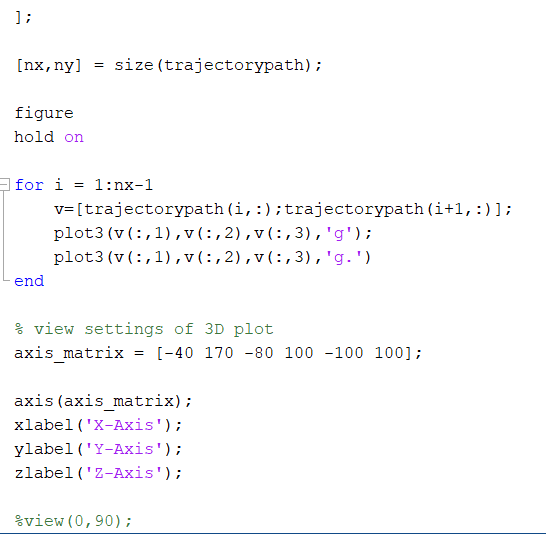


After that, the matrix that we have store with the coordinates needs to be translated.

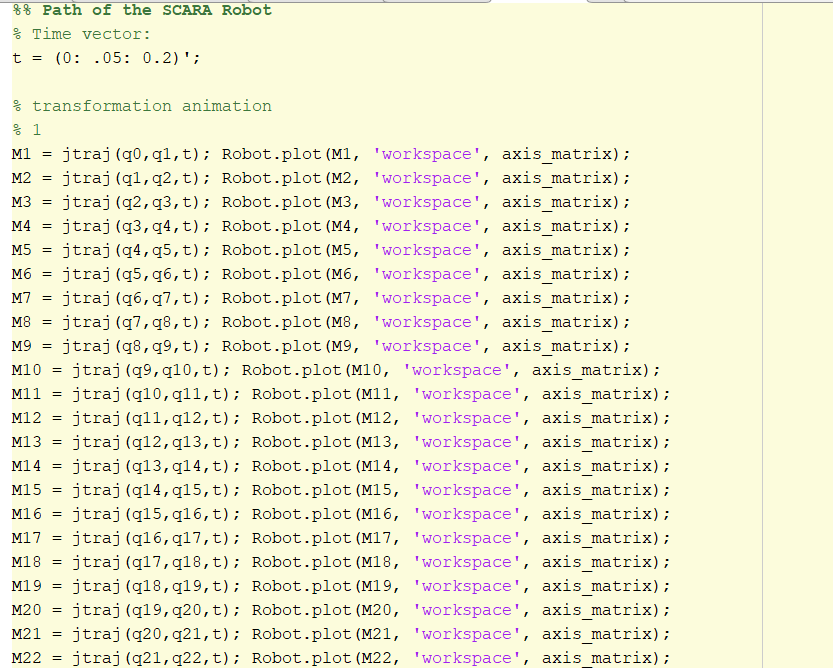


We also need to find the inverse kinematics of the system by utilising the function ikine.





Next, we set the trajectory path of the robot.



Finally, a 3D motion of the robot is done by using the function jtrat().

SIMULATION RESULTS:

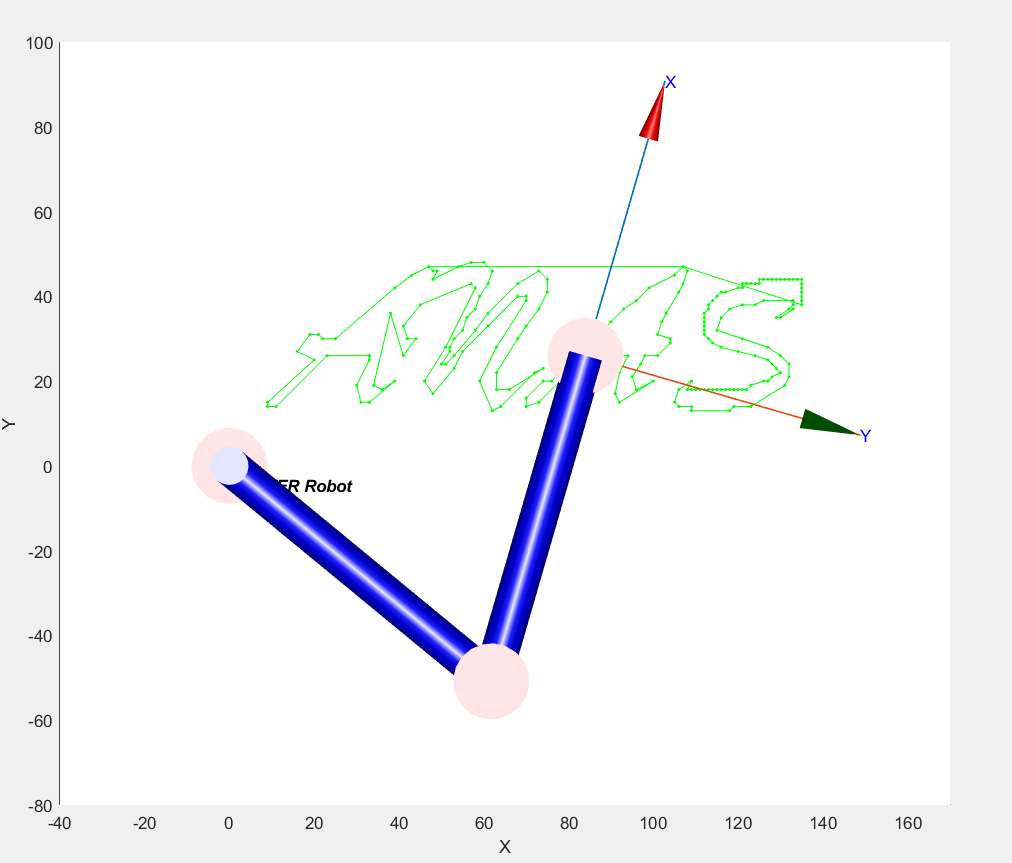


Figure 5: 2D Motion

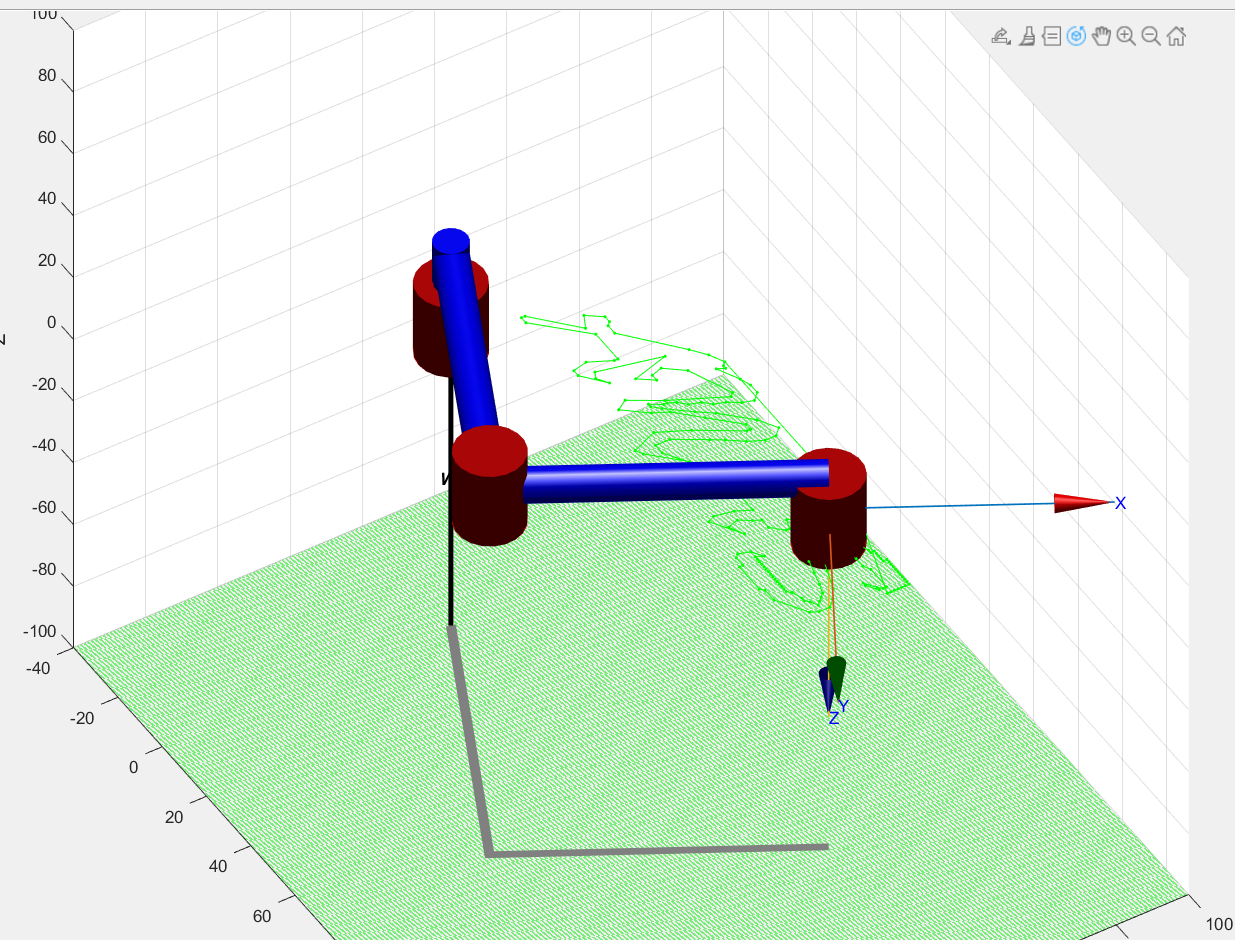


Figure 6: 3D Motion

**5.0 DISCUSSION**

Along the experiment, the z-coordinates are made as constant. This is made in order to make the design process easier. The number of coordinates for each letter plays a role in the accuracy of the letter. One of the mistakes that I do is that I do not have a consistent number of coordinates for each letter. For the part AN, I use 87 coordinates while A uses 27 coordinates and S uses 91 coordinates. Thus we can see that the letter S is most accurate followed by AN, and A. As number of coordinates increase, the accuracy of the letter also increase.

For the first objective, I partially achieve the objective as I have a slight confusion when determining the type of robot. At first, I decided to follow the SCARA Robot which I have encountered during the lab session. However, SCARA Robot is not a 3DOF robot as it have a prismatic joint instead of all revolute joint. For future works, I should change the prismatic joint into a non-rotating revolute joint to satisfy the demand of the project.

The second objective is achieved as the line graph constructed via Excel matches the one in the simulation.

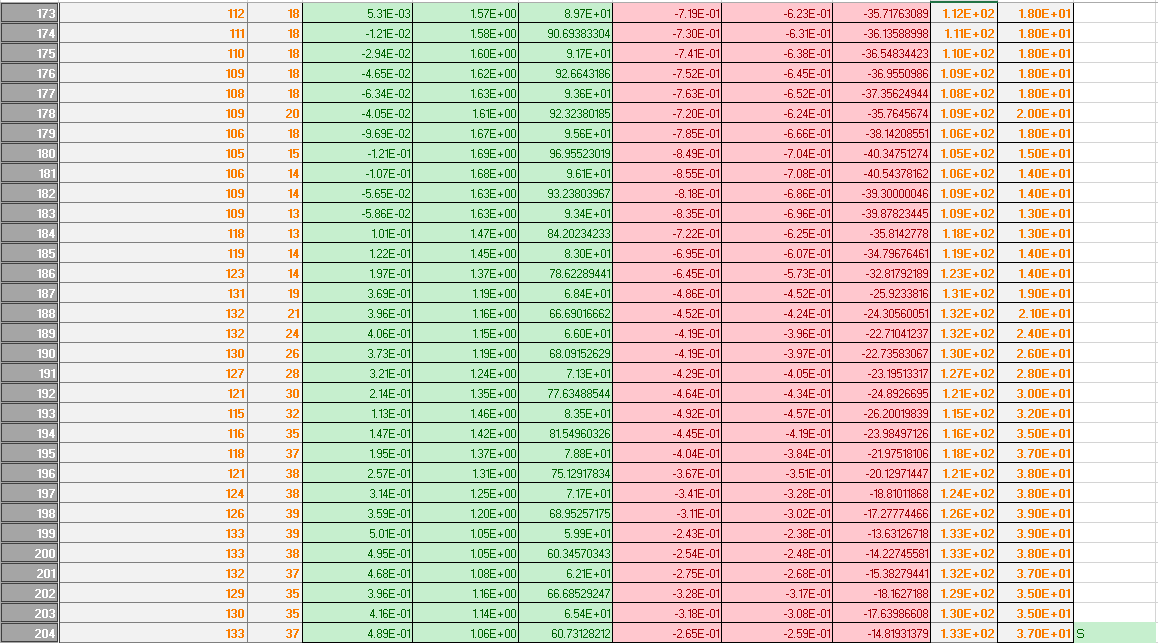
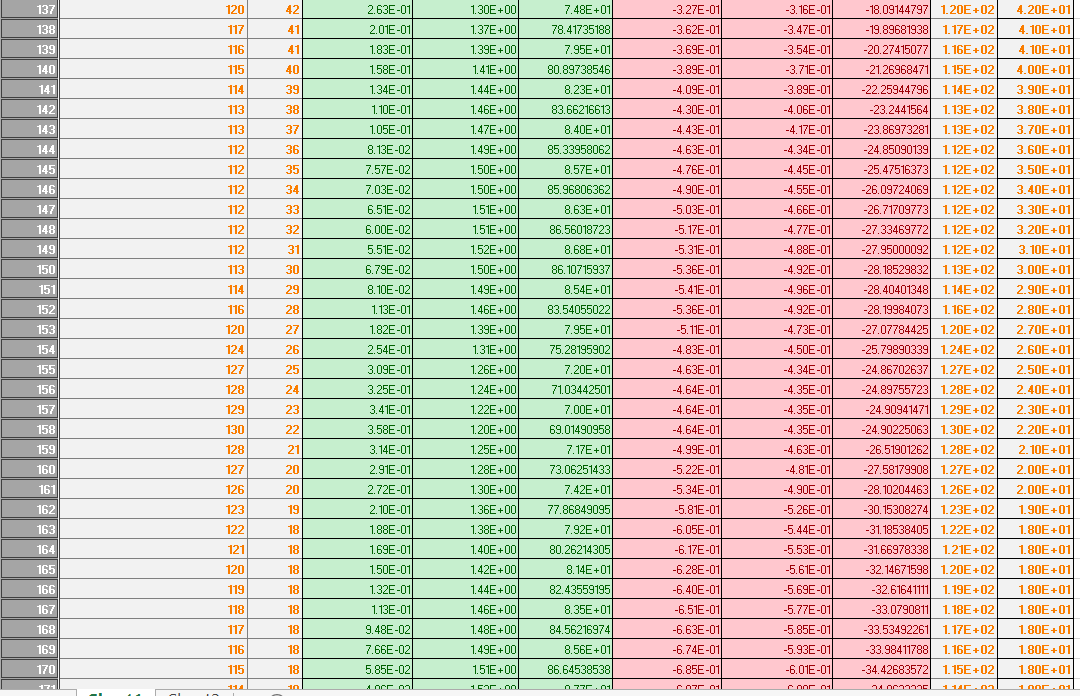
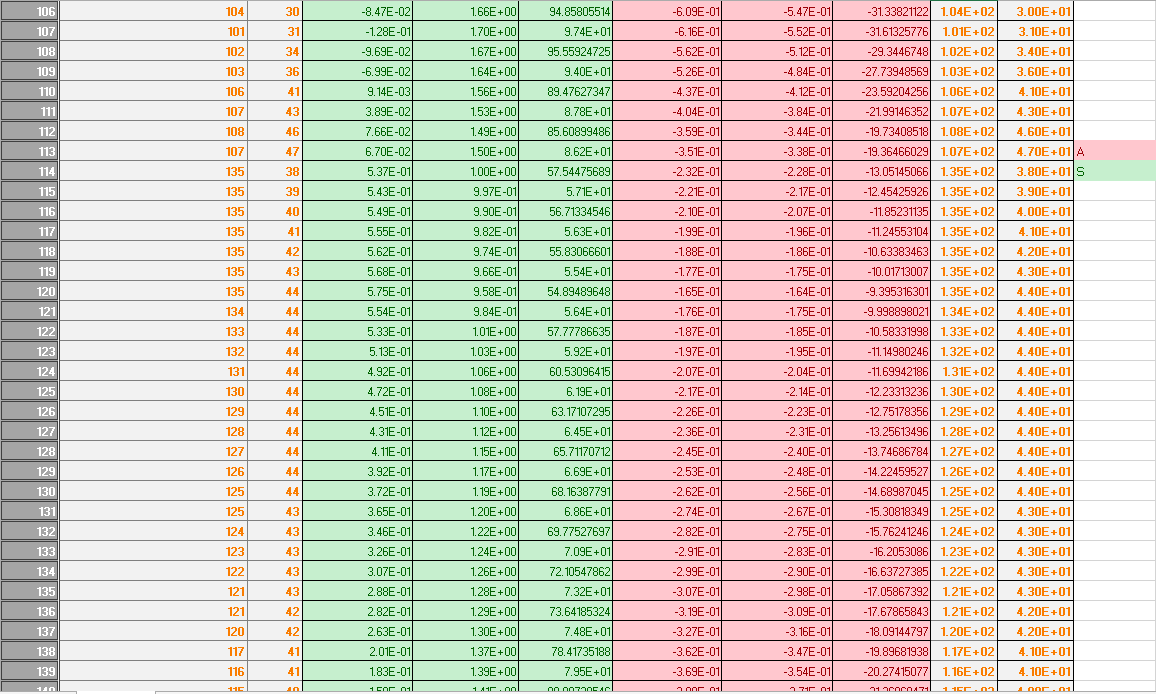
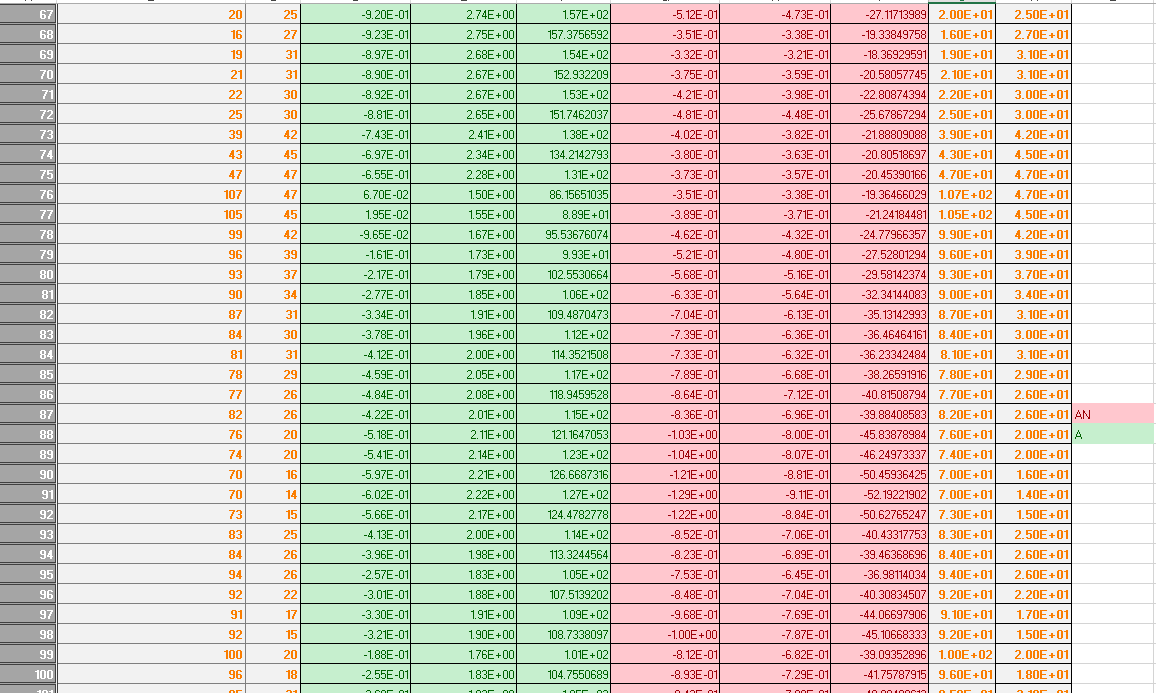
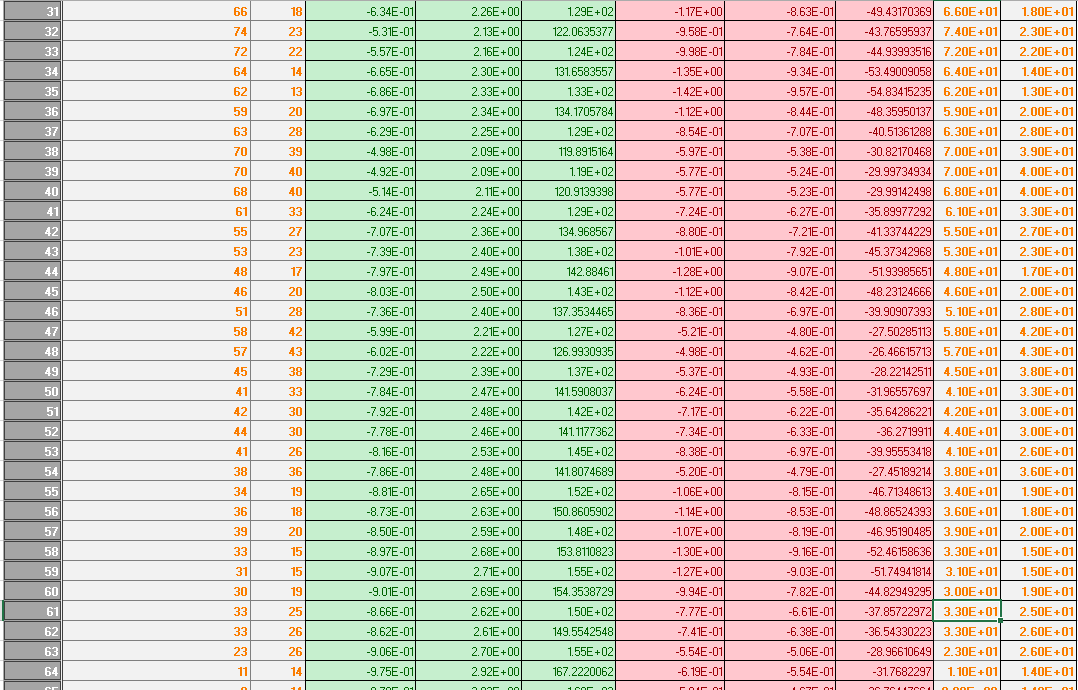
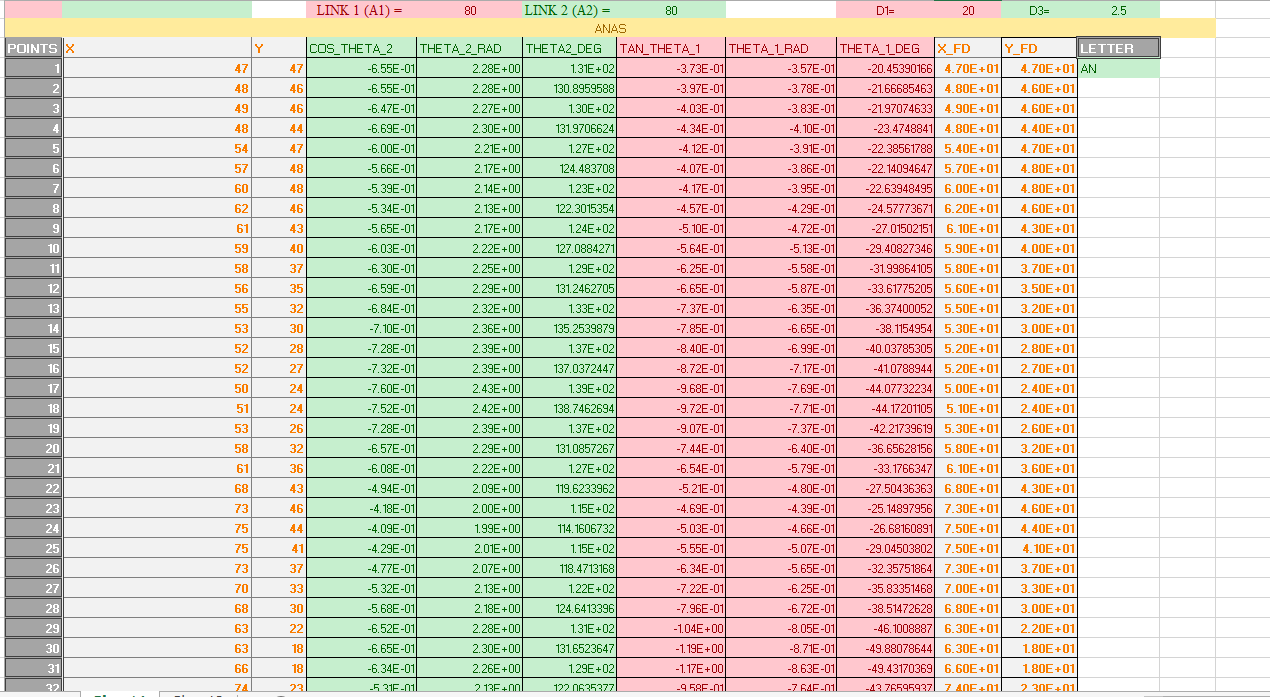
The inverse kinematics of the robot is obtained via calculation

Finally, a complete Anas name is constructed by the robot with the font of Brush Script MT.

**6.0 CONCLUSION**

In conclusion, the project can be called a success unless for a minor error where the robot is RRP and not RRR.

**APPENDIX A (IK CALCULATION)**

****

**APPENDIX B (FULL CODING)**

Full coding can be viewed at GITHUB:

https://github.com/AnasHafi07/ROBOTICS-PROJECT