Pick and Place Robot

Ziad Sherif*, Kirollos George*,
Ahmed Gobran*, Mohannad Noaman* ,
Mohamed Farid* and Mazen Moataz*
*German University in Cairo (GUC), Egypt
Email: Ziad.haridi@student.edu.eg

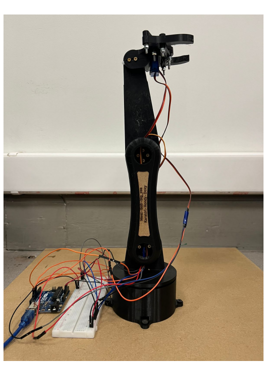
***Abstract*—This project involves designing, modeling, and controlling a 4-DOF robotic system for precision and adaptability in tasks like assembly and trajectory track. It includes kinematic analysis, MATLAB/Simulink simulations, and Coppeliasim Simulatiosns. Hardware implementation validates the system's real-world applicability, aiming to deliver reliable and cost-effective robotic solutions.**

I. Introduction

Robotic systems have become an integral part of modern industries, offering enhanced precision, efficiency, and adaptability in performing complex tasks. Among these, robots with four degrees of freedom (4-DOF) are particularly versatile, finding applications in areas such as material handling, automated assembly, and trajectory tracking. The ability to control and optimize the motion of each joint independently enables these systems to operate effectively in dynamic and constrained environments.

This project focuses on designing, analyzing, and controlling a 4-DOF robotic system. The robot's motion is modeled by integrating forward and inverse kinematics to ensure precise end-effector positioning and path planning. Velocity kinematics is further employed to refine the system's dynamic response, optimizing joint movements for smooth and efficient operation.

To validate the system's performance, MATLAB/Simulink and CoppeliaSim simulations are conducted, providing insights into the robot's behavior under various operational scenarios. This work aims to contribute to the development of cost-effective, reliable, and high-performance robotic solutions for industrial and research applications.



This paper is going to represent Hardware components, electrical wiring simulations, DH-Convention, simulink multibody simulations, CoppeliaSim Simulations, Hardware implementation.

II. Hardware Design and Implementation

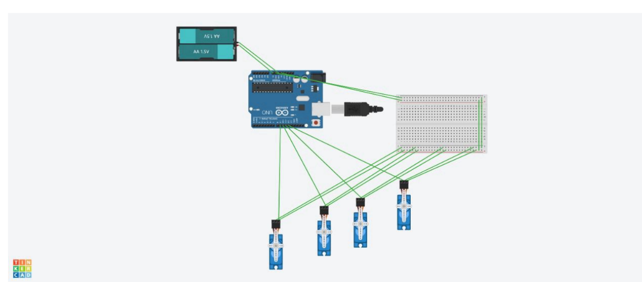
In order to build the hardware of the system, certain components are required as well as the circuit diagram of the system is built.

A. Hardware Components

1) Arduino Uno.
2) Power Supply 5V.
3) 3D-Printed Parts.
4) 2 Servo Motor SG90 180°
5) 3 MG995 Servo Motor 180°
6) Screws for fixation.

B. Circuit Diagram

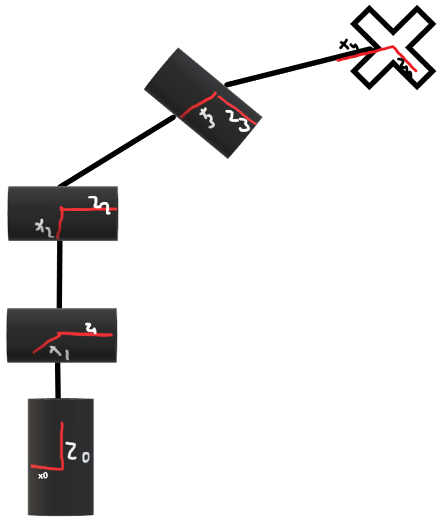
The figure below represents the simulation of the electrical wiring used in the project using TinkerCad.



III. Robot Kinematics

A. Robot's Frame Assignment

This section presents the frames used in calculating the DH-convention and the total homogeneous transformation matrix.



B. DH Convention

This section presents the total homogeneous transformation matrix.

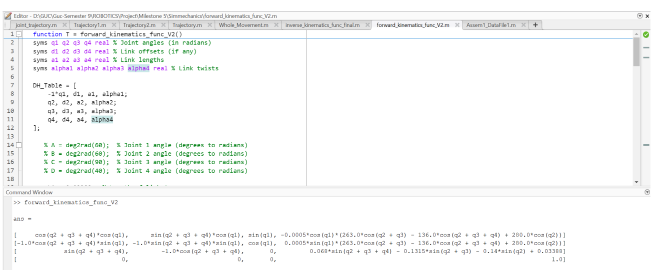
```
T =

[     cos(q2 + q3 + q4)*cos(q1),      sin(q2 + q3 + q4)*cos(q1), sin(q1), -0.0005*cos(q1)*(263.0*cos(q2 + q3) - 136.0*cos(q2 + q3 + q4) + 280.0*cos(q2))]
[-1.0*cos(q2 + q3 + q4)*sin(q1), -1.0*sin(q2 + q3 + q4)*sin(q1), cos(q1),  0.0005*sin(q1)*(263.0*cos(q2 + q3) - 136.0*cos(q2 + q3 + q4) + 280.0*cos(q2))]
[             sin(q2 + q3 + q4),         -1.0*cos(q2 + q3 + q4),       0,          0.068*sin(q2 + q3 + q4) - 0.1315*sin(q2 + q3) - 0.14*sin(q2) + 0.03388]
[                             0,                              0,       0,                                                                             1.0]
```

IV. Simulation Results

This section will represent the simulations done using Simulink multibody and Coppeliasim

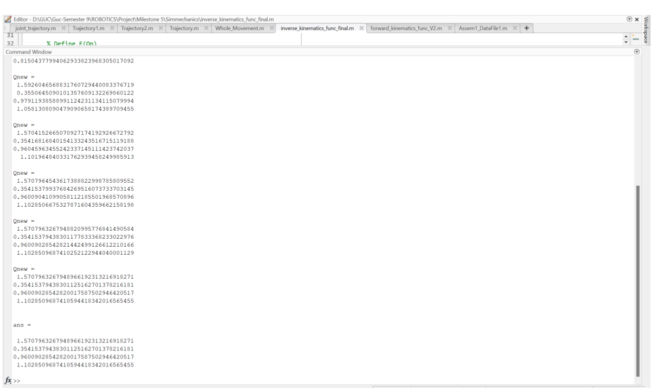
A. Simulink multibody

1) Forward kinematics code:

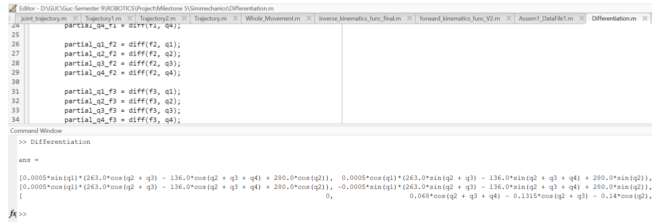


```
Editor - D:\GUC\Guc-Semester 9\ROBOTICS\Project\Milestone 5\Simmechanics\forward_kinematics_func_V2.m
joint_trajectory.m | Trajectory1.m | Trajectory2.m | Trajectory.m | Whole_Movement.m | inverse_kinematics_func_final.m | forward_kinematics_func_V2.m | Assem1_DataFile1.m

function T = forward_kinematics_func_V2()
syms q1 q2 q3 q4 real % Joint angles (in radians)
syms d1 d2 d3 d4 real % Link offsets (if any)
syms a1 a2 a3 a4 real % Link lengths
syms alpha1 alpha2 alpha3 alpha4 real % Link twists

DH_Table = [
    -1*q1, d1, a1, alpha1;
    q2, d2, a2, alpha2;
    q3, d3, a3, alpha3;
    q4, d4, a4, alpha4
];

  % A = deg2rad(60);  % Joint 1 angle (degrees to radians)
  % B = deg2rad(60);  % Joint 2 angle (degrees to radians)
  % C = deg2rad(90);  % Joint 3 angle (degrees to radians)
  % D = deg2rad(40);  % Joint 4 angle (degrees to radians)

Command Window
>> forward_kinematics_func_V2

ans =

[     cos(q2 + q3 + q4)*cos(q1),      sin(q2 + q3 + q4)*cos(q1), sin(q1), -0.0005*cos(q1)*(263.0*cos(q2 + q3) - 136.0*cos(q2 + q3 + q4) + 280.0*cos(q2))]
[-1.0*cos(q2 + q3 + q4)*sin(q1), -1.0*sin(q2 + q3 + q4)*sin(q1), cos(q1),  0.0005*sin(q1)*(263.0*cos(q2 + q3) - 136.0*cos(q2 + q3 + q4) + 280.0*cos(q2))]
[             sin(q2 + q3 + q4),         -1.0*cos(q2 + q3 + q4),       0,          0.068*sin(q2 + q3 + q4) - 0.1315*sin(q2 + q3) - 0.14*sin(q2) + 0.03388]
[                             0,                              0,       0,                                                                             1.0]

fx >>
```

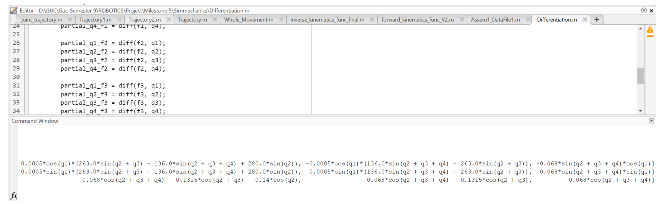
2) Inverse Kinematics:



```
Editor - D:\GUC\Guc-Semester 9\ROBOTICS\Project\Milestone 5\Simmechanics\inverse_kinematics_func_final.m
joint_trajectory.m | Trajectory1.m | Trajectory2.m | Trajectory.m | Whole_Movement.m | inverse_kinematics_func_final.m | forward_kinematics_func_V2.m | Assem1_DataFile1.m

      % Define F(Qn)

Command Window
0.81504377994062933823968305017092

Qnew =
 1.5926046568831760729440083376719
 0.3550645090101357609132269860122
0.97911938588991124231134115079994
 1.0581308090479090658174389709455

Qnew =
 1.5704152665070927174192926672792
0.35416816840154133243516715119188
0.96045963455242337145111423742037
  1.101964840331762939458249985913

Qnew =
 1.5707964543617388822998785809552
0.35415379937684269516073733703145
0.96009041099058112185501968570896
 1.1028506675327871604359662158198

Qnew =
 1.5707963267948820995776841490584
0.35415379438301177833368233022976
0.96009028542821442499126612210166
 1.1028509687410252122944040001129

Qnew =
 1.5707963267948966192313216918271
0.35415379438301125162701378216181
0.96009028542820017587502946420517
 1.1028509687410594418342016565455


ans =

 1.5707963267948966192313216918271
0.35415379438301125162701378216181
0.96009028542820017587502946420517
 1.1028509687410594418342016565455

fx >>
```

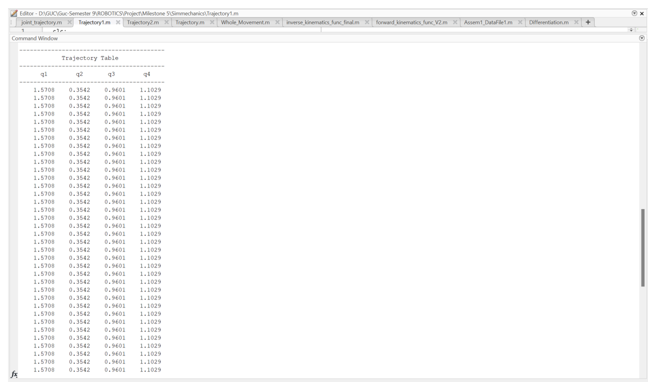


```
Editor - D:\GUC\Guc-Semester 9\ROBOTICS\Project\Milestone 5\Simmechanics\Differentiation.m
joint_trajectory.m | Trajectory1.m | Trajectory2.m | Trajectory.m | Whole_Movement.m | inverse_kinematics_func_final.m | forward_kinematics_func_V2.m | Assem1_DataFile1.m | Differentiation.m

        partial_q4_f1 = diff(f1, q4);

        partial_q1_f2 = diff(f2, q1);
        partial_q2_f2 = diff(f2, q2);
        partial_q3_f2 = diff(f2, q3);
        partial_q4_f2 = diff(f2, q4);

        partial_q1_f3 = diff(f3, q1);
        partial_q2_f3 = diff(f3, q2);
        partial_q3_f3 = diff(f3, q3);
        partial_q4_f3 = diff(f3, q4);

Command Window
>> Differentiation

ans =

[0.0005*sin(q1)*(263.0*cos(q2 + q3) - 136.0*cos(q2 + q3 + q4) + 280.0*cos(q2)),  0.0005*cos(q1)*(263.0*sin(q2 + q3) - 136.0*sin(q2 + q3 + q4) + 280.0*sin(q2)),
[0.0005*cos(q1)*(263.0*cos(q2 + q3) - 136.0*cos(q2 + q3 + q4) + 280.0*cos(q2)), -0.0005*sin(q1)*(263.0*sin(q2 + q3) - 136.0*sin(q2 + q3 + q4) + 280.0*sin(q2)),
[                                                                            0,          0.068*cos(q2 + q3 + q4) - 0.1315*cos(q2 + q3) - 0.14*cos(q2),

fx >>
```

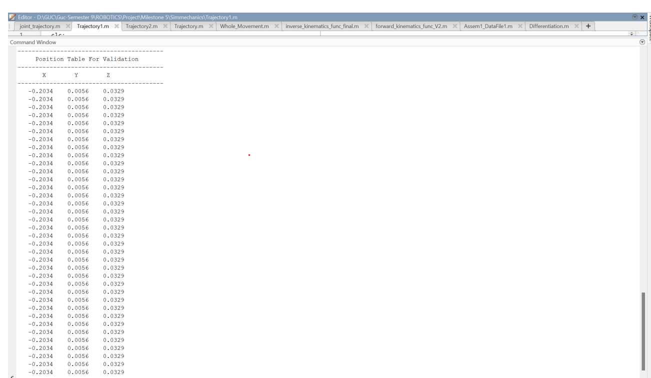


```
Editor - D:\GUC\Guc-Semester 9\ROBOTICS\Project\Milestone 5\Simmechanics\Differentiation.m
joint_trajectory.m | Trajectory1.m | Trajectory2.m | Trajectory.m | Whole_Movement.m | inverse_kinematics_func_final.m | forward_kinematics_func_V2.m | Assem1_DataFile1.m | Differentiation.m

        partial_q4_f1 = diff(f1, q4);

        partial_q1_f2 = diff(f2, q1);
        partial_q2_f2 = diff(f2, q2);
        partial_q3_f2 = diff(f2, q3);
        partial_q4_f2 = diff(f2, q4);

        partial_q1_f3 = diff(f3, q1);
        partial_q2_f3 = diff(f3, q2);
        partial_q3_f3 = diff(f3, q3);
        partial_q4_f3 = diff(f3, q4);

Command Window

 0.0005*cos(q1)*(263.0*sin(q2 + q3) - 136.0*sin(q2 + q3 + q4) + 280.0*sin(q2)), -0.0005*cos(q1)*(136.0*sin(q2 + q3 + q4) - 263.0*sin(q2 + q3)), -0.068*sin(q2 + q3 + q4)*cos(q1)]
-0.0005*sin(q1)*(263.0*sin(q2 + q3) - 136.0*sin(q2 + q3 + q4) + 280.0*sin(q2)),  0.0005*sin(q1)*(136.0*sin(q2 + q3 + q4) - 263.0*sin(q2 + q3)),  0.068*sin(q2 + q3 + q4)*sin(q1)]
          0.068*cos(q2 + q3 + q4) - 0.1315*cos(q2 + q3) - 0.14*cos(q2),           0.068*cos(q2 + q3 + q4) - 0.1315*cos(q2 + q3),          0.068*cos(q2 + q3 + q4)]
fx
```

3) Trajectory 1:

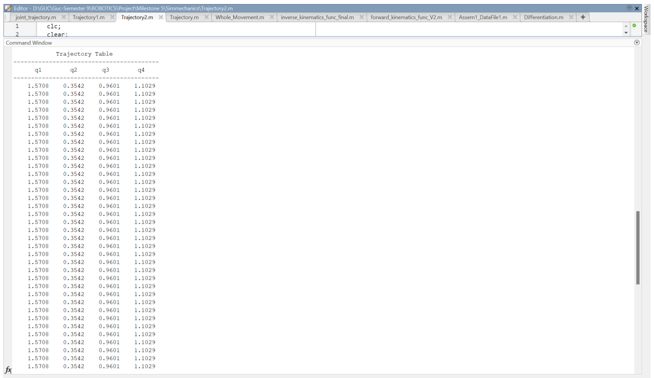


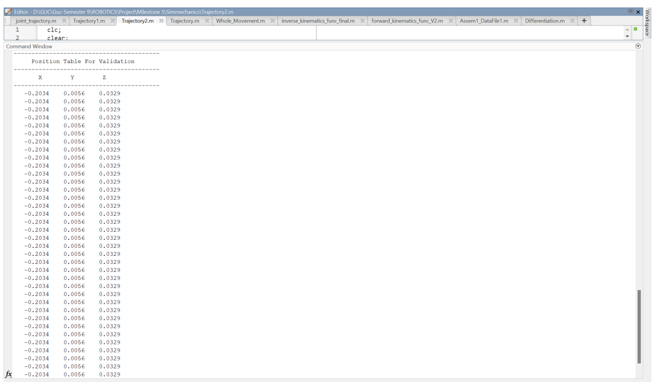
```
Editor - D:\GUC\Guc-Semester 9\ROBOTICS\Project\Milestone 5\Simmechanics\Trajectory1.m
joint_trajectory.m | Trajectory1.m | Trajectory2.m | Trajectory.m | Whole_Movement.m | inverse_kinematics_func_final.m | forward_kinematics_func_V2.m | Assem1_DataFile1.m | Differentiation.m

Command Window
----------------------------------------
         Trajectory Table
----------------------------------------
      q1        q2        q3        q4
----------------------------------------
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
    1.5700    0.3542    0.9601    1.1029
fx
```



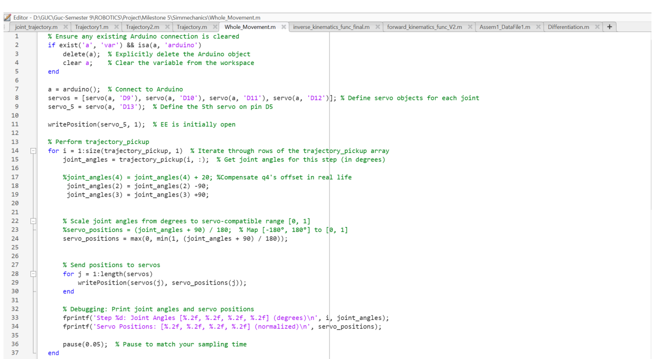
```
Editor - D:\GUC\Guc-Semester 9\ROBOTICS\Project\Milestone 5\Simmechanics\Trajectory1.m
joint_trajectory.m | Trajectory1.m | Trajectory2.m | Trajectory.m | Whole_Movement.m | inverse_kinematics_func_final.m | forward_kinematics_func_V2.m | Assem1_DataFile1.m | Differentiation.m

Command Window
----------------------------------------
     Position Table For Validation
----------------------------------------
       X         Y         Z
----------------------------------------
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
   -0.2034    0.0056    0.0329
fx
```

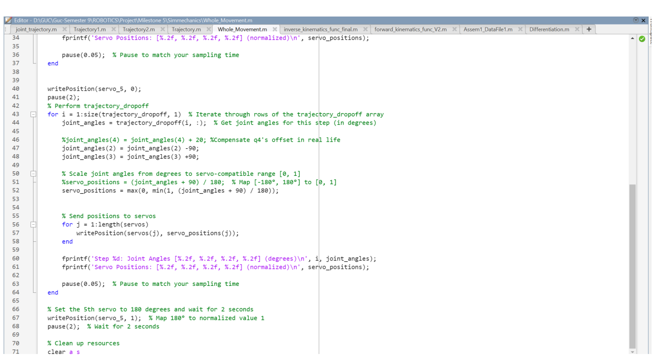
4) Trajectory 2:



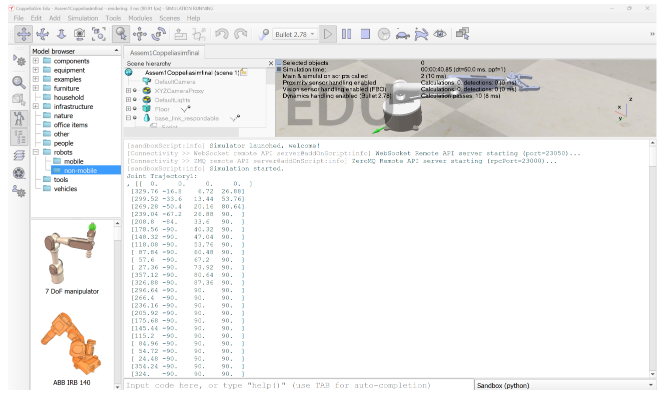


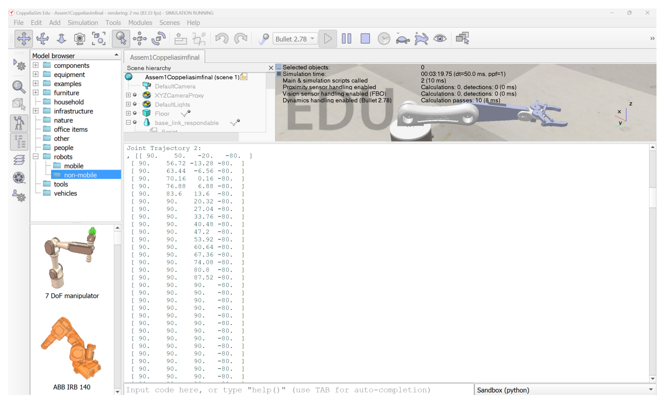
5) Whole Movement:
   This code is responsible of sending the data to the arduino





B. *CoppeliaSim*





V. Conclusions and Future Recommendations

As stated in the last sections, the full hardware 4 degrees of freedom robot is implemented, also multiple simulations are done through several applications like matlab simulink mutibody and Coppeliasim. After, all these simulations there are matlab scripts that calculates the kinematics of the system whether position or velocity, and the results of these scripts were compared to the ones obtained from the simulations to make sure that all the outputs are the same. At the end, the joint angles were sent to the Arduino from the trajectory code using Matlab support package for arduino and Simulink support package for arduino. At the end, the robot moved with the trajectiry given.

References