HW3

Andy Tsai

March 14, 2025

1 Q1: Forward Kinematics of a SCARA robot

1.1 Perform Forward Kinematics

Transformation Matrix

$$
{}^{0}T_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0.4\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0.4\sin\theta_1 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{1}T_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0.3\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0.3\sin\theta_2 \\ 0 & 0 & 1 & 0.25 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}T_{3(\text{E.E.})} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 0.15\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0.15\sin\theta_3 \\ 0 & 0 & 1 & 0.15 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Final Transformation

$$
{}^{0}T_{\text{E.E.}} = {}^{0}T_1 \times {}^{1}T_2 \times {}^{2}T_{3(\text{E.E.})}
$$

$$
= \begin{bmatrix} \cos(\theta_1+\theta_2+\theta_3) & -\sin(\theta_1+\theta_2+\theta_3) & 0 & 0.15\cos(\theta_1+\theta_2+\theta_3) + 0.3\cos(\theta_1+\theta_2) + 0.4\cos\theta_1 \\ \sin(\theta_1+\theta_2+\theta_3) & \cos(\theta_1+\theta_2+\theta_3) & 0 & 0.15\sin(\theta_1+\theta_2+\theta_3) + 0.3\sin(\theta_1+\theta_2) + 0.4\sin\theta_1 \\ 0 & 0 & 1 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

DH-Table

Links	θ	d	a	α
Base (0) - 1	θ_1	0.2	0.4	0
1 - 2	θ_2	0.25	0.3	0
2 - 3 (E.E.)	θ_3	0.15	0.15	0

Table 1: My DH-Table

Verify the result (MATLAB)

```
Editor - createDH_T.m                                    Mechanics Explorers
createDH_T.m  +
function T = createDH_T(theta, d, a, alpha)
    T = [cos(theta) -sin(theta)*cos(alpha) sin(theta)*sin(alpha) a*cos(theta);
        sin(theta) cos(theta)*cos(alpha) -cos(theta)*sin(alpha) a*sin(theta)
        0 sin(alpha) cos(alpha) d
        0 0 0 1];
end
```

Figure 1: My function to create DH transformation matrix.

```
Editor - Q1.m                Mechanics Explorers - Mechanics Explorer-SCARA
Q1_1.m  Q1_2.m  Q1.m  +
clc; clear all

% theta
syms th1 th2 th3;

% DH-table
DH = [[th1 0.2 0.4 0];
    [th2 0.25 0.3 0];
    [th3 0.15 0.15 0]];

len = length(DH);

for i=1:len-1
    if i==1
        T_final = createDH_T(DH(1, 1), DH(1, 2), DH(1, 3), DH(1, 4));
    else
        T_final = T_final * createDH_T(DH(i, 1), DH(i, 2), DH(i, 3), DH(i, 4));
    end
end

T_final = simplify(T_final)
```

```
Command Window

T_final =

[cos(th1 + th2 + th3), -sin(th1 + th2 + th3), 0, (3*cos(th1 + th2 + th3))/20 + (3*cos(th1 + th2))/10 + (2*cos(th1))/5]
[sin(th1 + th2 + th3),  cos(th1 + th2 + th3), 0, (3*sin(th1 + th2 + th3))/20 + (3*sin(th1 + th2))/10 + (2*sin(th1))/5]
[                   0,                     0, 1,                                                                    3/5]
[                   0,                     0, 0,                                                                      1]

fx >>
```

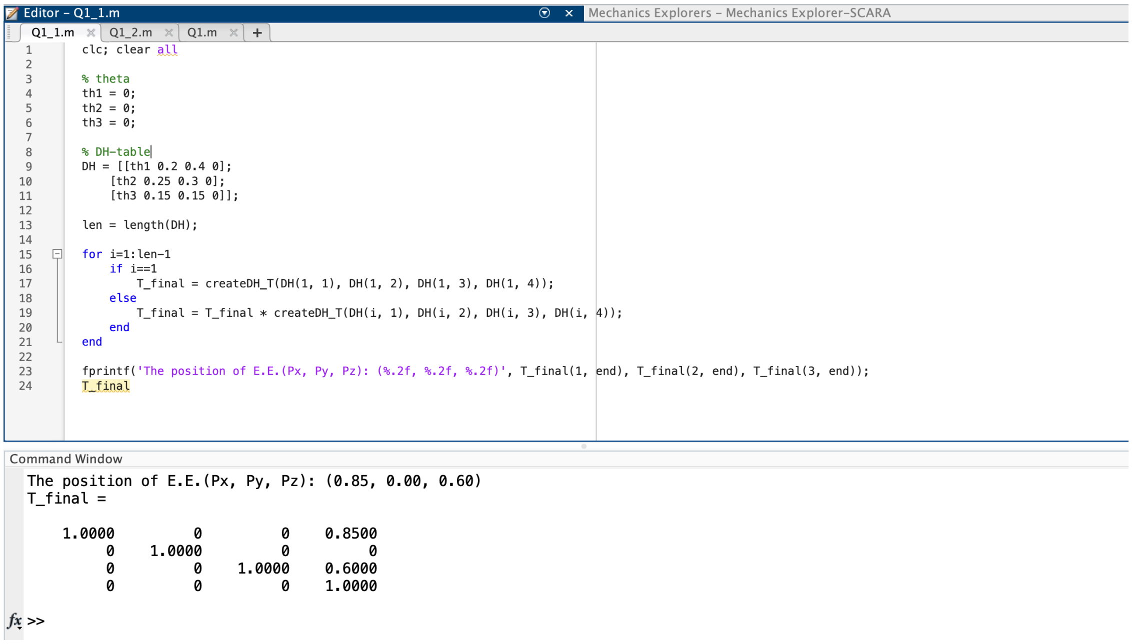
1.2 Home position

When $\theta_1 = 0°, \theta_2 = 0°, \theta_3 = 0°$, the homogeneous transformation looks like this:

$$^{0}T_{\text{E.E.}} = \begin{bmatrix} 1 & 0 & 0 & 0.85 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The translation in $^{0}T_{\text{E.E.}}$ matches the given end effector position (x, y, z) = (0.85, 0, 0.6).

Verify the result (MATLAB)


Editor - Q1_1.m   Mechanics Explorers - Mechanics Explorer-SCARA

Q1_1.m Q1_2.m Q1.m

```
clc; clear all

% theta
th1 = 0;
th2 = 0;
th3 = 0;

% DH-table
DH = [[th1 0.2 0.4 0];
    [th2 0.25 0.3 0];
    [th3 0.15 0.15 0]];

len = length(DH);

for i=1:len-1
    if i==1
        T_final = createDH_T(DH(1, 1), DH(1, 2), DH(1, 3), DH(1, 4));
    else
        T_final = T_final * createDH_T(DH(i, 1), DH(i, 2), DH(i, 3), DH(i, 4));
    end
end

fprintf('The position of E.E.(Px, Py, Pz): (%.2f, %.2f, %.2f)', T_final(1, end), T_final(2, end), T_final(3, end));
T_final
```

Command Window

```
The position of E.E.(Px, Py, Pz): (0.85, 0.00, 0.60)
T_final =

    1.0000         0         0    0.8500
         0    1.0000         0         0
         0         0    1.0000    0.6000
         0         0         0    1.0000

fx >>
```

1.3 Workspace of SCARA

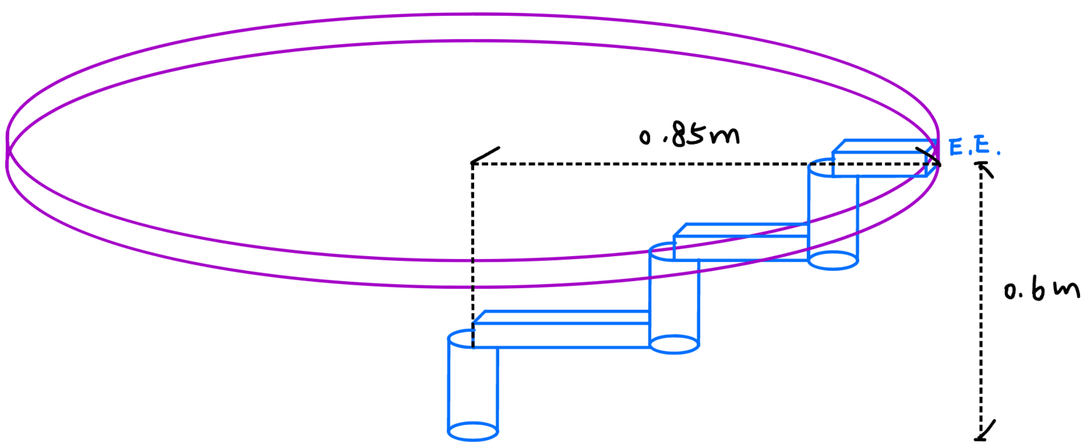
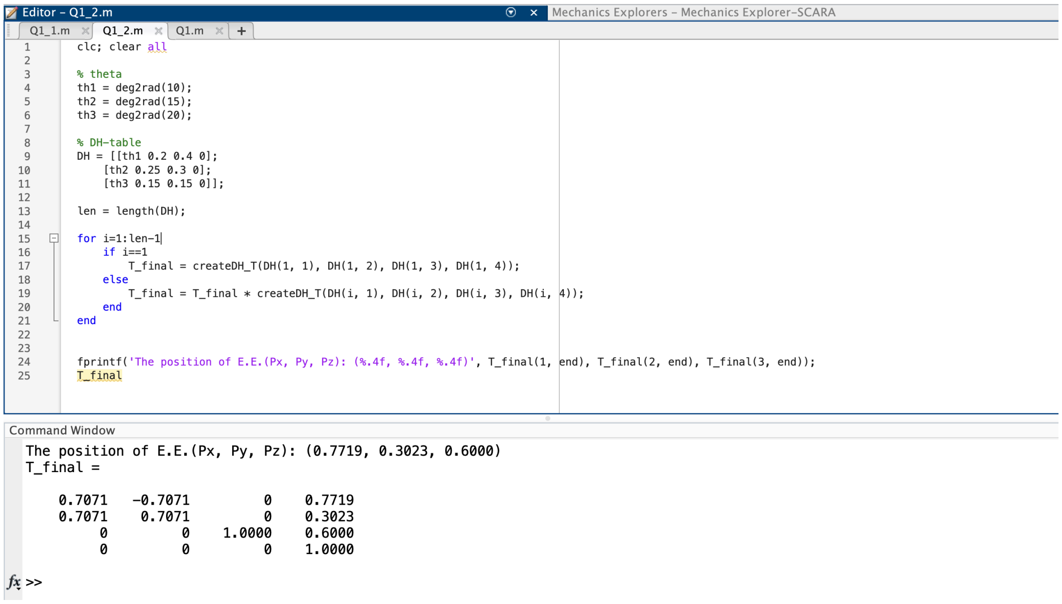


Figure 2: Hand drawn workspace of SCARA (the purple region).

When $\theta_1 = 10°, \theta_2 = 15°, \theta_3 = 20°$, the homogeneous transformation looks like this:

$$^{0}T_{\text{E.E.}} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0.7719 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0.3023 \\ 0 & 0 & 1 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Verify the result (MATLAB)



2 Q2:Forward Kinematics Simulator in Simulink

2.1 Create SCARA simulation model.

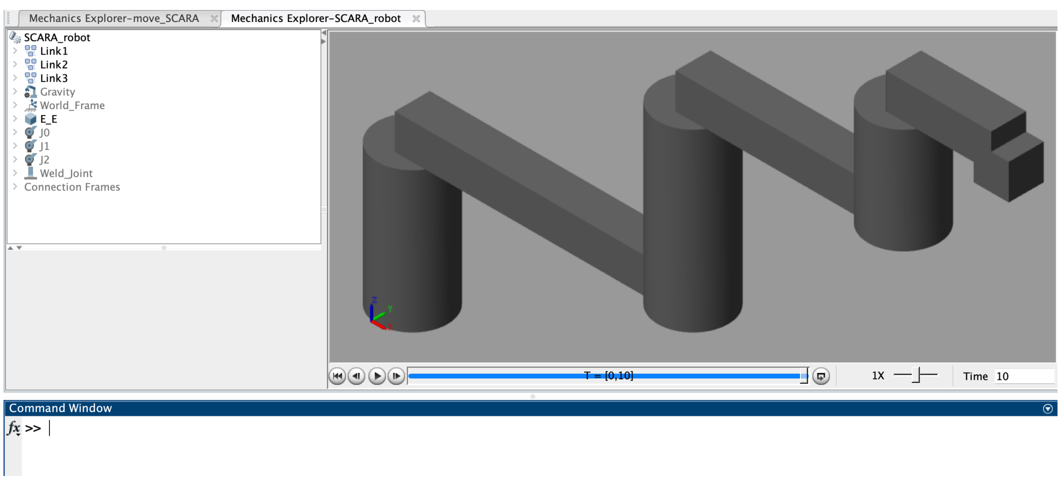
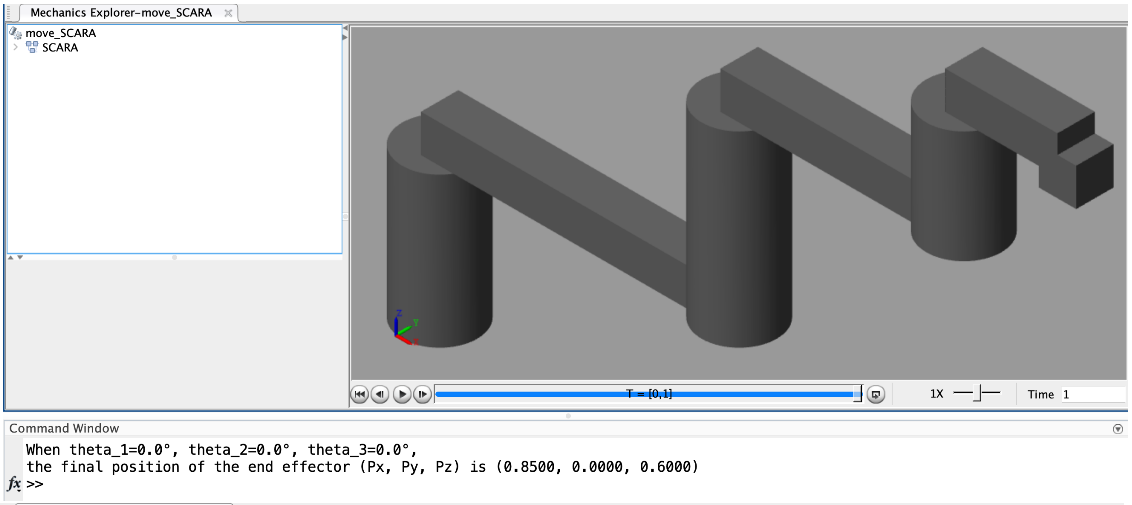
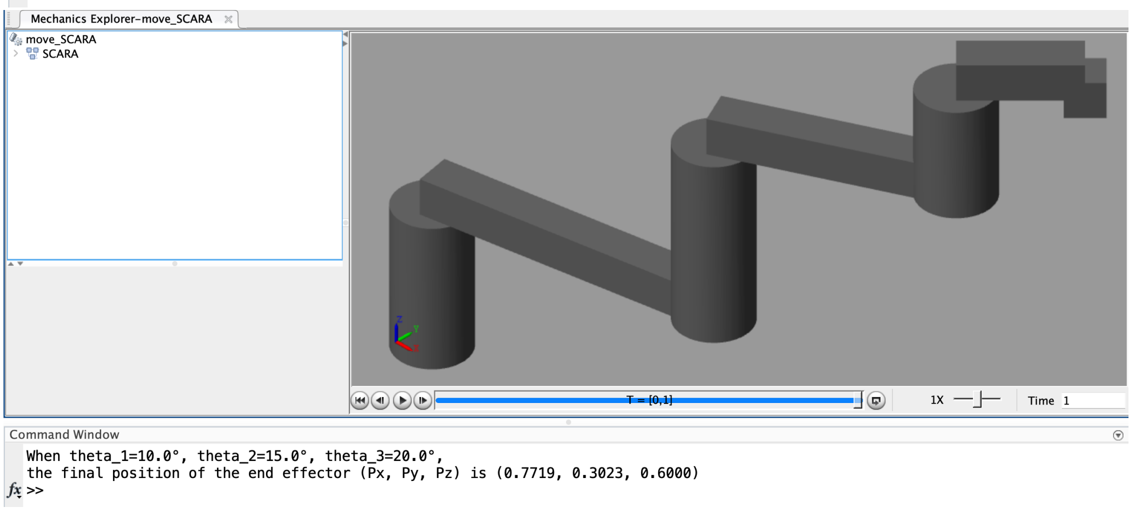


Figure 3: The SCARA robot's simulation model I built.

2.2 Verify the FK result (MATLAB & Simulink)





The top image is the validation of 1.2, and the bottom one is the validation of 1.3. Both answers match the calculated results in 1.2 and 1.3.

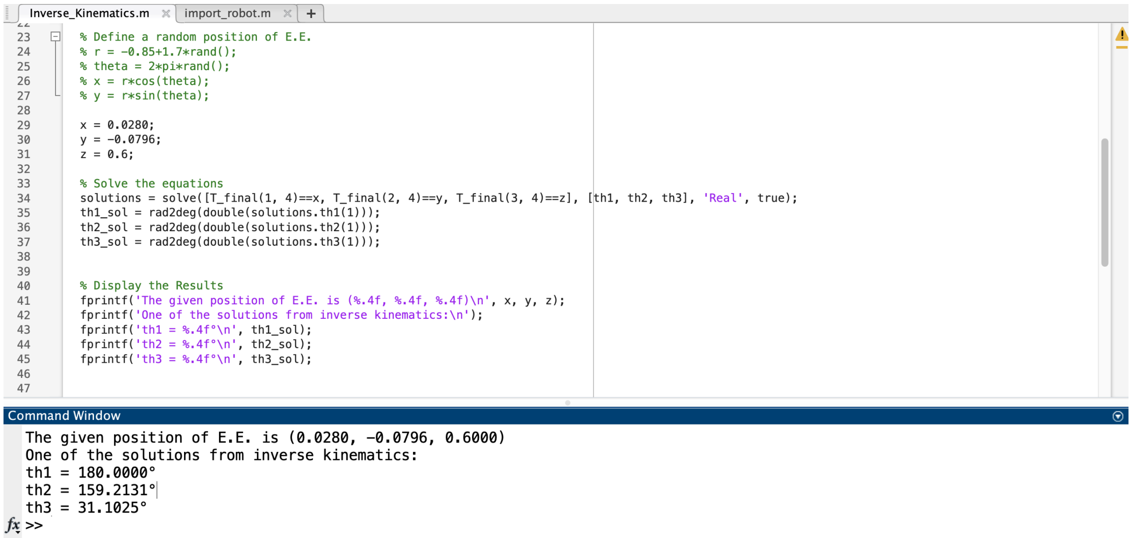
3 Q3: Inverse Kinematics Simulation

3.1 Z coordinates in workspace

The Z coordinate of the end effector is always at 0.6 m due to the design of the robot.

3.2 Inverse Kinematics with MATLAB code

In this subproblem, I chose a random position (0.0280, -0.0796, 0.6000) for the robot first and perform inverse kinematics. The result can be seen in Figure 3.2.



Inverse_Kinematics.m import_robot.m

```
% Define a random position of E.E.
% r = -0.85+1.7*rand();
% theta = 2*pi*rand();
% x = r*cos(theta);
% y = r*sin(theta);

x = 0.0280;
y = -0.0796;
z = 0.6;

% Solve the equations
solutions = solve([T_final(1, 4)==x, T_final(2, 4)==y, T_final(3, 4)==z], [th1, th2, th3], 'Real', true);
th1_sol = rad2deg(double(solutions.th1(1)));
th2_sol = rad2deg(double(solutions.th2(1)));
th3_sol = rad2deg(double(solutions.th3(1)));


% Display the Results
fprintf('The given position of E.E. is (%.4f, %.4f, %.4f)\n', x, y, z);
fprintf('One of the solutions from inverse kinematics:\n');
fprintf('th1 = %.4f°\n', th1_sol);
fprintf('th2 = %.4f°\n', th2_sol);
fprintf('th3 = %.4f°\n', th3_sol);
```

Command Window

```
The given position of E.E. is (0.0280, -0.0796, 0.6000)
One of the solutions from inverse kinematics:
th1 = 180.0000°
th2 = 159.2131°
th3 = 31.1025°
>>
```

3.3 Using End Effector to draw a square

The coordinates of each vertex the square I chose is (0.5, 0.2, 0.6), (0.5, -0.2, 0.6), (0.8, -0.2, 0.6) and (0.8, 0.2, 0.6). The input position signal can be seen in Figure 4. The generated θ_1, θ_2, θ_3 from inverse kinematics is plotted in Figure 5. Run the attached file 'draw_square.m' in folder Q3 to activate the simulator.

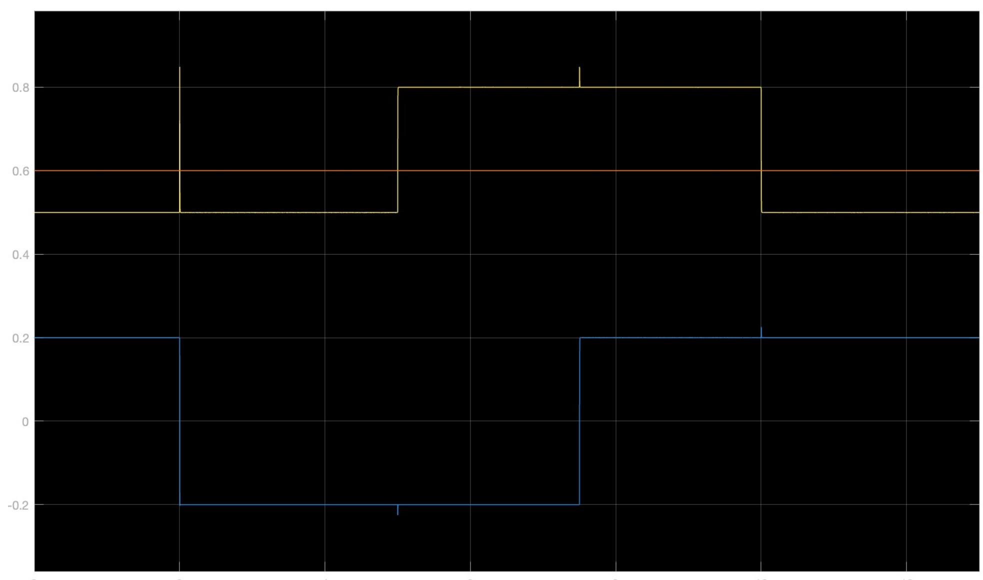


Figure 4: The position of the end effector over time ($0 \sim 13s$). The yellow line is the X coordinate, the blue line is the Y coordinate, and the red line is the Z coordinate.

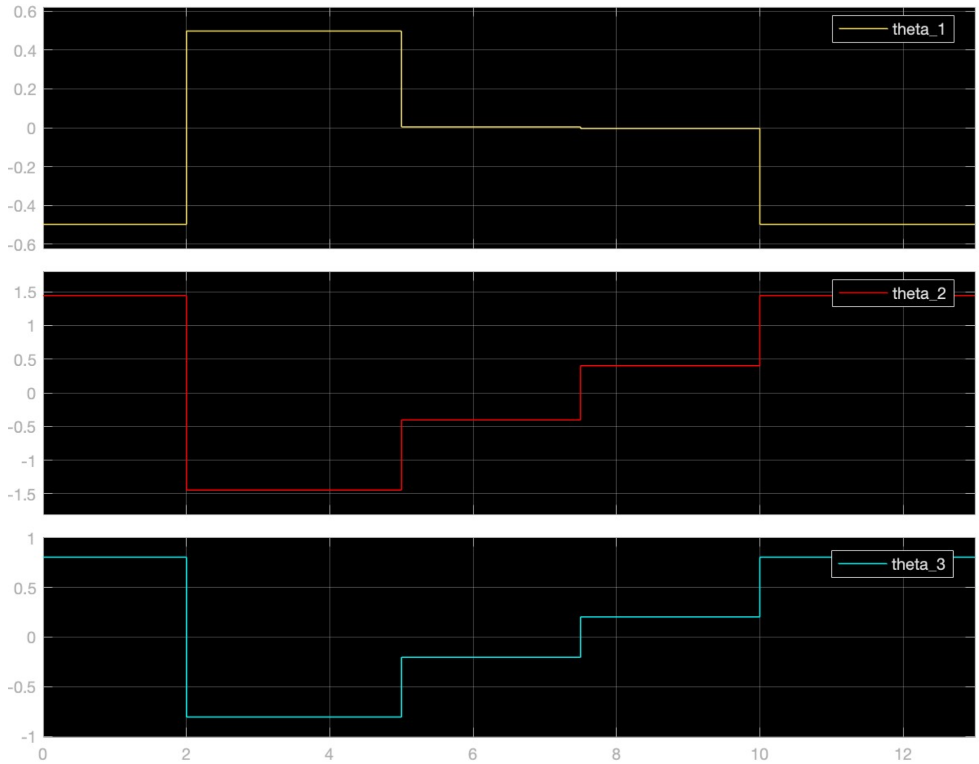


Figure 5: The calculated theta from the inverse kinematics simulations.

Top view of the position of the end effector

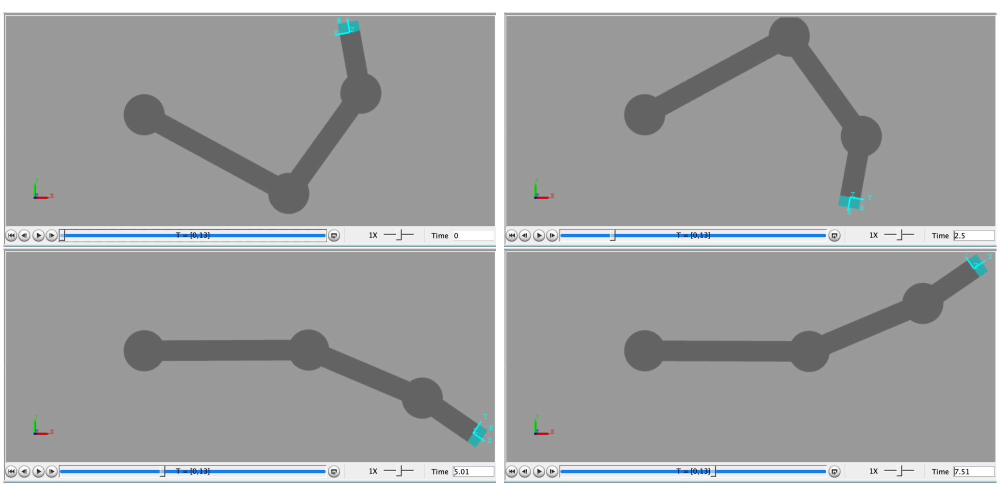


Figure 6: The top left image is the starting position of drawing a square, where $t = 0s$. The top right image is when $t = 2.5s$. The bottom left image represents $t = 5s$, while the bottom right one shows when $t = 7.5s$.