

EE594

Assignment 1: Industrial Manipulator

E/18/023

Amarasinghe E.G.C.L.

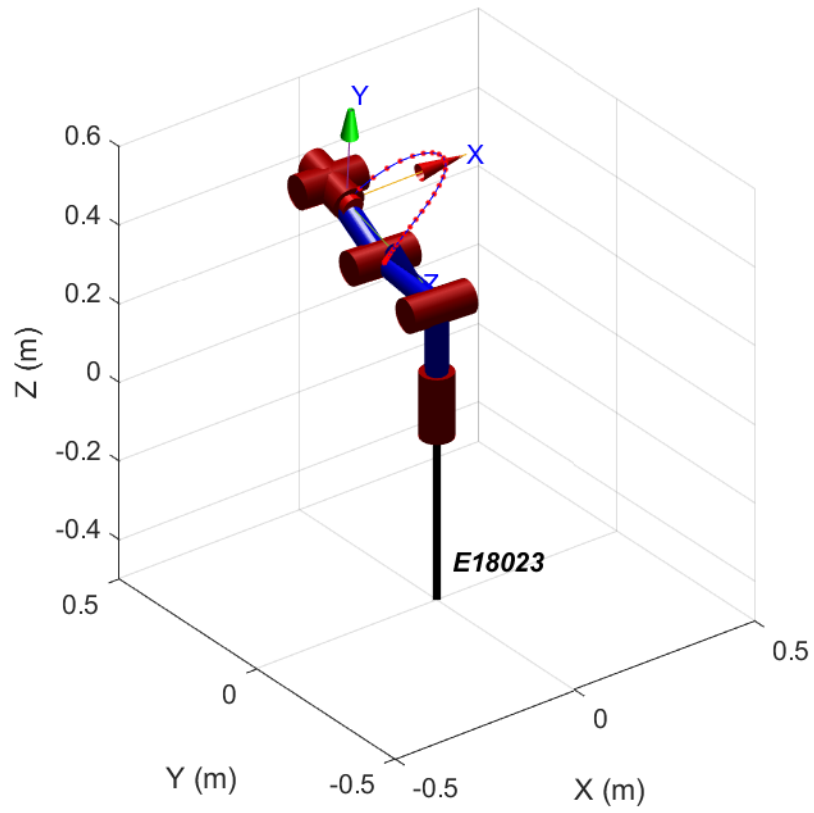
```
1 % E/18/023
2 % Amarasinghe E.G.C.L.
3
4
5 EN = 023; % My reg no
6
7 A_y = 0.30; % since EN is odd
8 A_z = 0.010 + EN / 420 * 0.40;
9
10 % DH parameters
11 L1 = Link('d', 0.24765, 'a', 0, 'alpha', pi/2);
12 L2 = Link('d', 0, 'a', 0.2286, 'alpha', 0);
13 L3 = Link('d', 0, 'a', 0.2286, 'alpha', 0);
14 L4 = Link('d', 0, 'a', 0, 'alpha', -pi/2);
15 L5 = Link('d', 0.073025, 'a', 0, 'alpha', 0);
16
17 Robot = SerialLink([L1 L2 L3 L4 L5], 'name', 'E18023');
18
19 A_pos = [-0.17, A_y, A_z];
20 A_rpy = [-180, 0, 60];
21 B_pos = [0.181, 0.313, 0.345];
22 B_rpy = [-125, 26, 106];
23 C_pos = [0.420, 0.000, 0.540];
24 C_rpy = [0, 70, 0];
25 D_pos = [0.237, -0.338, 0.100];
26 D_rpy = [180, 0, -125];
27
28 start_pos = [-0.146, 0, 0.409];
29 start_rpy = [0, -90, -180];
30
31 Home_T = transl(start_pos) * rpy2tr(deg2rad(start_rpy));
32 A_T = transl(A_pos) * rpy2tr(deg2rad(A_rpy));
33 B_T = transl(B_pos) * rpy2tr(deg2rad(B_rpy));
34 C_T = transl(C_pos) * rpy2tr(deg2rad(C_rpy));
35 D_T = transl(D_pos) * rpy2tr(deg2rad(D_rpy));
36
37 q_home = Robot.ikcon(Home_T);
38 q_A = Robot.ikcon(A_T);
39 q_B = Robot.ikcon(B_T);
40 q_C = Robot.ikcon(C_T);
41 q_D = Robot.ikcon(D_T);
42
43
44 t = 0:0.04:2;
45
46 traj_home_A = jtraj(q_home, q_A, t);
47 traj_A_B = jtraj(q_A, q_B, t);
48 traj_B_C = jtraj(q_B, q_C, t);
49 traj_C_D = jtraj(q_C, q_D, t);
50 traj_D_home = jtraj(q_D, q_home, t);
```

```

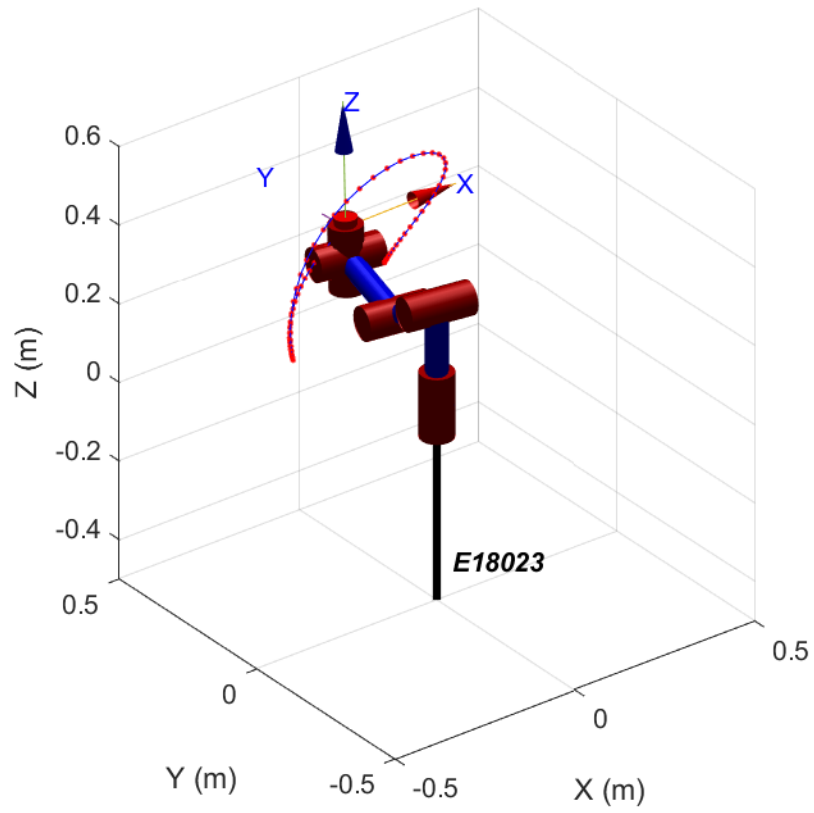
53 segments = {traj_home_A, traj_A_B, traj_B_C, traj_C_D, traj_D_home};
54 segment_names = {'Home_to_A', 'A_to_B', 'B_to_C', 'C_to_D', 'D_to_Home'};
55 save_directory = './frameDIR/';
56
57 figure;
58 hold on;
59
60 for idx = 1:length(segments)
61     plot_trajectory(Robot, segments{idx}, segment_names{idx}, save_directory);
62 end
63
64 hold off;
65
66 % Function to plot and save end-effector trajectory
67 function plot_trajectory(RobotModel, trajectory, segmentName, saveDirectory)
68     endEffectorPositions = zeros(size(trajectory, 1), 3); % End-effector positions array
69
70     for stepIndex = 1:size(trajectory, 1)
71         endEffectorTransform = double(RobotModel.fkine(trajectory(stepIndex, :)));
72         endEffectorPositions(stepIndex, :) = endEffectorTransform(1:3, 4)';
73
74         plot3(endEffectorPositions(1:stepIndex, 1), endEffectorPositions(1:stepIndex, 2), endEffectorPositions(1:stepIndex, 3), 'b');
75         plot2(endEffectorPositions(stepIndex, :), 'r. ');
76
77         RobotModel.plot(trajectory(stepIndex, :));
78
79         frameFileName = fullfile(saveDirectory, [segmentName, '_frame_', num2str(stepIndex), '.png']);
80         saveas(gcf, frameFileName);
81
82         grid on;
83         xlim([-0.5, 0.5]);
84         ylim([-0.5, 0.5]);
85         zlim([-0.5, 0.6]);
86         xlabel('X (m)');
87         ylabel('Y (m)');
88         zlabel('Z (m)');
89         title(['End-Effector Trajectory: ', segmentName]);
90
91         pause(0.05);
92     end
93 end
94
95

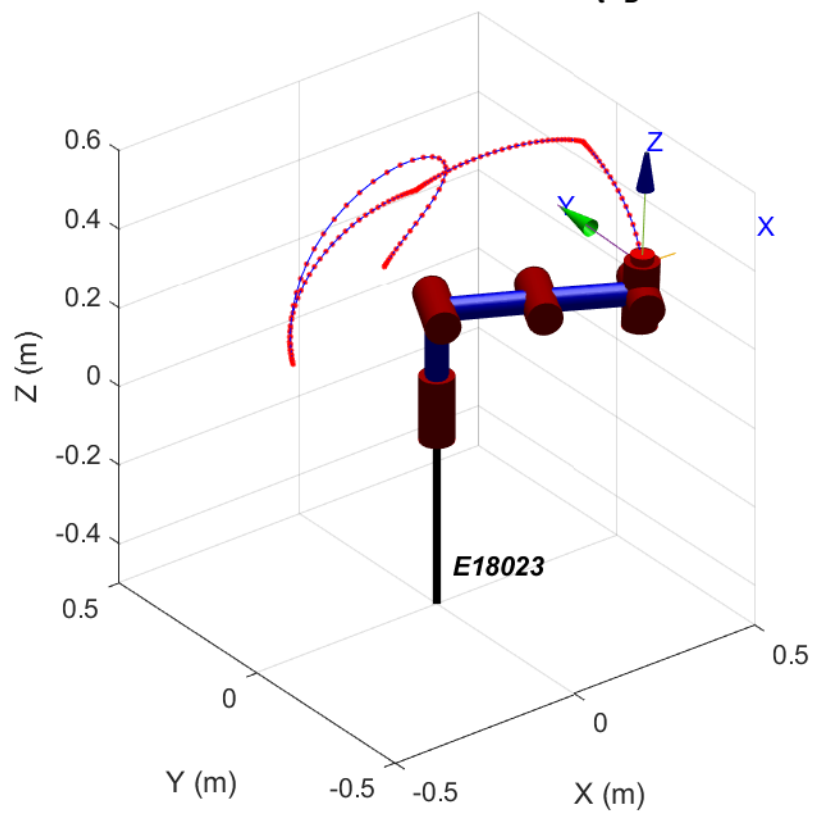
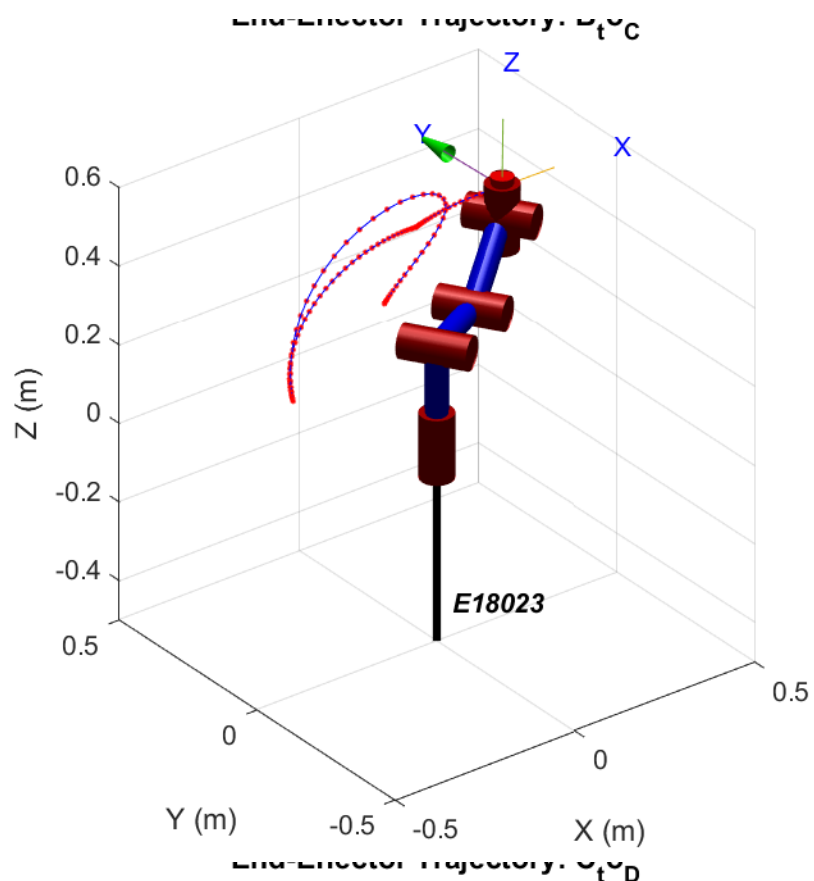
```

End-Effector Trajectory: ${}^{H_0}t_A$



End-Effector Trajectory: ${}^{H_0}t_B$





End-Effector Trajectory: $\mathcal{D}_t \mathcal{O}_{Home}$

