Table of Contents

PA #2: Graduate Group Assignment

Group 8: Akarsh Mohan Konaje, Abigail Hua

```
clear
clc
%%Default Values
theta1 = 180; %180
theta2 = 90; %90
theta3 = 90; %90
D4 = 1; %(Prismatic Joint)
theta4 = D4;
Theta = [theta1,theta2,theta3,theta4];
L2 = 1;
% We also store all the given joint types available in this program.
jointTypeDict = {'R', 'P'};
```

User Inputs

Here we ask user to input parameters,to specify parameters of each joint, such as the type of joint, the value of joint variable, etc.

```
%{
n = input("How many joints are there in the mechanism?\n");
while (~isequal(size(n), [1,1]) || n <= 0)
    fprintf("Errors: The number of joints should be a positive number
\n");
n = input("How many joints are there in the mechanism?\n");
end
Theta = [];
for i = 1:n
    joint_type = input('Please specify the type of the joint (e.g. R,
 P):\n','s');
    while (~any(strcmp(jointTypeDict, joint_type)))
        fprintf('Error: Invalid joint type!\n');
        joint_type = input('Please specify the type of the joint (e.g.
 R, P):\
n','s');
    end
    if (joint_type == 'P')
        fprintf('Please enter the D value of the %d-th joint
 (prismatic) in base
frame:\n',i);
        d_i = input('');
        while (~isequal(size(d_i), [1,1]))
            fprintf('Error: Invalid value!\n');
            fprintf('Please specify the value of d_%d:\n', i);
            d_i = input('');
        end
        Theta(i) = d_i;
    elseif (joint_type == 'R')
        fprintf('Please specify the Theta value of %d-th joint
 (revolute) in base
frame:\n',i);
        theta_i = input('');
        while (~isequal(size(theta_i), [1,1]))
            fprintf('Error: Invalid value!\n');
            fprintf('Please specify the value of theta_%d:\n', i);
            theta_i = input('');
        end
        Theta(i) = theta_i;
    end
end
%}
```

FK Calculations - Final End-Effector Position and Orientation

Transformation Matrix DHmat(thetai,d,a,alpha)

```
T01 = DHmat(Theta(1),    0,  0,  0);
T12 = DHmat(Theta(2),    0,  0, 90);
T23 = DHmat(Theta(3)+90, 0,  L2, 0);
T34 = DHmat(0,    Theta(4), 0, 90);
T = T01*T12*T23*T34;
p_ee = T(1:3,4);
fprintf('Here is the Matrix Orientation of the End-Effector: \n');
T
fprintf('With the xyz co-ordinates of the End-Effector as: \n');
p_ee'
%Axis angle method
Rot_mat = T(1:3,1:3);
tr = trace(Rot_mat);
Rot_angle_ee = acosd((tr-1)/2);
fprintf('The Rotation angle of the End-Effector w.r.t. the base frame
 is: \n');
Rot_angle_ee
k = (1/2*sind(Rot_angle_ee)).*[Rot_mat(3,2)-Rot_mat(2,3);
                               Rot_mat(1,3)-Rot_mat(3,1);
                               Rot_mat(2,1)-Rot_mat(1,2)];
k; %Base frame check [0,0,0]
% T-matrix w.r.t. Base Frame 0
T02=T01*T12;
T03=T02*T23;
```

```
T04=T03*T34;
% Joint Co-ordinates w.r.t. Base Frame
X=[T01(1,4) T02(1,4) T03(1,4) T04(1,4)];
Y=[T01(2,4) T02(2,4) T03(2,4) T04(2,4)];
Z=[T01(3,4) T02(3,4) T03(3,4) T04(3,4)];
```

Simulation

```
a1 = linspace(0,Theta(1),3);  %Theta1 = 180
a2 = linspace(0,Theta(2),3);   %Theta2 = 90
a3 = linspace(0,Theta(3),3);   %Theta3 = 90
disp4 = linspace(0,Theta(4),3); %Theta4 or D4  = 1
for i=1:length(a1)
    for j = 1:length(a2)
        for k = 1:length(a3)
            for l = 1:length(disp4)
                T01 = DHmat(a1(i),    0,  0,  0);
                T12 = DHmat(a2(j),    0,  0, 90);
                T23 = DHmat(a3(k)+90, 0,  L2, 0);
                T34 = DHmat(0, disp4(l), 0, 90);
                T02=T01*T12;
                T03=T02*T23;
                T04=T03*T34;
                X=[T01(1,4) T02(1,4) T03(1,4) T04(1,4)];
                Y=[T01(2,4) T02(2,4) T03(2,4) T04(2,4)];
                Z=[T01(3,4) T02(3,4) T03(3,4) T04(3,4)];
                plot3(X,Y,Z,'-
o','LineWidth',2,'MarkerSize',6,'MarkerFaceColor','auto');
                xlabel("x");
                ylabel("y");
                zlabel("z");
                title("Simulation of the Open Chain Manipulator");
                grid on;
                axis([-2,2,-2,2,0,2]);
                pause(0.01);
            end
        end
    end
end
T = T01*T12*T23*T34;
fprintf('Here is the Matrix Orientation of the End-Effector after
 simulation: \n');
T
```

Functions

```
%DH Matrix Function
function [DH] = DHmat(thetai,d,a,alpha)
DH = [cosd(thetai) -sind(thetai) 0 a;
      sind(thetai)*cosd(alpha) cosd(thetai)*cosd(alpha)  -sind(alpha)
 -d*sind(alpha);
      sind(thetai)*sind(alpha) cosd(thetai)*sind(alpha) cosd(alpha)
 d*cosd(alpha);
```

```
      0 0 0 1];
end
```

```
Here is the Matrix Orientation of the End-Effector:

T =

     0     0     1     1
     0     1     0     0
    -1     0     0     1
     0     0     0     1

With the xyz co-ordinates of the End-Effector as:

ans =

     1     0     1

The Rotation angle of the End-Effector w.r.t. the base frame is:

Rot_angle_ee =

    90
```

Published with MATLAB® R2020b