Table of Contents

Problem 2

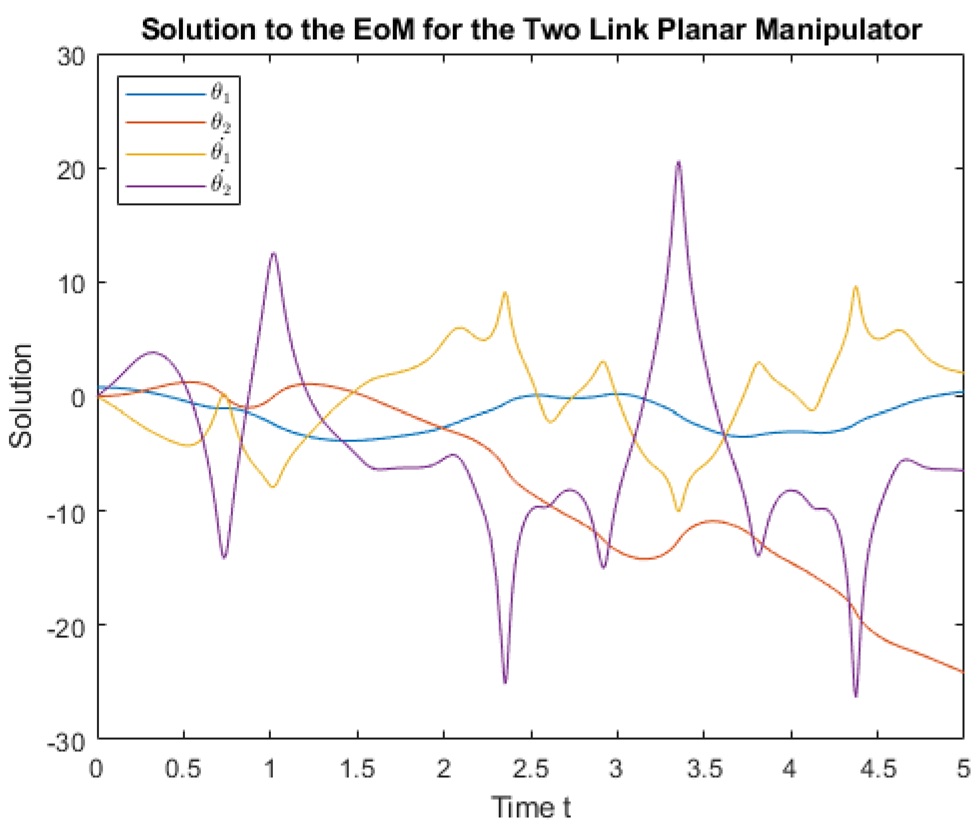
```
clear
clc
```

Solve the differential equation system

```
[t, y] = solveode(pi/4, 0, 0, 0);
```

Plot

```
plotode(t, y);
```



Simulation

```
f2 = figure;
xlim([-2 2]);
ylim([-2 2]);
title('Simulation');
pbaspect([1 1 1]);
hold on
for i=1:size(y, 1)
    th1 = y(i, 1);
    th2 = y(i, 2);
    x1 = 1 * cos(th1);
    x2 = x1 + 1 * cos(th1 + th2);
    y1 = 1 * sin(th1);
    y2 = y1 + 1 * sin(th1 + th2);
    a1 = plot([0;x1], [0;y1], 'm', 'LineWidth', 2);
    a2 = plot([x1;x2], [y1;y2], 'c', 'LineWidth', 2);
    if i < size(y, 1)
        pause(0.02);
        set(a1, 'Visible', 'off');
        set(a2, 'Visible', 'off');
    end
end
hold off
pause(1)
close(f2);
```

Compare different IC

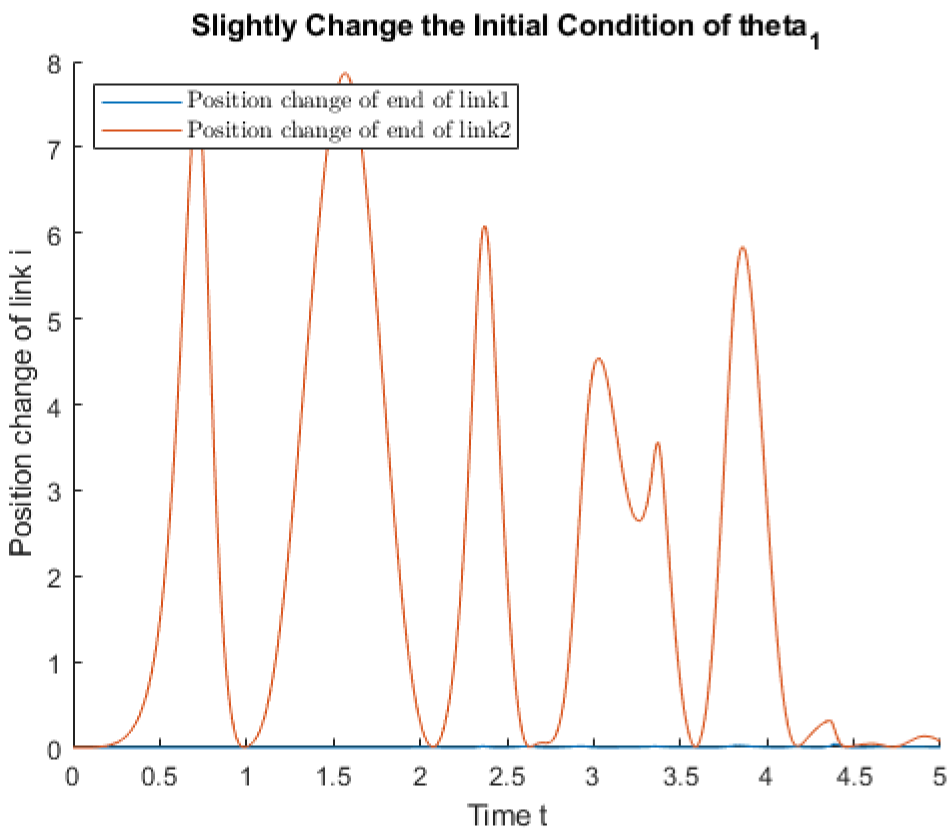
In this section, I slightly change the initial condition and see which joint is more sensitive to initial condition.

I model the sensitivity of joint i as the difference in the position of the end of link i when changing the initial condition.

```
t0 = (0:.01:10)';
[t1, y1] = solveode(pi/4*1.01, 0, 0, 0);
[t2, y2] = solveode(pi/4, pi/400, 0, 0);
y = interp1(t, y, t0);
y1 = interp1(t1, y1, t0);
y2 = interp1(t2, y2, t0);
```

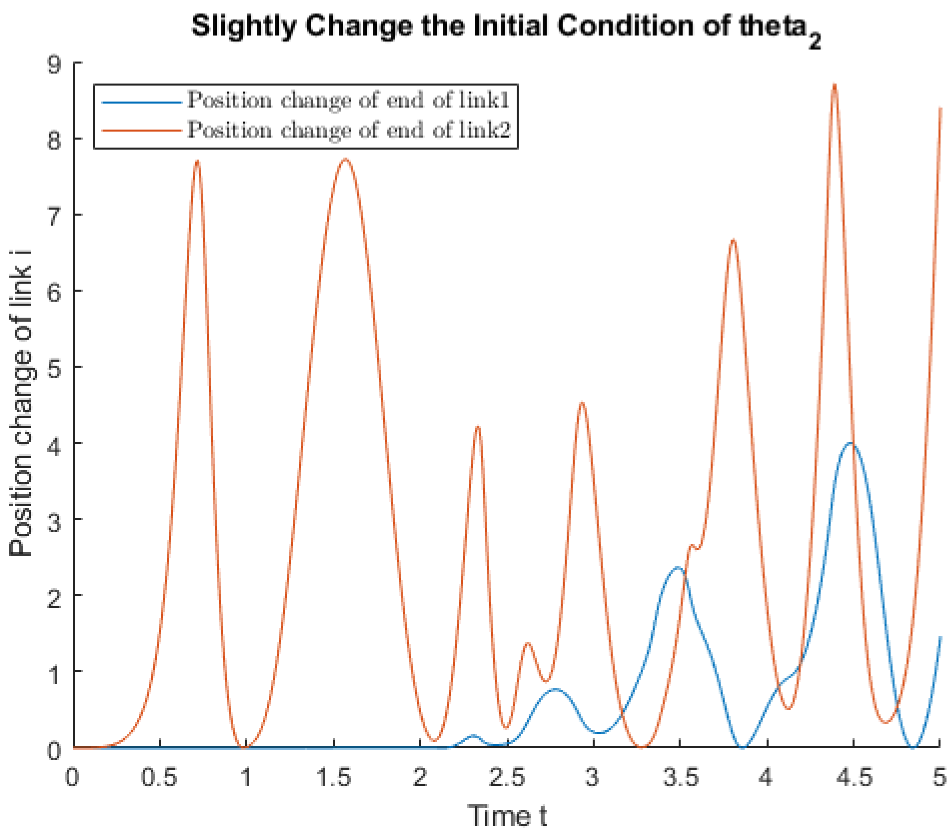
First I slightly change the initial condition of theta_1, i.e., theta_1(0) = pi/4*1.01, theta_2(0) = 0.

```
f3 = figure;
hold on
plotPos(t0, y, y1, 1);
hold off
```



Then I slightly change the initial condition of theta_2, i.e., theta_1(0) = pi/4, theta_2(0) = pi/4*0.01.

```
f4 = figure;
hold on
plotPos(t0, y, y2, 2);
hold off
```



As we can see from the graphs, in both cases, the deviation of link 2 is larger than that of link 1, indicating that joint 2 is more sensitive to initial condition, which also complies with the instinction that joint 2 is affected by more variables.

Functions

I use the following function to get the symbolic expression of theta_1_dotdot and theta_2_dotdot. However, for performance issue, I run it in a separate file and do not use it directly in simulation process.

```
function [t1dd, t2dd] = getEoM(t1, t2)
    clear
    clc
    syms l1 l2 th1 th2 m1 m2 I_c1 I_c2 g th1_dd th2_dd th1_d th2_d
    l1 = 1;
    l2 = 1;
    m1 = 1;
    m2 = 1;
    I_c1 = 1/12;
    I_c2 = 1/12;

    p_c1 = [l1*cos(th1)/2; l1*sin(th1)/2; 0];
    p_c2 = [l1*cos(th1) + l2*cos(th1+th2)/2; l1*sin(th1) +
 l2*sin(th1+th2)/2; 0];
    J_v1 = [diff(p_c1, th1) zeros(3, 1)];
    J_v2 = [diff(p_c2, th1) diff(p_c2, th2)];
    J_w1 = [0 0; 0 0; 1 0];
```

```
    J_w2 = [0 0; 0 0; 1 1];
    M = simplify(m1 * (J_v1.' * J_v1) + J_w1' * I_c1 * J_w1 + m2 *
 (J_v2.' * J_v2) + J_w2' * I_c2 * J_w2);

    mxxx = sym('m', [2, 2, 2]);
    mxxx(:, :, 1) = diff(M, th1);
    mxxx(:, :, 2) = diff(M, th2);
    bxxx = sym('b', [2, 2, 2]);
    for i=1:2
        for j=1:2
            for k=1:2
                bxxx(i,j,k) = (mxxx(i,j,k)+mxxx(i,k,j)-mxxx(j,k,i))/2;
            end
        end
    end
    gamma = [bxxx(1,1,1)*th1_d^2 +
 (bxxx(1,1,2)+bxxx(1,2,1))*th1_d*th2_d + bxxx(1,2,2)*th2_d^2;
             bxxx(2,1,1)*th1_d^2 +
 (bxxx(2,1,2)+bxxx(2,2,1))*th1_d*th2_d + bxxx(2,2,2)*th2_d^2];

    vec_g = [0; -g; 0];
    G = - J_v1.'*m1*vec_g - J_v2.'*m2*vec_g;
    EoM = M * [th1_dd; th2_dd] + gamma + G;
    EoMAnswers = solve(EoM == zeros(2, 1), th1_dd, th2_dd);
    t1dd = subs(EoMAnswers.th1_dd);
    t2dd = subs(EoMAnswers.th2_dd);
end
```

Function to plot the result of ode

```
function plotode(t, y)
    f1 = figure;
    plot(t,y(:,1),t,y(:,2), t, y(:,3), t, y(:,4));
    title('Solution to the EoM for the Two Link Planar Manipulator');
    xlabel('Time t');
    ylabel('Solution');
    legend({'$\theta_1$', '$\theta_2$', '$\dot{\theta_1}$', '$
\dot{\theta_2}$'}, 'Location', 'northwest', 'Interpreter','latex')
    pause(3);
end
```

ode solver

```
function [t, y] = solveode(th1_0, th2_0, th1_dot_0, th2_dot_0)
    [t, y] = ode45(@(t,Y) odefcn(t,Y) , [0, 5] , [th1_0, th2_0,
 th1_dot_0, th2_dot_0]');
end

function dYdt = odefcn(t,Y)
    g = 9.8;
    t1dd = -(3*(2*Y(3)^2*sin(Y(2)) + 2*Y(4)^2*sin(Y(2))
 - 6*g*cos(Y(1)) + 4*Y(3)*Y(4)*sin(Y(2)) + 3*g*cos(Y(1) +
 Y(2))*cos(Y(2)) + 3*Y(3)^2*cos(Y(2))*sin(Y(2))))/(9*cos(Y(2))^2 -
 16);
```

```
    t2dd = (3*(10*Y(3)^2*sin(Y(2)) + 2*Y(4)^2*sin(Y(2)) +
 8*g*cos(Y(1) + Y(2)) - 6*g*cos(Y(1)) + 4*Y(3)*Y(4)*sin(Y(2))
 + 3*g*cos(Y(1) + Y(2))*cos(Y(2)) - 9*g*cos(Y(1))*cos(Y(2)) +
 6*Y(3)^2*cos(Y(2))*sin(Y(2)) + 3*Y(4)^2*cos(Y(2))*sin(Y(2)) +
 6*Y(3)*Y(4)*cos(Y(2))*sin(Y(2))))/(9*cos(Y(2))^2 - 16);

    dYdt = [ Y(3);
             Y(4);
             t1dd;
             t2dd];
end

function plotPos(t, Y0, Y, i)
    l1 = 1;
    l2 = 1;
    x1 = l1*cos(Y(:, 1)) - l1*cos(Y0(:, 1));
    y1 = l1*sin(Y(:, 1)) - l1*sin(Y0(:, 1));
    x2 = l1*cos(Y(:, 1)) + l2*cos(Y(:, 1)+Y(:, 2)) - (l1*cos(Y0(:, 1))
 + l2*cos(Y0(:, 1)+Y0(:, 2)));
    y2 = l1*sin(Y(:, 1)) + l2*sin(Y(:, 1)+Y(:, 2)) - (l1*cos(Y0(:, 1))
 + l2*cos(Y0(:, 1)+Y0(:, 2)));
    diff1 = x1 .* x1 + y1 .* y1;
    diff2 = x2 .* x2 + y2 .* y2;
    plot(t, diff1, t, diff2);
    title(sprintf('Slightly Change the Initial Condition of theta_%d',
 i));
    xlabel('Time t');
    ylabel('Position change of link i');
    legend({'Position change of end of link1', 'Position change of end
 of link2'}, 'Location', 'northwest', 'Interpreter','latex')
end
```

Reference

1. Mathworks forum about how to solve differential equation system with ode45:

https://www.mathworks.com/matlabcentral/answers/305700-solve-ode-system-with-ode45

2. Documentation of syms in Matlab: https://www.mathworks.com/help/symbolic/syms.html

3. Documentation of subs in Matlab: https://www.mathworks.com/help/symbolic/subs.html

Published with MATLAB® R2020b