

```

1 clear all
2 close all
3 clc
4
5 %% Declarations
6 % Parameters
7 L = 1;
8 M = 1;
9 m = 1;
10
11 % Initial position
12 x0 = 0;
13 theta1_0 = pi/4;
14 theta2_0 = pi/2;
15 q0 = [x0; theta1_0; theta2_0];
16
17 % Initial velocity
18 dq0 = zeros(3,1);
19
20 % Initial state
21 %      1  2      3      4  5      6
22 % x = [x; theta1; theta2; dx; dtheta1; dtheta2];
23 state = [q0; dq0];
24
25
26 %% Simulation
27 tf = 45;
28
29 % Function declarations
30 W = @(x) prob1_W_hessian(L,M,m,x(2),x(3));
31 other = @(x) prob1_other_vector(L,M,x(5),x(6),x(4),x(2),x(3),x(1));
32 simFunc = @(t, x) [x(4:6); W(x) \ other(x)];
33
34 [time, statetrajectory] = ode45(simFunc, [0 tf], state);
35
36
37 %% 3D animation
38 DoublePlot = true;
39 FS = 30;
40 scale = 0.1;
41
42 % Create Objects
43 % Cube
44 vert{1} = 3*[ -1, -1, 0; %1
45              1, -1, 0; %2
46              1, 1, 0; %3
47              -1, 1, 0; %4
48              -1, -1, 2; %5
49              1, -1, 2; %6
50              1, 1, 2; %7
51              -1, 1, 2]/2; %8
52 fac{1} = [1 2 3 4;
53           5 6 7 8;
54           1 4 8 5;
55           1 2 6 5;
56           2 3 7 6;
57           3 4 8 7];
58 Lrail = 1.2*max(abs(statetrajectory(:,1)))/scale;

```

```

59 % Rail
60 a = 1.5;
61 vert{2} = [-Lrail,-a,-0.1;
62            -Lrail, a,-0.1;
63            Lrail, a,-0.1;
64            Lrail,-a,-0.1];
65 fac{2} = [1,2,3,4];
66 % Sphere
67 [X,Y,Z] = sphere(20);
68 [fac{3},vert{3},c] = surf2patch(3*X/2,3*Y/2,3*Z/2);
69 % Animation
70 tic
71 t_disp = 0;
72 SimSpeed = 1;
73 while t_disp < tf/SimSpeed
74     % Interpolate state
75     state_animate = interp1(time,statetraj,SimSpeed*t_disp)';
76
77     x = state_animate(1);
78     theta1 = state_animate(2);
79     theta2 = state_animate(3);
80
81     p0 = [x; 0]; % box
82     p1 = p0 + [L*sin(theta1); -L*cos(theta1)]; % sphere 1
83     p2 = p1 + [L*sin(theta2); -L*cos(theta2)]; % sphere 2
84
85     % shift coords
86     p0_3d = [-p0(1); 0; p0(2)];
87     p1_3d = [-p1(1); 0; p1(2)];
88     p2_3d = [-p2(1); 0; p2(2)];
89
90     % Input argument for DrawPendulum
91     pos_disp = [p0_3d(1); p1_3d; p2_3d];
92
93     figure(1);clf;hold on
94     if DoublePlot
95         subplot(1,2,1);hold on
96         DrawPendulum( pos_disp, vert, fac, scale);
97         campos(scale*[15 15 -70])
98         camtarget(scale*[0,0,1.5])
99         camva(30)
100        camproj('perspective')
101        subplot(1,2,2);hold on
102    end
103    DrawPendulum( pos_disp, vert, fac, scale);
104    campos(scale*[1 70 20])
105    camtarget(scale*[0,0,1.5])
106    camva(30)
107    camproj('perspective')
108    drawnow
109    if t_disp == 0
110        display('Hit a key to start animation')
111        pause
112        tic
113    end
114    t_disp = toc;
115 end
116

```

