HW3 Gabe Colangelo

clear

close all

warning off

clc

syms x1 x2 x3 x4 x5 x6 u1 u2 u3 x(t) x\_dot(t) theta1(t) theta\_dot\_1(t)...

theta2(t) theta\_dot\_2(t) t theta\_ddot\_1 x\_ddot theta\_ddot\_2...

m1 m2 M L1 L2 g real

% Numeric System Parameters

m1\_num = 0.5;

L1\_num = 0.5;

m2\_num = 0.75;

L2\_num = 0.75;

M\_num = 1.5;

g\_num = 9.81;

# Problem 1 - Equilibrium Input for single input

% Equilibrium States

xe = [0.1, deg2rad(60), deg2rad(45), 0, 0, 0 ]';

% Non-linear Model with single input

xdot = DIPC\_1([],xe,u1,m1\_num,m2\_num,M\_num,L1\_num,L2\_num,g\_num);

% Solve for ue if it exists

ue\_1 = solve(xdot == 0,u1);

if isempty(ue\_1)

disp('There does not exist a u\_e such that a single input can acheive the desired equilibirum state')

end

There does not exist a u\_e such that a single input can acheive the desired equilibirum state

# Problem 2 - Equilibrium Input for two inputs

% Non-linear Model with two inputs

xdot\_2 = DIPC\_2([],xe,[u1;u2],m1\_num,m2\_num,M\_num,L1\_num,L2\_num,g\_num);

% Solve for ue if it exists

sol\_2 = solve(xdot\_2 == 0,[u1;u2]);

ue\_2 = [sol\_2.u1;sol\_2.u2];

if isempty(ue\_2)

disp('There does not exist a u\_e such that two inputs can achieve the desired equilibrium state')

end

There does not exist a u\_e such that two inputs can achieve the desired equilibrium state

# Problem 3 - Equilibrium Input for Three inputs

% Call Lagrangian for DIPC

L = DIPC\_Lagrangian(t,x,x\_dot, theta1, theta\_dot\_1, theta2, theta\_dot\_2, M, m1,m2, L1, L2, g);

% Solve Lagrange's Equations of Motion

% q = x, Q = u1

eqn\_x = subs(simplify(diff(diff(L,x\_dot),t) - diff(L,x)),[diff(x(t),t), diff(theta1(t),t)...

,diff(theta2(t),t), diff(x\_dot,t), diff(theta\_dot\_1(t), t), diff(theta\_dot\_2(t), t)],...

[x\_dot, theta\_dot\_1, theta\_dot\_2, x\_ddot, theta\_ddot\_1, theta\_ddot\_2]) == u1;

% q = theta1, Q = u2

eqn\_theta1 = subs(simplify(diff(diff(L,theta\_dot\_1),t) - diff(L,theta1)),[diff(x(t),t), diff(theta1(t),t)...

,diff(theta2(t),t), diff(x\_dot,t), diff(theta\_dot\_1(t), t), diff(theta\_dot\_2(t), t)],...

[x\_dot, theta\_dot\_1, theta\_dot\_2, x\_ddot, theta\_ddot\_1, theta\_ddot\_2]) == u2;

% q = theta2, Q = u3

eqn\_theta2 = subs(simplify(diff(diff(L,theta\_dot\_2),t) - diff(L,theta2)),[diff(x(t),t), diff(theta1(t),t)...

,diff(theta2(t),t), diff(x\_dot,t), diff(theta\_dot\_1(t), t), diff(theta\_dot\_2(t), t)],...

[x\_dot, theta\_dot\_1, theta\_dot\_2, x\_ddot, theta\_ddot\_1, theta\_ddot\_2]) == u3;

% Solve system of equations for 2nd derivative of states

sys\_eqn = solve([eqn\_x,eqn\_theta1,eqn\_theta2],[x\_ddot,theta\_ddot\_1,theta\_ddot\_2]);

% Put EOM into state space form

x4\_dot = subs(simplify(sys\_eqn.x\_ddot),[x theta1 theta2 x\_dot theta\_dot\_1 theta\_dot\_2],[x1 x2 x3 x4 x5 x6]);

x5\_dot = subs(simplify(sys\_eqn.theta\_ddot\_1),[x theta1 theta2 x\_dot theta\_dot\_1 theta\_dot\_2],[x1 x2 x3 x4 x5 x6]);

x6\_dot = subs(simplify(sys\_eqn.theta\_ddot\_2),[x theta1 theta2 x\_dot theta\_dot\_1 theta\_dot\_2],[x1 x2 x3 x4 x5 x6]);

% Use fsolve to find ue if it exists

fsol\_opt = optimset('Display','off');

fun = @(u)DIPC\_3([],xe,u,m1\_num,m2\_num,M\_num,L1\_num,L2\_num,g\_num);

disp('There does not exist a u\_e such that two inputs can achieve the desired equilibrium state')

There does not exist a u\_e such that two inputs can achieve the desired equilibrium state

ue = fsolve(fun,[0 0 0]',fsol\_opt)

ue = 3×1

0.0000

-5.3098

-3.9019

# Problem 4 - Taylor Series Expansion

% Non-linear System

f = DIPC\_3([],[x1;x2;x3;x4;x5;x6],[u1;u2;u3],m1\_num,m2\_num,M\_num,L1\_num,L2\_num,g\_num);

h = [x1;x2;x3];

% Equilibrium output

ye = double(subs(h,[x1;x2;x3;x4;x5;x6],xe));

disp('The linearized state space model matrices are given by')

The linearized state space model matrices are given by

% Jacobian Matrices/ Linearized Model about origin

A = double(subs(jacobian(f,[x1;x2;x3;x4;x5;x6]),[x1;x2;x3;x4;x5;x6;u1;u2;u3],[xe;ue]))

A = 6×6

0 0 0 1.0000 0 0

0 0 0 0 1.0000 0

0 0 0 0 0 1.0000

0 -0.5341 -1.1264 0 0 0

0 22.5046 -17.8041 0 0 0

0 -13.9883 21.7759 0 0 0

B = double(subs(jacobian(f,[u1;u2;u3]),[x1;x2;x3;x4;x5;x6;u1;u2;u3],[xe;ue]))

B = 6×3

0 0 0

0 0 0

0 0 0

0.4252 -0.1742 -0.2887

-0.1742 7.3409 -4.5629

-0.2887 -4.5629 5.5808

C = double(jacobian(h,[x1;x2;x3;x4;x5;x6]))

C = 3×6

1 0 0 0 0 0

0 1 0 0 0 0

0 0 1 0 0 0

D = double(jacobian(h,[u1;u2;u3]))

D = 3×3

0 0 0

0 0 0

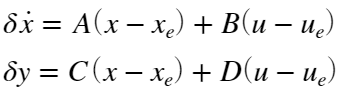
0 0 0

del\_x = [x1;x2;x3;x4;x5;x6] - xe;

del\_u = [u1;u2;u3] - ue;

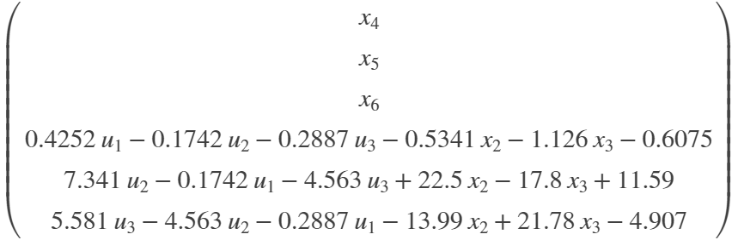
disp('The Taylor series expansion about (xe,ue) is: ')

The Taylor series expansion about (xe,ue) is:



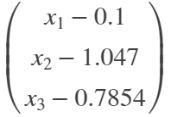
del\_xdot= vpa(A\*del\_x + B\*del\_u,4)

del\_xdot =



del\_y = vpa(C\*del\_x + D\*del\_u,4)

del\_y =



# Problem 5 - State Feedback Controller Design

% Check system controllability

co = ctrb(A,B);

if rank(co) == length(A)

disp('The pair (A,B) is controllable')

end

The pair (A,B) is controllable

% Get dimensions of B

[n, m] = size(B);

% Controller Robustness Term

alpha\_K= 2;

% Use CVX to solve matrix inequality and determine K

cvx\_begin sdp quiet

% Variable definition

variable S(n, n) symmetric

variable Z(m, n)

% LMIs

S\*A' + A\*S -Z'\*B'- B\*Z +2\*alpha\_K\*S <= -eps\*eye(n);

S >= eps\*eye(n);

cvx\_end

disp('The linear state-feedback controller applied to the non-linear model is: u = -K\*del\_x + u\_e')

The linear state-feedback controller applied to the non-linear model is: u = -K\*del\_x + u\_e



disp('The control gains for the applied control law are:')

The control gains for the applied control law are:

% compute K matrix

K = Z/S

K = 3×6

46.5967 5.9210 7.6456 20.5268 2.6616 3.3223

4.0451 8.0484 3.3714 2.2284 2.2719 2.0270

5.2509 2.7333 11.4612 2.9236 2.0400 3.1918

% State IC [m, rad, rad, m/s, rad/s, rad/s]

x0 = [0 .01 .02 0 0 0]';

% Time interval and vector

dt = 1/200;

time = (0:dt:3)';

% ODE solver options

options = odeset('AbsTol',1e-8,'RelTol',1e-8);

% ODE45 Function call

[~, X\_ode45] = ode45(@(t,x) ControlledDIPC\_3([], x, xe, ue, K, m1\_num, m2\_num, M\_num, L1\_num, L2\_num, g\_num), time, x0, options);

% ODE23 Function call

[~, X\_ode23] = ode23(@(t,x) ControlledDIPC\_3([], x, xe, ue, K, m1\_num, m2\_num, M\_num, L1\_num, L2\_num, g\_num), time, x0, options);

figure

plot(time,X\_ode23(:,1),time, X\_ode45(:,1),'--')

yline(xe(1),'--r')

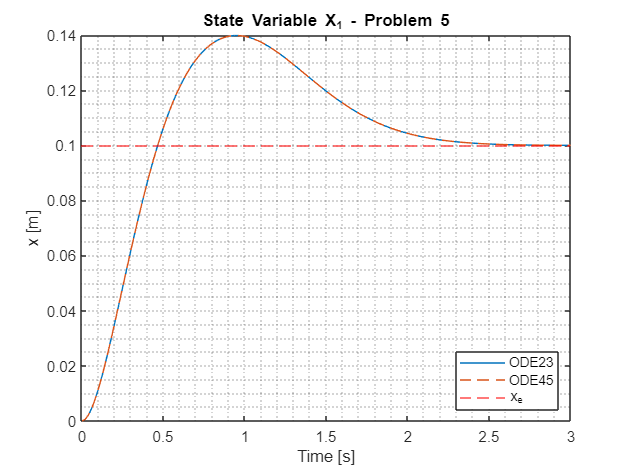
title('State Variable X\_1 - Problem 5')

legend('ODE23','ODE45','x\_1\_e','Location','southeast')

ylabel('x [m]')

grid minor

xlabel('Time [s]')



figure

plot(time,X\_ode23(:,2)\*180/pi,time, X\_ode45(:,2)\*180/pi,'--')

yline(xe(2)\*180/pi,'--r')

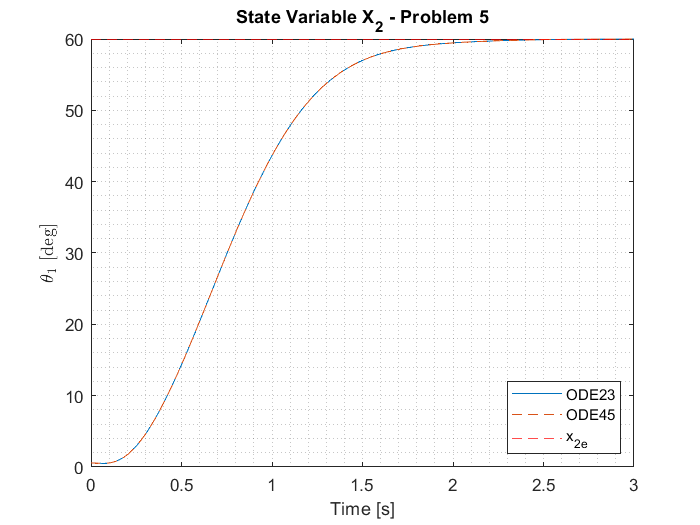
title('State Variable X\_2 - Problem 5')

ylabel('$\theta\_1$ [deg]','Interpreter','latex')

grid minor

xlabel('Time [s]')

legend('ODE23','ODE45','x\_2\_e','Location','southeast')



figure

plot(time,X\_ode23(:,3)\*180/pi,time, X\_ode45(:,3)\*180/pi,'--')

yline(xe(3)\*180/pi,'--r')

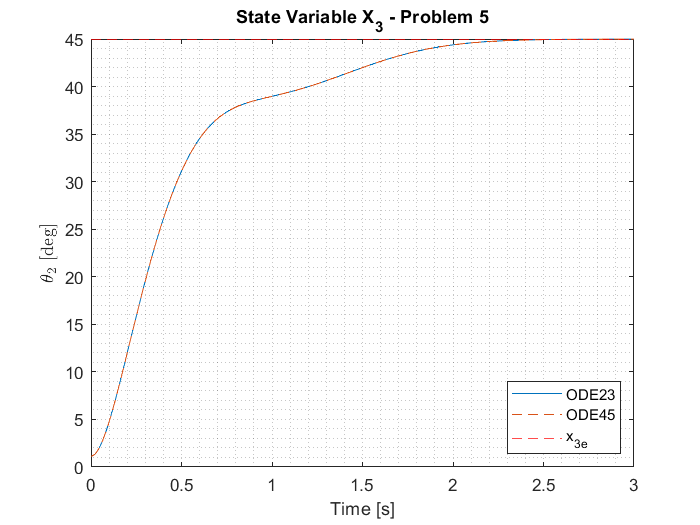
title('State Variable X\_3 - Problem 5')

ylabel('$\theta\_2$ [deg]','Interpreter','latex')

grid minor

xlabel('Time [s]')

legend('ODE23','ODE45','x\_3\_e','Location','southeast')



figure

plot(time,X\_ode23(:,4),time, X\_ode45(:,4),'--')

yline(xe(4),'--r')

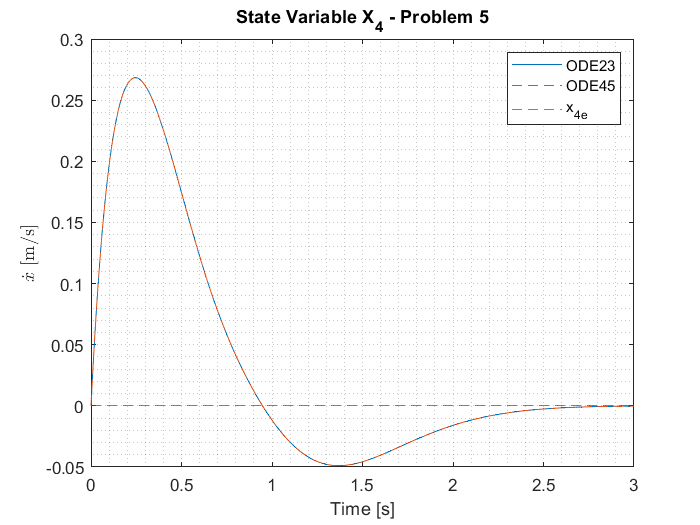
title('State Variable X\_4 - Problem 5')

ylabel('$\dot{x}$ [m/s]','Interpreter','latex')

grid minor

xlabel('Time [s]')

legend('ODE23','ODE45','x\_4\_e')



figure

plot(time,X\_ode23(:,5)\*180/pi,time, X\_ode45(:,5)\*180/pi,'--')

yline(xe(5)\*180/pi,'--r')

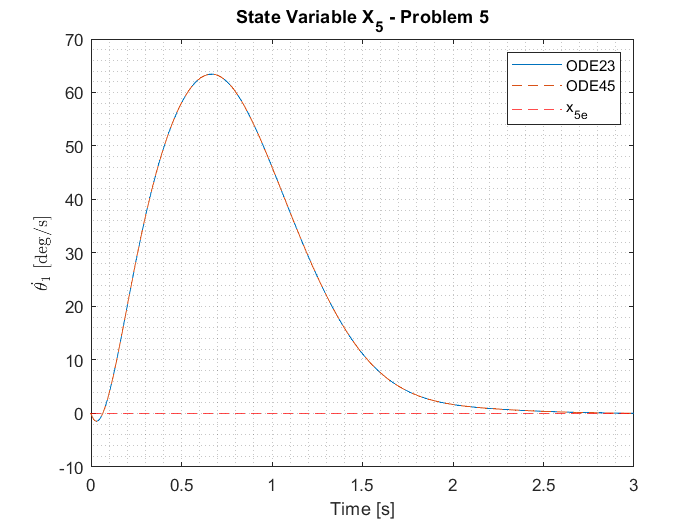
title('State Variable X\_5 - Problem 5')

ylabel('$\dot{\theta\_1}$ [deg/s]','Interpreter','latex')

xlabel('Time [s]')

grid minor

legend('ODE23','ODE45','x\_5\_e')



figure

plot(time,X\_ode23(:,6)\*180/pi,time, X\_ode45(:,6)\*180/pi,'--')

yline(xe(6)\*180/pi,'--r')

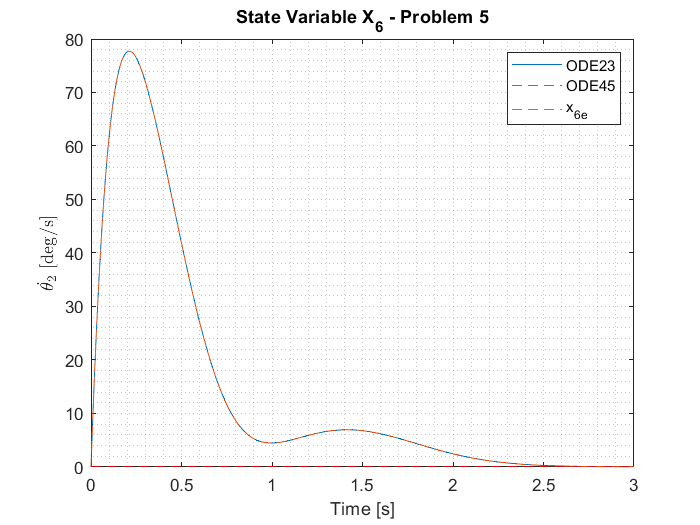
title('State Variable X\_6 - Problem 5')

ylabel('$\dot{\theta\_2}$ [deg/s]','Interpreter','latex')

grid minor

xlabel('Time [s]')

legend('ODE23','ODE45','x\_6\_e')



Upon inspection, there does not appear to be a noticeable difference between the performance of the ODE45 solver and the ODE23 solver. This held true for any initial condition, x0, and it should be noted that I increased the tolerances on both solvers to be 1e-8. This increase of the tolerances may have prevented numerical differences.

# Problem 6 - Output Feedback Controller Design

% Get dimensions of C

[p,~] = size(C);

cvx\_begin sdp quiet

% Variable definition

variable P(n, n) symmetric

variable N(m, p)

variable M(m, m)

% LMIs - output feedback

P\*A + A'\*P - B\*N\*C - C'\*N'\*B' <= -eps\*eye(n)

B\*M == P\*B

P >= eps\*eye(n);

cvx\_end

disp('The output feedback controller applied to the non-linear model is: u = -K\*del\_y + u\_e')

The output feedback controller applied to the non-linear model is: u = -K\*del\_y + u\_e



disp('The control gains for the applied control law are:')

The control gains for the applied control law are:

% compute K matrix for output feedback controller

K0 = M\N

K0 = 3×3

1.1673 0.3313 0.4428

-0.0413 4.1491 0.8682

-0.0446 0.7837 5.4127

% ODE45 Function call

[~, X\_output] = ode45(@(t,x) OutputControlledDIPC\_3([], x, ye, ue, K0, m1\_num, m2\_num, M\_num, L1\_num, L2\_num, g\_num), time, x0, options);

figure

subplot(611)

plot(time,X\_output(:,1))

yline(xe(1),'--r')

legend('x','x\_e')

title('State Variables - Problem 6')

ylabel('x [m]')

grid minor

xlabel('Time [s]')

subplot(612)

plot(time,X\_output(:,2)\*180/pi)

yline(xe(2)\*180/pi,'--r')

ylabel('$\theta\_1$ [deg]','Interpreter','latex')

grid minor

xlabel('Time [s]')

subplot(613)

plot(time,X\_output(:,3)\*180/pi)

yline(xe(3)\*180/pi,'--r')

ylabel('$\theta\_2$ [deg]','Interpreter','latex')

grid minor

xlabel('Time [s]')

subplot(614)

plot(time,X\_output(:,4))

ylabel('$\dot{x}$ [m/s]','Interpreter','latex')

yline(xe(4),'--r')

grid minor

xlabel('Time [s]')

subplot(615)

plot(time,X\_output(:,5)\*180/pi)

yline(xe(5)\*180/pi,'--r')

ylabel('$\dot{\theta\_1}$ [deg/s]','Interpreter','latex')

xlabel('Time [s]')

grid minor

subplot(616)

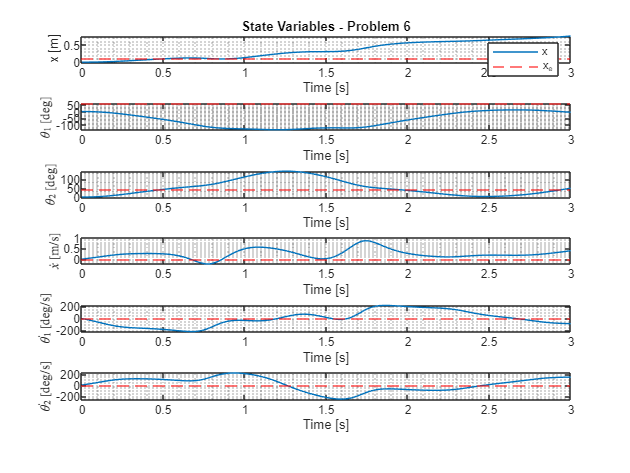
plot(time,X\_output(:,6)\*180/pi)

yline(xe(6)\*180/pi,'--r')

ylabel('$\dot{\theta\_2}$ [deg/s]','Interpreter','latex')

grid minor

xlabel('Time [s]')



It is clearly observed that the output feedback controller doesn’t stabilize the system about the desired equilibrium state, or even stabilize at all for that matter. From the Kimura*-*Davison condition we can see that a stabilizing output feedback controller does not exist (n = 6 which is not less than or equal to (r + m -1 = 5).

# Problem 7 - Combined Controller- Observer Compensator

% Check system observability

ob = obsv(A,C);

if rank(ob) == length(A)

disp('The pair (A,C) is observable')

end

The pair (A,C) is observable

% Observer Robustness Term

alpha\_L = 8;

% Use CVX to solve matrix inequality and determine L

cvx\_begin sdp quiet

% Variable definition

variable P(n, n) symmetric

variable Y(n, p)

% LMI with robustness term (all eigenvalues less than -2)

A'\*P + P\*A - C'\*Y' - Y\*C + 4\*alpha\_L\*P <= -eps\*eye(n);

P >= eps\*eye(n)

cvx\_end

disp('The control gains for the Luenberger observer are:')

The control gains for the Luenberger observer are:

% solver for observer gain matrix

L = P\Y

L = 6×3

103 ×

0.0496 0.0001 -0.0002

0.0032 0.1169 -0.0522

-0.0014 -0.0504 0.1087

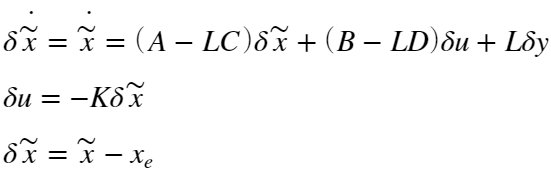
1.0603 0.0023 -0.0083

0.0602 2.6755 -1.2292

-0.0321 -1.1452 2.4280

disp('The Luenberger observer takes the form of:')

The Luenberger observer takes the form of:



% Observer IC

z0 = zeros(6,1);

% ODE45 Function call

[~, X\_comp] = ode45(@(t,x) CombinedCompensatorDIPC([], x, xe, ye, ue, K, L, A, B, C, D, m1\_num, m2\_num, M\_num, L1\_num, L2\_num, g\_num),...

time, [x0;z0], options);

figure

plot(time,X\_comp(:,1),time, X\_comp(:,7),'--')

yline(xe(1),'--r')

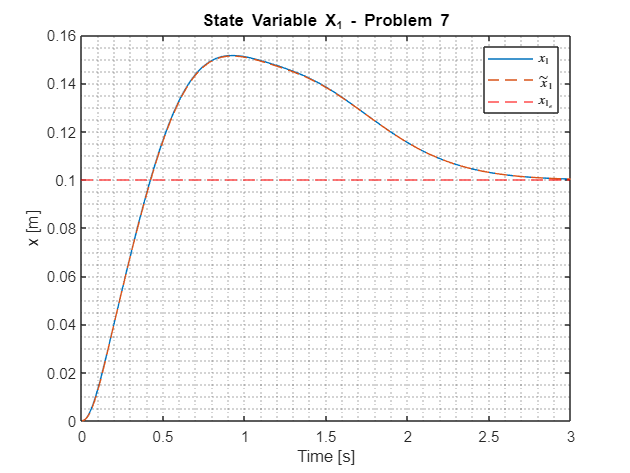
title('State Variable X\_1 - Problem 7')

legend('$x\_1$','$\tilde{x}\_1$','$x\_{1\_e}$','Interpreter','latex')

ylabel('x [m]')

grid minor

xlabel('Time [s]')



figure

plot(time,X\_comp(:,2)\*180/pi,time, X\_comp(:,8)\*180/pi,'--')

yline(xe(2)\*180/pi,'--r')

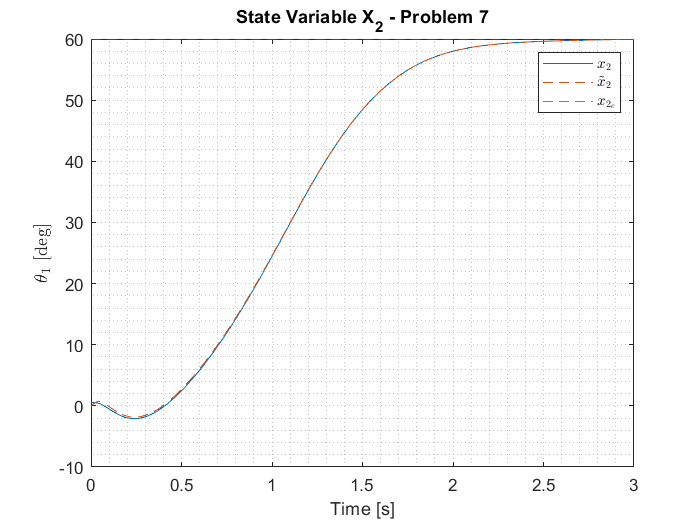
title('State Variable X\_2 - Problem 7')

legend('$x\_2$','$\tilde{x}\_2$','$x\_{2\_e}$','Interpreter','latex')

ylabel('$\theta\_1$ [deg]','Interpreter','latex')

grid minor

xlabel('Time [s]')



figure

plot(time,X\_comp(:,3)\*180/pi,time, X\_comp(:,9)\*180/pi,'--')

yline(xe(3)\*180/pi,'--r')

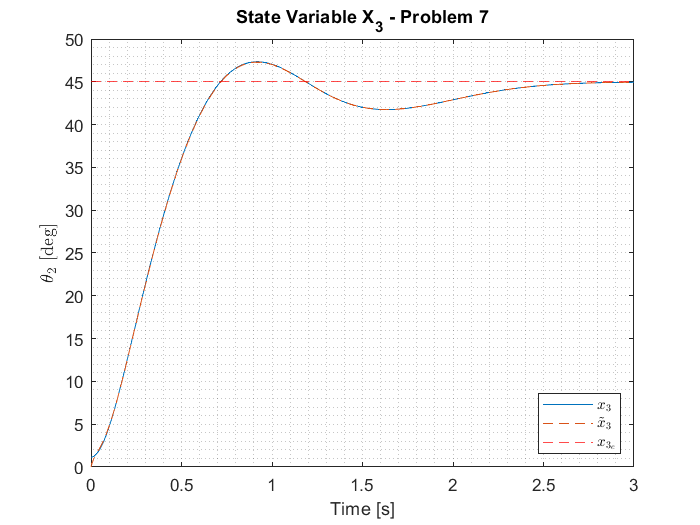
title('State Variable X\_3 - Problem 7')

legend('$x\_3$','$\tilde{x}\_3$','$x\_{3\_e}$','Interpreter','latex','Location','southeast')

ylabel('$\theta\_2$ [deg]','Interpreter','latex')

grid minor

xlabel('Time [s]')



figure

plot(time,X\_comp(:,4),time, X\_comp(:,10),'--')

yline(xe(4),'--r')

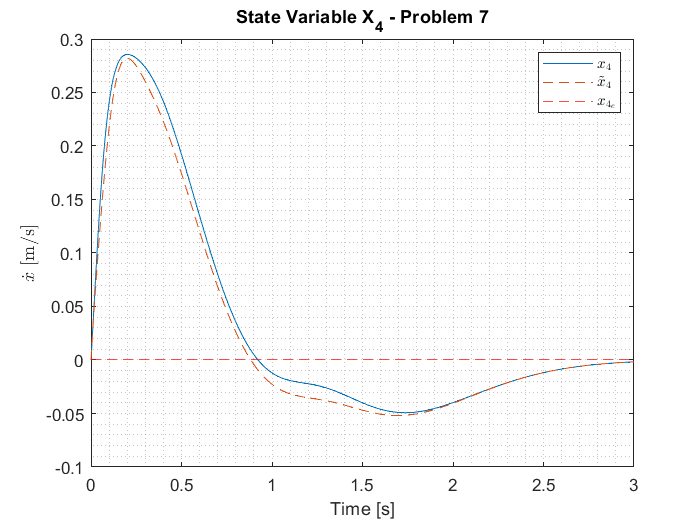
title('State Variable X\_4 - Problem 7')

legend('$x\_4$','$\tilde{x}\_4$','$x\_{4\_e}$','Interpreter','latex')

ylabel('$\dot{x}$ [m/s]','Interpreter','latex')

grid minor

xlabel('Time [s]')



figure

plot(time,X\_comp(:,5)\*180/pi,time, X\_comp(:,11)\*180/pi,'--')

yline(xe(5)\*180/pi,'--r')

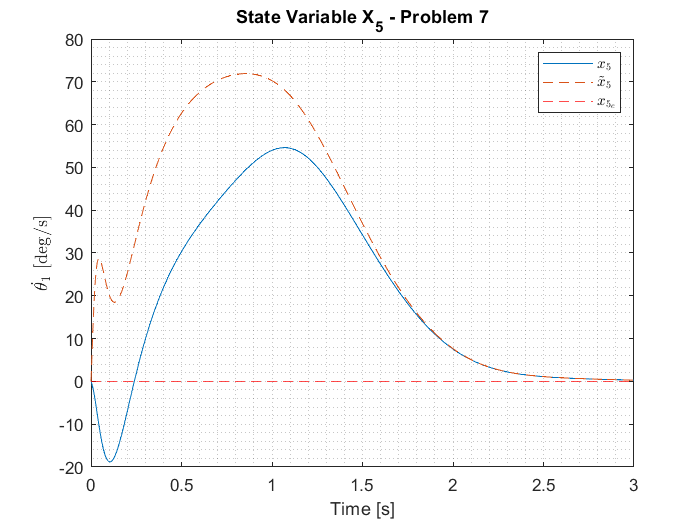
title('State Variable X\_5 - Problem 7')

ylabel('$\dot{\theta\_1}$ [deg/s]','Interpreter','latex')

xlabel('Time [s]')

grid minor

legend('$x\_5$','$\tilde{x}\_5$','$x\_{5\_e}$','Interpreter','latex')



figure

plot(time,X\_comp(:,6)\*180/pi,time, X\_comp(:,12)\*180/pi,'--')

yline(xe(6)\*180/pi,'--r')

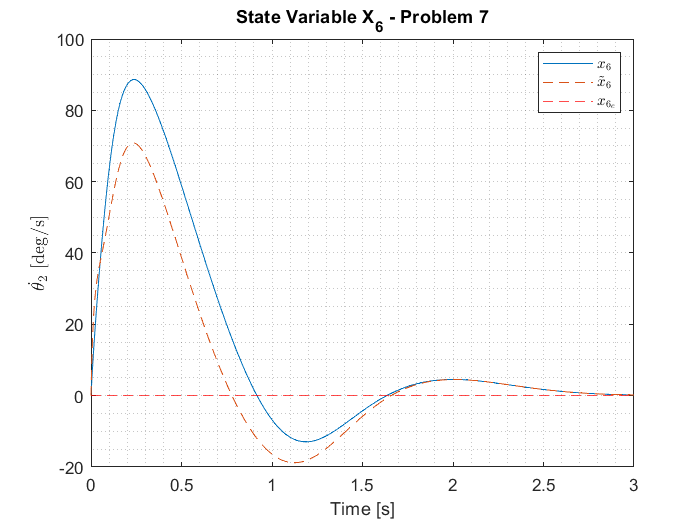
title('State Variable X\_6 - Problem 7')

ylabel('$\dot{\theta\_2}$ [deg/s]','Interpreter','latex')

grid minor

xlabel('Time [s]')

legend('$x\_6$','$\tilde{x}\_6$','$x\_{6\_e}$','Interpreter','latex')



The combined state-feedback controller-observer compensator clearly outperforms the output feedback controller. The combined controller-observer compensator successfully stabilizes the system about the desired equilibrium state. The combined controller-observer compensator can outperform the output feedback controller because the observer produces accurate estimates for the unobserved states (x4, x5, x6). Then the estimated states can be used to perform full state feedback. We can see that once the observer error dynamics die out, the system stabilizes at the desired xe.

# Functions

% Non-linear DIPC Model with single input

function xdot = DIPC\_1(t, x ,u, m1, m2, M, L1, L2, g)

% States and inputs

x1 = x(1,1); % x

x2 = x(2,1); % theta\_1

x3 = x(3,1); % theta\_2

x4 = x(4,1); % xdot

x5 = x(5,1); % theta\_1\_dot

x6 = x(6,1); % theta\_2\_dot

% Equations of Motion

x1dot = x4; % xdot

x2dot = x5; % theta\_1\_dot

x3dot = x6; % theta\_2\_dot

% x\_ddot

x4dot = (2\*m1\*u + m2\*u - m2\*u\*cos(2\*x2 - 2\*x3) - g\*m1^2\*sin(2\*x2) +...

2\*L1\*m1^2\*x5^2\*sin(x2) - g\*m1\*m2\*sin(2\*x2) +...

2\*L1\*m1\*m2\*x5^2\*sin(x2) + L2\*m1\*m2\*x6^2\*sin(x3) +...

L2\*m1\*m2\*x6^2\*sin(2\*x2 - x3))/(2\*M\*m1 + M\*m2 + m1\*m2 -...

m1^2\*cos(2\*x2) + m1^2 - m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3));

% theta\_1\_ddot

x5dot = -(m1\*u\*cos(x2) + (m2\*u\*cos(x2))/2 - (m2\*u\*cos(x2 - 2\*x3))/2 -...

g\*m1^2\*sin(x2) - M\*g\*m1\*sin(x2) - (M\*g\*m2\*sin(x2))/2 -...

g\*m1\*m2\*sin(x2) + (L1\*m1^2\*x5^2\*sin(2\*x2))/2 - (M\*g\*m2\*...

sin(x2 - 2\*x3))/2 + (L2\*m1\*m2\*x6^2\*sin(x2 + x3))/2 + ...

L2\*M\*m2\*x6^2\*sin(x2 - x3) + (L2\*m1\*m2\*x6^2\*sin(x2 - x3))/2 +...

(L1\*m1\*m2\*x5^2\*sin(2\*x2))/2 + (L1\*M\*m2\*x5^2\*sin(2\*x2 - 2\*x3))/2)...

/(L1\*(M\*m1 + (M\*m2)/2 + (m1\*m2)/2 - (m1^2\*cos(2\*x2))/2 + m1^2/2 ...

- (m1\*m2\*cos(2\*x2))/2 - (M\*m2\*cos(2\*x2 - 2\*x3))/2));

% theta\_2\_ddot

x6dot = ((m1\*u\*cos(2\*x2 - x3))/2 - (m2\*u\*cos(x3))/2 - (m1\*u\*cos(x3))/2 +...

(m2\*u\*cos(2\*x2 - x3))/2 - (M\*g\*m1\*sin(2\*x2 - x3))/2 - ...

(M\*g\*m2\*sin(2\*x2 - x3))/2 + (M\*g\*m1\*sin(x3))/2 + ...

(M\*g\*m2\*sin(x3))/2 + L1\*M\*m1\*x5^2\*sin(x2 - x3) +...

L1\*M\*m2\*x5^2\*sin(x2 - x3) + (L2\*M\*m2\*x6^2\*sin(2\*x2 - 2\*x3))/2)/...

(L2\*(M\*m1 + (M\*m2)/2 + (m1\*m2)/2 - (m1^2\*cos(2\*x2))/2 + m1^2/2 -...

(m1\*m2\*cos(2\*x2))/2 - (M\*m2\*cos(2\*x2 - 2\*x3))/2));

xdot = [x1dot;x2dot;x3dot;x4dot;x5dot;x6dot];

end

% Non-linear DIPC Model with two inputs

function xdot = DIPC\_2(t, x, u, m1, m2, M, L1, L2, g)

% Define State and Input Vectors

x1 = x(1,1); % x

x2 = x(2,1); % theta\_1

x3 = x(3,1); % theta\_2

x4 = x(4,1); % xdot

x5 = x(5,1); % theta\_1\_dot

x6 = x(6,1); % theta\_2\_dot

u1 = u(1,1);

u2 = u(2,1);

% State Dynamics

x1dot = x4; % xdot

x2dot = x5; % theta\_1\_dot

x3dot = x6; % theta\_2\_dot

% x\_ddot

x4dot = ((m2\*u2\*cos(x2 - 2\*x3))/2 - (m2\*u2\*cos(x2))/2 - m1\*u2\*cos(x2) +...

L1\*m1\*u1 + (L1\*m2\*u1)/2 - (L1\*g\*m1^2\*sin(2\*x2))/2 + L1^2\*m1^2\*x5^2\*sin(x2)...

- (L1\*m2\*u1\*cos(2\*x2 - 2\*x3))/2 - (L1\*g\*m1\*m2\*sin(2\*x2))/2 +...

L1^2\*m1\*m2\*x5^2\*sin(x2) + (L1\*L2\*m1\*m2\*x6^2\*sin(2\*x2 - x3))/2 + ...

(L1\*L2\*m1\*m2\*x6^2\*sin(x3))/2)/(L1\*(M\*m1 + (M\*m2)/2 + (m1\*m2)/2 -...

(m1^2\*cos(2\*x2))/2 + m1^2/2 - (m1\*m2\*cos(2\*x2))/2 - (M\*m2\*cos(2\*x2 - 2\*x3))/2));

% theta\_1\_ddot

x5dot = (M\*u2 + m1\*u2 + (m2\*u2)/2 - (m2\*u2\*cos(2\*x3))/2 - L1\*m1\*u1\*cos(x2) -...

(L1\*m2\*u1\*cos(x2))/2 + (L1\*m2\*u1\*cos(x2 - 2\*x3))/2 + L1\*g\*m1^2\*sin(x2) -...

(L1^2\*m1^2\*x5^2\*sin(2\*x2))/2 - (L1^2\*m1\*m2\*x5^2\*sin(2\*x2))/2 -...

(L1^2\*M\*m2\*x5^2\*sin(2\*x2 - 2\*x3))/2 + L1\*M\*g\*m1\*sin(x2) +...

(L1\*M\*g\*m2\*sin(x2))/2 + L1\*g\*m1\*m2\*sin(x2) +...

(L1\*M\*g\*m2\*sin(x2 - 2\*x3))/2 - (L1\*L2\*m1\*m2\*x6^2\*sin(x2 + x3))/2 -...

L1\*L2\*M\*m2\*x6^2\*sin(x2 - x3) - (L1\*L2\*m1\*m2\*x6^2\*sin(x2 - x3))/2)...

/(L1^2\*(M\*m1 + (M\*m2)/2 + (m1\*m2)/2 - (m1^2\*cos(2\*x2))/2 + m1^2/2 -...

(m1\*m2\*cos(2\*x2))/2 - (M\*m2\*cos(2\*x2 - 2\*x3))/2));

% theta\_2\_ddot

x6dot = (m1\*u2\*cos(x2 + x3) - m1\*u2\*cos(x2 - x3) - m2\*u2\*cos(x2 - x3) -...

2\*M\*u2\*cos(x2 - x3) + m2\*u2\*cos(x2 + x3) - L1\*m1\*u1\*cos(x3) - ...

L1\*m2\*u1\*cos(x3) + L1\*m1\*u1\*cos(2\*x2 - x3) + L1\*m2\*u1\*cos(2\*x2 - x3) -...

L1\*M\*g\*m1\*sin(2\*x2 - x3) - L1\*M\*g\*m2\*sin(2\*x2 - x3) + L1\*M\*g\*m1\*sin(x3) +...

L1\*M\*g\*m2\*sin(x3) + 2\*L1^2\*M\*m1\*x5^2\*sin(x2 - x3) +...

2\*L1^2\*M\*m2\*x5^2\*sin(x2 - x3) + L1\*L2\*M\*m2\*x6^2\*sin(2\*x2 - 2\*x3))/...

(L1\*L2\*(2\*M\*m1 + M\*m2 + m1\*m2 - m1^2\*cos(2\*x2) + m1^2 ...

- m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

xdot = [x1dot;x2dot;x3dot;x4dot;x5dot;x6dot];

end

% Non-linear DIPC model with three inputs

function xdot = DIPC\_3(t, x, u, m1, m2, M, L1, L2, g)

% Define State and Input Vectors

x1 = x(1,1); % x

x2 = x(2,1); % theta\_1

x3 = x(3,1); % theta\_2

x4 = x(4,1); % xdot

x5 = x(5,1); % theta\_1\_dot

x6 = x(6,1); % theta\_2\_dot

u1 = u(1,1);

u2 = u(2,1);

u3 = u(3,1);

% State Dynamics

x1dot = x4; % xdot

x2dot = x5; % theta\_1\_dot

x3dot = x6; % theta\_2\_dot

% x\_ddot

x4dot = (L2\*m2\*u2\*cos(x2 - 2\*x3) - L1\*m1\*u3\*cos(x3) - L2\*m2\*u2\*cos(x2)...

- L1\*m2\*u3\*cos(x3) - 2\*L2\*m1\*u2\*cos(x2) + 2\*L1\*L2\*m1\*u1 + L1\*L2\*m2\*u1...

+ L1\*m1\*u3\*cos(2\*x2 - x3) + L1\*m2\*u3\*cos(2\*x2 - x3) -...

L1\*L2\*m2\*u1\*cos(2\*x2 - 2\*x3) - L1\*L2\*g\*m1^2\*sin(2\*x2) +...

2\*L1^2\*L2\*m1^2\*x5^2\*sin(x2) - L1\*L2\*g\*m1\*m2\*sin(2\*x2) +...

L1\*L2^2\*m1\*m2\*x6^2\*sin(2\*x2 - x3) + 2\*L1^2\*L2\*m1\*m2\*x5^2\*sin(x2) +...

L1\*L2^2\*m1\*m2\*x6^2\*sin(x3))/(L1\*L2\*(2\*M\*m1 + M\*m2 + m1\*m2 -...

m1^2\*cos(2\*x2) + m1^2 - m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

% theta\_1\_ddot

x5dot = -(L2\*m2\*u2\*cos(2\*x3) - 2\*L2\*m1\*u2 - L2\*m2\*u2 - 2\*L2\*M\*u2 -...

2\*L1\*L2\*g\*m1^2\*sin(x2) + 2\*L1\*M\*u3\*cos(x2)\*cos(x3) +...

2\*L1\*M\*u3\*sin(x2)\*sin(x3) + L1^2\*L2\*m1^2\*x5^2\*sin(2\*x2) +...

2\*L1\*m1\*u3\*sin(x2)\*sin(x3) + 2\*L1\*m2\*u3\*sin(x2)\*sin(x3) +...

2\*L1\*L2\*m1\*u1\*cos(x2) + L1\*L2\*m2\*u1\*cos(x2) - 2\*L1\*L2\*M\*g\*m1\*sin(x2)...

- L1\*L2\*M\*g\*m2\*sin(x2) - L1\*L2\*m2\*u1\*sin(2\*x3)\*sin(x2) -...

2\*L1\*L2\*g\*m1\*m2\*sin(x2) + L1^2\*L2\*m1\*m2\*x5^2\*sin(2\*x2) -...

L1\*L2\*m2\*u1\*cos(2\*x3)\*cos(x2) - L1\*L2\*M\*g\*m2\*cos(2\*x3)\*sin(x2) +...

L1\*L2\*M\*g\*m2\*sin(2\*x3)\*cos(x2) - L1^2\*L2\*M\*m2\*x5^2\*cos(2\*x2)\*sin(2\*x3) +...

L1^2\*L2\*M\*m2\*x5^2\*cos(2\*x3)\*sin(2\*x2) - 2\*L1\*L2^2\*M\*m2\*x6^2\*cos(x2)\*sin(x3) +...

2\*L1\*L2^2\*M\*m2\*x6^2\*cos(x3)\*sin(x2) + 2\*L1\*L2^2\*m1\*m2\*x6^2\*cos(x3)\*sin(x2))/...

(L1^2\*L2\*(2\*M\*m1 + M\*m2 + m1\*m2 - m1^2\*cos(2\*x2) + m1^2 -...

m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

% theta\_2\_ddot

x6dot = (L1\*m1^2\*u3 + L1\*m2^2\*u3 - L1\*m1^2\*u3\*cos(2\*x2) - L1\*m2^2\*u3\*cos(2\*x2) +...

L2\*m2^2\*u2\*cos(x2 + x3) + 2\*L1\*M\*m1\*u3 + 2\*L1\*M\*m2\*u3 + 2\*L1\*m1\*m2\*u3 -...

L2\*m2^2\*u2\*cos(x2 - x3) - L2\*m1\*m2\*u2\*cos(x2 - x3) - L1\*L2\*m2^2\*u1\*cos(x3) -...

2\*L1\*m1\*m2\*u3\*cos(2\*x2) + L1\*L2\*m2^2\*u1\*cos(2\*x2 - x3) +...

L2\*m1\*m2\*u2\*cos(x2 + x3) - 2\*L2\*M\*m2\*u2\*cos(x2 - x3) - L1\*L2\*m1\*m2\*u1\*cos(x3) +...

L1\*L2\*M\*g\*m2^2\*sin(x3) + 2\*L1^2\*L2\*M\*m2^2\*x5^2\*sin(x2 - x3) +...

L1\*L2\*m1\*m2\*u1\*cos(2\*x2 - x3) + L1\*L2^2\*M\*m2^2\*x6^2\*sin(2\*x2 - 2\*x3) -...

L1\*L2\*M\*g\*m2^2\*sin(2\*x2 - x3) - L1\*L2\*M\*g\*m1\*m2\*sin(2\*x2 - x3) +...

L1\*L2\*M\*g\*m1\*m2\*sin(x3) + 2\*L1^2\*L2\*M\*m1\*m2\*x5^2\*sin(x2 - x3))/...

(L1\*L2^2\*m2\*(2\*M\*m1 + M\*m2 + m1\*m2 - m1^2\*cos(2\*x2) + m1^2 -...

m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

xdot = [x1dot;x2dot;x3dot;x4dot;x5dot;x6dot];

end

% DIPC Lagrangian

function L = DIPC\_Lagrangian(t,x, x\_dot, theta1, theta\_dot\_1, theta2, theta\_dot\_2, M, m1,m2, L1, L2, g)

% Lagrangian for DIPC from HW1

L = (m2\*(x\_dot(t) + L1\*cos(theta1(t))\*theta\_dot\_1(t) + L2\*cos(theta2(t))\*...

theta\_dot\_2(t))^2)/2 + (m1\*(x\_dot(t) + L1\*cos(theta1(t))\*...

theta\_dot\_1(t))^2)/2 + (m2\*(L1\*sin(theta1(t))\*theta\_dot\_1(t) +...

L2\*sin(theta2(t))\*theta\_dot\_2(t))^2)/2 + (M\*x\_dot(t)^2)/2 + ...

(L1^2\*m1\*sin(theta1(t))^2\*theta\_dot\_1(t)^2)/2 - ...

L1\*g\*m1\*cos(theta1(t)) - L1\*g\*m2\*cos(theta1(t)) - L2\*g\*m2\*cos(theta2(t));

end

% Controlled Non-linear model with 3 inputs

function xdot = ControlledDIPC\_3(t, x, xe, ue, K, m1, m2, M, L1, L2, g)

% Define State and Input Vectors

x1 = x(1,1); % x

x2 = x(2,1); % theta\_1

x3 = x(3,1); % theta\_2

x4 = x(4,1); % xdot

x5 = x(5,1); % theta\_1\_dot

x6 = x(6,1); % theta\_2\_dot

% Control Law

u = -K\*x(:,1) + K\*xe + ue;

u1 = u(1);

u2 = u(2);

u3 = u(3);

% State Dynamics

x1dot = x4; % xdot

x2dot = x5; % theta\_1\_dot

x3dot = x6; % theta\_2\_dot

% x\_ddot

x4dot = (L2\*m2\*u2\*cos(x2 - 2\*x3) - L1\*m1\*u3\*cos(x3) - L2\*m2\*u2\*cos(x2)...

- L1\*m2\*u3\*cos(x3) - 2\*L2\*m1\*u2\*cos(x2) + 2\*L1\*L2\*m1\*u1 + L1\*L2\*m2\*u1...

+ L1\*m1\*u3\*cos(2\*x2 - x3) + L1\*m2\*u3\*cos(2\*x2 - x3) -...

L1\*L2\*m2\*u1\*cos(2\*x2 - 2\*x3) - L1\*L2\*g\*m1^2\*sin(2\*x2) +...

2\*L1^2\*L2\*m1^2\*x5^2\*sin(x2) - L1\*L2\*g\*m1\*m2\*sin(2\*x2) +...

L1\*L2^2\*m1\*m2\*x6^2\*sin(2\*x2 - x3) + 2\*L1^2\*L2\*m1\*m2\*x5^2\*sin(x2) +...

L1\*L2^2\*m1\*m2\*x6^2\*sin(x3))/(L1\*L2\*(2\*M\*m1 + M\*m2 + m1\*m2 -...

m1^2\*cos(2\*x2) + m1^2 - m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

% theta\_1\_ddot

x5dot = -(L2\*m2\*u2\*cos(2\*x3) - 2\*L2\*m1\*u2 - L2\*m2\*u2 - 2\*L2\*M\*u2 -...

2\*L1\*L2\*g\*m1^2\*sin(x2) + 2\*L1\*M\*u3\*cos(x2)\*cos(x3) +...

2\*L1\*M\*u3\*sin(x2)\*sin(x3) + L1^2\*L2\*m1^2\*x5^2\*sin(2\*x2) +...

2\*L1\*m1\*u3\*sin(x2)\*sin(x3) + 2\*L1\*m2\*u3\*sin(x2)\*sin(x3) +...

2\*L1\*L2\*m1\*u1\*cos(x2) + L1\*L2\*m2\*u1\*cos(x2) - 2\*L1\*L2\*M\*g\*m1\*sin(x2)...

- L1\*L2\*M\*g\*m2\*sin(x2) - L1\*L2\*m2\*u1\*sin(2\*x3)\*sin(x2) -...

2\*L1\*L2\*g\*m1\*m2\*sin(x2) + L1^2\*L2\*m1\*m2\*x5^2\*sin(2\*x2) -...

L1\*L2\*m2\*u1\*cos(2\*x3)\*cos(x2) - L1\*L2\*M\*g\*m2\*cos(2\*x3)\*sin(x2) +...

L1\*L2\*M\*g\*m2\*sin(2\*x3)\*cos(x2) - L1^2\*L2\*M\*m2\*x5^2\*cos(2\*x2)\*sin(2\*x3) +...

L1^2\*L2\*M\*m2\*x5^2\*cos(2\*x3)\*sin(2\*x2) - 2\*L1\*L2^2\*M\*m2\*x6^2\*cos(x2)\*sin(x3) +...

2\*L1\*L2^2\*M\*m2\*x6^2\*cos(x3)\*sin(x2) + 2\*L1\*L2^2\*m1\*m2\*x6^2\*cos(x3)\*sin(x2))/...

(L1^2\*L2\*(2\*M\*m1 + M\*m2 + m1\*m2 - m1^2\*cos(2\*x2) + m1^2 -...

m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

% theta\_2\_ddot

x6dot = (L1\*m1^2\*u3 + L1\*m2^2\*u3 - L1\*m1^2\*u3\*cos(2\*x2) - L1\*m2^2\*u3\*cos(2\*x2) +...

L2\*m2^2\*u2\*cos(x2 + x3) + 2\*L1\*M\*m1\*u3 + 2\*L1\*M\*m2\*u3 + 2\*L1\*m1\*m2\*u3 -...

L2\*m2^2\*u2\*cos(x2 - x3) - L2\*m1\*m2\*u2\*cos(x2 - x3) - L1\*L2\*m2^2\*u1\*cos(x3) -...

2\*L1\*m1\*m2\*u3\*cos(2\*x2) + L1\*L2\*m2^2\*u1\*cos(2\*x2 - x3) +...

L2\*m1\*m2\*u2\*cos(x2 + x3) - 2\*L2\*M\*m2\*u2\*cos(x2 - x3) - L1\*L2\*m1\*m2\*u1\*cos(x3) +...

L1\*L2\*M\*g\*m2^2\*sin(x3) + 2\*L1^2\*L2\*M\*m2^2\*x5^2\*sin(x2 - x3) +...

L1\*L2\*m1\*m2\*u1\*cos(2\*x2 - x3) + L1\*L2^2\*M\*m2^2\*x6^2\*sin(2\*x2 - 2\*x3) -...

L1\*L2\*M\*g\*m2^2\*sin(2\*x2 - x3) - L1\*L2\*M\*g\*m1\*m2\*sin(2\*x2 - x3) +...

L1\*L2\*M\*g\*m1\*m2\*sin(x3) + 2\*L1^2\*L2\*M\*m1\*m2\*x5^2\*sin(x2 - x3))/...

(L1\*L2^2\*m2\*(2\*M\*m1 + M\*m2 + m1\*m2 - m1^2\*cos(2\*x2) + m1^2 -...

m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

xdot = [x1dot;x2dot;x3dot;x4dot;x5dot;x6dot];

end

% Output Controlled Non-linear model with 3 inputs

function xdot = OutputControlledDIPC\_3(t, x, ye, ue, K, m1, m2, M, L1, L2, g)

% Define State and Input Vectors

x1 = x(1,1); % x

x2 = x(2,1); % theta\_1

x3 = x(3,1); % theta\_2

x4 = x(4,1); % xdot

x5 = x(5,1); % theta\_1\_dot

x6 = x(6,1); % theta\_2\_dot

% output

y = [x1;x2;x3];

% Control Law

u = -K\*y + K\*ye + ue;

u1 = u(1);

u2 = u(2);

u3 = u(3);

% State Dynamics

x1dot = x4; % xdot

x2dot = x5; % theta\_1\_dot

x3dot = x6; % theta\_2\_dot

% x\_ddot

x4dot = (L2\*m2\*u2\*cos(x2 - 2\*x3) - L1\*m1\*u3\*cos(x3) - L2\*m2\*u2\*cos(x2)...

- L1\*m2\*u3\*cos(x3) - 2\*L2\*m1\*u2\*cos(x2) + 2\*L1\*L2\*m1\*u1 + L1\*L2\*m2\*u1...

+ L1\*m1\*u3\*cos(2\*x2 - x3) + L1\*m2\*u3\*cos(2\*x2 - x3) -...

L1\*L2\*m2\*u1\*cos(2\*x2 - 2\*x3) - L1\*L2\*g\*m1^2\*sin(2\*x2) +...

2\*L1^2\*L2\*m1^2\*x5^2\*sin(x2) - L1\*L2\*g\*m1\*m2\*sin(2\*x2) +...

L1\*L2^2\*m1\*m2\*x6^2\*sin(2\*x2 - x3) + 2\*L1^2\*L2\*m1\*m2\*x5^2\*sin(x2) +...

L1\*L2^2\*m1\*m2\*x6^2\*sin(x3))/(L1\*L2\*(2\*M\*m1 + M\*m2 + m1\*m2 -...

m1^2\*cos(2\*x2) + m1^2 - m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

% theta\_1\_ddot

x5dot = -(L2\*m2\*u2\*cos(2\*x3) - 2\*L2\*m1\*u2 - L2\*m2\*u2 - 2\*L2\*M\*u2 -...

2\*L1\*L2\*g\*m1^2\*sin(x2) + 2\*L1\*M\*u3\*cos(x2)\*cos(x3) +...

2\*L1\*M\*u3\*sin(x2)\*sin(x3) + L1^2\*L2\*m1^2\*x5^2\*sin(2\*x2) +...

2\*L1\*m1\*u3\*sin(x2)\*sin(x3) + 2\*L1\*m2\*u3\*sin(x2)\*sin(x3) +...

2\*L1\*L2\*m1\*u1\*cos(x2) + L1\*L2\*m2\*u1\*cos(x2) - 2\*L1\*L2\*M\*g\*m1\*sin(x2)...

- L1\*L2\*M\*g\*m2\*sin(x2) - L1\*L2\*m2\*u1\*sin(2\*x3)\*sin(x2) -...

2\*L1\*L2\*g\*m1\*m2\*sin(x2) + L1^2\*L2\*m1\*m2\*x5^2\*sin(2\*x2) -...

L1\*L2\*m2\*u1\*cos(2\*x3)\*cos(x2) - L1\*L2\*M\*g\*m2\*cos(2\*x3)\*sin(x2) +...

L1\*L2\*M\*g\*m2\*sin(2\*x3)\*cos(x2) - L1^2\*L2\*M\*m2\*x5^2\*cos(2\*x2)\*sin(2\*x3) +...

L1^2\*L2\*M\*m2\*x5^2\*cos(2\*x3)\*sin(2\*x2) - 2\*L1\*L2^2\*M\*m2\*x6^2\*cos(x2)\*sin(x3) +...

2\*L1\*L2^2\*M\*m2\*x6^2\*cos(x3)\*sin(x2) + 2\*L1\*L2^2\*m1\*m2\*x6^2\*cos(x3)\*sin(x2))/...

(L1^2\*L2\*(2\*M\*m1 + M\*m2 + m1\*m2 - m1^2\*cos(2\*x2) + m1^2 -...

m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

% theta\_2\_ddot

x6dot = (L1\*m1^2\*u3 + L1\*m2^2\*u3 - L1\*m1^2\*u3\*cos(2\*x2) - L1\*m2^2\*u3\*cos(2\*x2) +...

L2\*m2^2\*u2\*cos(x2 + x3) + 2\*L1\*M\*m1\*u3 + 2\*L1\*M\*m2\*u3 + 2\*L1\*m1\*m2\*u3 -...

L2\*m2^2\*u2\*cos(x2 - x3) - L2\*m1\*m2\*u2\*cos(x2 - x3) - L1\*L2\*m2^2\*u1\*cos(x3) -...

2\*L1\*m1\*m2\*u3\*cos(2\*x2) + L1\*L2\*m2^2\*u1\*cos(2\*x2 - x3) +...

L2\*m1\*m2\*u2\*cos(x2 + x3) - 2\*L2\*M\*m2\*u2\*cos(x2 - x3) - L1\*L2\*m1\*m2\*u1\*cos(x3) +...

L1\*L2\*M\*g\*m2^2\*sin(x3) + 2\*L1^2\*L2\*M\*m2^2\*x5^2\*sin(x2 - x3) +...

L1\*L2\*m1\*m2\*u1\*cos(2\*x2 - x3) + L1\*L2^2\*M\*m2^2\*x6^2\*sin(2\*x2 - 2\*x3) -...

L1\*L2\*M\*g\*m2^2\*sin(2\*x2 - x3) - L1\*L2\*M\*g\*m1\*m2\*sin(2\*x2 - x3) +...

L1\*L2\*M\*g\*m1\*m2\*sin(x3) + 2\*L1^2\*L2\*M\*m1\*m2\*x5^2\*sin(x2 - x3))/...

(L1\*L2^2\*m2\*(2\*M\*m1 + M\*m2 + m1\*m2 - m1^2\*cos(2\*x2) + m1^2 -...

m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

xdot = [x1dot;x2dot;x3dot;x4dot;x5dot;x6dot];

end

function xdot = CombinedCompensatorDIPC(t, x, xe, ye, ue, K, L, A, B, C, D, m1, m2, M, L1, L2, g)

% Define State, state estimates, and input Vectors

x1 = x(1,1); % x

x2 = x(2,1); % theta\_1

x3 = x(3,1); % theta\_2

x4 = x(4,1); % xdot

x5 = x(5,1); % theta\_1\_dot

x6 = x(6,1); % theta\_2\_dot

x1\_tilde = x(7,1); % x1\_tilde - estimate of x

x2\_tilde = x(8,1); % x2\_tilde - estimate of theta\_1

x3\_tilde = x(9,1); % x3\_tilde - estimate of theta\_2

x4\_tilde = x(10,1); % x4\_tilde - estimate of xdot

x5\_tilde = x(11,1); % x5\_tilde - estimate of theta\_1\_dot

x6\_tilde = x(12,1); % x6\_tilde - estimate of theta\_2\_dot

x\_tilde = [x1\_tilde;x2\_tilde;x3\_tilde;x4\_tilde;x5\_tilde;x6\_tilde];

% Estimated state pertubation: z = delta\_xtilde

z = x\_tilde - xe;

% Control Law: u = -K\*z + ue

del\_u = -K\*z;

u = del\_u + ue;

u1 = u(1);

u2 = u(2);

u3 = u(3);

% State Dynamics

x1dot = x4; % xdot

x2dot = x5; % theta\_1\_dot

x3dot = x6; % theta\_2\_dot

% x\_ddot

x4dot = (L2\*m2\*u2\*cos(x2 - 2\*x3) - L1\*m1\*u3\*cos(x3) - L2\*m2\*u2\*cos(x2)...

- L1\*m2\*u3\*cos(x3) - 2\*L2\*m1\*u2\*cos(x2) + 2\*L1\*L2\*m1\*u1 + L1\*L2\*m2\*u1...

+ L1\*m1\*u3\*cos(2\*x2 - x3) + L1\*m2\*u3\*cos(2\*x2 - x3) -...

L1\*L2\*m2\*u1\*cos(2\*x2 - 2\*x3) - L1\*L2\*g\*m1^2\*sin(2\*x2) +...

2\*L1^2\*L2\*m1^2\*x5^2\*sin(x2) - L1\*L2\*g\*m1\*m2\*sin(2\*x2) +...

L1\*L2^2\*m1\*m2\*x6^2\*sin(2\*x2 - x3) + 2\*L1^2\*L2\*m1\*m2\*x5^2\*sin(x2) +...

L1\*L2^2\*m1\*m2\*x6^2\*sin(x3))/(L1\*L2\*(2\*M\*m1 + M\*m2 + m1\*m2 -...

m1^2\*cos(2\*x2) + m1^2 - m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

% theta\_1\_ddot

x5dot = -(L2\*m2\*u2\*cos(2\*x3) - 2\*L2\*m1\*u2 - L2\*m2\*u2 - 2\*L2\*M\*u2 -...

2\*L1\*L2\*g\*m1^2\*sin(x2) + 2\*L1\*M\*u3\*cos(x2)\*cos(x3) +...

2\*L1\*M\*u3\*sin(x2)\*sin(x3) + L1^2\*L2\*m1^2\*x5^2\*sin(2\*x2) +...

2\*L1\*m1\*u3\*sin(x2)\*sin(x3) + 2\*L1\*m2\*u3\*sin(x2)\*sin(x3) +...

2\*L1\*L2\*m1\*u1\*cos(x2) + L1\*L2\*m2\*u1\*cos(x2) - 2\*L1\*L2\*M\*g\*m1\*sin(x2)...

- L1\*L2\*M\*g\*m2\*sin(x2) - L1\*L2\*m2\*u1\*sin(2\*x3)\*sin(x2) -...

2\*L1\*L2\*g\*m1\*m2\*sin(x2) + L1^2\*L2\*m1\*m2\*x5^2\*sin(2\*x2) -...

L1\*L2\*m2\*u1\*cos(2\*x3)\*cos(x2) - L1\*L2\*M\*g\*m2\*cos(2\*x3)\*sin(x2) +...

L1\*L2\*M\*g\*m2\*sin(2\*x3)\*cos(x2) - L1^2\*L2\*M\*m2\*x5^2\*cos(2\*x2)\*sin(2\*x3) +...

L1^2\*L2\*M\*m2\*x5^2\*cos(2\*x3)\*sin(2\*x2) - 2\*L1\*L2^2\*M\*m2\*x6^2\*cos(x2)\*sin(x3) +...

2\*L1\*L2^2\*M\*m2\*x6^2\*cos(x3)\*sin(x2) + 2\*L1\*L2^2\*m1\*m2\*x6^2\*cos(x3)\*sin(x2))/...

(L1^2\*L2\*(2\*M\*m1 + M\*m2 + m1\*m2 - m1^2\*cos(2\*x2) + m1^2 -...

m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

% theta\_2\_ddot

x6dot = (L1\*m1^2\*u3 + L1\*m2^2\*u3 - L1\*m1^2\*u3\*cos(2\*x2) - L1\*m2^2\*u3\*cos(2\*x2) +...

L2\*m2^2\*u2\*cos(x2 + x3) + 2\*L1\*M\*m1\*u3 + 2\*L1\*M\*m2\*u3 + 2\*L1\*m1\*m2\*u3 -...

L2\*m2^2\*u2\*cos(x2 - x3) - L2\*m1\*m2\*u2\*cos(x2 - x3) - L1\*L2\*m2^2\*u1\*cos(x3) -...

2\*L1\*m1\*m2\*u3\*cos(2\*x2) + L1\*L2\*m2^2\*u1\*cos(2\*x2 - x3) +...

L2\*m1\*m2\*u2\*cos(x2 + x3) - 2\*L2\*M\*m2\*u2\*cos(x2 - x3) - L1\*L2\*m1\*m2\*u1\*cos(x3) +...

L1\*L2\*M\*g\*m2^2\*sin(x3) + 2\*L1^2\*L2\*M\*m2^2\*x5^2\*sin(x2 - x3) +...

L1\*L2\*m1\*m2\*u1\*cos(2\*x2 - x3) + L1\*L2^2\*M\*m2^2\*x6^2\*sin(2\*x2 - 2\*x3) -...

L1\*L2\*M\*g\*m2^2\*sin(2\*x2 - x3) - L1\*L2\*M\*g\*m1\*m2\*sin(2\*x2 - x3) +...

L1\*L2\*M\*g\*m1\*m2\*sin(x3) + 2\*L1^2\*L2\*M\*m1\*m2\*x5^2\*sin(x2 - x3))/...

(L1\*L2^2\*m2\*(2\*M\*m1 + M\*m2 + m1\*m2 - m1^2\*cos(2\*x2) + m1^2 -...

m1\*m2\*cos(2\*x2) - M\*m2\*cos(2\*x2 - 2\*x3)));

xdot(1:6,1) = [x1dot;x2dot;x3dot;x4dot;x5dot;x6dot];

% Output vector - x, theta\_1, theta\_2

y = [x1;x2;x3];

% Output pertubation vector

del\_y = y - ye;

% Observer Dynamics

del\_y\_tilde = C\*z + D\*del\_u;

zdot = A\*z + B\*del\_u + L\*(del\_y - del\_y\_tilde);

xdot(7:12,1)= zdot;

end