Contents

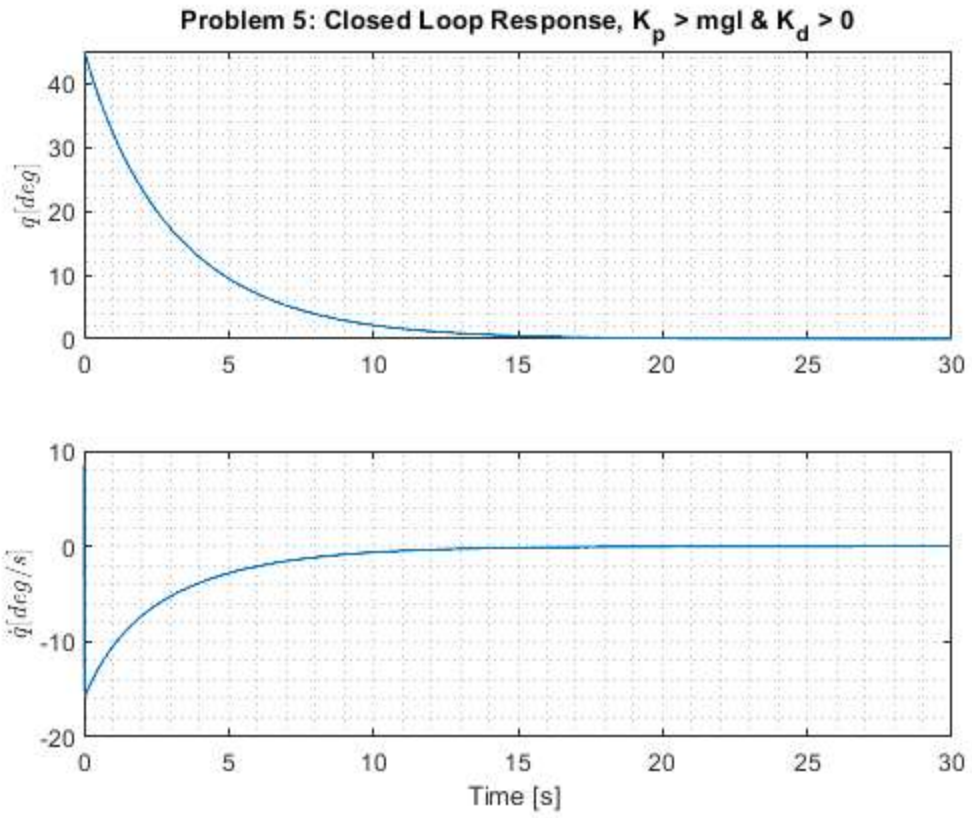
Gabriel Colangelo HW 8

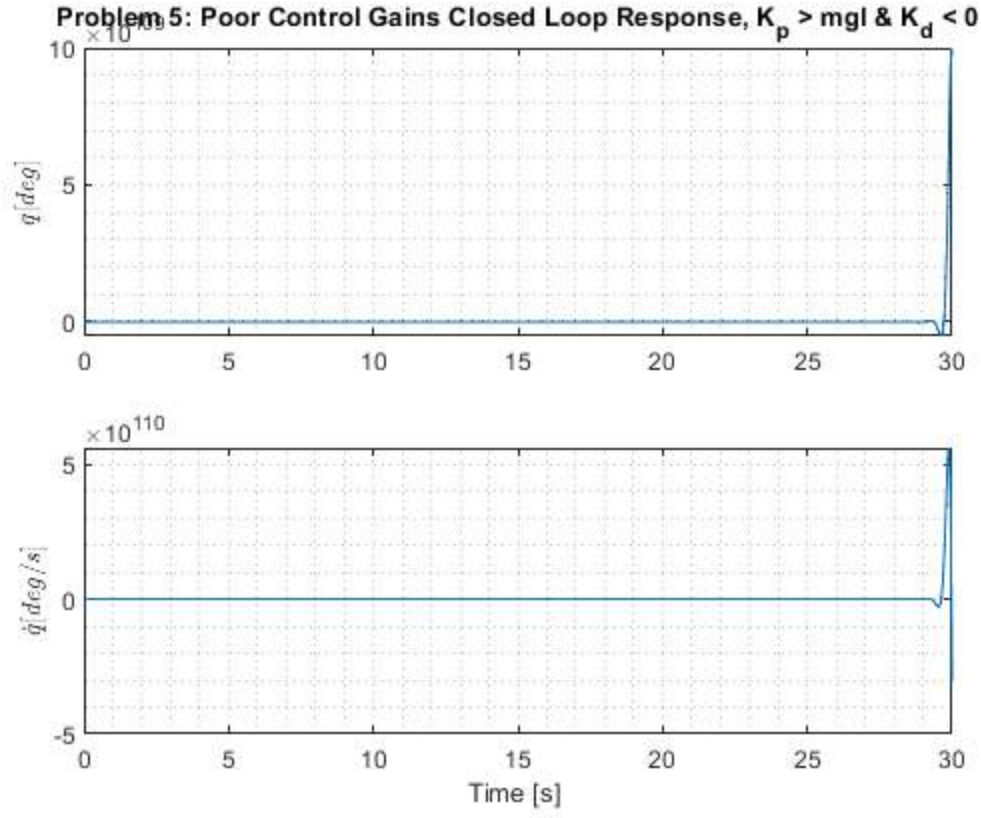
```
clear
close all
clc
```

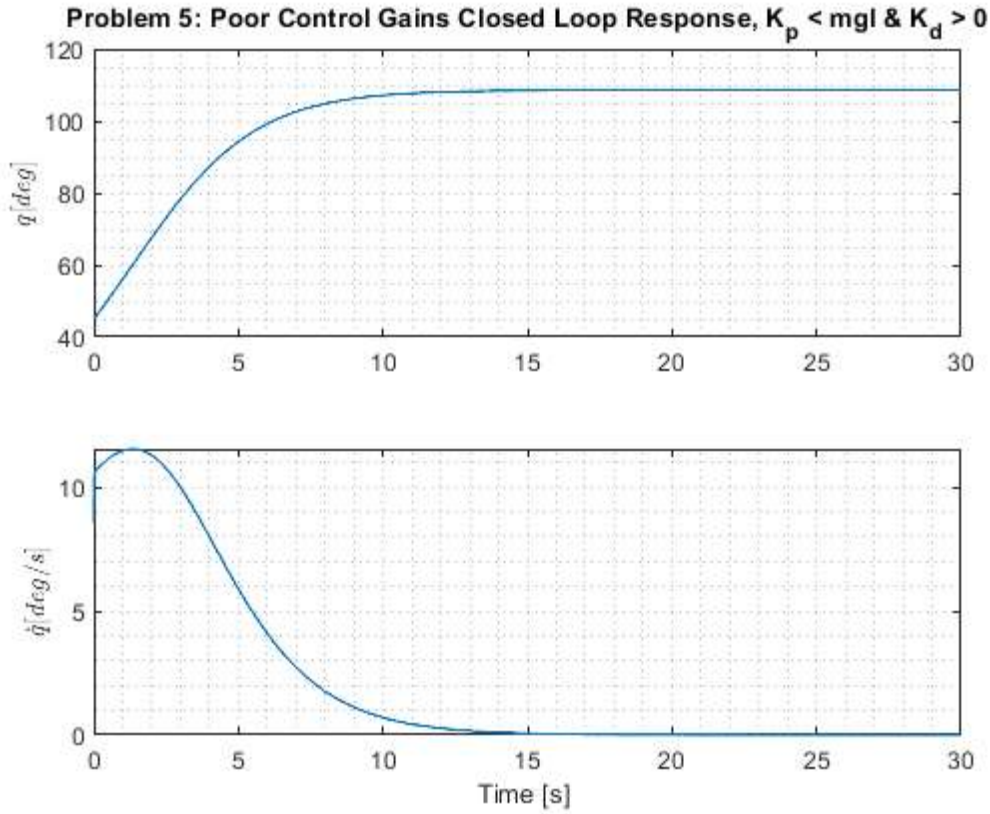
Problem 5

```
% Numerical parameters
m           = 0.2;          % [kg] Pendulum mass
l           = 0.3;          % [m] Distance to center of mass
I           = .006;         % [kg-m^2]
g           = 9.81;         % [m/s^2]
a           = m*g*l/I;
b           = 1/I;

% sim time
time        = (0:.005:30)';

% ODE45 solver options
opts        = odeset('AbsTol',1e-8,'RelTol',1e-8);

% Initial Conditions
x0          = [45*pi/180; 0.15];

% Choose control gains
Kp          = 1.5*m*g*l;    % Kp > m*g*l
Kd          = 1;            % Kd > 0
K           = [Kp Kd];

% Choose bad control gains
Kd_bad      = [Kp -0.1];
Kp_bad      = [0.5*m*g*l 1];

% ODE45 Function calls
[T, X]      = ode45(@(t,x) InvertedPendulumPD(t, x, a, b, K),...
                time, x0, opts);
[~, X_kd]   = ode45(@(t,x) InvertedPendulumPD(t, x, a, b, Kd_bad),...
                time, x0, opts);
[~, X_kp]   = ode45(@(t,x) InvertedPendulumPD(t, x, a, b, Kp_bad),...
                time, x0, opts);

figure
subplot(211)
plot(T,X(:,1)*180/pi)
grid minor
ylabel('$q [deg]$','Interpreter','latex')
title('Problem 5: Closed Loop Response, K_p > mgl & K_d > 0')
subplot(212)
```

```
plot(T,X(:,2)*180/pi)
grid minor
ylabel('$\dot{q} [deg/s]$','Interpreter','latex')
xlabel('Time [s] ')

figure
subplot(211)
plot(T,X_kd(:,1)*180/pi)
grid minor
ylabel('$q [deg]$','Interpreter','latex')
title('Problem 5: Poor Control Gains Closed Loop Response, K_p > mgl & K_d < 0')
subplot(212)
plot(T,X_kd(:,2)*180/pi)
grid minor
ylabel('$\dot{q} [deg/s]$','Interpreter','latex')
xlabel('Time [s] ')

figure
subplot(211)
plot(T,X_kp(:,1)*180/pi)
grid minor
ylabel('$q [deg]$','Interpreter','latex')
title('Problem 5: Poor Control Gains Closed Loop Response, K_p < mgl & K_d > 0')
subplot(212)
plot(T,X_kp(:,2)*180/pi)
grid minor
ylabel('$\dot{q} [deg/s]$','Interpreter','latex')
xlabel('Time [s] ')
```


Problem 5: Closed Loop Response, $K_p > mgl$ & $K_d > 0$
$q\,[deg]$
$\dot{q}\,[deg/s]$
Time [s]

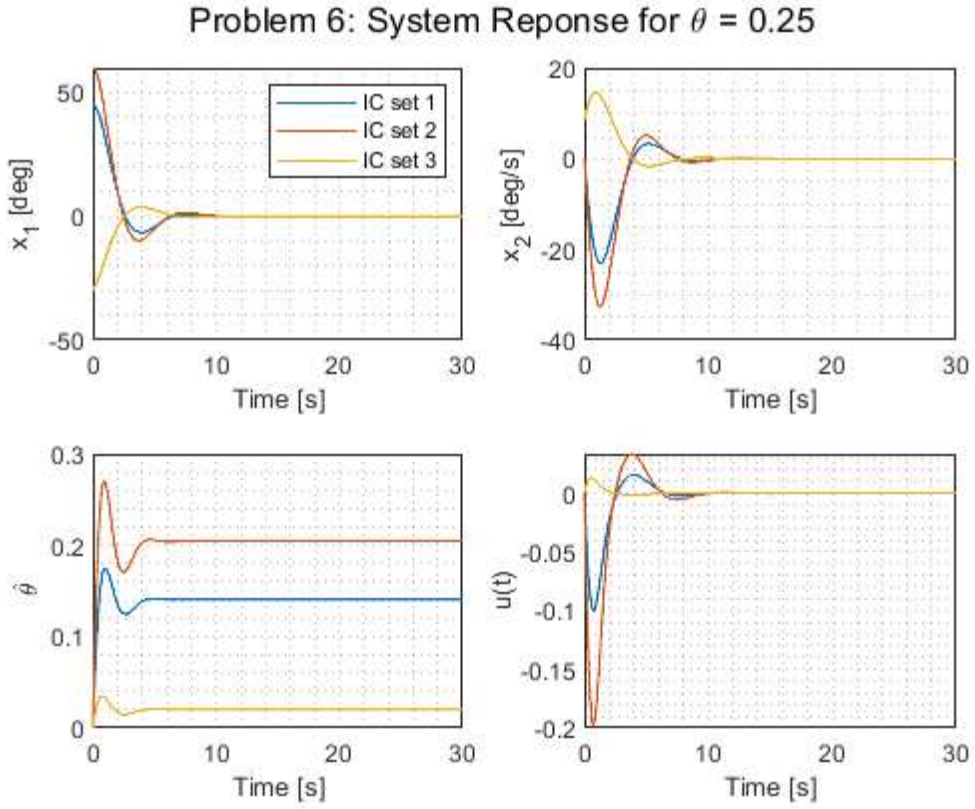

Problem 5: Poor Control Gains Closed Loop Response, $K_p > mgl$ & $K_d < 0$
$q\,[deg]$
$\dot{q}\,[deg/s]$
Time [s]

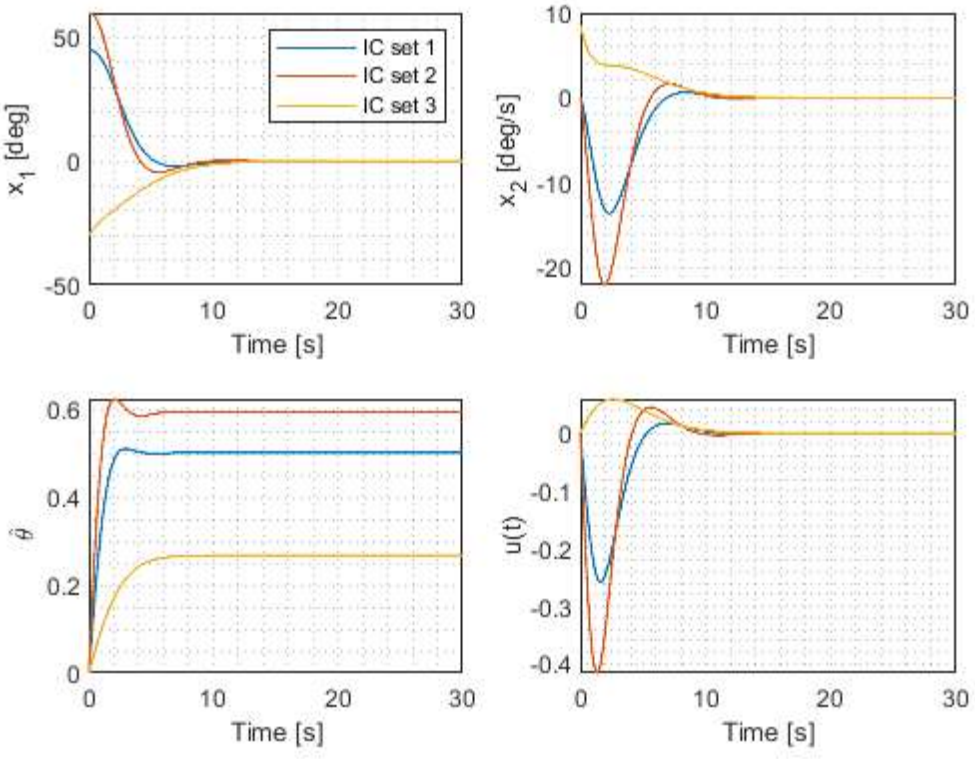


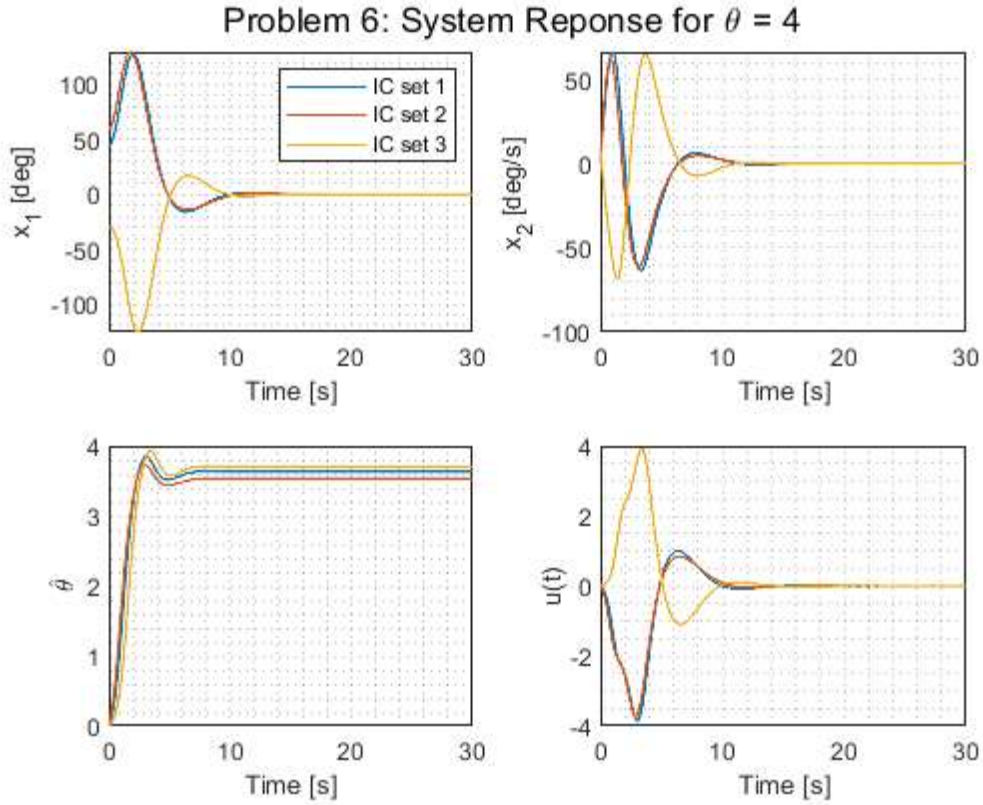
Problem 6

```
% Unknown parameter vector
theta       = [0.25, 1, 4];

% Constant tuning parameter (0 < lambda < 1)
lambda      = 0.7;

% Matrix of Initial Conditions: [x1;x2;theta_hat]
IC          = [45*pi/180 60*pi/180 -30*pi/180; 0 0 0.15; 0 0 0];

% Loop through various unknown parameters
for j = 1:length(theta)

    % Initialize vectors
    x1          = zeros(length(time),length(IC));
    x2          = x1;
    theta_hat   = x1;
    u           = x1;

    for i = 1:length(IC)
        % ODE45 Function call
        [T, Y]          = ode45(@(t,x) AdaptiveController(t,x, theta(j), lambda),...
                            time, IC(:,i), opts);
        % Extract and Store States
        x1(:,i)         = Y(:,1)*180/pi;
        x2(:,i)         = Y(:,2)*180/pi;
        theta_hat(:,i)  = Y(:,3);

        % Calculate control input
        u(:,i)          = -theta_hat(:,i).*sind(x1(:,i));
    end
```

```
    figure
    subplot(221)
    plot(T,x1)
    grid minor
    legend('IC set 1','IC set 2', 'IC set 3')
    xlabel('Time [s]')
    ylabel('x_1 [deg]')
    subplot(222)
    plot(T,x2)
    xlabel('Time [s]')
    grid minor
    ylabel('x_2 [deg/s]')
    subplot(223)
    plot(T,theta_hat)
    xlabel('Time [s]')
    grid minor
    ylabel('$$\hat{\theta}$$','Interpreter','latex')
    subplot(224)
    plot(T,u)
    grid minor
    ylabel('u(t)')
    xlabel('Time [s]')
    sgtitle(['Problem 6: System Reponse for \theta = ' num2str(theta(j))])
end
```


Problem 6: System Reponse for θ = 0.25
IC set 1
IC set 2
IC set 3
x_1 [deg]
x_2 [deg/s]
$\hat{\theta}$
u(t)
Time [s]


Problem 6: System Reponse for θ = 1
IC set 1
IC set 2
IC set 3
x_1 [deg]
x_2 [deg/s]
$\hat{\theta}$
u(t)
Time [s]



Functions

```
function xdot = InvertedPendulumPD(t, x, a, b, K)
    % K = [Kp Kd]
    u       = -K*x;   % PD Controller

    % Plant: x2dot - asin(x1) = bu
    xdot    = [x(2,1); a*sin(x(1,1)) + b*u];
end

function xdot = AdaptiveController(t, x, theta, lambda)
    % States: [x1, x2, theta_hat]
    x1              = x(1,1);
    x2              = x(2,1);
    theta_hat       = x(3,1);

    % Adaptive Controller
    u               = -theta_hat*sin(x1);
    theta_hat_dot   = x2*sin(x1) + lambda*x1*sin(x1);

    % Plant
    x1dot           = x2;
    x2dot           = -x1 -x2 + theta*sin(x1) + u;

    % State dynamics
    xdot            = [x1dot;x2dot;theta_hat_dot];
end
```

Published with MATLAB® R2021b