Table of Contents

Constants

```
clear; clc; format compact; close all;
k_t = 0.0077; % Motor Torque Constant, N*m/A
R_a = 2.6; %Armature resistance, Ohms
N = 70; %Gear ratio
I_1 = .83e-3; % Inertia, link 1 kg*m^2
J_m = .65e-6; % Motor Inertia, 2 kg*m^2
b = 3.1e-6; % mechanical damping, N*m*s/rad
gravity_torque = 0.067; % gravitational torque constant (m1*g*r01) N*m/rad
```

Problem 1 Work

```
N_maximize_torque = sqrt(I_1/J_m);

fprintf('Maximize Torque Gear Ratio: %i\n', round(N_maximize_torque));
```

Maximize Torque Gear Ratio: 36

Problem 2.1

```
disp("")
disp("Problem 2a")
disp("")
% syms I_1, J_m, N, b, k_t, R_a, gravity_torque
inertial_terms = (I_1 + J_m*N^2);
num = [0, 0, (N*k_t)/(R_a*inertial_terms)]; % numerator Open loop
den = [1, (N^2*(b + k_t^2/R_a)) / inertial_terms, gravity_torque/
inertial_terms]; % Denominator open loop

% G = tf(num, den); % transfer function
omega_n = sqrt(den(3));
zeta = den(2) / (2*omega_n);
% P = pole(G);
% tau = 1 / -P(2);
% G_steady_state = num(3)/ den(3);
% 0.63*G_steady_state;
```

Problem 2a

Problem 2.2

```
disp("")
disp("Problem 2b")
disp("")
num = [0, 0, N*k_t/inertial_terms];
den = [1, N^2*b/inertial_terms, gravity_torque/inertial_terms];
% H = tf(num, den);
% omega_n = sqrt(den(3));
% zeta = den(2) / (2*omega_n);
% rlocus(H); %works, just commented to avoid seeing it
% stepinfo(H);
%
% ts = 4 / (zeta*omega_n);
% tp = pi / (omega_n*sqrt(1-zeta^2));
% os = exp(-pi*zeta/(sqrt(1-zeta^2)));

% H_steady_state = num(3) / den(3);
% step(H); %works, just commented to avoid seeing it + speed up script
```

Problem 2b

Problem 3.1

```
disp("")
disp("Problem 3.1")
disp("")
% G_p = H

% P = pole(G_p)

num = [0, 0, N*k_t/inertial_terms];
den = [1, N^2*b/inertial_terms, gravity_torque/inertial_terms];


% Determine desired characteristic eqn and poles
overshoot_des = .1
zeta_des = -log(overshoot_des) / sqrt(pi^2 + (log(overshoot_des))^2)
ts_des = 0.2
omega_n_des = 4/ (ts_des*zeta_des)
omega_d_des = omega_n_des*sqrt(1-zeta_des^2)
den_des = [1, 2*zeta_des*omega_n_des, omega_n_des^2]
real_des = zeta_des*omega_n_des
img_des = omega_d_des

% Factor poles for testing RL equations


% Angle Condition Checks
p = [1.89, 3.62]
p_x = 1.89
p_y = 3.62
```

```
theta1 = 180- atand((img_des-p_y)/(real_des - p_x) )
theta2 = 180 - atand((img_des+p_y)/(real_des - p_x) )
l1 = sqrt((img_des-p_y)^2 + (real_des - p_x)^2 )
l2 = sqrt((img_des+p_y)^2 + (real_des - p_x)^2 )

phi = theta1 + theta2 - 180

kp_over_kd = img_des/tand(phi) + real_des

l3 = sqrt(img_des^2 + (kp_over_kd-real_des)^2)

kd = (l1*l2 / l3)/134.2
kp = kd* kp_over_kd
```

Problem 3.1

```
overshoot_des =
    0.1000
zeta_des =
    0.5912
ts_des =
    0.2000
omega_n_des =
   33.8321
omega_d_des =
   27.2875
den_des =
   1.0e+03 *
    0.0010    0.0400    1.1446
real_des =
    20
img_des =
   27.2875
p =
    1.8900    3.6200
p_x =
    1.8900
p_y =
    3.6200
theta1 =
  127.4226
theta2 =
  120.3678
l1 =
   29.8014
l2 =
   35.8224
phi =
   67.7904
kp_over_kd =
   31.1412
l3 =
   29.4743
kd =
```

```
    0.2699
kp =
    8.4049
```

3.2

We need a new l3

```
l3 = sqrt(real_des^2 + img_des^2)
theta3 = atan2d(27.278, -20)
theta_remaining = -180 + theta1 + theta2 + theta3
phi = theta_remaining/2
z = img_des/tand(phi) + 20
Kd_PID = ( (l1*l2*l3) / (2*sqrt((z-real_des)^2 + img_des^2)) ) / 134.2

Kp_PID = 2*z * Kd_PID
Ki_PID = z^2*Kd_PID
```

```
l3 =
   33.8321
theta3 =
  126.2486
theta_remaining =
  194.0390
phi =
   97.0195
z =
   16.6401
Kd_PID =
    4.8945
Kp_PID =
  162.8891
Ki_PID =
   1.3552e+03
```

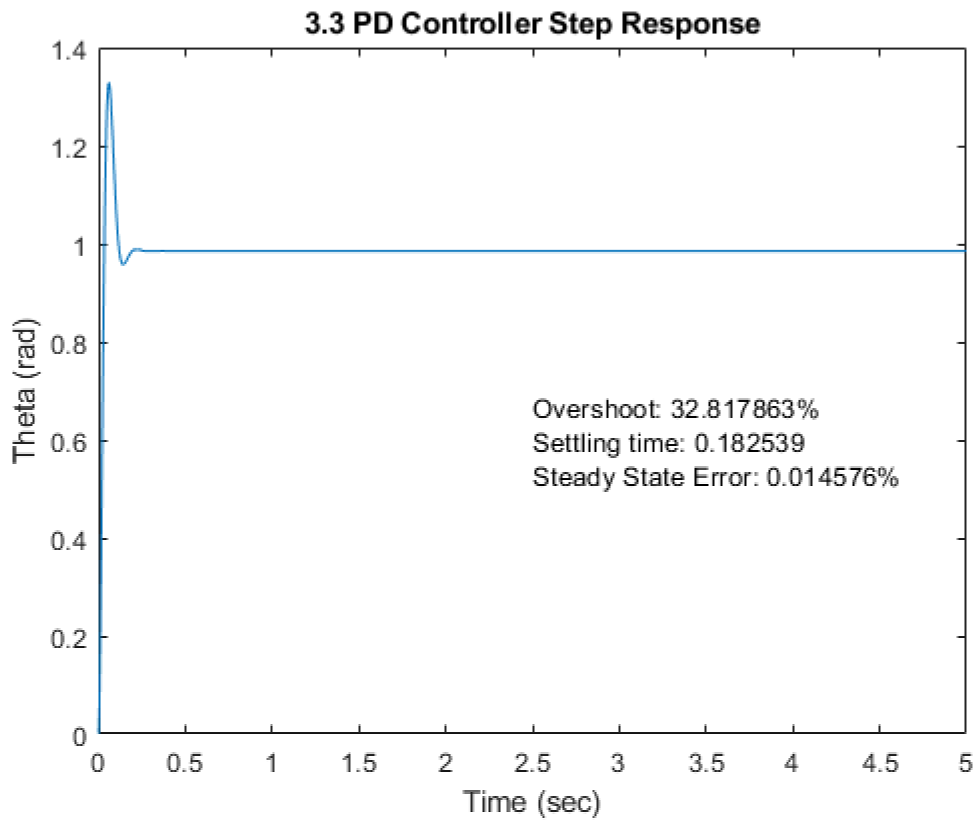
3.3

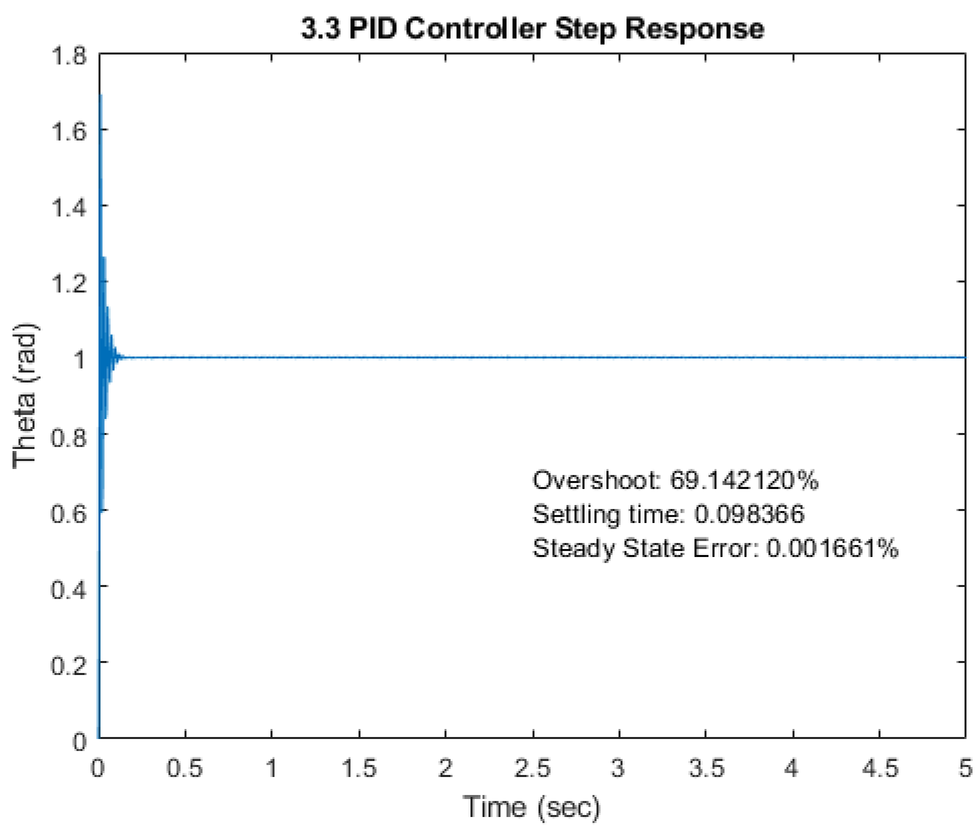
```
out = sim("PS4_sim.slx", 5);
fig1 = figure(1)
fig1 = out.PD_controller.plot()
xlabel("Time (sec)")
ylabel("Theta (rad)")
title("3.3 PD Controller Step Response")
S = stepinfo(out.PD_controller.Data, out.tout, 1)
% Unpack S for values
values = [S.Overshoot, S.SettlingTime, (1-out.PD_controller.Data(end))]
str = sprintf("Overshoot: %f%%\nSettling time: %f\nSteady State Error: %f%%",
 values)
text(2.5, .6, str)

fig2 = figure(2)
fig2 = out.PID_controller.plot()
xlabel("Time (sec)")
```

```
ylabel("Theta (rad)")
title("3.3 PID Controller Step Response")
S = stepinfo(out.PID_controller.Data, out.tout, 1)
% Unpack S for values
values = [S.Overshoot, S.SettlingTime, (1-out.PID_controller.Data(end))]
str = sprintf("Overshoot: %f%%\nSettling time: %f\nSteady State Error: %f%%",
 values)
text(2.5, .6, str)
```

```
fig1 =
  Figure (1) with properties:

      Number: 1
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [1000 918 560 420]
       Units: 'pixels'

  Use GET to show all properties
fig1 =
  Line with properties:

              Color: [0 0.4470 0.7410]
          LineStyle: '-'
          LineWidth: 0.5000
             Marker: 'none'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [0 3.1554e-30 1.4828e-07 8.8966e-07 4.5966e-06 … ]
              YData: [0 2.3656e-56 5.2234e-11 1.8804e-09 5.0191e-08 … ]

  Use GET to show all properties
S =
  struct with fields:

         RiseTime: 0.0233
    TransientTime: 0.1825
     SettlingTime: 0.1825
      SettlingMin: 0.9086
      SettlingMax: 1.3282
        Overshoot: 32.8179
       Undershoot: 0
             Peak: 1.3282
         PeakTime: 0.0580
values =
   32.8179    0.1825    0.0146
str =
    "Overshoot: 32.817863%
     Settling time: 0.182539
     Steady State Error: 0.014576%"
fig2 =
  Figure (2) with properties:

      Number: 2
```

```
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [1000 918 560 420]
       Units: 'pixels'

  Use GET to show all properties
fig2 =
  Line with properties:

              Color: [0 0.4470 0.7410]
          LineStyle: '-'
          LineWidth: 0.5000
             Marker: 'none'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [0 3.1554e-30 1.4828e-07 8.8966e-07 4.5966e-06 … ]
              YData: [0 4.3598e-55 9.6269e-10 3.4656e-08 9.2504e-07 … ]

  Use GET to show all properties
S =
  struct with fields:

         RiseTime: 0.0041
    TransientTime: 0.0984
     SettlingTime: 0.0984
      SettlingMin: 0.5923
      SettlingMax: 1.6914
        Overshoot: 69.1421
       Undershoot: 0
             Peak: 1.6914
         PeakTime: 0.0111
values =
   69.1421    0.0984    0.0017
str =
    "Overshoot: 69.142120%
     Settling time: 0.098366
     Steady State Error: 0.001661%"
```


3.3 PD Controller Step Response
Theta (rad)
Time (sec)
Overshoot: 32.817863%
Settling time: 0.182539
Steady State Error: 0.014576%


3.3 PID Controller Step Response
Theta (rad)
Time (sec)
Overshoot: 69.142120%
Settling time: 0.098366
Steady State Error: 0.001661%

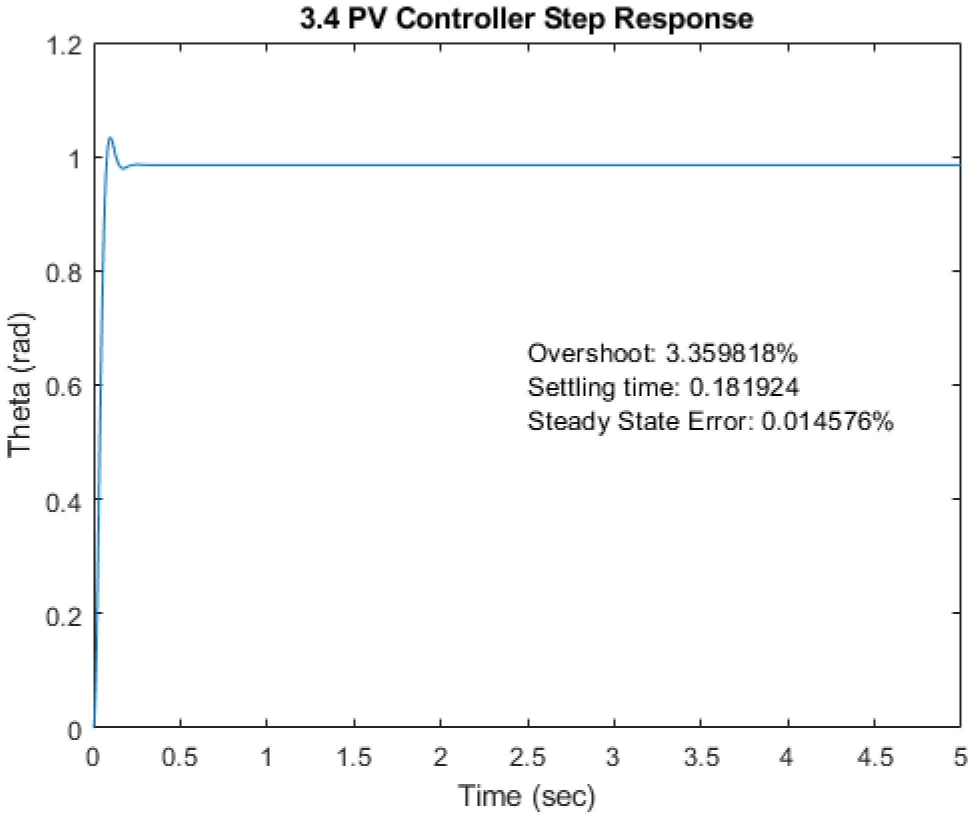
3.4

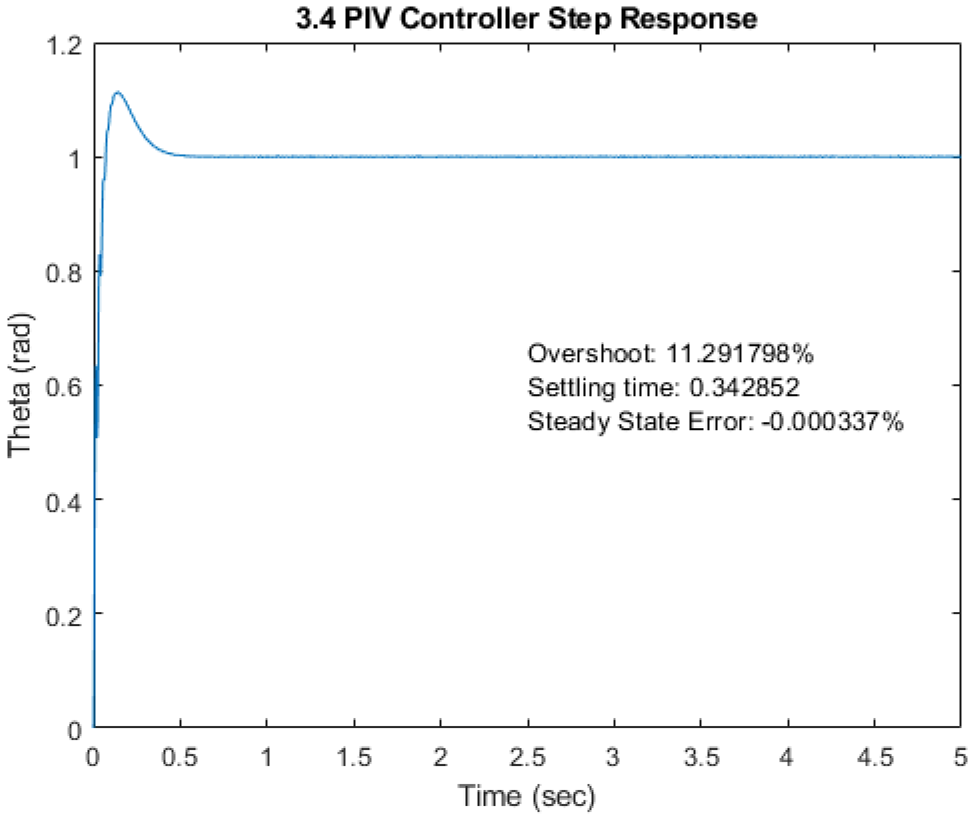
```
out = sim("PS4_sim2.slx", 5);
fig3 = figure(3)
fig3 = out.PV_controller.plot()
xlabel("Time (sec)")
ylabel("Theta (rad)")
title("3.4 PV Controller Step Response")
S = stepinfo(out.PV_controller.Data, out.tout, 1)
% Unpack S for values
values = [S.Overshoot, S.SettlingTime, (1-out.PV_controller.Data(end))]
str = sprintf("Overshoot: %f%%\nSettling time: %f\nSteady State Error: %f%%",
 values)
text(2.5, .6, str)

fig4 = figure(4)
fig4 = out.PIV_controller.plot()
xlabel("Time (sec)")
ylabel("Theta (rad)")
title("3.4 PIV Controller Step Response")
S = stepinfo(out.PIV_controller.Data, out.tout, 1)
% Unpack S for values
values = [S.Overshoot, S.SettlingTime, (1-out.PIV_controller.Data(end))]
str = sprintf("Overshoot: %f%%\nSettling time: %f\nSteady State Error: %f%%",
 values)
text(2.5, .6, str)
```

```
fig3 =
  Figure (3) with properties:

      Number: 3
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [1000 918 560 420]
       Units: 'pixels'

  Use GET to show all properties
fig3 =
  Line with properties:

              Color: [0 0.4470 0.7410]
          LineStyle: '-'
          LineWidth: 0.5000
             Marker: 'none'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [0 3.1554e-30 1.4828e-07 8.8966e-07 4.5966e-06 … ]
              YData: [0 5.6173e-57 1.2404e-11 4.4653e-10 1.1920e-08 … ]

  Use GET to show all properties
S =
  struct with fields:
```

```
         RiseTime: 0.0491
    TransientTime: 0.1819
     SettlingTime: 0.1819
      SettlingMin: 0.9191
      SettlingMax: 1.0336
        Overshoot: 3.3598
       Undershoot: 0
             Peak: 1.0336
         PeakTime: 0.0962
values =
    3.3598    0.1819    0.0146
str =
    "Overshoot: 3.359818%
     Settling time: 0.181924
     Steady State Error: 0.014576%"
fig4 =
  Figure (4) with properties:

      Number: 4
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [1000 918 560 420]
       Units: 'pixels'

  Use GET to show all properties
fig4 =
  Line with properties:

              Color: [0 0.4470 0.7410]
          LineStyle: '-'
          LineWidth: 0.5000
             Marker: 'none'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [0 3.1554e-30 1.4828e-07 8.8966e-07 4.5966e-06 … ]
              YData: [0 1.0886e-55 2.4039e-10 8.6539e-09 2.3101e-07 … ]

  Use GET to show all properties
S =
  struct with fields:

         RiseTime: 0.0475
    TransientTime: 0.3429
     SettlingTime: 0.3429
      SettlingMin: 0.9114
      SettlingMax: 1.1129
        Overshoot: 11.2918
       Undershoot: 0
             Peak: 1.1129
         PeakTime: 0.1440
values =
   11.2918    0.3429   -0.0003
str =
    "Overshoot: 11.291798%
```

```
Settling time: 0.342852
Steady State Error: -0.000337%"
```



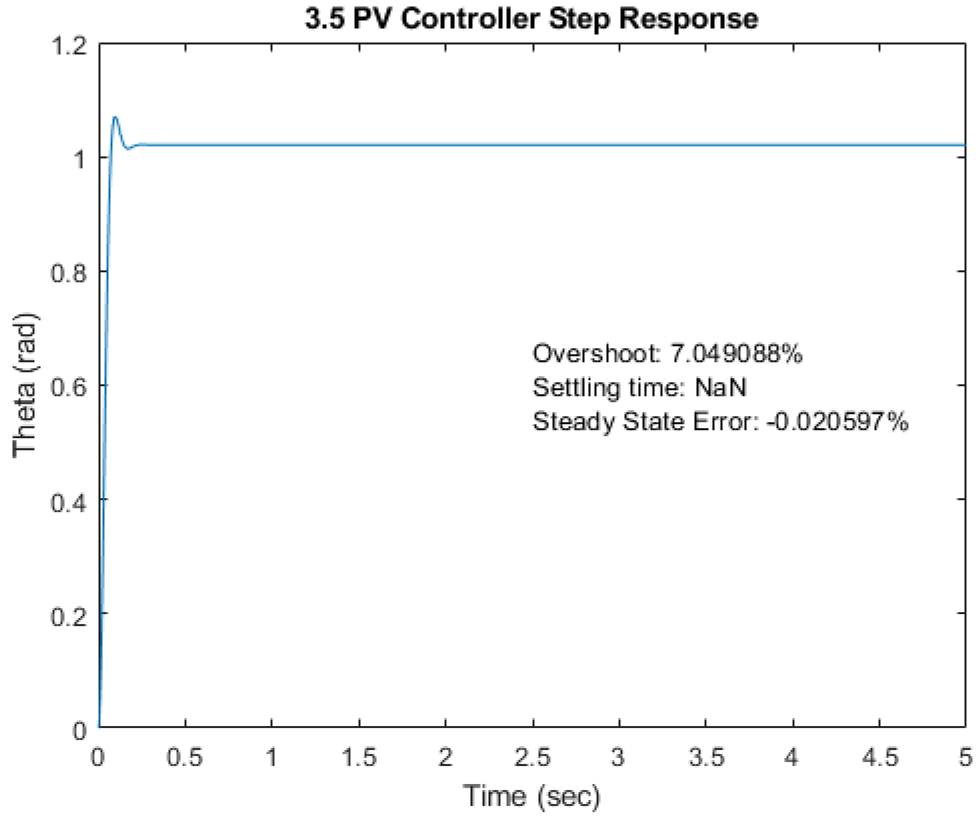


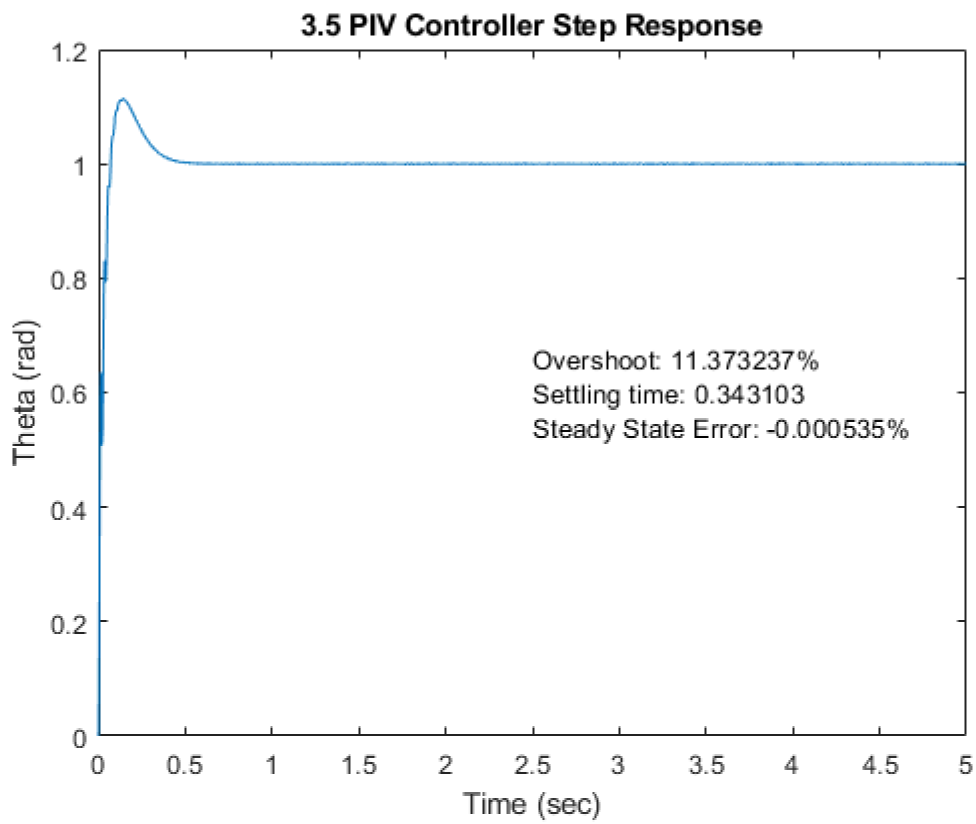
3.5

```
disturbance_amplitude = .3

out = sim("PS4_sim3.slx", 5);
fig5 = figure(5)
fig5 = out.PV_controller.plot()
xlabel("Time (sec)")
ylabel("Theta (rad)")
title("3.5 PV Controller Step Response")
S = stepinfo(out.PV_controller.Data, out.tout, 1)
% Unpack S for values
values = [S.Overshoot, S.SettlingTime, (1-out.PV_controller.Data(end))]
str = sprintf("Overshoot: %f%%\nSettling time: %f\nSteady State Error: %f%%",
 values)
text(2.5, .6, str)

fig6 = figure(6)
fig6 = out.PIV_controller.plot()
xlabel("Time (sec)")
ylabel("Theta (rad)")
title("3.5 PIV Controller Step Response")
S = stepinfo(out.PIV_controller.Data, out.tout, 1)
% Unpack S for values
values = [S.Overshoot, S.SettlingTime, (1-out.PIV_controller.Data(end))]
```

```
str = sprintf("Overshoot: %f%%\nSettling time: %f\nSteady State Error: %f%%",
 values)
text(2.5, .6, str)

disturbance_amplitude =
    0.3000
fig5 =
  Figure (5) with properties:

      Number: 5
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [1000 918 560 420]
       Units: 'pixels'

  Use GET to show all properties
fig5 =
  Line with properties:

              Color: [0 0.4470 0.7410]
          LineStyle: '-'
          LineWidth: 0.5000
             Marker: 'none'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [0 3.1554e-30 1.4828e-07 8.8966e-07 4.5966e-06 … ]
              YData: [0 5.8178e-57 1.2846e-11 4.6247e-10 1.2345e-08 … ]

  Use GET to show all properties
S =
  struct with fields:

         RiseTime: 0.0462
    TransientTime: NaN
     SettlingTime: NaN
      SettlingMin: 0.9084
      SettlingMax: 1.0705
        Overshoot: 7.0491
       Undershoot: 0
             Peak: 1.0705
         PeakTime: 0.0962
values =
    7.0491       NaN   -0.0206
str =
    "Overshoot: 7.049088%
     Settling time: NaN
     Steady State Error: -0.020597%"
fig6 =
  Figure (6) with properties:

      Number: 6
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [1000 918 560 420]
```

```
       Units: 'pixels'

  Use GET to show all properties
fig6 =
  Line with properties:

              Color: [0 0.4470 0.7410]
          LineStyle: '-'
          LineWidth: 0.5000
             Marker: 'none'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [0 3.1554e-30 1.4828e-07 8.8966e-07 4.5966e-06 … ]
              YData: [0 1.0906e-55 2.4083e-10 8.6699e-09 2.3144e-07 … ]

  Use GET to show all properties
S =
  struct with fields:

         RiseTime: 0.0475
    TransientTime: 0.3431
     SettlingTime: 0.3431
      SettlingMin: 0.9126
      SettlingMax: 1.1137
        Overshoot: 11.3732
       Undershoot: 0
             Peak: 1.1137
         PeakTime: 0.1368
values =
   11.3732    0.3431   -0.0005
str =
    "Overshoot: 11.373237%
     Settling time: 0.343103
     Steady State Error: -0.000535%"
```


3.5 PV Controller Step Response
Theta (rad)
Time (sec)
Overshoot: 7.049088%
Settling time: NaN
Steady State Error: -0.020597%


3.5 PIV Controller Step Response
Theta (rad)
Time (sec)
Overshoot: 11.373237%
Settling time: 0.343103
Steady State Error: -0.000535%

Published with MATLAB® R2022a