Fall 2024 – 16-642 Manipulation, Estimation, and Control
Problem Set 2

Ankit Aggarwal (ankitagg)

Question 1

To build an observer, we first check for observability of the system,

```
obsv = obsv(A_l, C);
rank(obsv)
```

The rank output is 4, hence the system is observable.

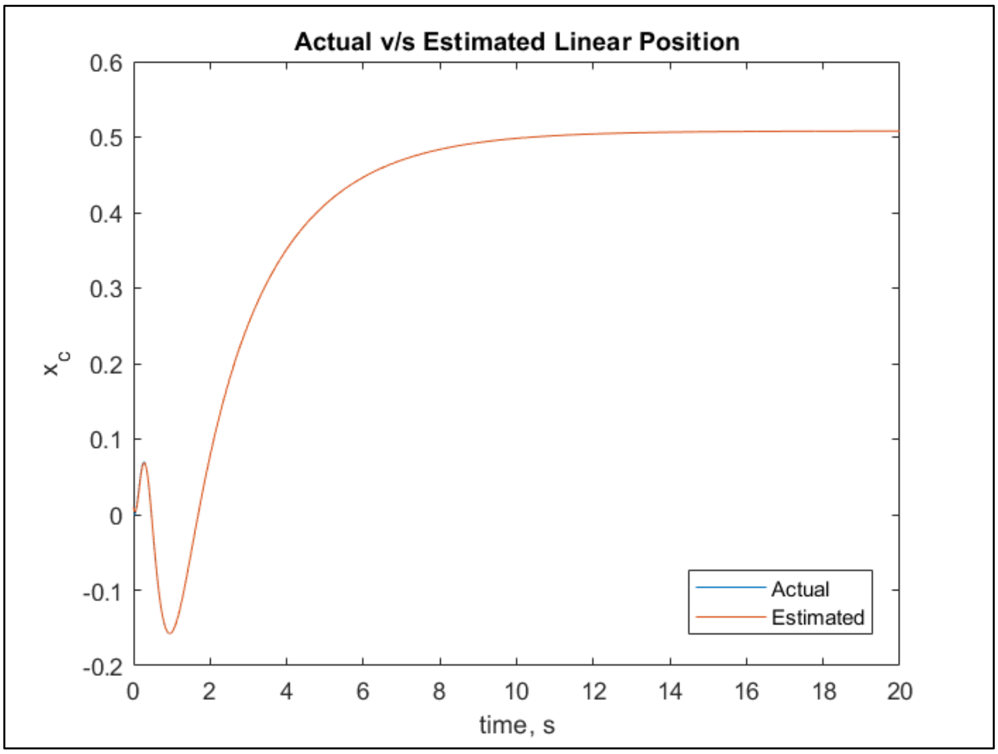
Building the observer system for the question, using the following poles,

```
% P = [-0.4571 -0.6601 -1.2558 -3.2915]'
poles_observed = [-3, -4, -6, -15];
K_not = (place(A_l',C',poles_observed))';
```

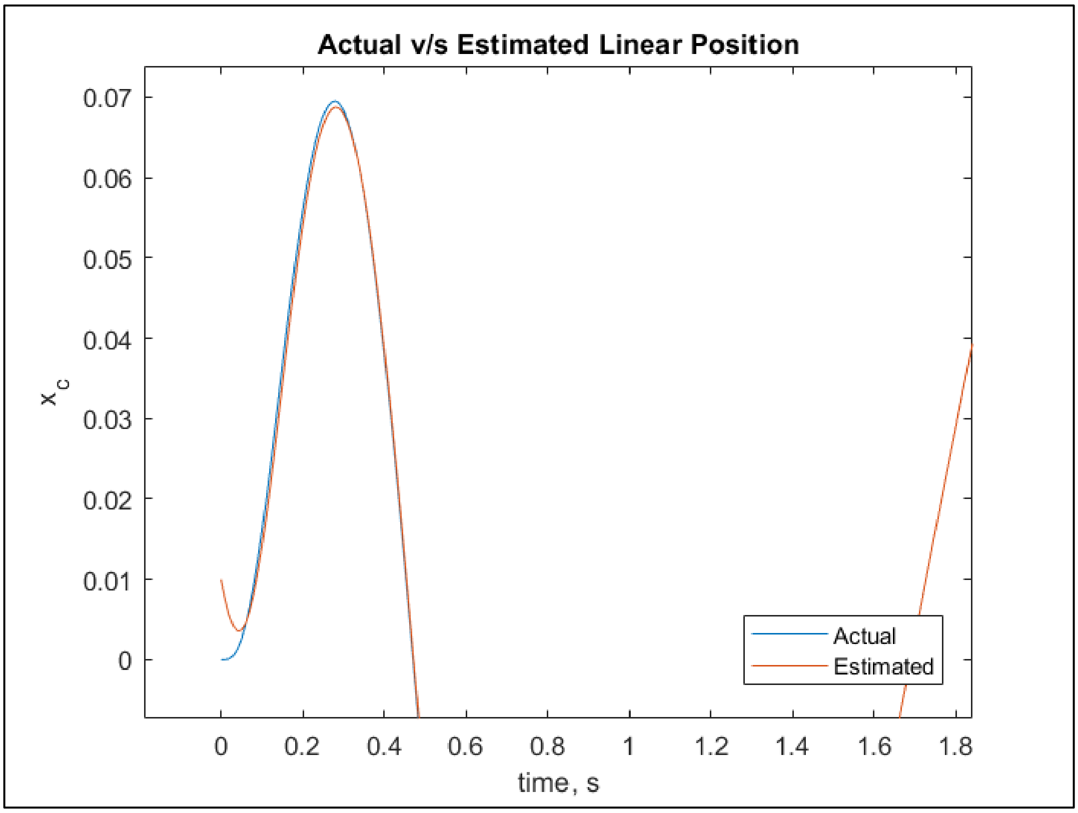
The system showed the following plots between actual state x and observed state $\hat{x}$,

```
x_0 = [0,0,0,0]';
xhat_0 = [0.01, 0.01, -0.03, 0.01]';
```

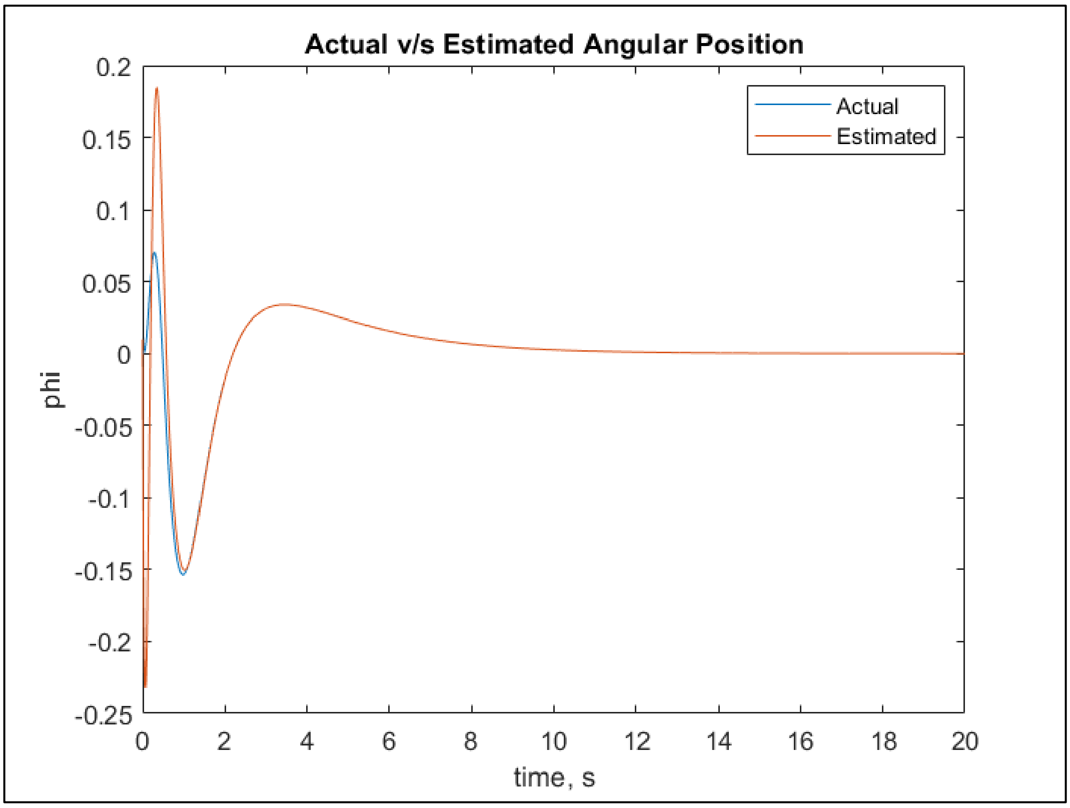
(a) Actual v/s Estimated Linear Position



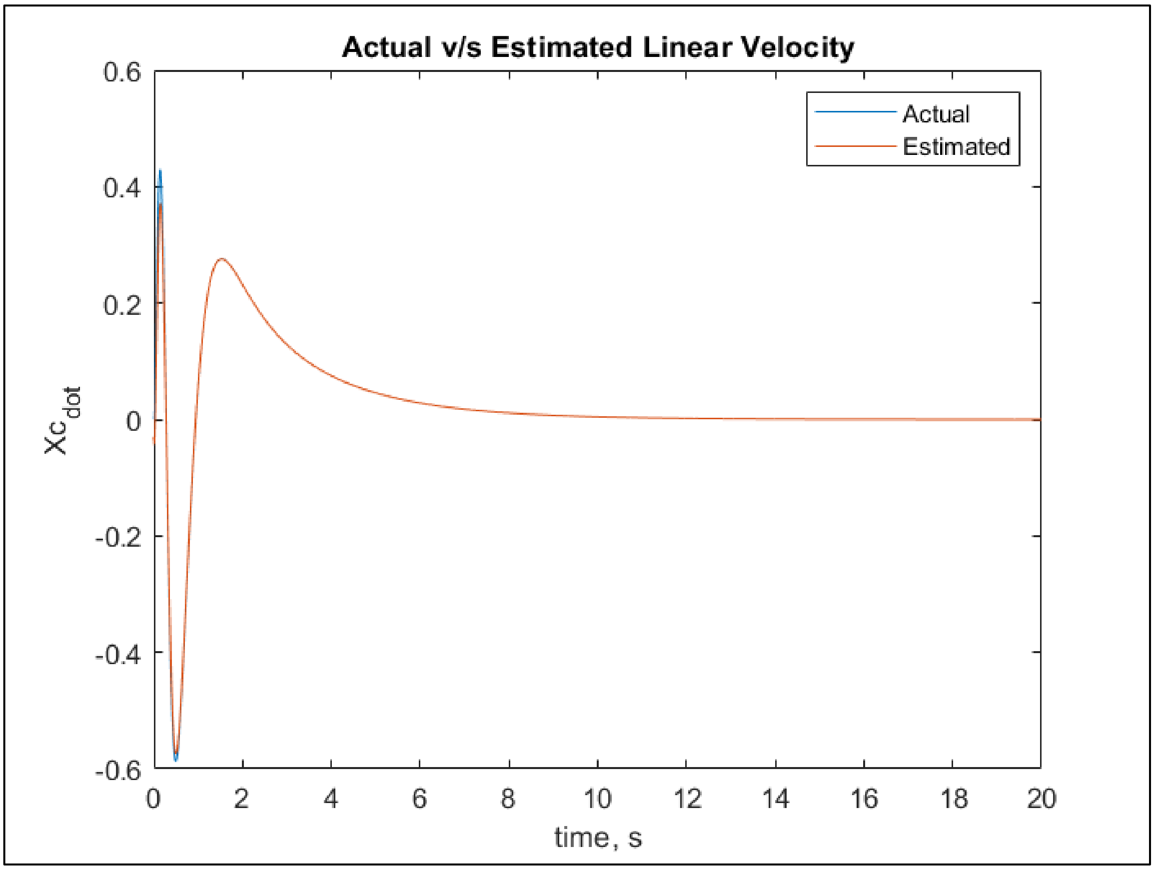
Zoomed in plot to highlight deviations is shown below,



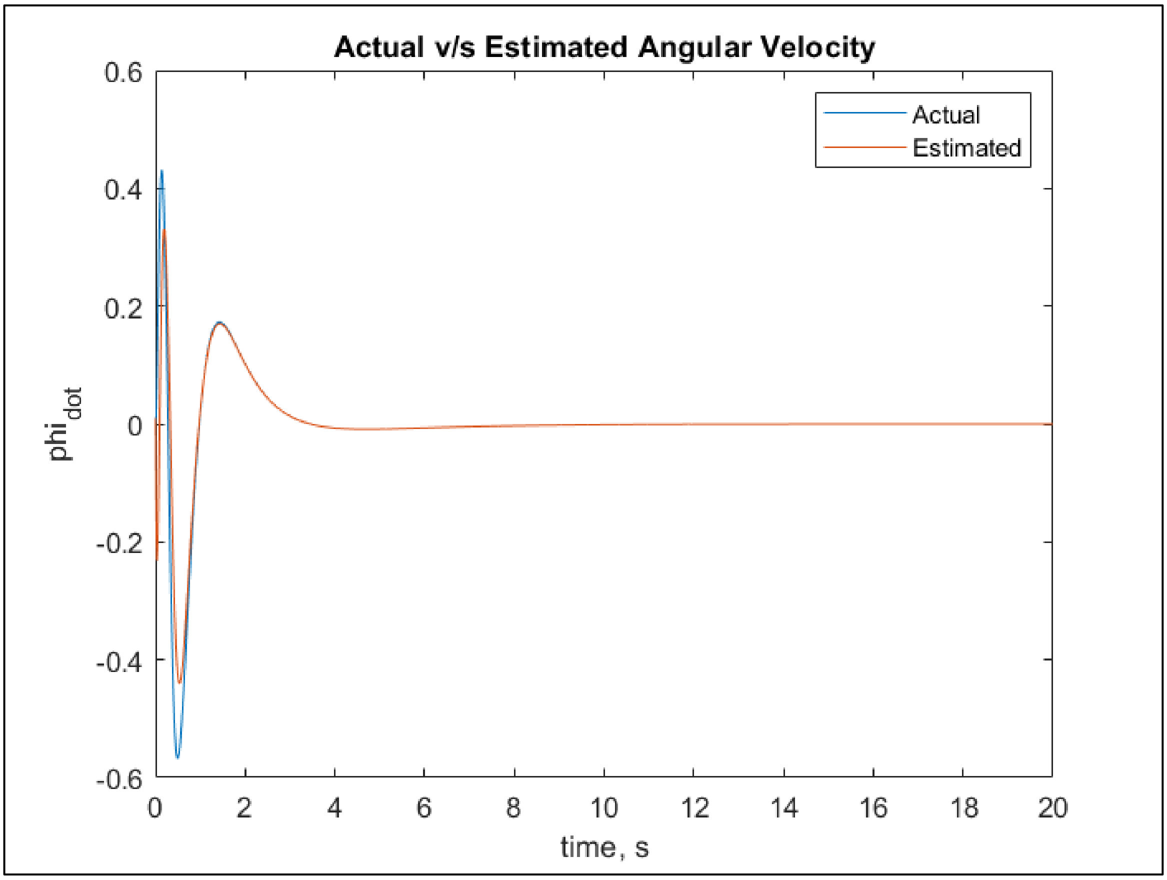
(b) Actual v/s Estimated Angular Position



(c) Actual v/s Estimated Linear Velocity



(d) Actual v/s Estimated Angular Velocity



Code Used

File Name: system_nl_error.m {Function}

```
function dx_xhat = system_nl_error(t,x)

% model set up
x_true = x(1:4);
x_estimate = x(5:8);

xc = x(1);
phi = x(2);
xc_dot = x(3);
phi_dot = x(4);

xc_hat = x(5);
phi_hat = x(6);
xc_dot_hat = x(7);
phi_dot_hat = x(8);

gamma = 2; alpha = 1; beta = 1; D = 1; mu = 3;

A_l = [0 0 1 0; 0 0 0 1; 0 1 -3 0; 0 2 -3 0];
B_l= [0 0 1 1]';
C = [39.37008 0 0 0];
Qu = 450;
Qx = [700 0 0 0; 0 700 0 0; 0 0 15 0; 0 0 0 10];
[Kc,~,~] = lqr(A_l,B_l,Qx,Qu);
% P = [-0.4571 -0.6601 -1.2558 -3.2915]'
Acl = A_l - B_l * Kc;
Acl_inv = inv(Acl);
y_des = 20 * square(2*pi*0.01*t);
K_f = -inv(C*Acl_inv*B_l);

% true model
F = K_f * y_des - Kc * x_estimate;
dx_true = [xc_dot;
    phi_dot;
    (- alpha*sin(phi)*beta*phi_dot^2 + F*alpha - alpha*mu*xc_dot +
cos(phi)*sin(phi)*D*beta)/(alpha*gamma - beta^2*cos(phi)^2);
    (- cos(phi)*sin(phi)*beta^2*phi_dot^2 + F*cos(phi)*beta + sin(phi)*D*gamma -
mu*xc_dot*cos(phi)*beta)/(alpha*gamma - beta^2*cos(phi)^2)
    ];

% estimated model
poles_observed = [-3, -4, -6, -15];
K_not = (place(A_l',C',poles_observed))';

dx_hat = [xc_dot_hat;
    phi_dot_hat;
    (- alpha*sin(phi_hat)*beta*phi_dot_hat^2 + F*alpha - alpha*mu*xc_dot_hat +
cos(phi_hat)*sin(phi_hat)*D*beta)/(alpha*gamma - beta^2*cos(phi_hat)^2);
    (- cos(phi_hat)*sin(phi_hat)*beta^2*phi_dot_hat^2 + F*cos(phi_hat)*beta +
sin(phi_hat)*D*gamma - mu*xc_dot_hat*cos(phi_hat)*beta)/(alpha*gamma - beta^2*cos(phi_hat)^2)
    ];
dx_hat = dx_hat + K_not * (C*x_true - C*x_estimate);

% final output
dx_xhat = [dx_true; dx_hat];

end
```

File Name: Observed.m {Script}

```
clear;
t_des = 0:0.01:20;
x_0 = [0,0,0,0]';
xhat_0 = [0.01, 0.01, -0.03, 0.01]';
x_xhat_0 = [x_0; xhat_0];


[t_sol, d_x_xhat_sol] = ode45(@system_nl_error, t_des, x_xhat_0);

figure()
plot(t_sol, d_x_xhat_sol(:,1))
hold on
plot(t_sol, d_x_xhat_sol(:,5))
xlabel('time, s')
ylabel('x_c')
legend('Actual','Estimated')
title('Actual v/s Estimated Linear Position')
hold off

figure()
plot(t_sol, d_x_xhat_sol(:,2))
hold on
plot(t_sol, d_x_xhat_sol(:,6))
xlabel('time, s')
ylabel('phi')
legend('Actual','Estimated')
title('Actual v/s Estimated Angular Position')
hold off

figure()
plot(t_sol, d_x_xhat_sol(:,3))
hold on
plot(t_sol, d_x_xhat_sol(:,7))
xlabel('time, s')
ylabel('Xc_{dot}')
legend('Actual','Estimated')
title('Actual v/s Estimated Linear Velocity')
hold off

figure()
plot(t_sol, d_x_xhat_sol(:,4))
hold on
plot(t_sol, d_x_xhat_sol(:,8))
xlabel('time, s')
ylabel('phi_{dot}')
legend('Actual','Estimated')
title('Actual v/s Estimated Angular Velocity')
hold off
```

Question 2

Q2 Plant Model G

$$G: u(t) \longrightarrow y(t)$$

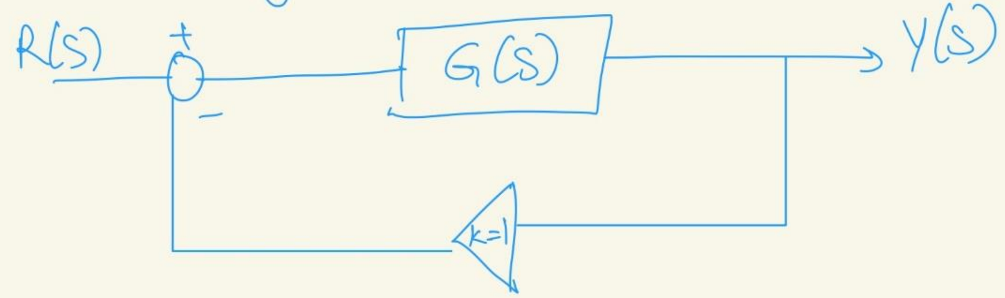
$$\ddot{y}(t) + 13\dot{y}(t) + 78y(t) = \ddot{u}(t) + 4\dot{u}(t) + 80u(t)$$

(a) Taking Laplace Transform,

$$(s^2+13s+78)\,Y(s) = (s^2+4s+80)\,U(s)$$

$$G(s) = \frac{Y(s)}{U(s)} = \boxed{\frac{s^2+4s+80}{s^2+13s+78}}$$

(b) The given system is



$$Y(S) = \frac{G(s)}{1+G(s)}\,R(S)$$

Hence, closed loop TF $T(s)$

$$= \frac{G(s)}{1+G(s)} = \boxed{\frac{s^2+4s+80}{2s^2+17s+158}}$$

(c) Code used:

```
system = tf([1,4,80],[2, 17, 158]);
p = pole(system)
z = zero(system)
```

Output:

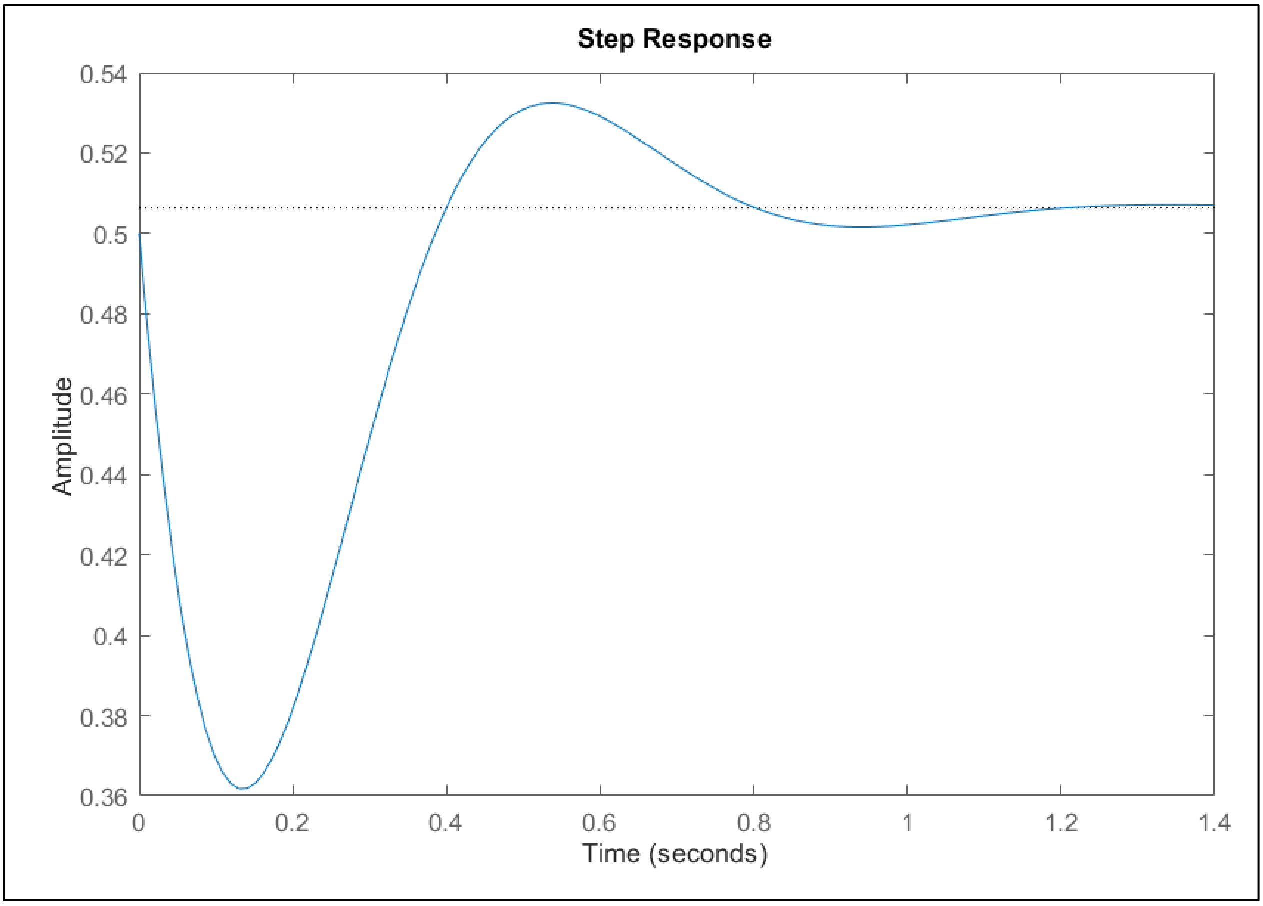
```
p = 2×1 complex
     -4.2500 + 7.8062i
     -4.2500 - 7.8062i

z = 2×1 complex
     -2.0000 + 8.7178i
     -2.0000 - 8.7178i
```

(d) The function's characteristics based on analysing the poles and zeros are given as,

- Both poles of the given system have negative real parts, signifying a stable system. *[Both poles lie on the left of the imaginary axis]*
- The system has complex conjugate poles, thereby, the system will exhibit oscillatory behaviour.
- As both zeros lie on the left of the imaginary axis and they are significant as they are close to the poles, they will increase the system overshoot but decrease the rise time, making the response faster.

(e) The plot is given below,



(f) Steady State Value

(f) Steady state value

$$y_{ss} = \lim_{s \to 0} s\,Y(s) = \lim_{s \to 0} s\,T(s)\,U(s)$$

Given that $U(s) = 1/s$ [step input]

$$y_{ss} = \lim_{s \to 0} \left(\frac{s^2 + 4s + 80}{2s^2 + 17s + 158} \right) \left(s \times \frac{1}{s} \right)$$

$$y_{ss} = \frac{80}{158} = \boxed{0.506}$$

Question 3

The given system is :

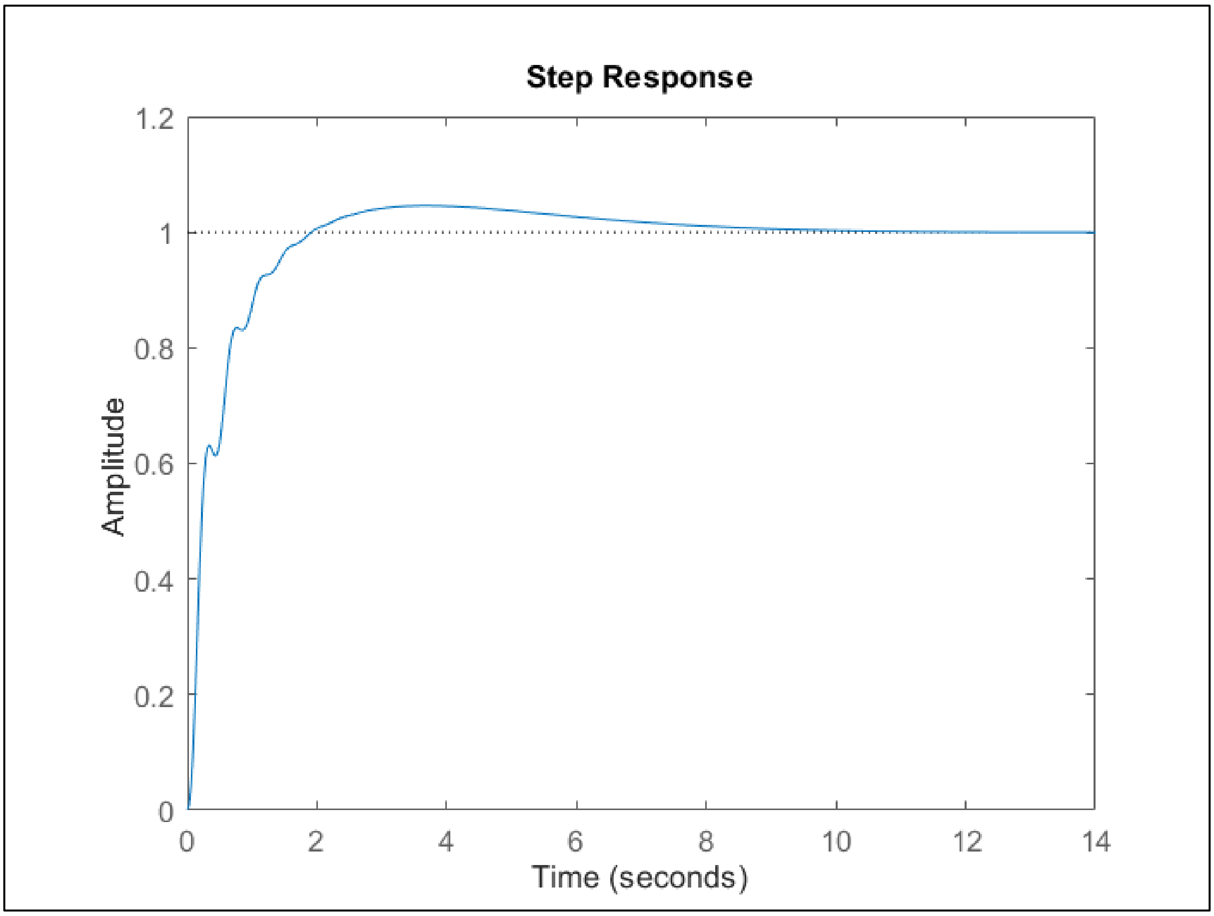
system_PID =

$$\frac{20 s + 17}{s^4 + 9 s^3 + 231 s^2 + 400 s + 60}$$

After PID tuning with parameters,

$K_p = 29$
$K_i = 15$
$K_d = 1$

The plot is given as,



The stepinfo was given as,

RiseTime: 0.9730
TransientTime: 6.6834
SettlingTime: 6.6834
SettlingMin: 0.9041
SettlingMax: 1.0458
Overshoot: 4.5810

Undershoot: 0
Peak: 1.0458
PeakTime: 3.6863

The system steady state value is 1. [ss_value]

Code used:

```
system_PID = tf([20,17],[1,9,231,400,60])
Kp = 29; Ki = 15; Kd = 1;
PID_C = pid(Kp, Ki, Kd);
T = feedback(PID_C*system_PID,1);

[Y,t] = step(T);
ss_value = Y(length(Y));
info = stepinfo(T)
```