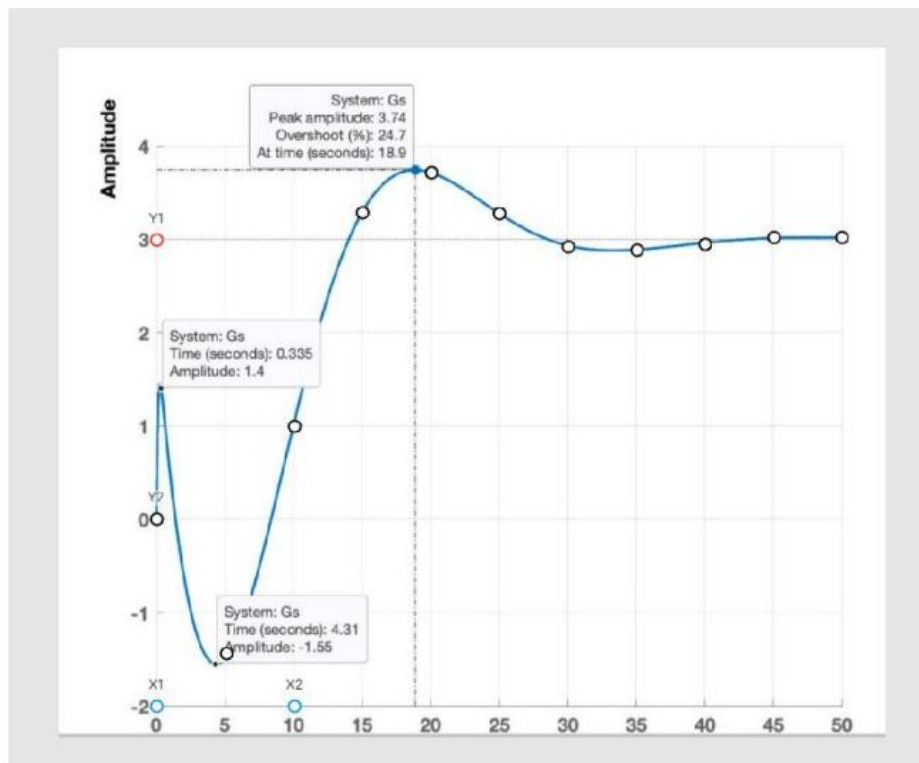


Q1 – Part a)

The general form of the required transfer function will be as given above. Data points are extracted from all the three given plots using an online Plot Digitizer as shown below for the step response



To find the parameters - $\tau_1, \tau_2, \tau_3, \tau_4$ and ξ we will do a least squares fit with the extracted data points of the step response using `lsqcurvefit` in MATLAB.

MATLAB Code:

```
clc; clear all; close all; format compact; format shortg;
```

```
%Data points extracted from the phase in Bode Plot
```

```
xdata_phase = [0.101081767
```

```
0.019909715
```

```
0.038795807
```

```
0.068619658
```

```
0.199097231
```

```
0.400685499
```

```
0.693620204
```

```
2.012511656
```

```
5.077009349
```

```
10.00000821
```

```
29.6457906
```

```
1
```

```
0.266215719
```

```
0.141101818
```

```
1.380978623
```

```
2.96457906
```

```
16.22859007
```

```
49.68923478
```

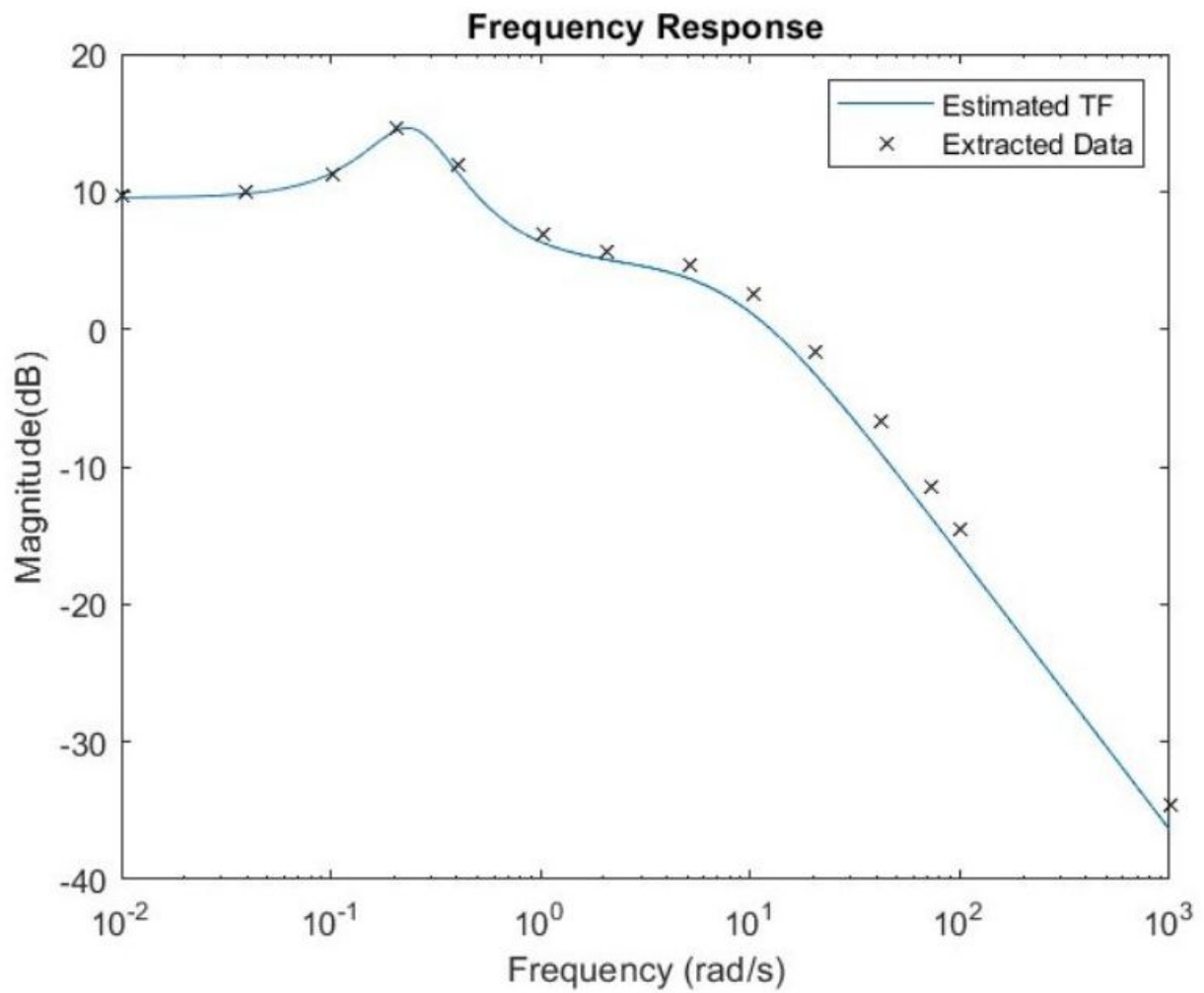
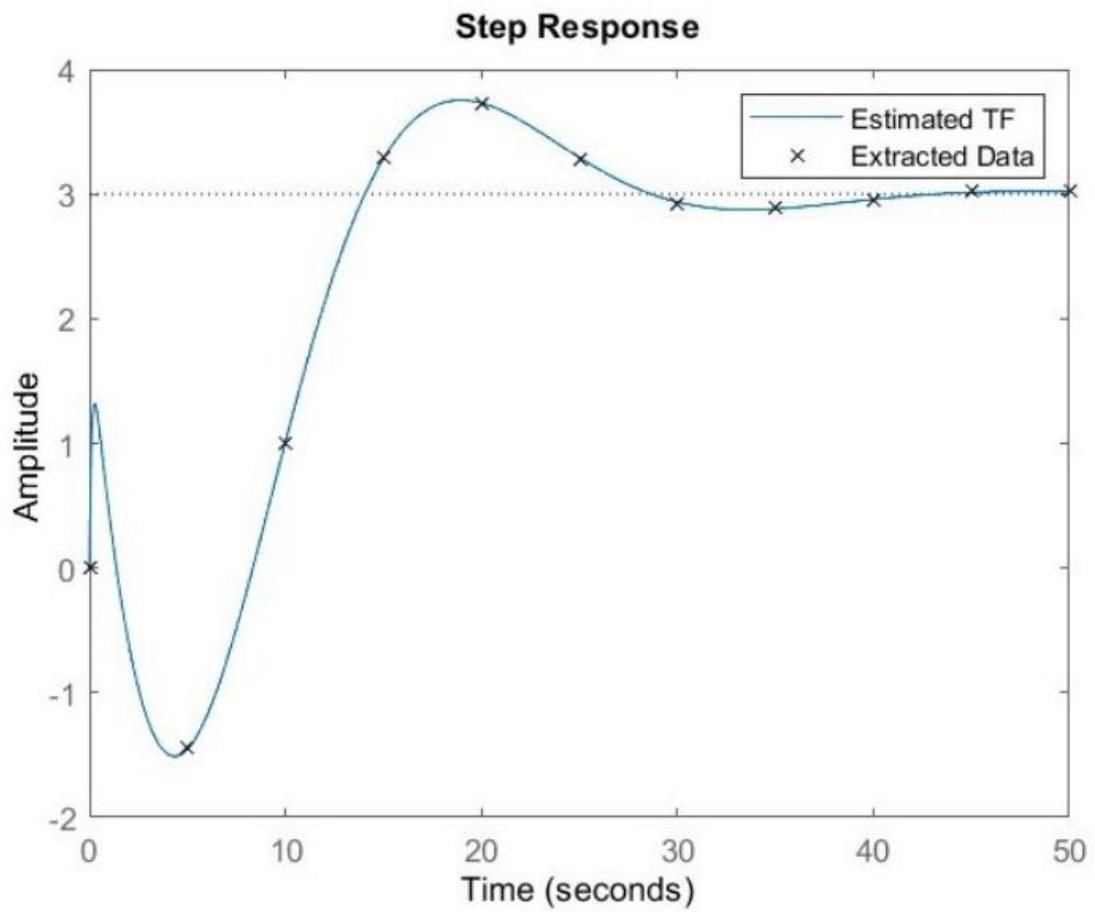
```
89.79912072
```

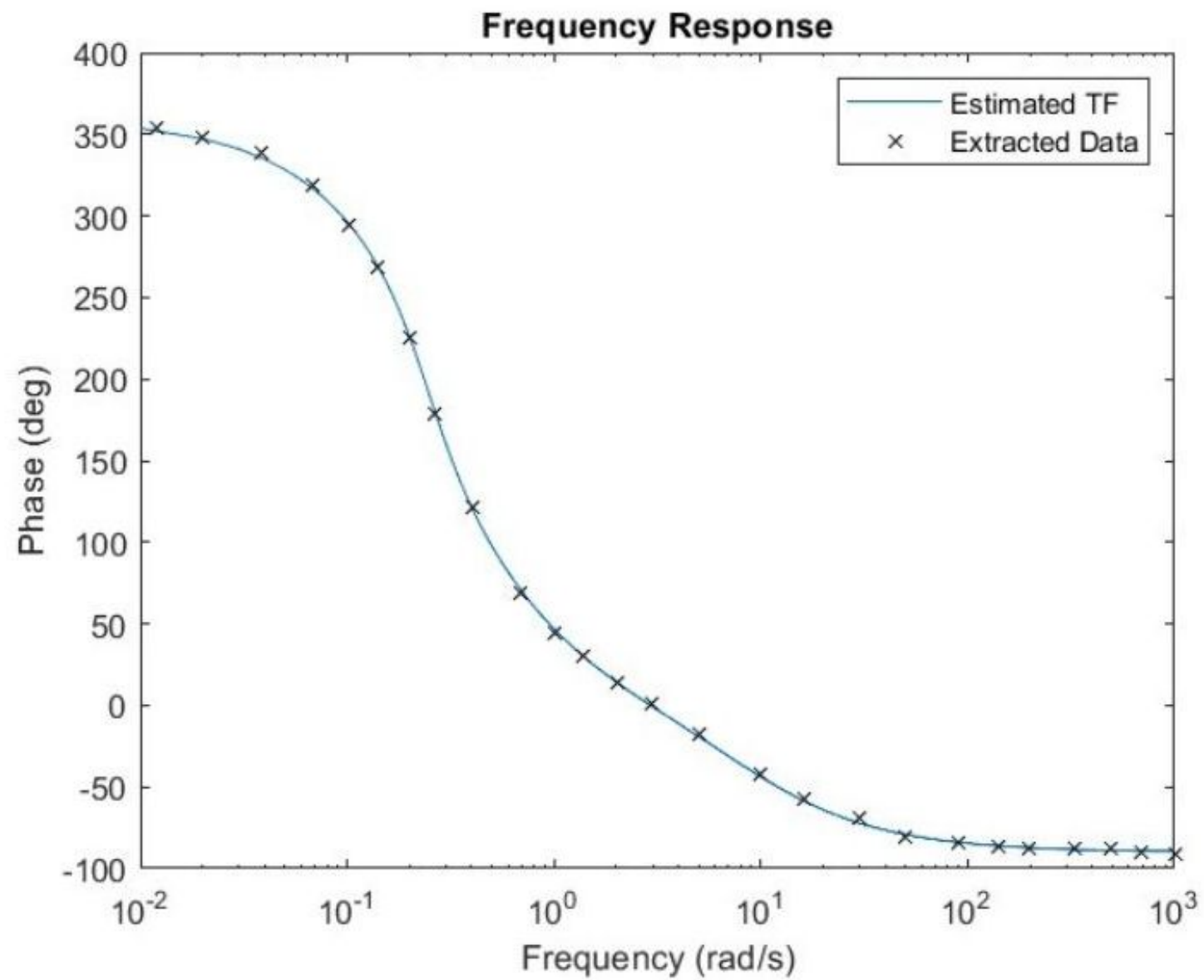
```
199.0977216
```

```
491.5746531
```

```
326.6028031
```

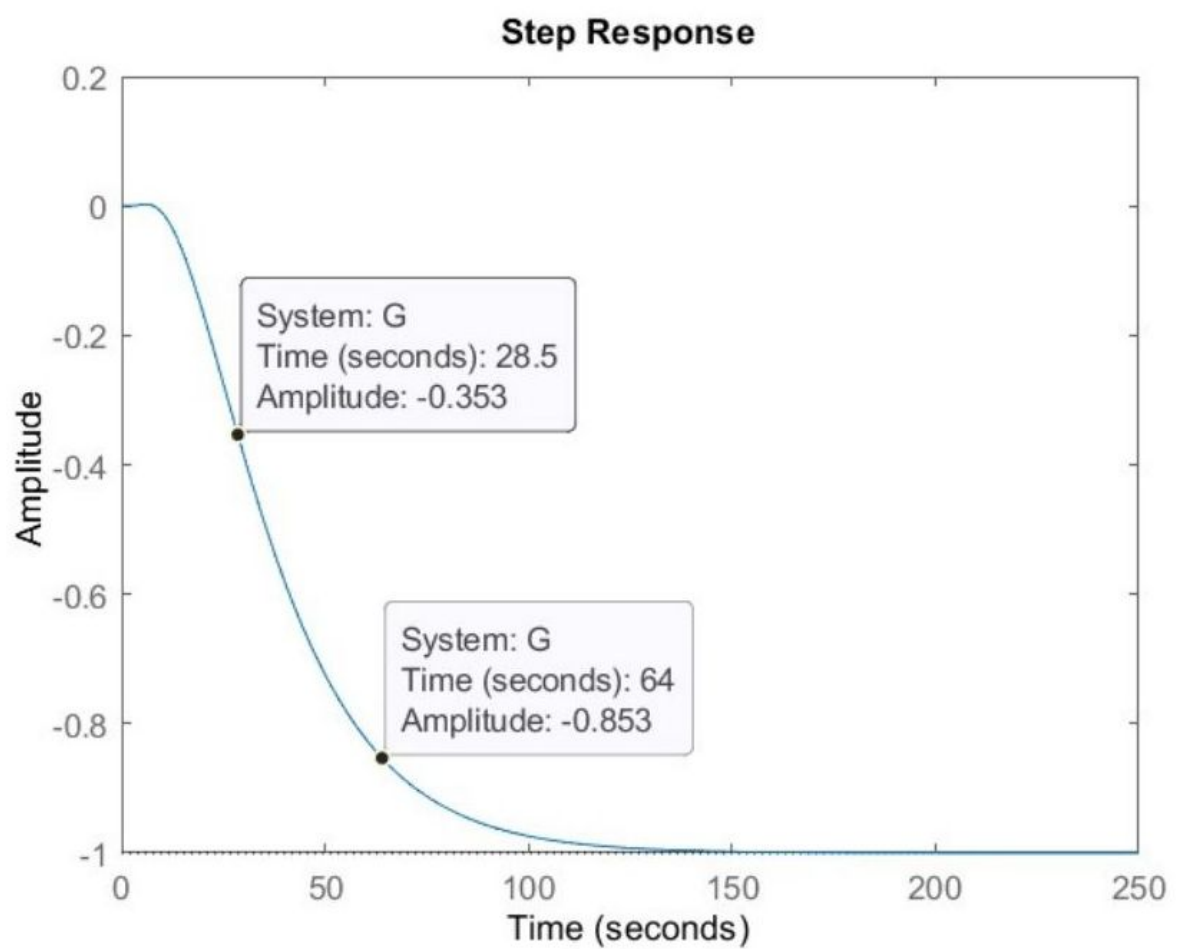
```
1000.002463
```





Q2 Part a)

Following is the step response of the system, required to find t_1 and t_2 for the Krishnaswamy and Sundaresan's two point method:



From the plot, we can see that

$t_1 = 28.5$ sec

$t_2 = 64$ sec

$$\tau_{P2} = 12 + \frac{4}{2} = 14$$

Second largest τ

Half of largest neglected τ

$$D = 2 + 2 + \frac{4}{2} + 1 = 7$$

Due to the zero delay

Existing delay

Half of largest neglected τ

Remaining time constants

$$\therefore G(s) = - \frac{e^{-7s}}{(20s+1)(14s+1)}$$

- c) We need to find an SOTD approximation using least squares for the frequency response. As the delay D does not affect the magnitude, we will find values of K_p , τ_p , τ_{p2} from the magnitude vs frequency Bode plot. Using these values for least squares fitting for Phase vs frequency we can find the delay.

From the MATLAB code, we get $K_p = 0.99977$. We know that K_p is -ve but it does not affect the magnitude.

\therefore We will take $K_p = -0.99977$ as we know that gain has a -ve sign which is not reflected in the magnitude plot, but can be observed in the phase plot.

MATLAB code and output is as follows -

Q2 Part c)

The MATLAB code for the least squares method is given below. The gain and time constants are found using the magnitude vs frequency response after which the delay is found using the phase vs frequency response

MATLAB Code:

```
clc; clear all; close all; format compact; format shortg;
G = tf([2,-1],[960,1328,404,37,1],'IOdelay',2); %Given Process TF

%Krishnaswami and Sundareshan's two point method
figure;
step(G)

%From the step response,
%t1 = 28.5 sec
%t2 = 64 sec
G_KS = tf(-1,[23.785,1],'IOdelay',18.49); %Krishnaswami and Sundareshan
Approximation - FOPTD

%Skogestad's Half Rule method
%FOPTD
G_foptd = tf(-1,[26,1],'IOdelay',15);
%SOPTD
G_soptd = tf(-1,[280,34,1],'IOdelay',7);

%SOPTD using least squares for frequency response
[mag,phase,w] = bode(G);
mag = squeeze(mag); phase = squeeze(phase);
magdb = 20*log10(mag);

%Least squares fitting for the magnitude to estimate tau1, tau2, Kp
pars = lsqcurvefit(@magfun,[1,1,1],w,mag);
%Least squares fitting for the phase to find D
D = lsqcurvefit(@(D,w) phasefun(D,w,pars),2,w,phase);

tau1 = pars(1);
tau2 = pars(2);
Kp = -pars(3); %Taking gain as negative as magnitude does not consider sign of Kp

%Approximate TF from least squares
Ghat = tf(Kp, [tau1*tau2,tau1+tau2,1], 'IOdelay',D);

%Plotting the step and frequency responses for all the approximations
figure; %Step Response
step(G, '-r', G_KS, '-b', G_foptd, '-g', G_soptd, '-y', Ghat, '-k')
ylabel('y(t)')
title('Comparison of Step responses')
legend('Process', 'KS FOPTD', 'Skogestad FOPTD', 'Skogestad SOPTD', 'Least squares FOPTD')

figure; %Frequency Response
bode(G, '-r', G_KS, '-b', G_foptd, '-g', G_soptd, '-y', Ghat, '-k')
title('Comparison of Frequency responses')
legend('Process', 'KS FOPTD', 'Skogestad FOPTD', 'Skogestad SOPTD', 'Least squares FOPTD')
```


From the code,

$$K_p = -0.99977$$

$$\tau_1 = 16.508$$

$$\tau_2 = 16.508$$

$$D = 2.1078$$

The transfer function from the least squares approximation is given as

$$G(s) = \frac{-0.99977 e^{-2.1078s}}{(16.508s+1)(16.508s+1)}$$

d) The step and frequency responses for the actual process and the various approximations are generated using MATLAB.

The plots are attached below and the observations for the step response are also tabulated.

Q3)

$$a) \quad G(s) = \frac{2(-s+2)e^{-2s}}{(8s+1)(5s+1)}$$

The Simulink model is attached below.

To select an appropriate sampling time T_s ,

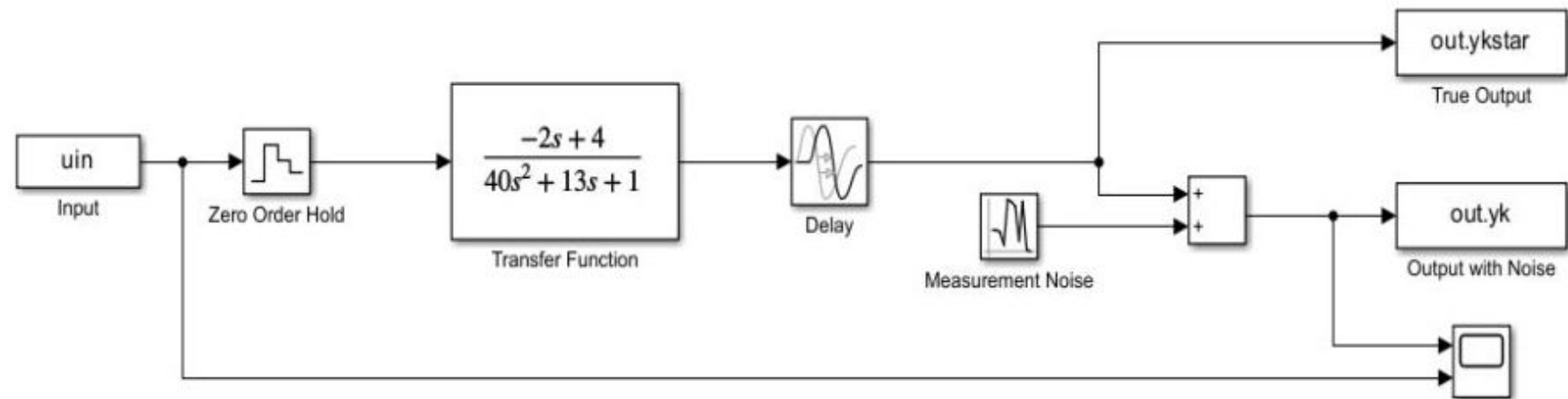
Dominating time const. $\tau = 8$

T_s should be between $\frac{\tau}{5}$ and $\frac{\tau}{10}$ i.e. 1.6 and

0.8

\therefore We will choose $T_s = 1$ sec. as the sampling time interval.

Q3-Part a)
Simulink Model:



Q3-Part c)

%Impulse response model using the training data

```
[ztrain,Tr]=detrend(data_train,0);
```

```
ztest=detrend(data_test,Tr);
```

```
impulse_est= impulseest(ztrain);
```

%Plotting the impusle response

```
figure
```

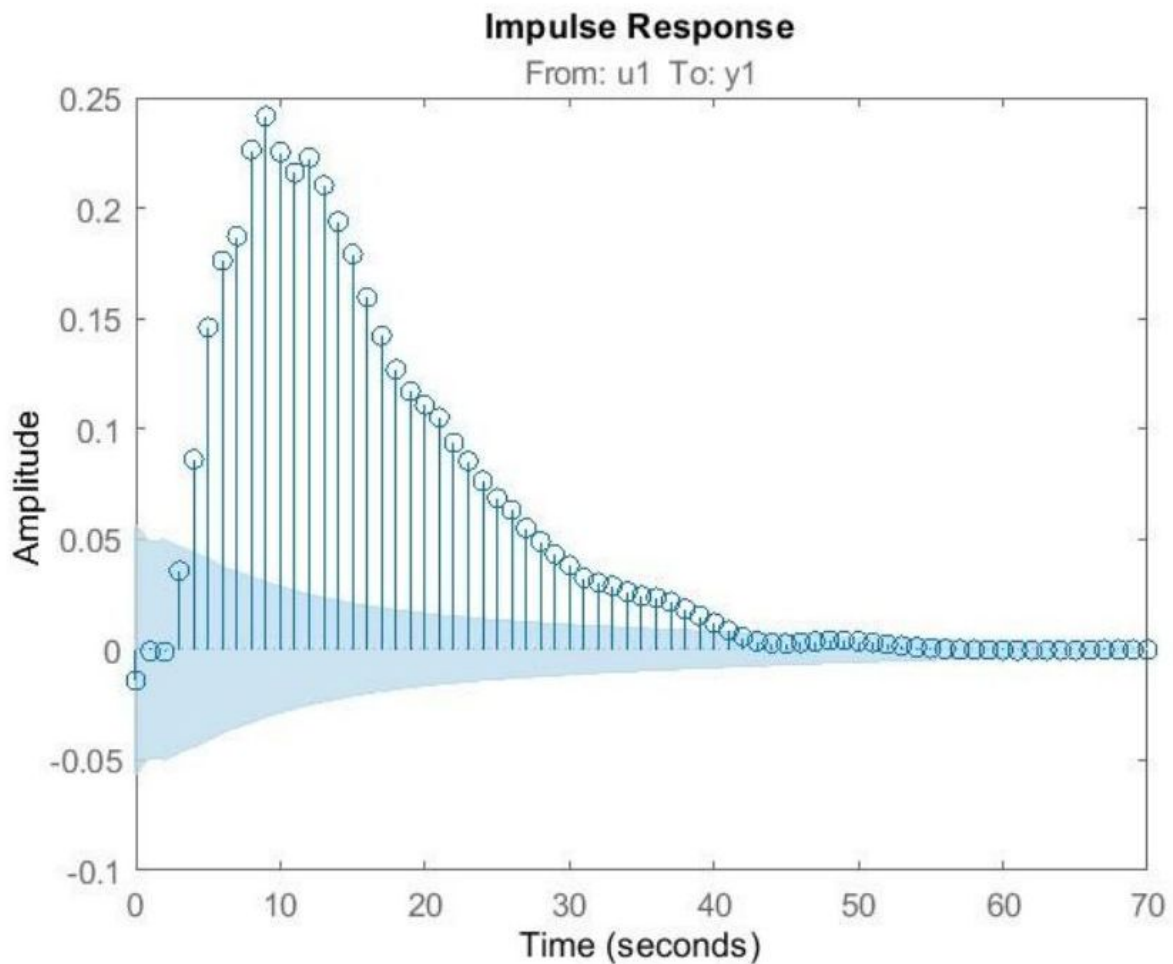
```
impulse(impulse_est,'sd',3);
```

%Plotting the step response based on the estimated impulse response

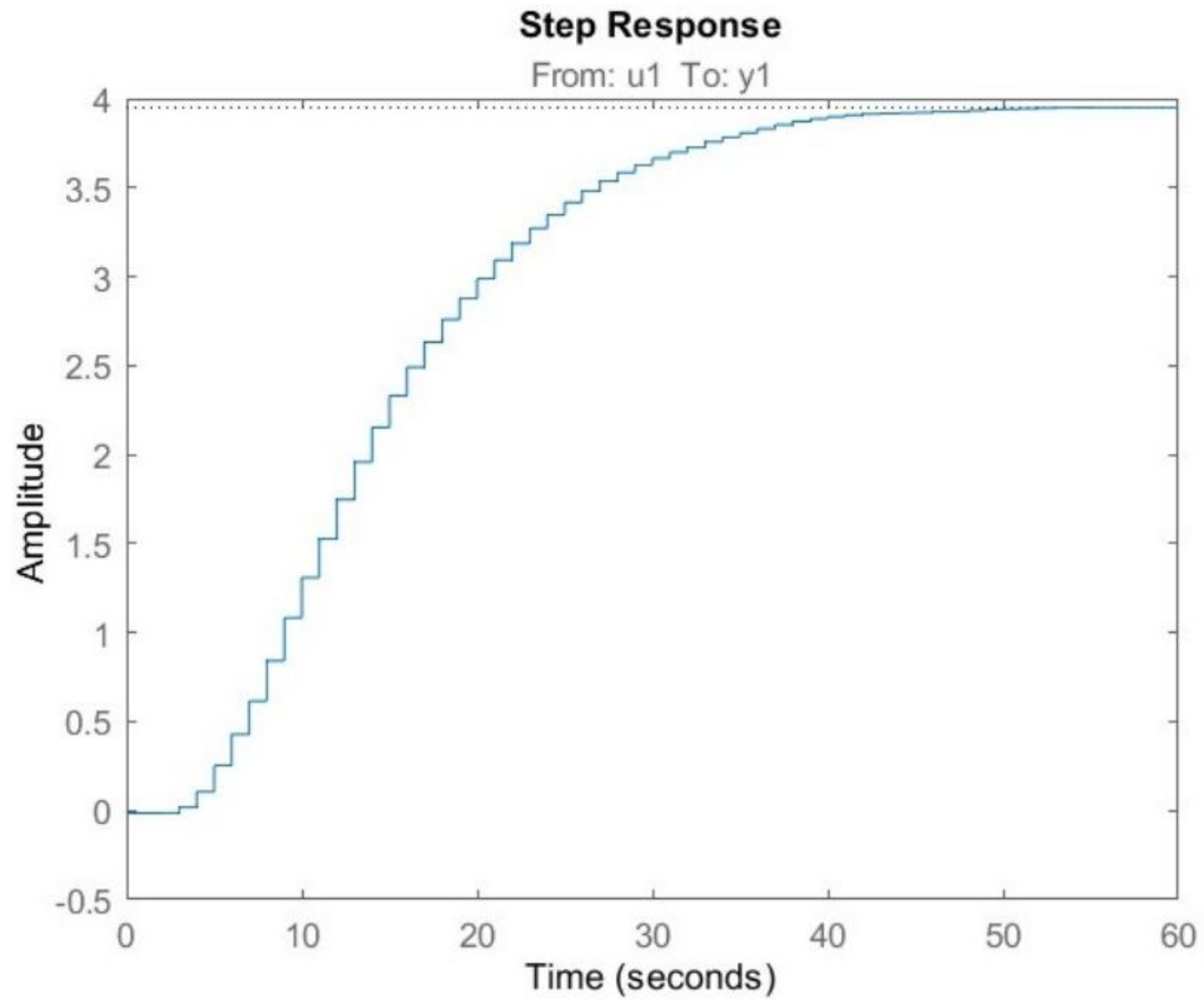
```
figure
```

```
step(impulse_est)
```

Plot of estimated impulse response



Plot of estimated step response



c) From the step response we can see that the system shows a small inverse response initially. So, it has a RHP zero.

From the impulse response we see a delay of 4 samples ie 4 sec as $T_s = 1 \text{ sec}$.

The gain from the step response is around 3.95 which is close to the actual gain of 4.

The time constant assuming a first order system is around 16 seconds. But based on the step response, the system can also be a overdamped second order system.

Q3-Part d)

MATLAB code:

```
%Guesses for the parameters of the OE model
m=2;
n=2;
d=4;

%Estimating the OE model
model_oe=oe(ztrain, [m, n, d]);

%Plotting the cross and auto residual corelations
figure
resid(model_oe, ztrain);

%Getting the estimated parameters for the OE model
present(model_oe);

%Comparison between the repsonse of OE model and actual system response
figure
compare(model_oe, ztest);
```

Model Properties

```
>> present(model_oe)
```

```
model_oe =
```

```
Discrete-time OE model:  $y(t) = [B(z)/F(z)]u(t) + e(t)$ 
```

```
     $B(z) = 0.08617 (+/- 0.009128) z^{-4} - 0.0006899 (+/- 0.01269) z^{-5}$ 
```

```
     $F(z) = 1 - 1.7 (+/- 0.01483) z^{-1} + 0.7215 (+/- 0.01383) z^{-2}$ 
```

```
Sample time: 1 seconds
```

From this the parameters for the OE model are

$a_1 = 1.7$

$a_2 = 0.7215$

$b_1, b_2, b_3 = 0$

$b_4 = 0.08617$

$b_5 = -0.0006899$

Q3-Part e)

Model Properties

```
>> present(model_oe)
```

```
model_oe =
```

```
Discrete-time OE model:  $y(t) = [B(z)/F(z)]u(t) + e(t)$ 
```

```
   $B(z) = 0.08617 \text{ (+/- } 0.009128) z^{-4} - 0.0006899 \text{ (+/- } 0.01269) z^{-5}$ 
```

```
   $F(z) = 1 - 1.7 \text{ (+/- } 0.01483) z^{-1} + 0.7215 \text{ (+/- } 0.01383) z^{-2}$ 
```

```
Sample time: 1 seconds
```

```
Parameterization:
```

```
  Polynomial orders:  nb=2  nf=2  nk=4
```

```
  Number of free coefficients: 4
```

```
  Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.
```

```
Status:
```

```
Termination condition: Near (local) minimum, (norm(g) < tol)..  
Number of iterations: 4, Number of function evaluations: 9
```

```
Estimated using OE on time domain data "ztrain".
```

```
Fit to estimation data: 68.86%
```

```
FPE: 1.297, MSE: 1.289
```

```
More information in model's "Report" property.
```