BY: S. VISHAL
CH18B020

CH3050 PDC ASSIGNMENT-4

1. FOPTD model : $G(s) = \dfrac{K_p e^{-Ds}}{(\tau_p s+1)}$ —— (1)

A step change of μ has been introduced

$\Rightarrow Y(s) = G(s)U(s) = \dfrac{K_p e^{-Ds}\mu}{(\tau_p s+1)(s)}$

$= \mu K_p e^{-Ds}\left(\dfrac{1}{s} - \dfrac{\tau_p}{\tau_p s+1}\right)$ —— (2)

Taking Inverse Laplace transform on both sides

$$y(t) = \begin{cases} \mu K_p (1 - e^{-(t-D)/\tau_p}) & t \geq D \\ 0 & t < D \end{cases}$$ —— (3)

Note that the values of μ & y are deviation variables wrt the initial steady state values.

Step change given at 9:05 AM
First non-zero instant : 9:09 AM.

$\therefore$ Delay = 4 minutes.

Given input = 540 L/min

$\Rightarrow \mu = 540 - 500 = 40$ L/min

~~Out~~ Steadystate temperature $= 55.7°C$

$\Rightarrow y = 55.7 - 50 = \underline{5.7°C}$

Let us assume that ~~at~~ at settling time, response is 95% of the true steadystate response

$t_{settle} = 34 - 5 = \underline{29 \text{ minutes}}$

Substituting the above values in the expression obtained (eqn 3)

$$0.95\, M K_p = M K_p \left(1 - e^{-\left(\frac{t - D}{\tau_P}\right)}\right)$$

$$\Rightarrow \frac{t - D}{\tau_P} = \ln \frac{1}{0.05} \Rightarrow \frac{29 - 4}{\tau_P} = \ln\left(\frac{1}{0.05}\right)$$

$$\Rightarrow \tau_P = \frac{25}{-\ln 0.05} = \underline{8.3452 \text{ minutes}}$$

Also $y_{steadystate} = M K_p$

$$\Rightarrow K_p = \frac{5.7}{40} = \underline{0.1425}$$

$$\therefore \boxed{G_{approx} = \frac{0.1425}{8.34528 + 1} e^{-4s}}$$

Since there were no maxima/minima, it can't be under damped systems; However it can be overdamped 2nd order or higher order system. If operator had noted the rise time we can do a SOPTD approximation.

And now since operator knows the gain

And now, since the operator knows the gain, the person can note down time at which 35%. & 85%. of steady state value is reached, so that we can use Krishnaswamy and Sundaresan's method to get an approximate transfer function

(2) a) Krishnaswamy and Sundaresan's Method.

t_1 : time at which 35.3%. of steady state output is reached ~~reached~~

= 27 units

t_2 : time at which 85.3%. of steady state output is reached

= 65.5 units

From these values,

$D = 1.3t_1 - 0.29t_2$

= 16.105 units

τ = ~~25~~ $0.67(t_2 - t_1)$

= 25.795 units

$K_p = 1$ (identified from the steady state value)

$$\therefore G_{approx} = \frac{K_p e^{-Ds}}{\tau s + 1}$$

$$\Rightarrow \boxed{G_{approx} = \frac{e^{-16.105 s}}{(25.795 s + 1)}}$$

b) Skogestad's half rule Method.

i) FOPTD

$$G_{actual} = \frac{(zs+1)e^{-Ds}}{(\tau_1 s+1)(\tau_2 s+1)(\tau_3 s+1)(\tau_4 s+1)}$$

$$\hat{\tau} = \tau_1 + \frac{\tau_2}{2} \; ; \; \hat{D} = D + \frac{\tau_2}{2} + \tau_3 + \tau_4 - Z$$

$K_p = 1$

$$\therefore G_{approx} = \frac{\cancel{}}{\cancel{27.5s+1}} \quad \boxed{\frac{e^{-13s}}{(27.5s+1)}}$$

ii) SOPTD

$$\hat{\tau}_1 = \tau_1 \; , \; \hat{\tau}_2 = \tau_2 + \frac{\tau_3}{2} \; ; \; K_p = 1$$

$$D = \frac{\tau_3}{2} + \tau_4 + D - Z$$

$$G_{approx} = \frac{e^{-\left(\frac{4}{2} + 0.5 + 3 - 2\right)s}}{(20s+1)\left((15+2)s+1\right)}$$

$$= \boxed{\frac{e^{-3.5s}}{(20s+1)(17s+1)}}$$

c) The magnitude, phase data (bode plots) was generated using MATLAB.

Let $G_{approx} = \dfrac{K_p e^{-Ds}}{(\tau_1 s+1)(\tau_2 s+1)}$

First using the magnitude data least squares was performed such that $\left(\| G_{approx} \| - \text{actual magnitude} \right)$ is minimised in a least square sense.

$$|G_{approx}(j\omega)| = \frac{K_p}{\sqrt{((\tau_1\omega)^2+1)((\tau_2\omega)^2+1)}} \quad \left(\because |e^{-Dj\omega}| = 1 \right)$$

$K_p = 1.0003$, $\tau_1 = 17.911$, $\tau_2 = 17.9335$

was found.

Next a least squares fit was done for $\cos(\phi)$

ϕ_{actual} is from data; $\phi_{approx} = \phi(G(j\omega))$

$$= \text{phase}\left(\frac{K_p}{(\tau_1 j\omega+1)(\tau_2 j\omega+1)} - D \right)$$

where D should be estimated.

It was estimated as $D = 1.006$

$$\therefore \boxed{G_{approx} = \frac{e^{-1.01s}}{(17.911s+1)(17.9335+1)}}$$

Question 2) part d)

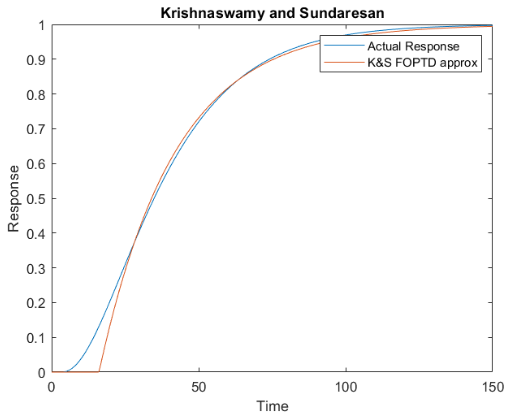

Figure 2.1: Comparing actual response with Krishnaswamy and Sundaresan approximation

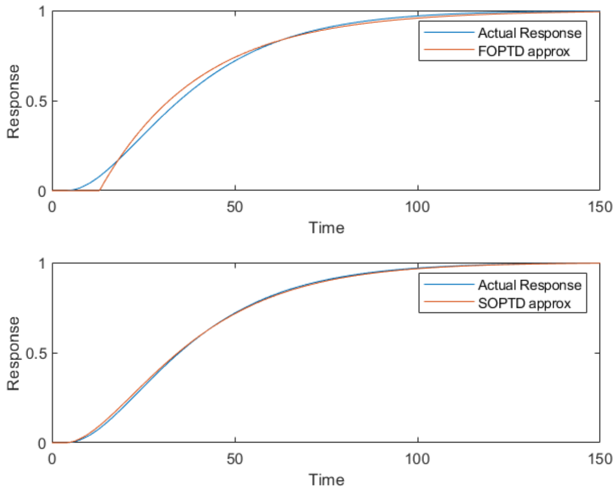

Figure 2.2: Comparing actual response with Skogestad's half point approximations

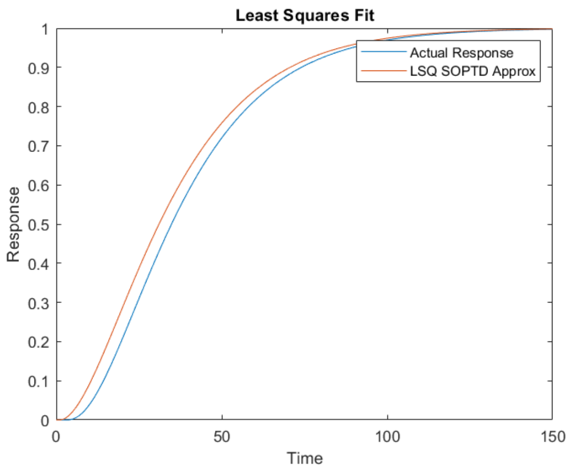

Figure 2.3: Comparing actual response with Frequency domain Least Squares approximation

Observations:

- All approximations capture the gain properly.
- As expected KS method will reach the 35% and 85% mark at the same time as the true process reaches.
- Both the FOPTD models pushed most of the higher order sluggishness as a delay, so the model response starts a long time after the true response of the process begins
- Least Squares has captured the delay most appropriately in the step response plots. This could be because we did a least squares fit to estimate delay alone.
- Comparing the step response plots, Skoegstad's half point SOPTD model is the best model (least MSE)

Questions 3)

Part a) SIMULINK DIAGRAM

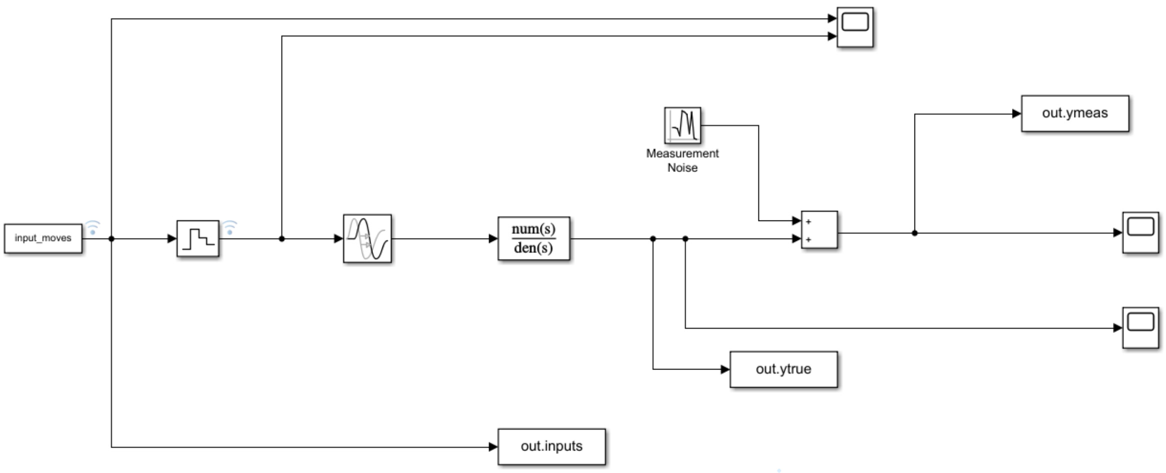

Figure 3.1: The SIMULINK diagram. Sampler was incorporated in the 'To Workspace' block

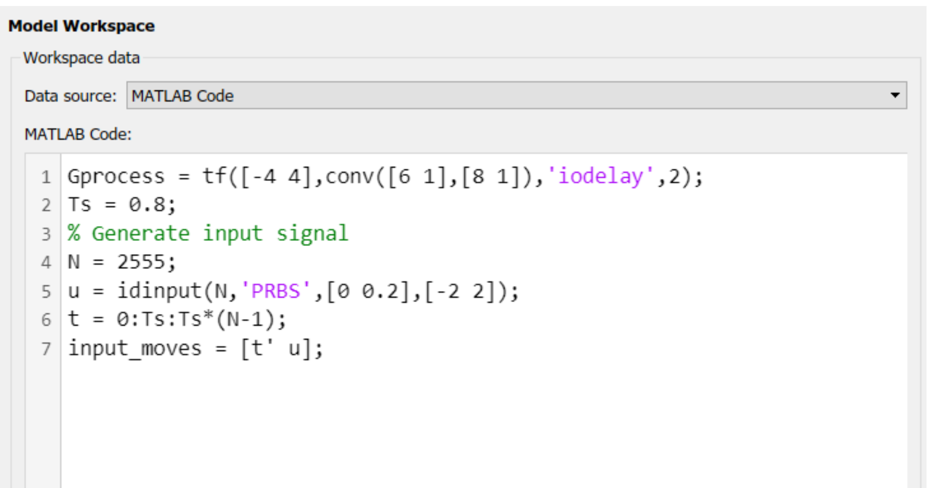

Figure 3.2: Workspace variables

Part b) Designing input and getting output

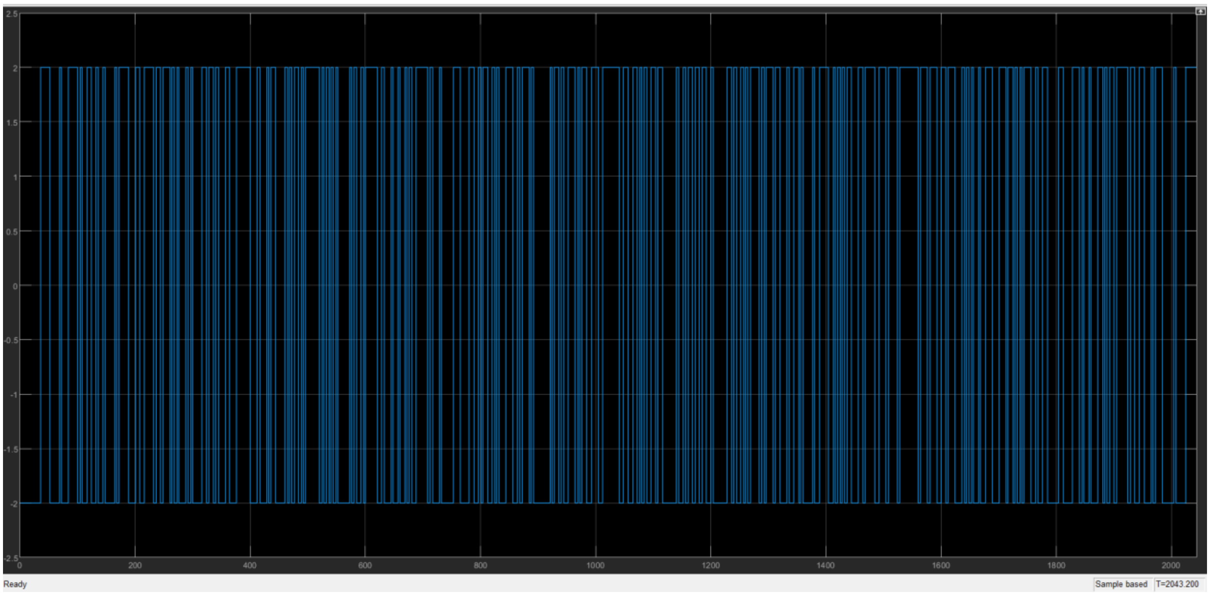

Figure 3.3: PRBS inputs after ZOH

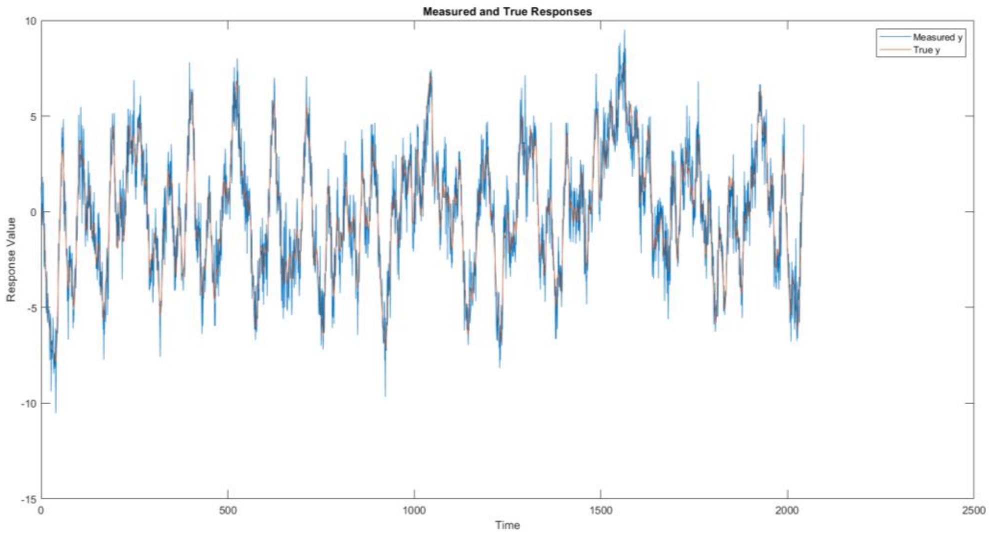

Figure 3.4: Measured and True responses

Part c) Response based model

Finite impulse response is built as depicted in live class session. The *'impulseest'* routine was used for that purpose.

From figure 3.5 (Impulse response) we can see that there only the response at the fifth time instant is significant. So there is a delay of **4 steps**. (In time units, it is 4*0.8 = 3.2s)

From figure 3.6 (Step response), gain was obtained to be 3.96 units.

Initially we see an inverse response (figure 3.5, and figure 3.7), this is a characteristic of a **zero**. There are no maxima/minima/oscillations in the step response, neither there is much sluggishness (only a delay is there). So for simplicity we can assume that the denominator is either a first order or second order polynomial. Therefore, it is a **first** or **second** order process with **a zero** and a **delay of 4 units**. Note that a delay of 4 units implies inputs only upto u[k-5] will affect the output at kth instant!

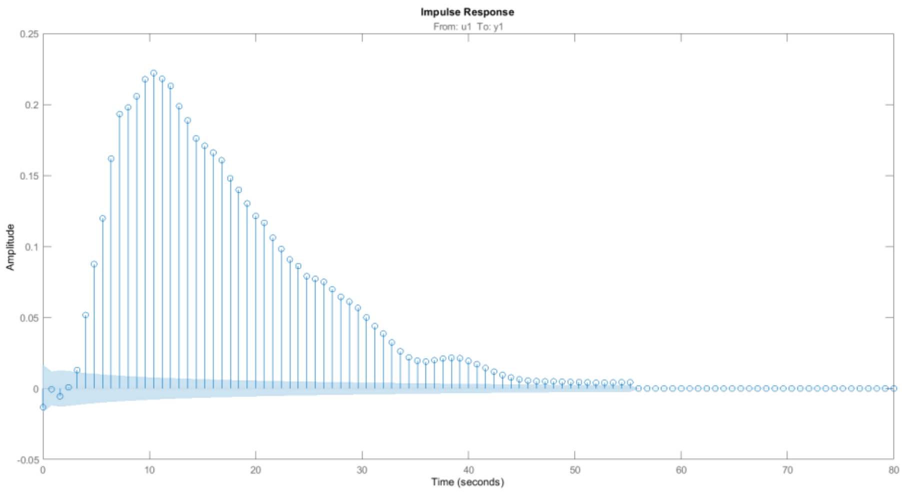

Figure 3.5: The finite impulse response model

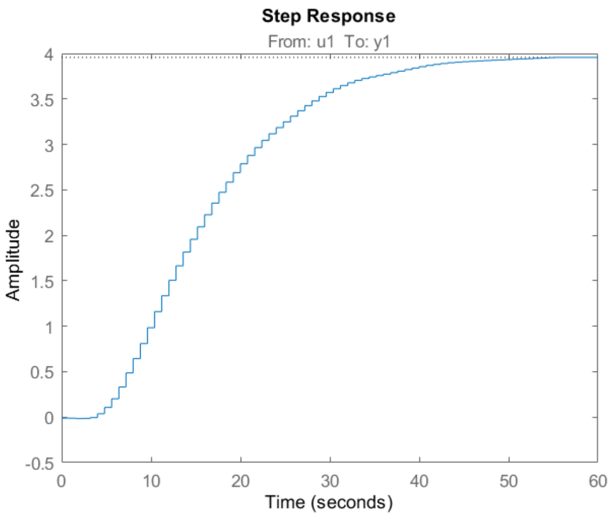

Figure 3.6: Step response obtained from the FIR model

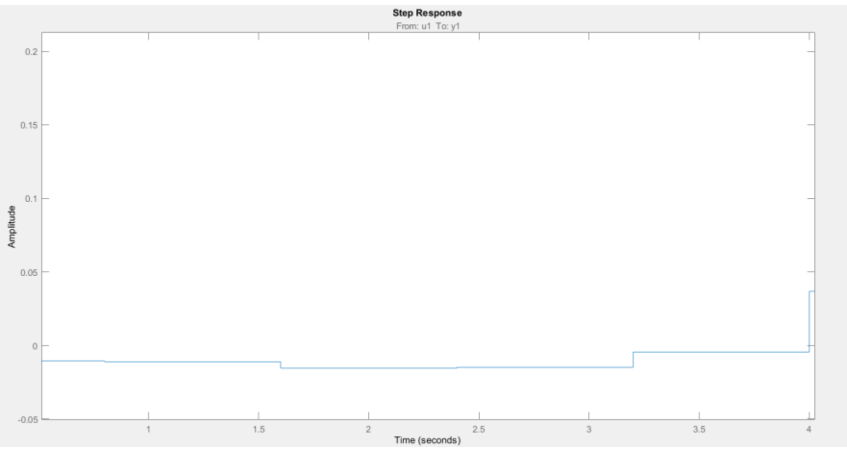

Figure 3.7: Initial inverted response

Part d) Parameterized model

First try: d = 5, m = 5, n =1

$$y[k] - 0.9593 * y[k-1] = 0.1921 * u[k-5]$$

Fit to estimation data: 54.21%

FPE: 2.108, MSE: 2.102

We see that the fit % is not very good.

Part e) Testing for underfit and over fit

To test for underfit we examine the residuals. Residuals should not be correlated among themselves (auto correlation) or correlated to the input (cross correlation). (Note that correlated to the input will take care of correlation to the output, because output itself is correlated to input.)

First try: d = 5, m = 5, n =1

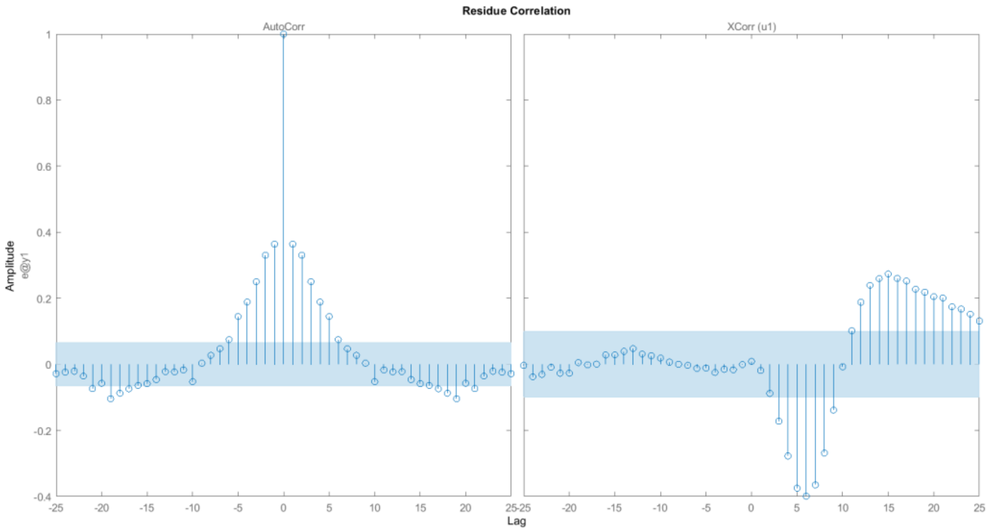

Figure 3.7 Residual correlations. (obtained using 'resid' routine)

We see that the auto correlations as well as the cross correlations are significant at multiple lags. So this model is an underfit!

Second try: d = 5, m = 1, n =2

Let's fit a model following second order dynamics.

$$y[k] - 1.784 * y[k-1] + 0.7951 * y[k-2] = 0.04677 * u[k-5]$$

Fit to estimation data: 64.54%

FPE: 1.266, MSE: 1.261

Both MSE and fit % have improved!

Let's examine the auto-correlations. From figure 3.8 we see that the cross and auto-correlation of residuals at all non-zero lags are zero!

This means that the new model is **not an underfit**!

Parameter estimates:

$a_1 = -1.784 \pm 0.00626$ $\quad a_2 = +0.7951 \pm 0.00604$

$b_5 = +0.04677 \pm 0.001276$

None of the confident intervals contain zero in them,

=> All parameters of the model are significant and hence the model is **not an overfit**!

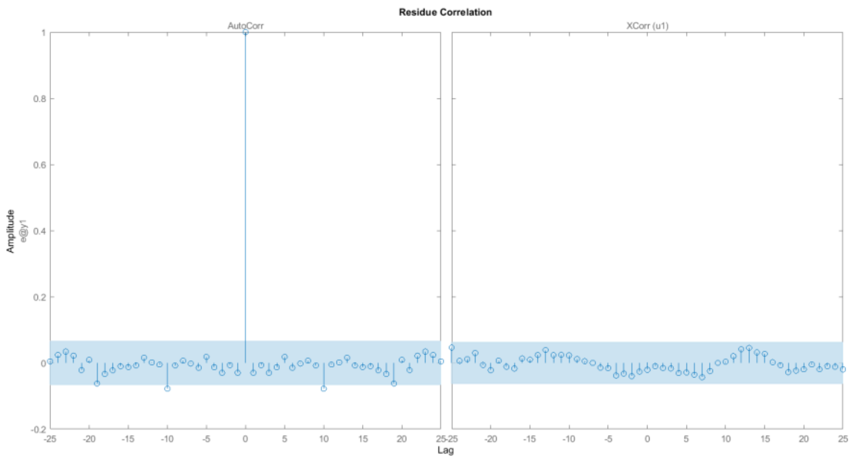

Figure 3.8 Residual correlations of the new model. (obtained using 'resid' routine)

Gain comparison with non-parametric model

In part c) we obtained a gain of 3.96. In this case we obtain a gain of 4.03 (refer figure 3.9). Both are close to the actual value of 4.

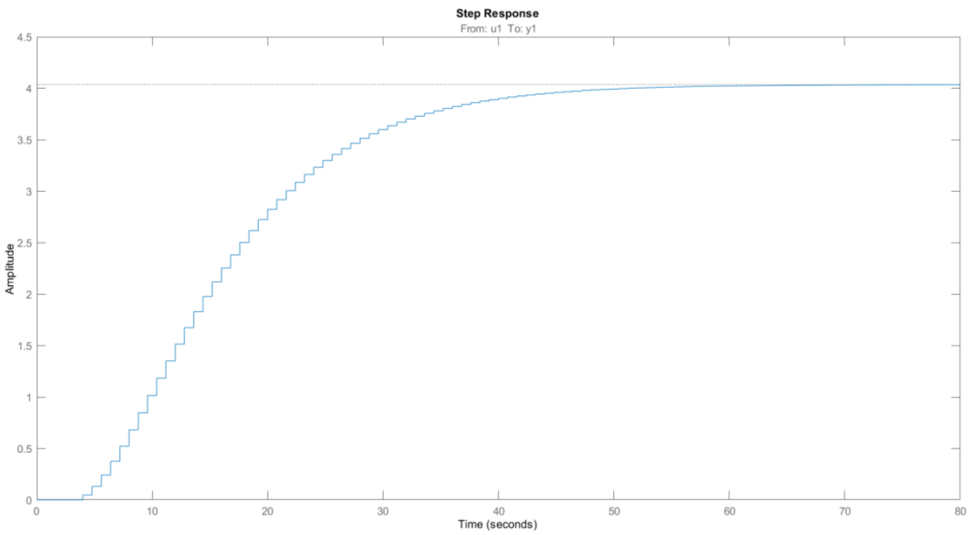

Figure 3.9: Step response the parametric model

Part f) Cross validation

Cross validation was performed using the compare routine (on the test data). The model had a Percentage Fit of about 65%.

Also, note that data has a certain amount of measurement noise, so we don't want a model that exactly mimics the data (Because it will end up including the effects of noise too.).

The model captures the trend of the data successfully!

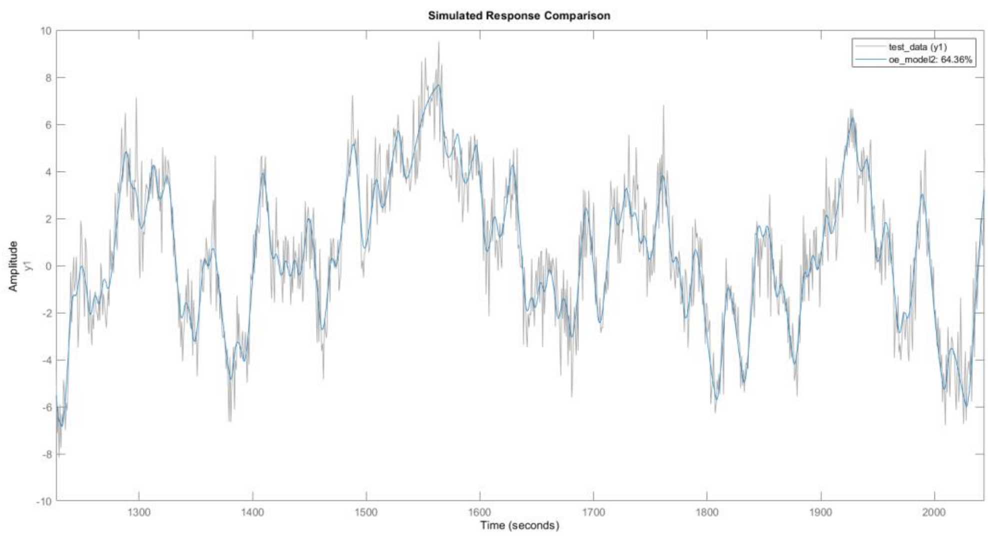

Figure 3.9: Cross validation using the test data

Final model:

$$y[k] - 1.784 * y[k-1] + 0.7951 * y[k-2] = 0.04677 * u[k-5]$$

```
oe_model2 =
Discrete-time OE model: y(t) = [B(z)/F(z)]u(t) + e(t)
  B(z) = 0.04677 (+/- 0.001276) z^-5

  F(z) = 1 - 1.784 (+/- 0.006257) z^-1 + 0.7951 (+/- 0.006045) z^-2

Sample time: 0.8 seconds

Parameterization:
   Polynomial orders:   nb=1   nf=2   nk=5
   Number of free coefficients: 3
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 5, Number of function evaluations: 12

Estimated using OE on time domain data "train_data".
Fit to estimation data: 64.54%
FPE: 1.266, MSE: 1.261
```

Figure 3.10: Final model details obtained using 'present' routine.

BY S.VISHAL
CH18B020

MATLAB codes

Question-2

```
clear; close all;
%% Simulating the actual process
G = tf([2 1],conv(conv(conv([20,1],[15,1]),[4 1]),[0.5 1]),'InputDelay',3);
t = 0:0.5:150;
[Yactual,Tactual] = step(G,t);
yfinal = 1;
%% Krishnaswamy and Sundaresan's method
[val,loc]= min(abs(Yactual-yfinal*0.3531));
t1 = Tactual(loc);
[val2,loc2] = min(abs(Yactual-yfinal*0.8531));
t2 = Tactual(loc2);
Dks = 1.3*t1 - 0.29*t2;
tauks = 0.67*(t2-t1);
G_ks = tf(1,[tauks 1],'InputDelay',Dks);
[Yks,Tks] = step(G_ks,t);
%% Skogestad's half rule method
% FOPTD
tau_s = 27.5; D_s = 13;
G_sk1 = tf(1,[tau_s 1],'InputDelay',D_s);
[Ysk1,Tsk1] = step(G_sk1,t);
% SOPTD
tau_1= 20; tau_2 = 17; D_s2 = 3.5;
G_sk2 = tf(1,conv([tau_1 1],[tau_2 1]),'InputDelay',D_s2);
[Ysk2,Tsk2] = step(G_sk2,t);
%% Least Squares (Frequency Domain)
[MAG,PHASE,W] = bode(G);
mpar = lsqcurvefit(@(mpar,wdata) magpred(mpar,wdata),[1 1 1]',W,squeeze(MAG));
Kp = mpar(1); tau1 = mpar(2); tau2 = mpar(3);
Delay = lsqcurvefit(@(D,wdata) phasepred(D,wdata,Kp,tau1,tau2),1,W,cos(squeeze(PHASE)));
Glsq = tf(Kp,conv([tau1 1],[tau2 1]),'InputDelay',Delay);
[Ylsq,Tlsq] = step(Glsq,t);
%% Compare step responses
% K&S
figure();
plot(Tactual,Yactual,Tks,Yks); xlabel('Time'); ylabel('Response');
title('Krishnaswamy and Sundaresan');
legend('Actual Response','K&S FOPTD approx');
% Half Rule
figure();
title("Skogestad's Half Rule");
subplot(2,1,1);
plot(Tactual,Yactual,Tsk1,Ysk1); xlabel('Time'); ylabel('Response');
legend('Actual Response','FOPTD approx');
subplot(2,1,2);
plot(Tactual,Yactual,Tsk2,Ysk2); xlabel('Time'); ylabel('Response');
legend('Actual Response','SOPTD approx');
% Least Squares in Frequency Domain
figure();
```

```
plot(Tactual,Yactual,Tlsq,Ylsq); xlabel('Time'); ylabel('Response');
title('Least Squares Fit');
legend('Actual Response','LSQ SOPTD Approx');
%% Functions for least squares fit
function Mag = magpred(mpar,wdata)
   Kp = mpar(1); tau1 = mpar(2);tau2 = mpar(3);
   Mag = Kp./sqrt(1+(tau1.*wdata).^2)./sqrt(1+(tau2.*wdata).^2);
end
function Phase = phasepred(D,wdata,Kp,tau1,tau2)
   Gw = Kp./(1+tau1*(1j*wdata))./(1+tau2*(1j*wdata));
   Phase = cos(phase(Gw)-D*wdata);
end
```

Question-5

```
clear; close all;
Ts = 0.8;
N = 2555;
%% Getting the data and splitting to train and test
open_system('model');
out = sim('model');
% Get the input and output
tk = out.ymeas.time;
uk = out.inputs.Data;
yk = out.ymeas.Data;
% Splitting the data: 60% train, rest 40% test
% tf is always in terms of deviation variables so we dont need to do any
% additional processing
ntrain = 0.6*N;
dataset = iddata(yk,uk,Ts);
train_data = dataset(1:ntrain);
test_data = dataset(ntrain+1:end);
%% Non parametric estimation
options = impulseestOptions;
options.Advanced.AROrder = 0;
fir_model = impulseest(train_data ,options);
[Y,T] = impulse(fir_model);
figure;
impulse(fir_model);
figure;
step(fir_model);
%% Parametric Estimation
d = 4;
% One zero because of inverse response
nb = 1;
% Looks overdamped so second order but can also be first. Trying both
nf1 = 1; nf2 = 2;
% First 4 responses are negligible
nk = 5;
oe_model1 = oe(train_data,[nb nf1 nk]);
figure;
resid(oe_model1,train_data);
```

```
oe_model2 = oe(train_data,[nb nf2 nk]);
figure;
resid(oe_model2,train_data);
% nf = 2 is not an underfit
% Check for overfit
present(oe_model2);
%% Cross Validation
figure;
compare(oe_model2,test_data);
figure;
step(oe_model2);
```