CH3050: Process Dynamics and Control

Assignment 4 Solutions

Question 1

$F_{i0} = 500$ l/min, $F_{i1} = 540$ l/min, $\Delta F_i = 40$ l/min

$T_0 = 50$ °C, $T_3 = 55.7$ °C, $\Delta T = 5.7$ °C

t_1 = 0 (9:05 am), t_2 = 4 (9:09 am), t_3 = 29 (9:34 am)

Flow rate changes from F_{i0} to F_{i1} at time t_1

No change in temperature (T) is observed till time t_2

Response in temperature is quite rapid, slowing down gradually until it appears to reach a steady-state value ($T_3 = 55.7$) . No change is observed after t_3 .

Based on these observations A FOPTD process would be a good choice to model the TF

$\theta = t_2 - t_1 = 4$

$t_3 \approx 5\tau$

$\tau = \frac{29-4}{5} = 5$

$K_p = \frac{\Delta T}{\Delta F_i} = \frac{5.7}{40} = 0.1425$

$$\therefore \quad \hat{G}(s) = \frac{K_p e^{-\theta s}}{\tau s + 1} = \frac{0.1425\, e^{-4s}}{5s+1}$$

To obtain a better estimate the operator can try the following:

- Record the temperature at more frequent intervals to get a better understaing of the step response.
- Obtain the impulse response and frequency response of the system. This will help in obtaining a qualitative guess of the process order

Question 2

The process is given by the transfer function:

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

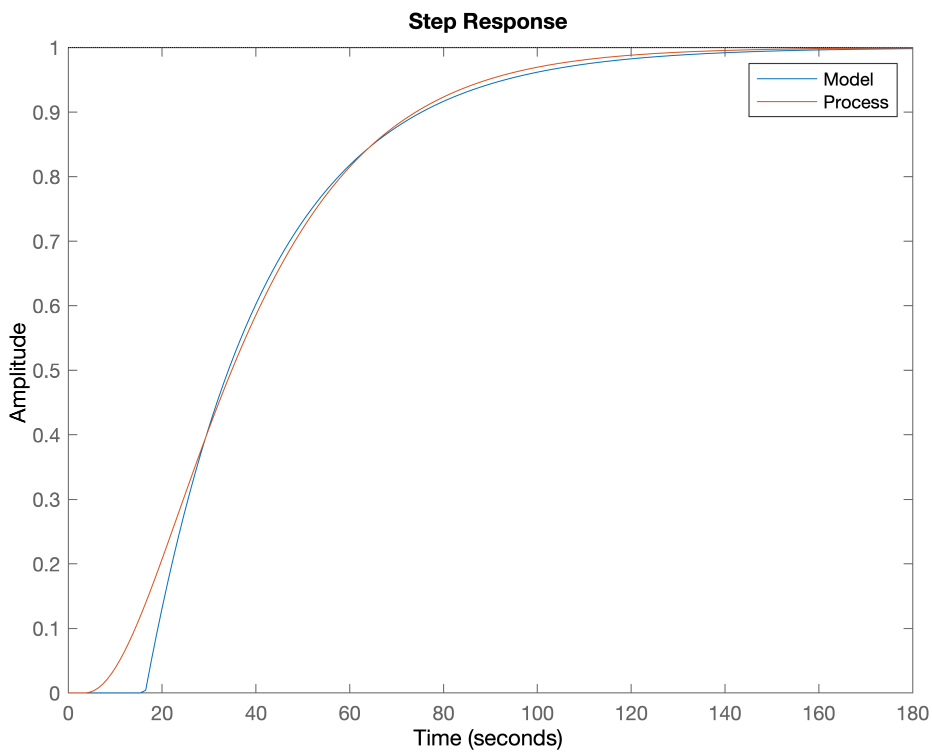
```
s = tf('s');
```

```
Gs = (2*s+1)*exp(-3*s)/((20*s+1)*(15*s+1)*(4*s+1)*(0.5*s+1));
```

Part a

To get Krishnaswamy and Sundaresan's FOPTD approximation

```
gain = 1;
[ystep, ts] = step(Gs, 0:0.2:200);
[~, t1_ind] = min(abs(0.353*gain-ystep));
[~, t2_ind] = min(abs(0.853*gain-ystep));
t1 = ts(t1_ind);
t2 = ts(t2_ind);

D = 1.3*t1-0.29*t2;
tau = 0.67*(t2-t1);
K = gain;

G_1_ks = K*exp(-D*s)/(tau*s+1);
step(G_1_ks, Gs)
legend('Model', 'Process')
```



Thus the approximate model is:

```
G_1_ks
```

```
G_1_ks =

                        1
  exp(-16.4*s) * -----------
                 25.59 s + 1
```

```
Continuous-time transfer function.
```

Part b

The time constants in descending order are: 20, 15, 4, 0.5.

For FOPTD, we retain only one time constant. Thus,

$\tau = 20 + 0.5 * 15 = 27.5$
$D = 0.5 * 15 + 4 + 0.5 - 2 + 3 = 10$
$K = 1$

Thus, the FOPTD approximation is

$$G_{\text{FOPTD}} = \frac{e^{-13s}}{27.5s + 1}$$

```
G_1_s = exp(-13*s)/(27.5*s+1);
```

For SOPTD, we retain two time constants. Thus,

$\tau_1 = 20$
$\tau_2 = 15 + 0.5 * 4 = 17$
$D = 0.5 * 4 + 0.5 - 2 + 3 = 3.5$
$K = 1$

$$G_{\text{SOPTD}}(s) = \frac{e^{-3.5s}}{(20s+1)(17s+1)} = \frac{e^{-3.5s}}{340s^2 + 37s + 1}$$

```
G_2_s = exp(-3.5*s)/(340*s^2+37*s+1);
```

Part c

```
[Gmag,Gphase,wvec] = bode(Gs); % Simulate frequency response
mpar = lsqcurvefit(@(mpar, wdata) magpred(mpar, wdata), [1 15 15], wvec, squeeze(Gmag))
```

```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

```
% magpred is defined at the end of the assignment in the Function Section
G_2_freq = mpar(1)/((mpar(2)*s+1)*(mpar(3)*s+1));

Gph = squeeze(Gphase);
D0 = 4.8; % Obtain initial guess after looking at final function value for different gu
options = optimoptions('lsqcurvefit', 'FunctionTolerance', 1e-10, 'MaxIterations', 100(
```

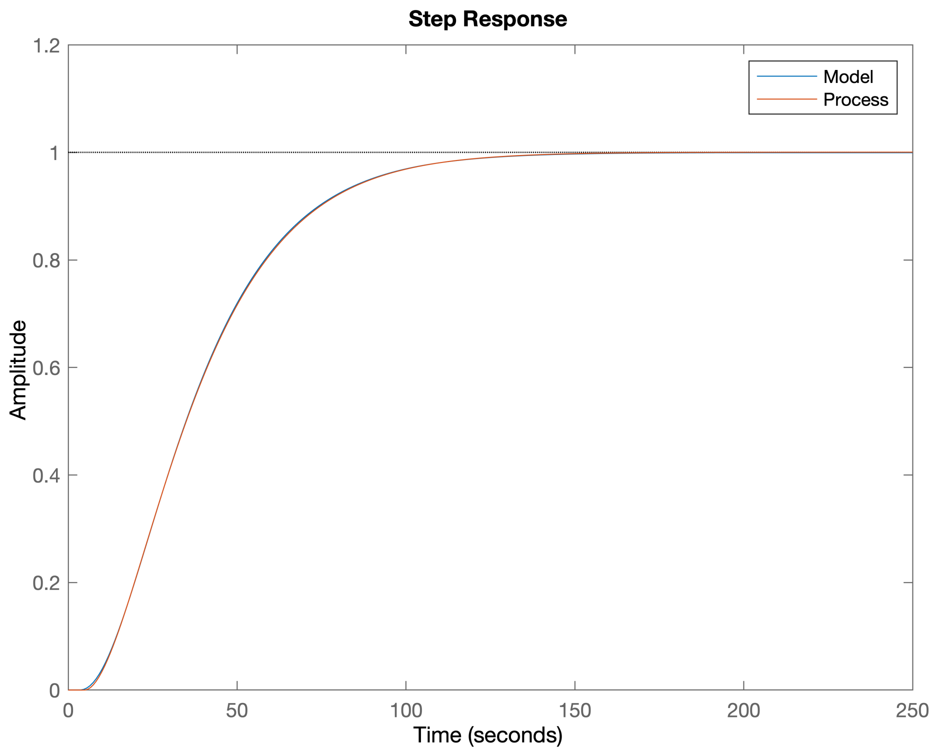
```
options =
  lsqcurvefit options:
```

```
Options used by current Algorithm ('trust-region-reflective'):
(Other available algorithms: 'levenberg-marquardt')

Set properties:
                  Display: 'iter'
        FunctionTolerance: 1.0000e-10
            MaxIterations: 1000
      OptimalityTolerance: 1.0000e-10

Default properties:
                Algorithm: 'trust-region-reflective'
           CheckGradients: 0
 FiniteDifferenceStepSize: 'sqrt(eps)'
     FiniteDifferenceType: 'forward'
      JacobianMultiplyFcn: []
   MaxFunctionEvaluations: '100*numberOfVariables'
                OutputFcn: []
                  PlotFcn: []
 SpecifyObjectiveGradient: 0
            StepTolerance: 1.0000e-06
     SubproblemAlgorithm: 'factorization'
                 TypicalX: 'ones(numberOfVariables,1)'
              UseParallel: 0
```

```
D = lsqcurvefit(@(d,wdata) phasepred(d,wdata,mpar(1),mpar(2),mpar(3)),D0,wvec,cos(Gph),
```

```
                                         Norm of      First-order
 Iteration  Func-count     f(x)          step          optimality
     0          2         38.4654                      4.13e+04
     1          4         37.7205    3.21454e-05       1.06e+04
     2          6         37.6589    1.02309e-05        2.8e+03
     3          8         37.6541     2.9888e-06            756
     4         10         37.6537    8.34506e-07            206

Local minimum possible.
lsqcurvefit stopped because the size of the current step is less than
the value of the step size tolerance.

<stopping criteria details>
```

```
G_2_freq.iodelay = D;

figure
step(Gs, G_2_freq)
legend('Model', 'Process')
```



The SOPTD approximation is

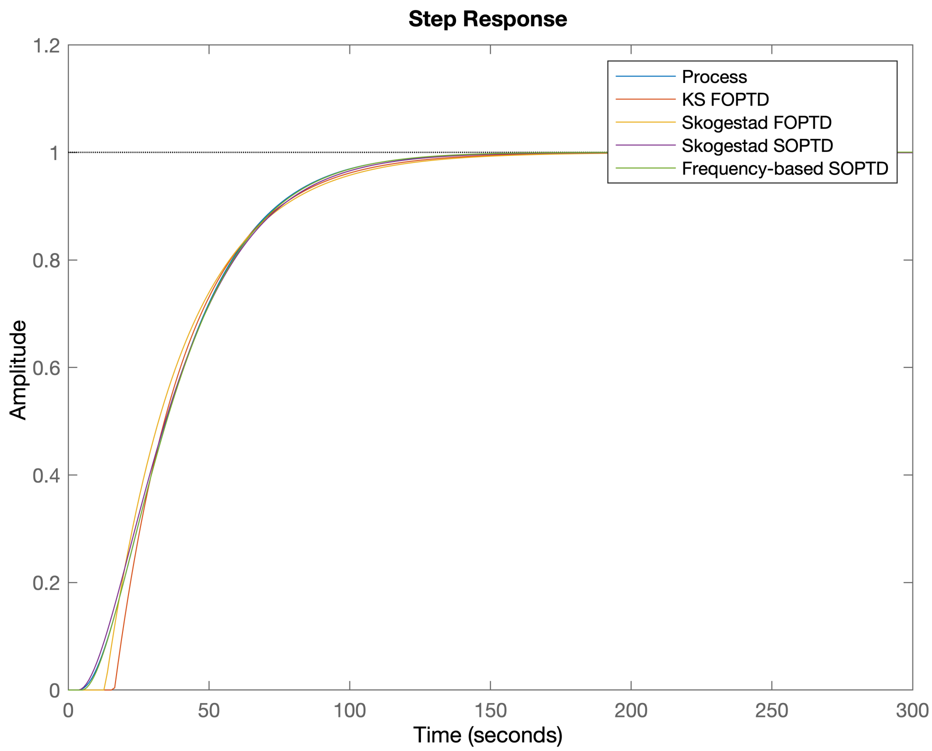
```
G_2_freq
```

```
G_2_freq =

                       1.001
  exp(-4.8*s) * -----------------------
                322.8 s^2 + 35.93 s + 1

Continuous-time transfer function.
```

We observe that the answers from Skogestad's method and the frequency-domain least sqaures methods are close.

Part d

```
step(Gs, G_1_ks, G_1_s, G_2_s, G_2_freq)
legend('Process', 'KS FOPTD', 'Skogestad FOPTD', 'Skogestad SOPTD', 'Frequency-based SO
```



The SOPTD models are able to best capture the transient characteristics of the process. The frequency-based method approximates the delay closer than the Skogestad SOPTD. Both the FOPTD models have slower settling than the process. The Krishnaswamy and Sundaresan's method is unable to get a good approximation of the delay but captures the rest of the step response characteristics well when compared to the Skogestad FOPTD model.

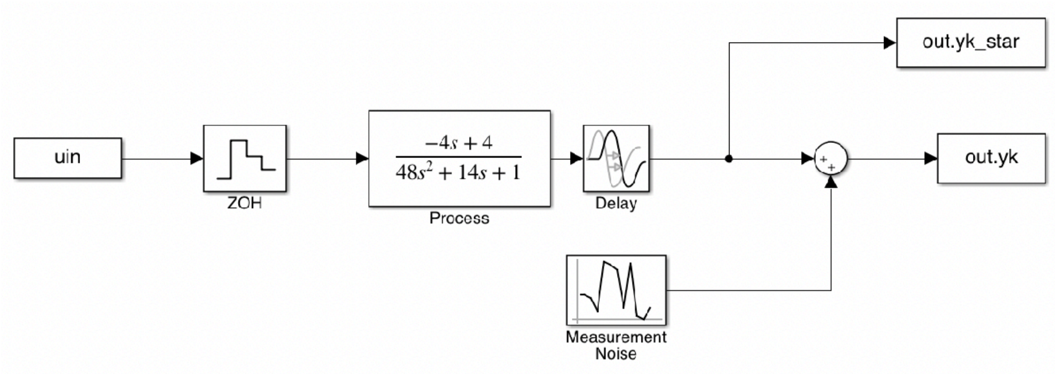
Model	**Delay**	**Gain**	**Settling Time**
KS FOPTD	Highest	Matched	Longer
Skogestad FOPT	Higher	Matched	Longest
Skogestad SOPTD	Shorter	Matched	Longer
Frequency – domain SOPTD	Close Match	Matched	Close Macth

Model	K	τ_1	τ_2	θ
KS FOPTD	1	25.59		16.4
Skogestad FOPT	1	27.5		13
Skogestad SOPTD	1	20	17	3.5
Frequency – domain SOPTD	1	17.9662	17.9662	4.8

Question 3

Part a

The SIMULINK setup is shown below:



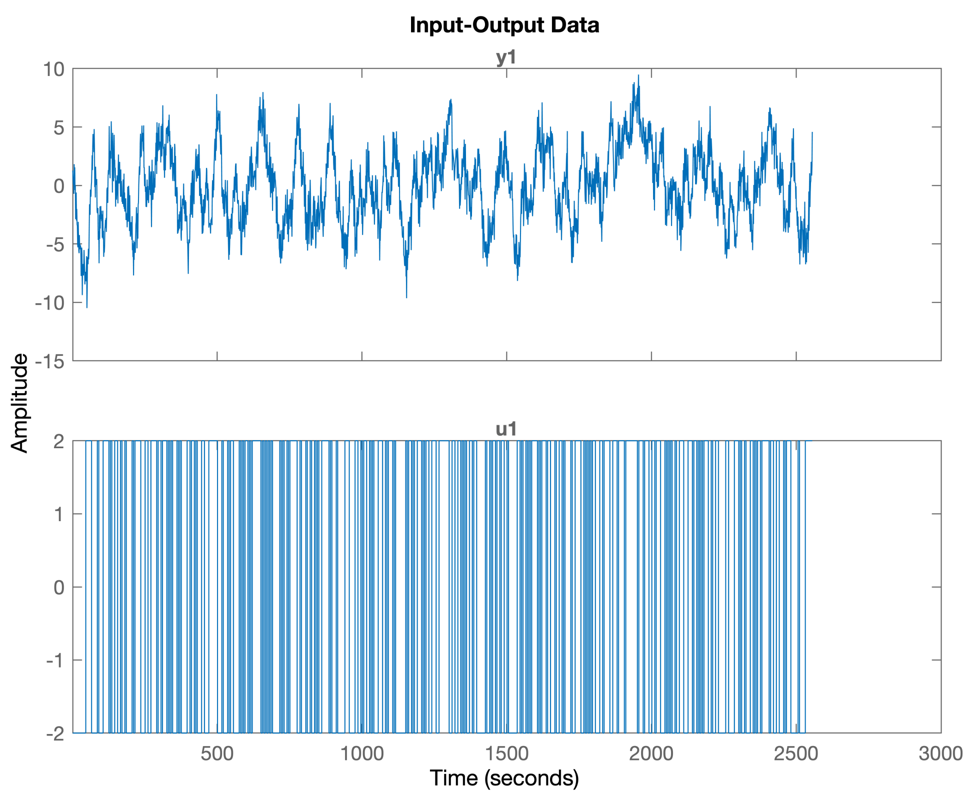
Part b

The input is designed using the code below:

```
B_max = 1/5;
Ts= 0.8; % sampling time
usig = idinput(2555,'prbs',[0 B_max],[-2 2]);
uin = [(0:1:length(usig)-1)'*Ts (usig)];
```

The input and output has been plotted below:

```
data=iddata(out.yk(1:length(usig)),uin(:, 2),1);
plot(data)
```



We partition the data into train and test

```
data_train=data(1:1300);
data_test=data(1300:end);
```

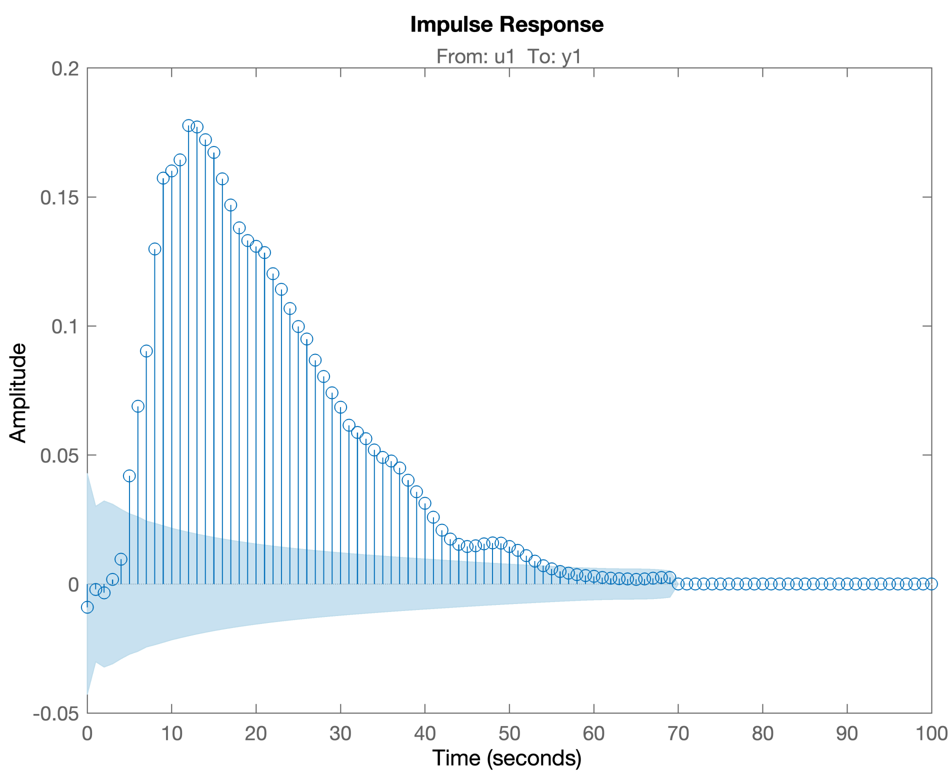
Part c

We build impulse-response model for the process based on the training data.

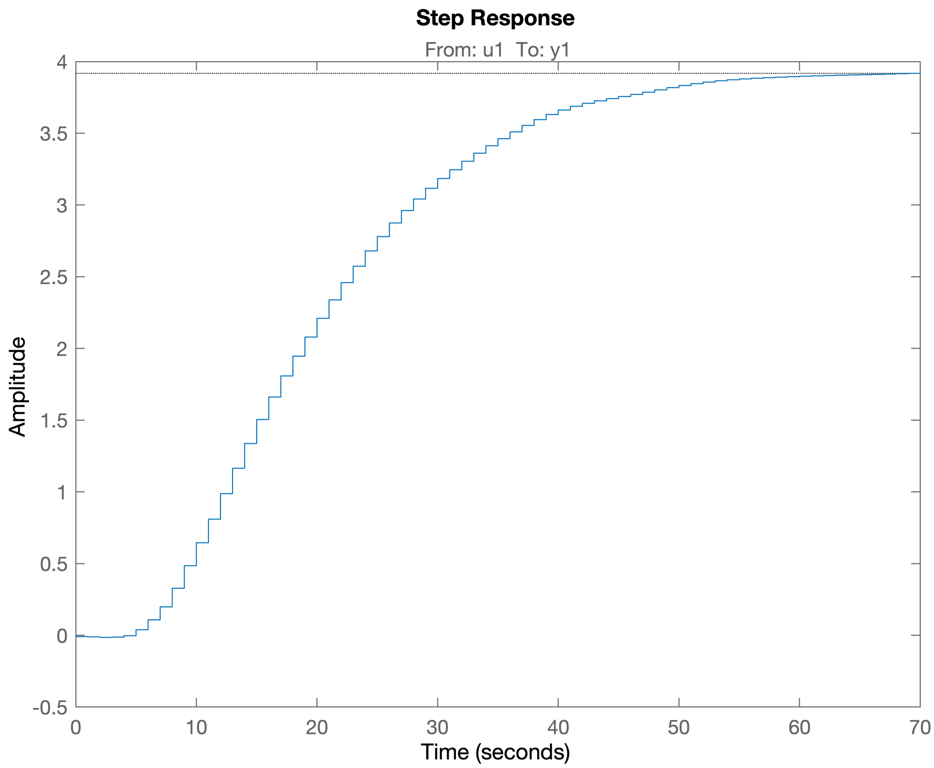
```
[ztrain,Tr]=detrend(data_train,0);
ztest=detrend(data_test,Tr);
impulse_est= impulseest(ztrain);
```

Let us plot the impulse and step responses:

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




```
figure
step(impulse_est)
```



We observe from both the impulse and step response plots an I/O delay of 4. So, we set **d=4**. The step response indicates either a 1st order or an over-damped 2nd order system. The impulse response shows the characteristics of an overdamped 2nd order system. So, we set **n=2**. We also observe a small inverse response and predict a RHP zero. Hence, we set **m=2**.

```
% Estimates
m=2;
n=2;
d=4;
```

Part d

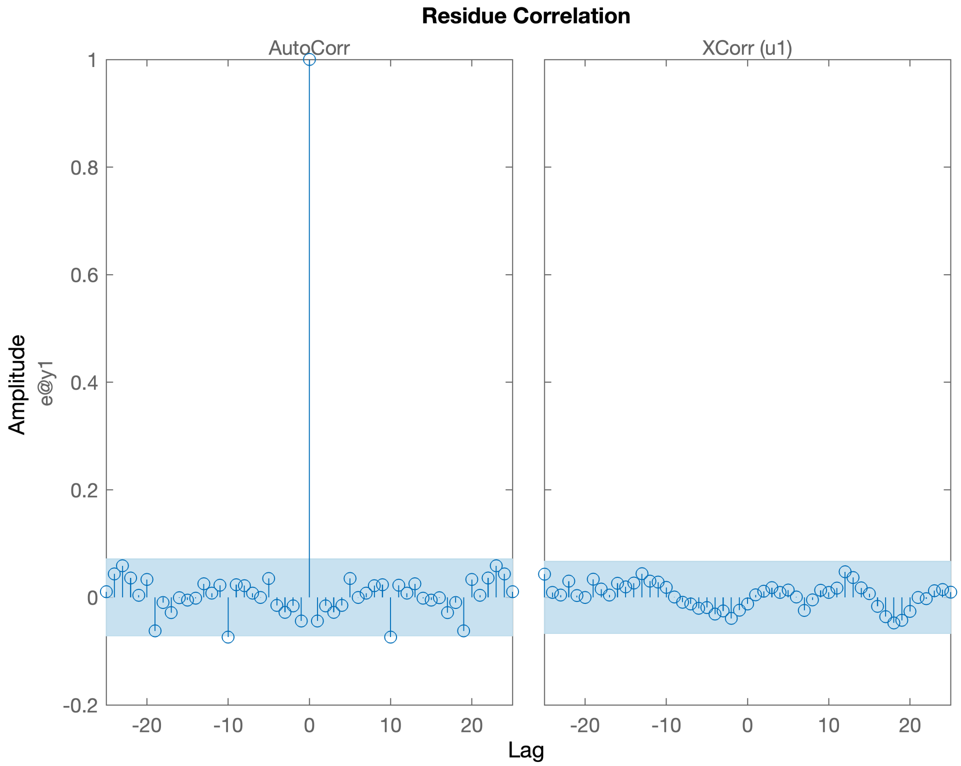
We estimate the oe model for the data

```
model_oe=oe(ztrain, [m, n, d]);
```

Part e

We perform residual analysis of the model

```
figure
resid(model_oe, ztrain);
```



We observe the residuals are uncorrelated with their past as well as with the input. Thus, the model does not have any **underfit.**

We now check the parameter estimates:

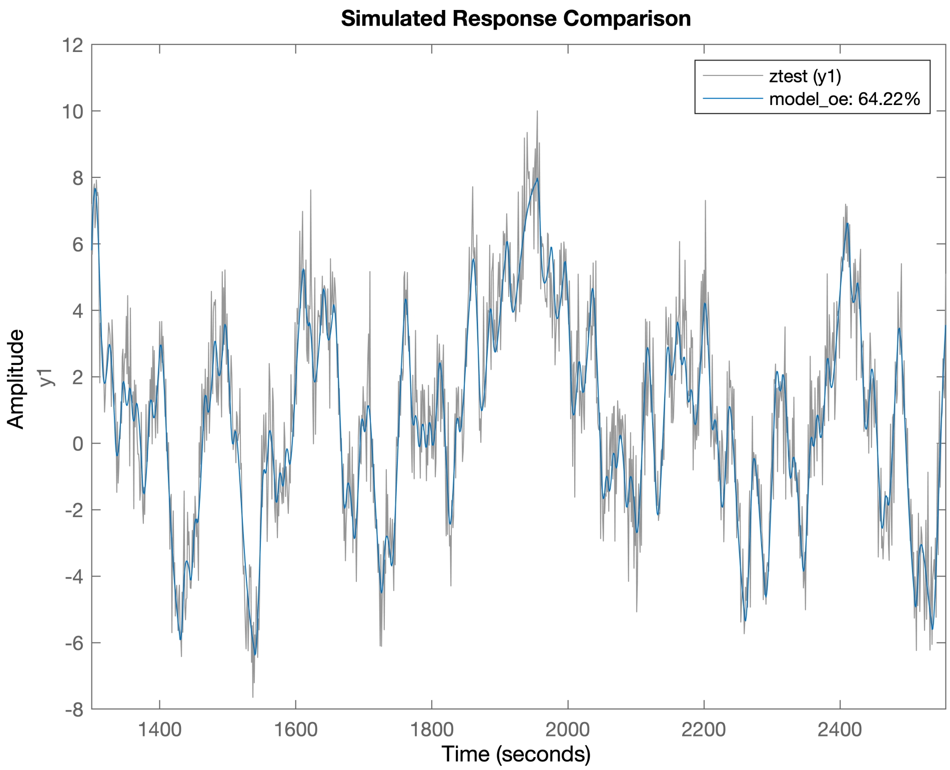
```
present(model_oe);
```

```
model_oe =
Discrete-time OE model: y(t) = [B(z)/F(z)]u(t) + e(t)
  B(z) = -0.0139 (+/- 0.007537) z^-4 + 0.06244 (+/- 0.009032) z^-5

  F(z) = 1 - 1.775 (+/- 0.009493) z^-1 + 0.7871 (+/- 0.009084) 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: 64.12%
FPE: 1.302, MSE: 1.294
More information in model's "Report" property.
```

We observe that the error in estimates is small. Also, the error bounds do not include 0. Thus our estimates are good and do not have any **overfit**.

We compare the model estimates on the test data:

```
figure
compare(model_oe, ztest);
```



We observe that the predictions are good. Thus, our model is satisfactory.

Part f

The final model obtained is:

$$y^*[k] - 1.775y^*[k-1] + 0.7871y^*[k-2] = -0.0139u[k-4] + 0.06244u[k-5]$$

FUNCTIONS

```
function Gwhat = magpred(mpar,wdata)
    Km = mpar(1); tau1 = mpar(2); tau2 = mpar(3);
    s = tf('s');
    Gm = Km/((tau1*s+1)*(tau2*s+1));
    [Gmag,~,~] = bode(Gm, wdata);
    Gwhat = squeeze(Gmag);
end

function Gwhat = phasepred(d,wdata,Km,tau1,tau2)
    s = tf('s');
```

```
    Gm = Km/((tau1*s+1)*(tau2*s+1));
    Gm.ioDelay = d;
    [~,Gphase] = bode(Gm, wdata);
    Gwhat = cos(squeeze(Gphase));
end
```