System Identification Tool (MATLAB) - Problem Statement

Generate time input and output time history data to be used in the Sys ID tool to create approximate models using just that data. Use a unit step input occurring at 0.01 seconds, with a simulation stop time of 20 seconds and a fixed step solver with step size of 0.01 seconds. Generate two variables, input and output needed by the tool.

$$y(t) = \frac{-s^2 - 0.9s + 1}{s^3 + 1.5s^2 + 1.5s + 1} u(t)$$

Problem 1) Polynomial functions ARX/ARMAX (30 points)
Use the System Identification App/toolbox to create input/output models of the data generated for the system described above. Here, though, you only need the input (the step function) and the output of the second order system.

a) Use the system identification app to create an ARX model of the system dynamics with the Focus set to Simulation. When you import the data, make sure to specify the begin time (0) and the sample time (0.01 sec). Then, choose to estimate a polynomial model, ARX structure. You get to pick the order.
b) Extract the A(z) and B(z) polynomials from the model and compare those polynomials with the result you find by using c2d to convert the original s-plane function. They should be close for at least the A(z) expression. The B(z) might be different, but let's see what the simulation looks like in part c).
c) Note that, in Simulink, you can put a polynomial model into an Idmodel block, found under the System Identification Toolbox pallet in Simulink. Export your model from the Sys ID tool for part c) to MATLAB and use the Idmodel block in Simulink to implement the approximated model. Compare the time histories.
d) Now add the zero mean, 0.001 variance noise to the output and use that data to repeat step a) and step c). What does noise do to the ability to produce an accurate model of the system? Feel free to play with the tool settings to make the model as good as you can get it. You can also try the ARMAX setting with various order models.

Problem 2) Transfer Function Approximation (30 points)
For the same system as above, use the System Identification app to identify a transfer function.

a) Use the system identification app to create third order transfer function denominator, second order numerator using the data without noise.
b) Compare the poles and zeros of the identified transfer function with the true values.
c) Export the model and use the Idmodel block to import the identified transfer function into Simulink. Compare the time histories of true vs identified.

d) Now repeat steps a), b), and c) when using the output which has the random noise added. What does noise do to the ability to produce an accurate model of the system in this case?

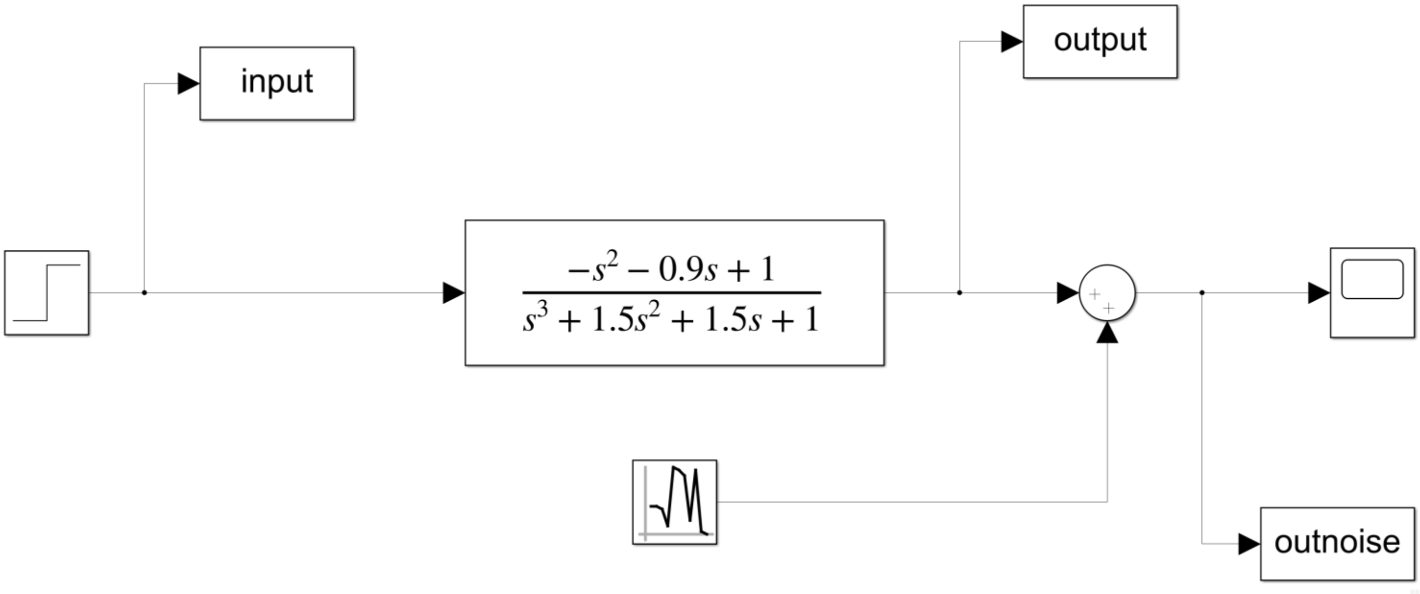
Problem 3) State Space ID (30 points)
Repeat the process as in Problem 1, but this time identify a state space model.

a) Use the system identification app to create a state space model using the data without noise. You might have to turn off the 'Allow unstable models' checkbox under the Estimation Options section of the GUI and set the Focus to Simulation.
b) Compare the poles and zeros of the identified model with the true values.
c) Export the model and use the Idmodel block to import the identified state space system into Simulink. Compare the time histories of true vs identified.
d) Now repeat steps a), b), and c) when using the output which has the random noise added. Feel free to experiment with the different tool settings to try to make the model as good as you can get it.
e) What does noise do to the problem, i.e. does it help or hurt?

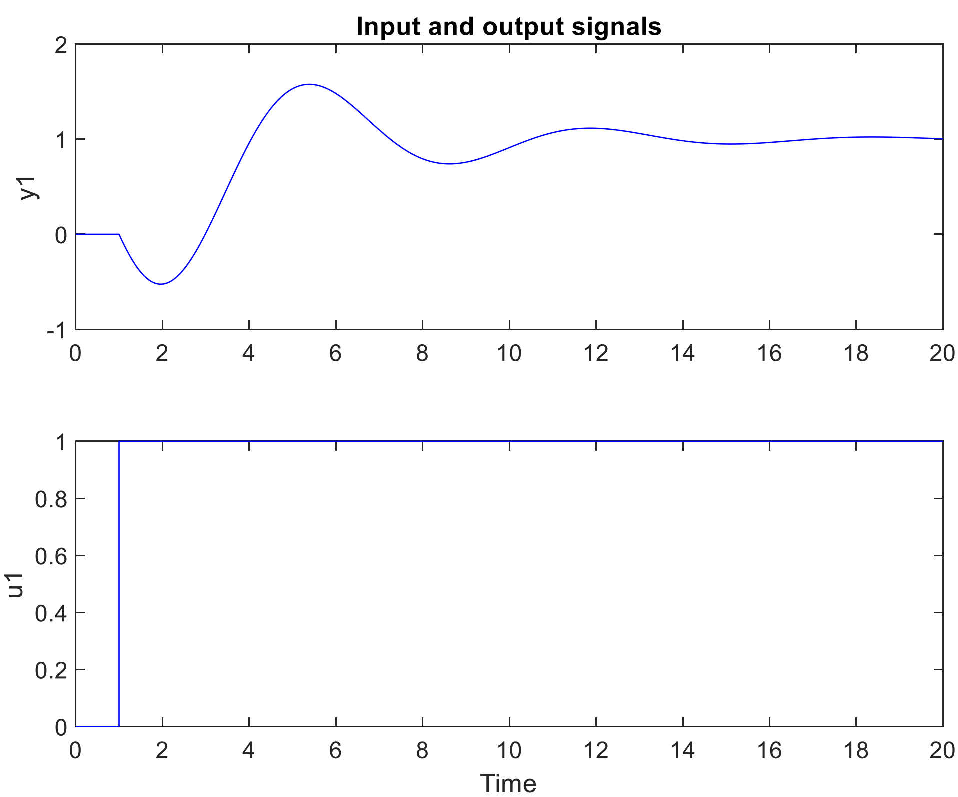
Problem 4) Discussion (10 points)
Comment on which of the three methods you tried above produced the best model fit for the noise-free case and for the noisy data case. Which would you probably use if you had to develop a model from data that has noise to it? Note this is probably problem-dependent so it would be best to try different methods to see what works best for your situation.

Problem – 1

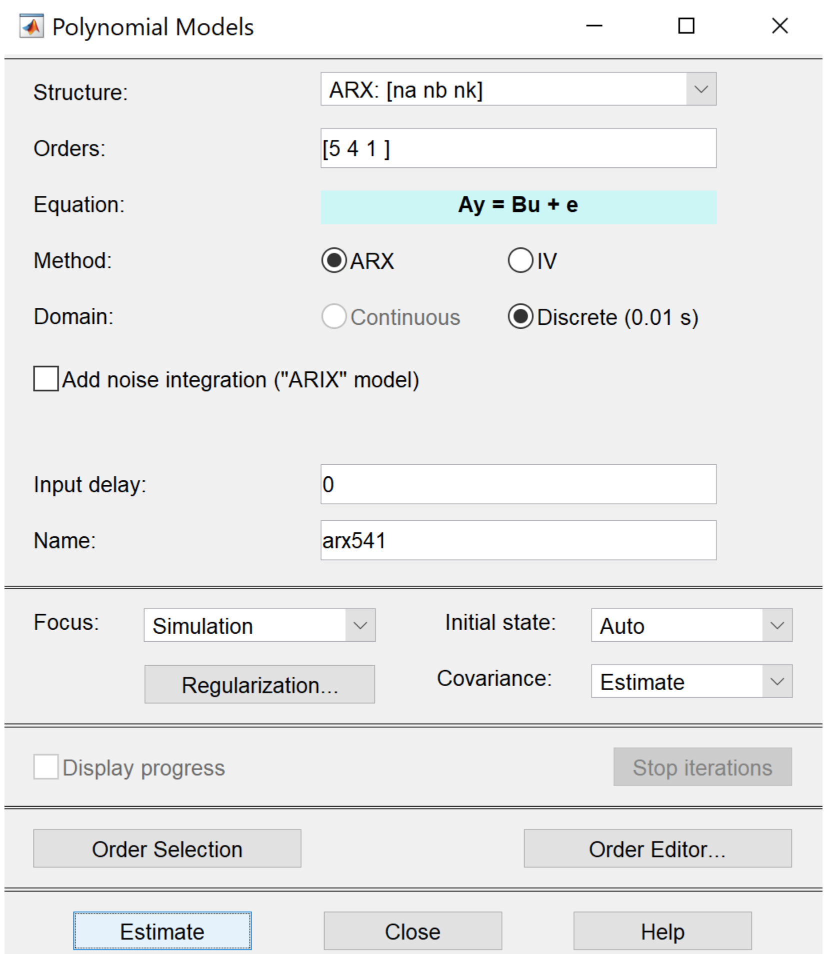
a.

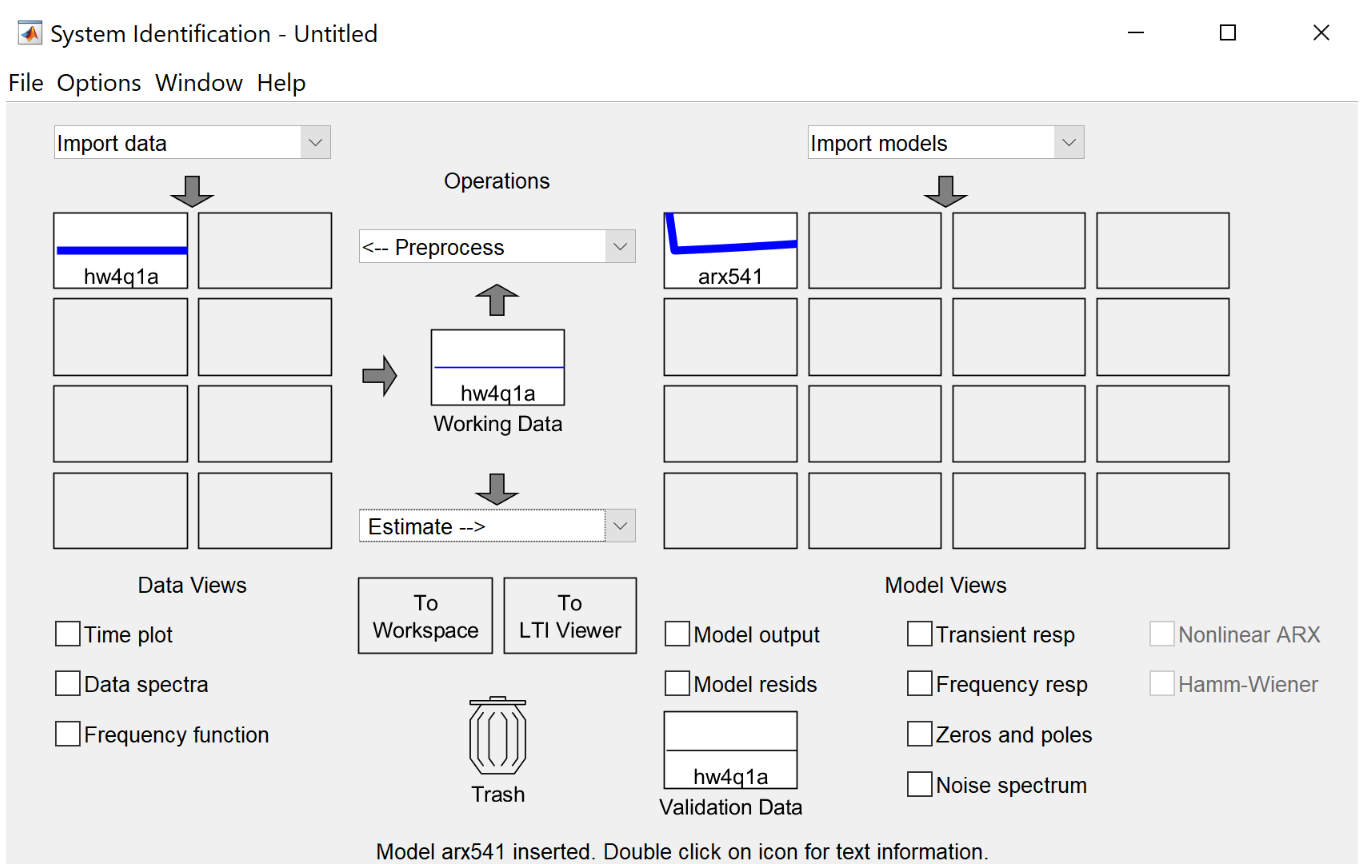


Simulink Diagram

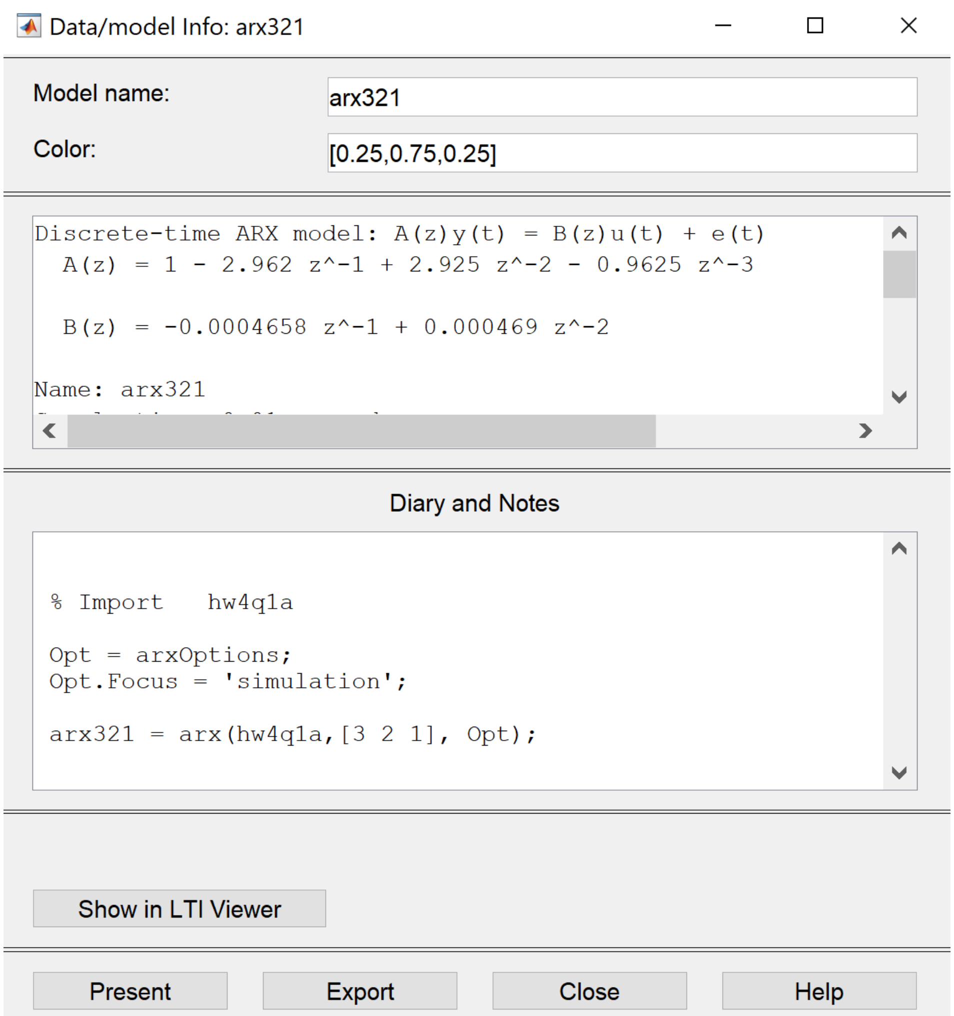


Plots for Input and Output signals





Importing data and estimating the polynomial function

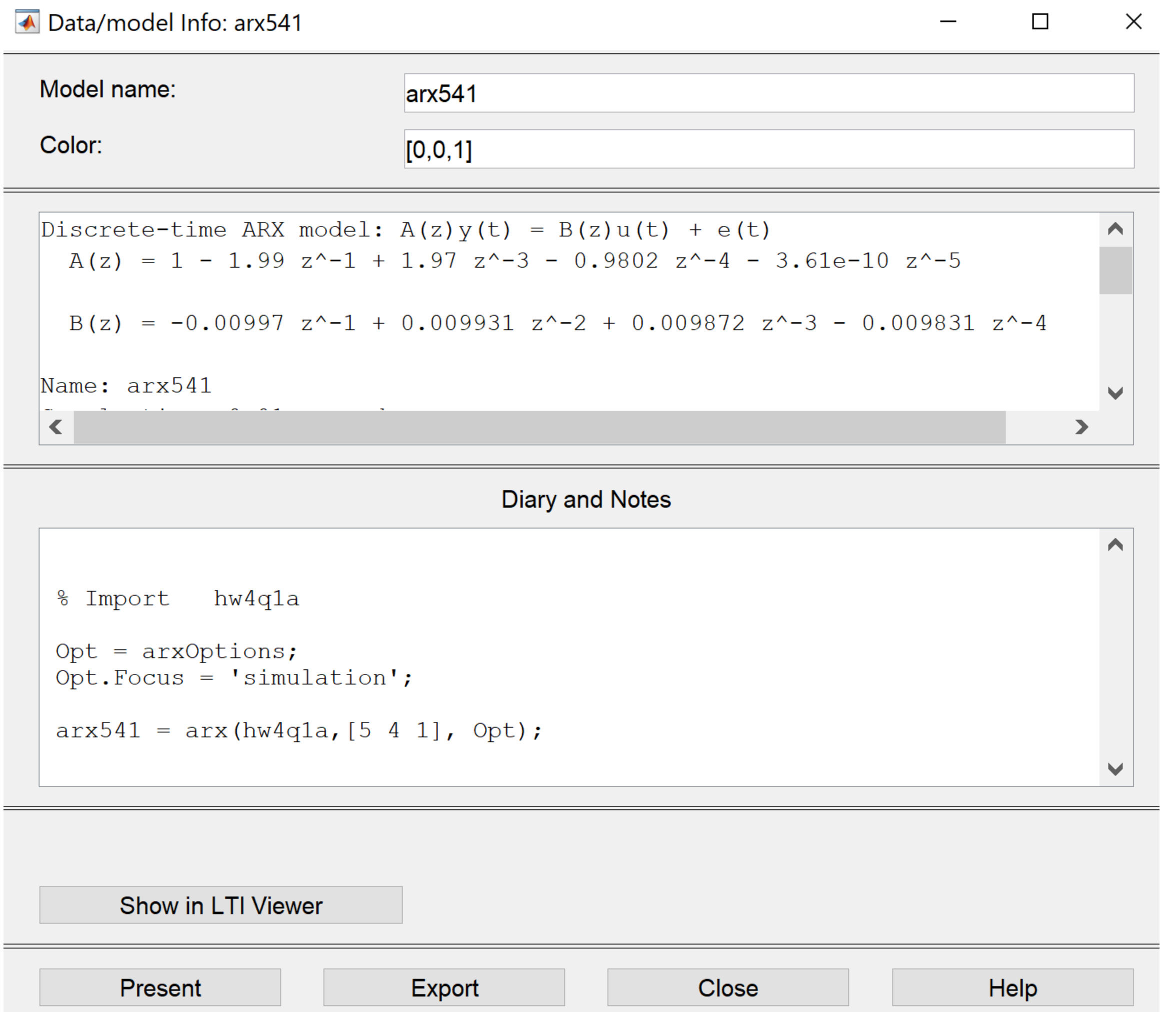


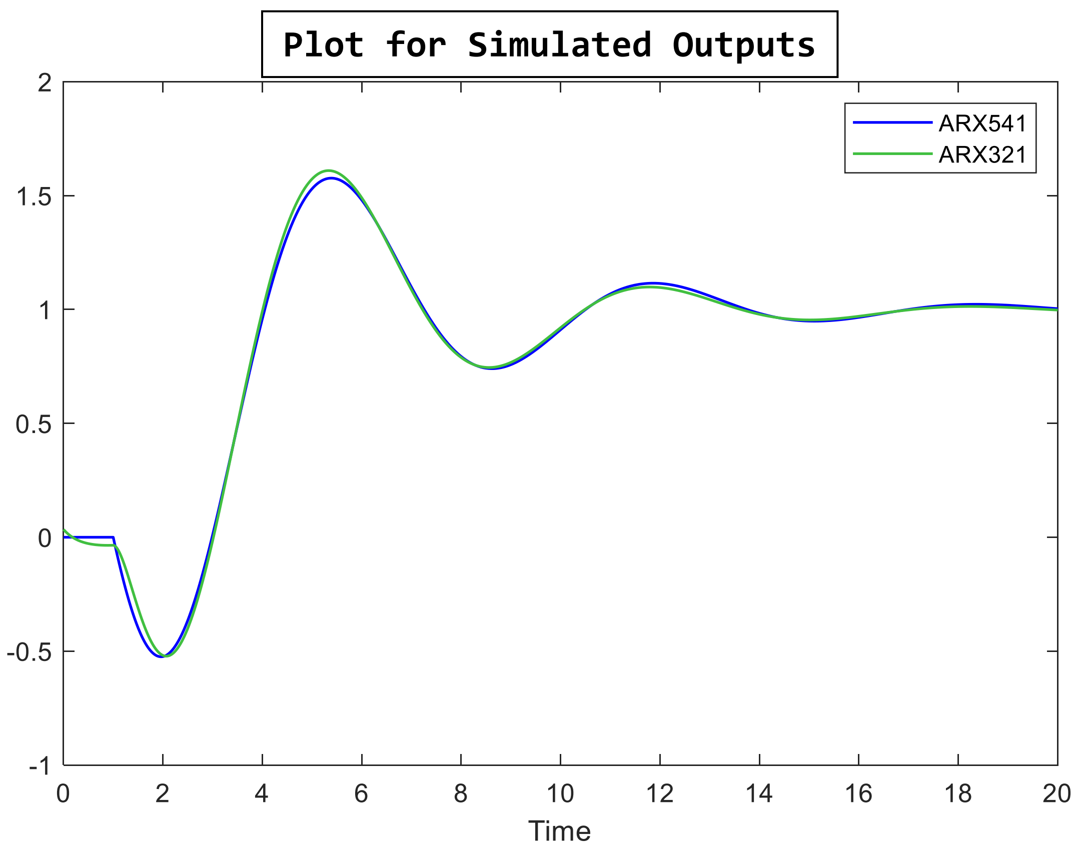
b.

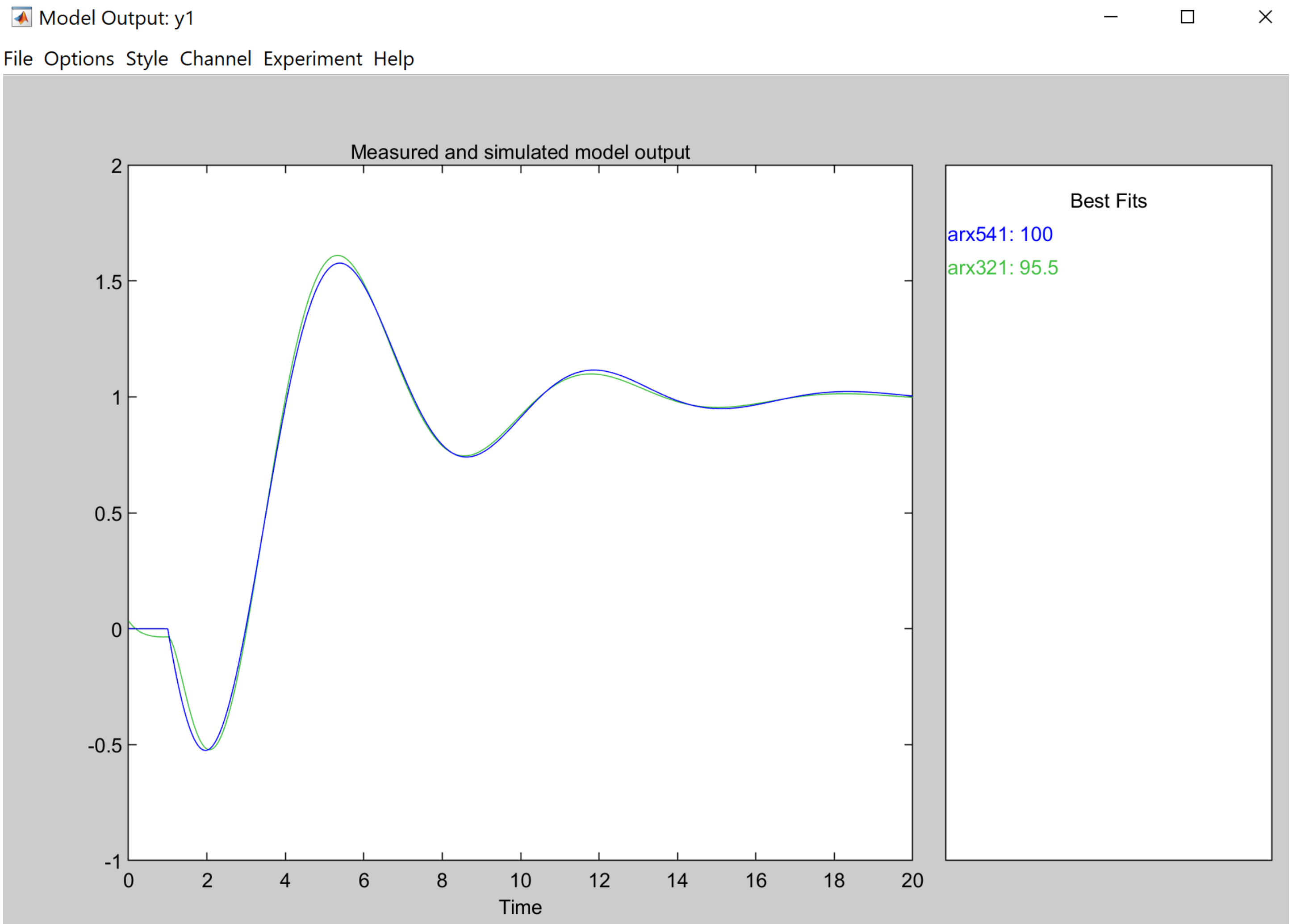
MATLAB Commands for c2d

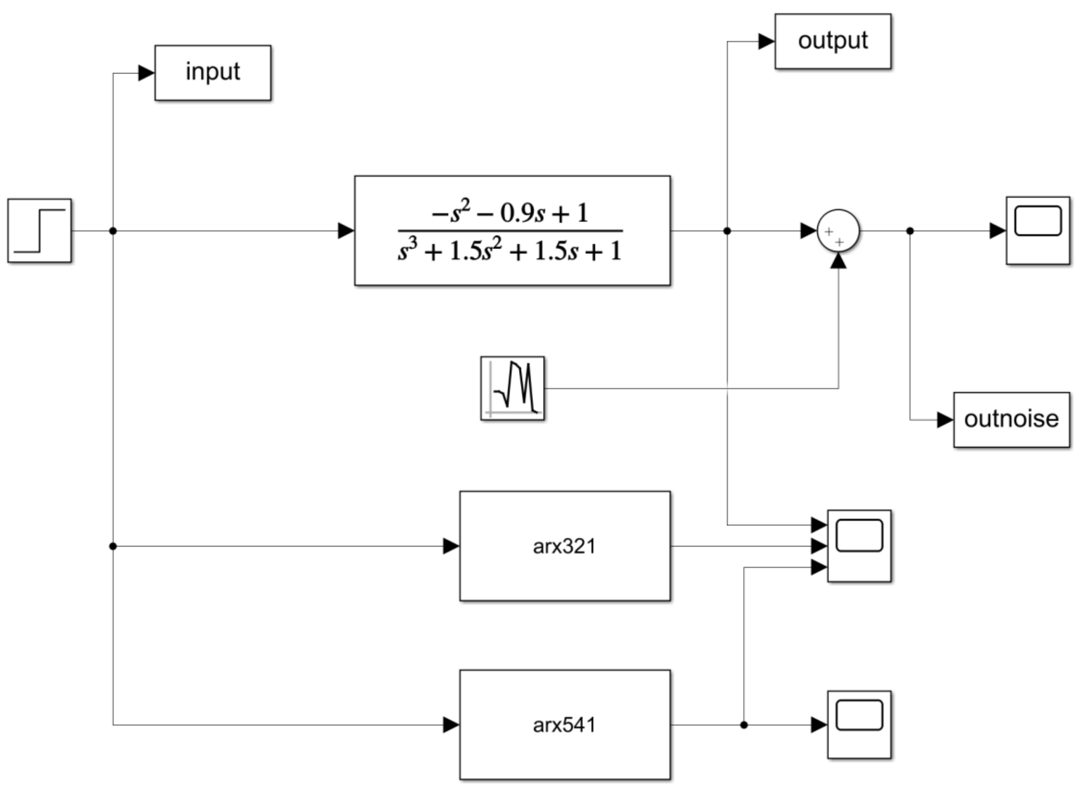
```
>> F=tf([-1 -0.9 1],[1 1.5 1.5 1])

F =

     -s^2 - 0.9 s + 1
  -------------------------
  s^3 + 1.5 s^2 + 1.5 s + 1

Continuous-time transfer function.

>> Fd=c2d(F,0.01)

Fd =

  -0.00997 z^2 + 0.01985 z - 0.00988
  ----------------------------------
  z^3 - 2.985 z^2 + 2.97 z - 0.9851

Sample time: 0.01 seconds
Discrete-time transfer function.
```

Conclusion – The A(z) is similar to the denominator of the discrete transfer function when it is estimated using the same orders for numerator and denominator. Higher order estimations cannot be compared easily.


Data/model Info: arx541
Model name:
arx541
Color:
[0,0,1]
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)
A(z) = 1 - 1.99 z^-1 + 1.97 z^-3 - 0.9802 z^-4 - 3.61e-10 z^-5
B(z) = -0.00997 z^-1 + 0.009931 z^-2 + 0.009872 z^-3 - 0.009831 z^-4
Name: arx541
Diary and Notes
% Import hw4q1a
Opt = arxOptions;
Opt.Focus = 'simulation';
arx541 = arx(hw4q1a,[5 4 1], Opt);
Show in LTI Viewer
Present
Export
Close
Help

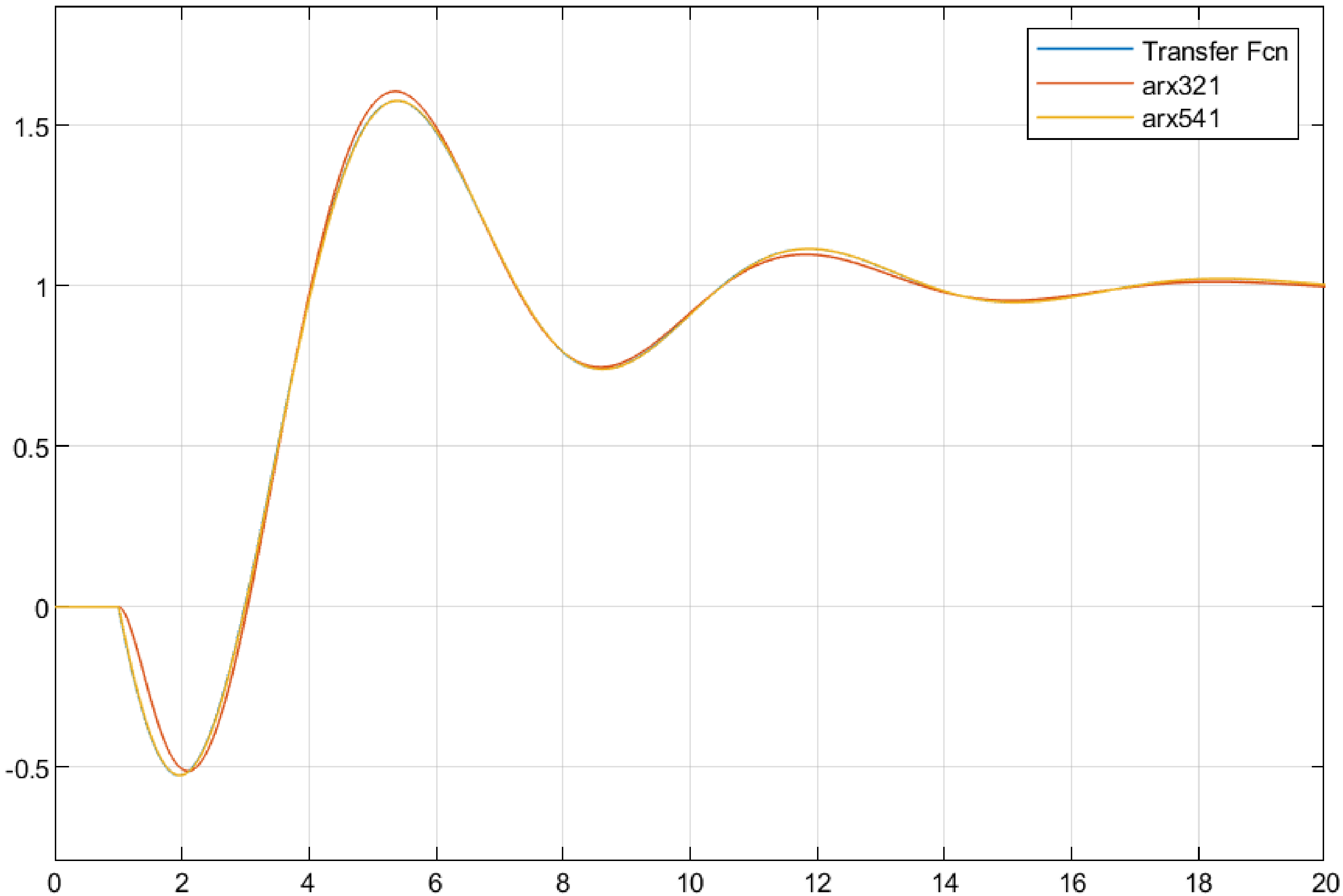

Plot for Simulated Outputs
ARX541
ARX321
2
1.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
16
18
20
Time

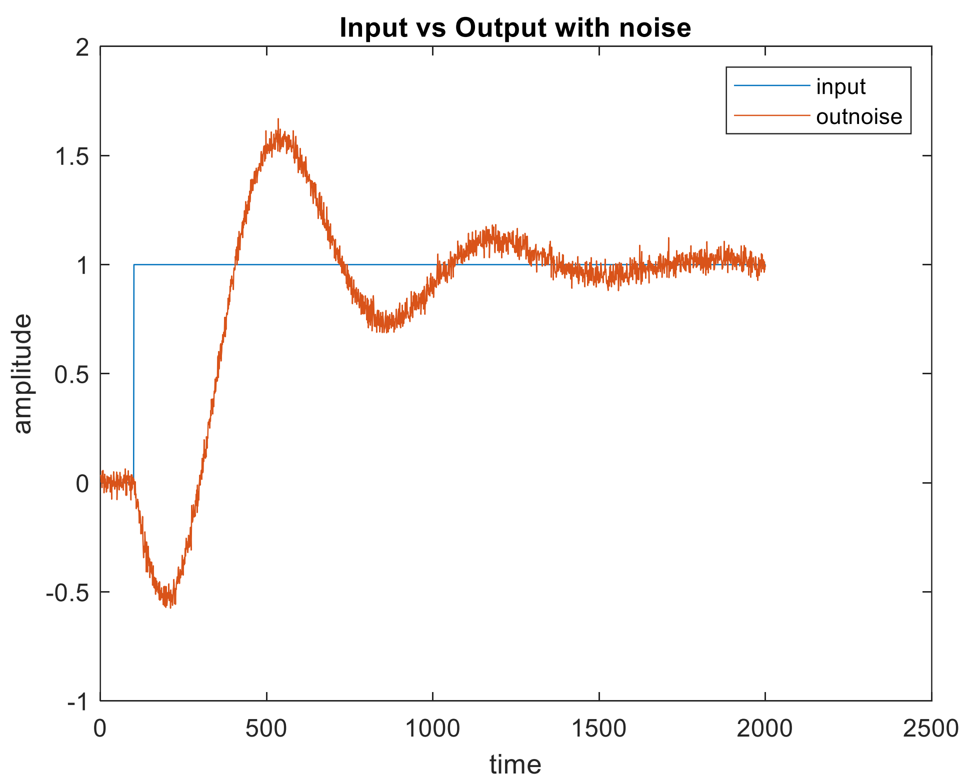

Model Output: y1
File Options Style Channel Experiment Help
Measured and simulated model output
Best Fits
arx541: 100
arx321: 95.5
Time
2
1.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
16
18
20


input
output
$\frac{-s^2-0.9s+1}{s^3+1.5s^2+1.5s+1}$
outnoise
arx321
arx541

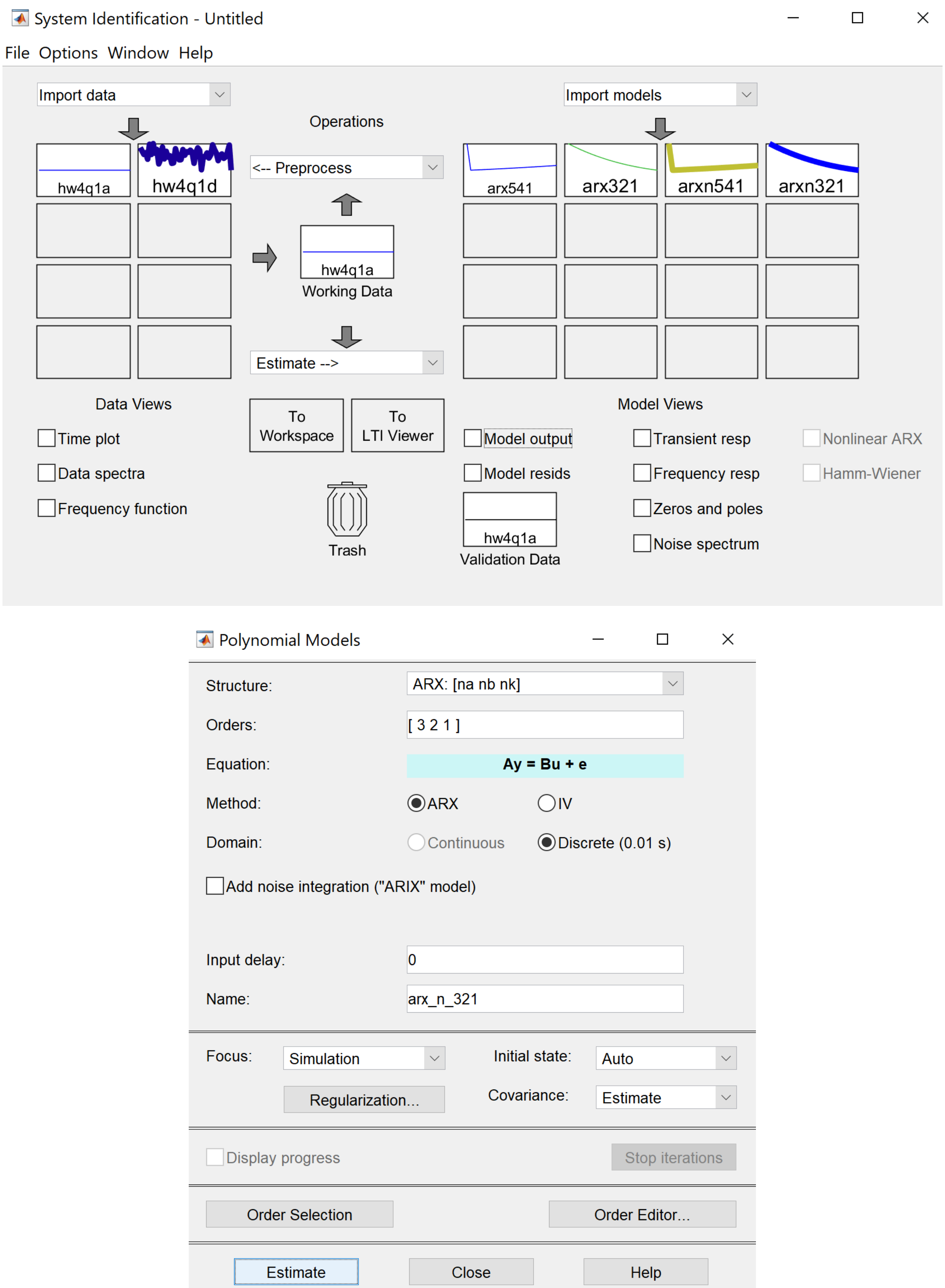
c.

Comparison plots for estimated and actual outputs

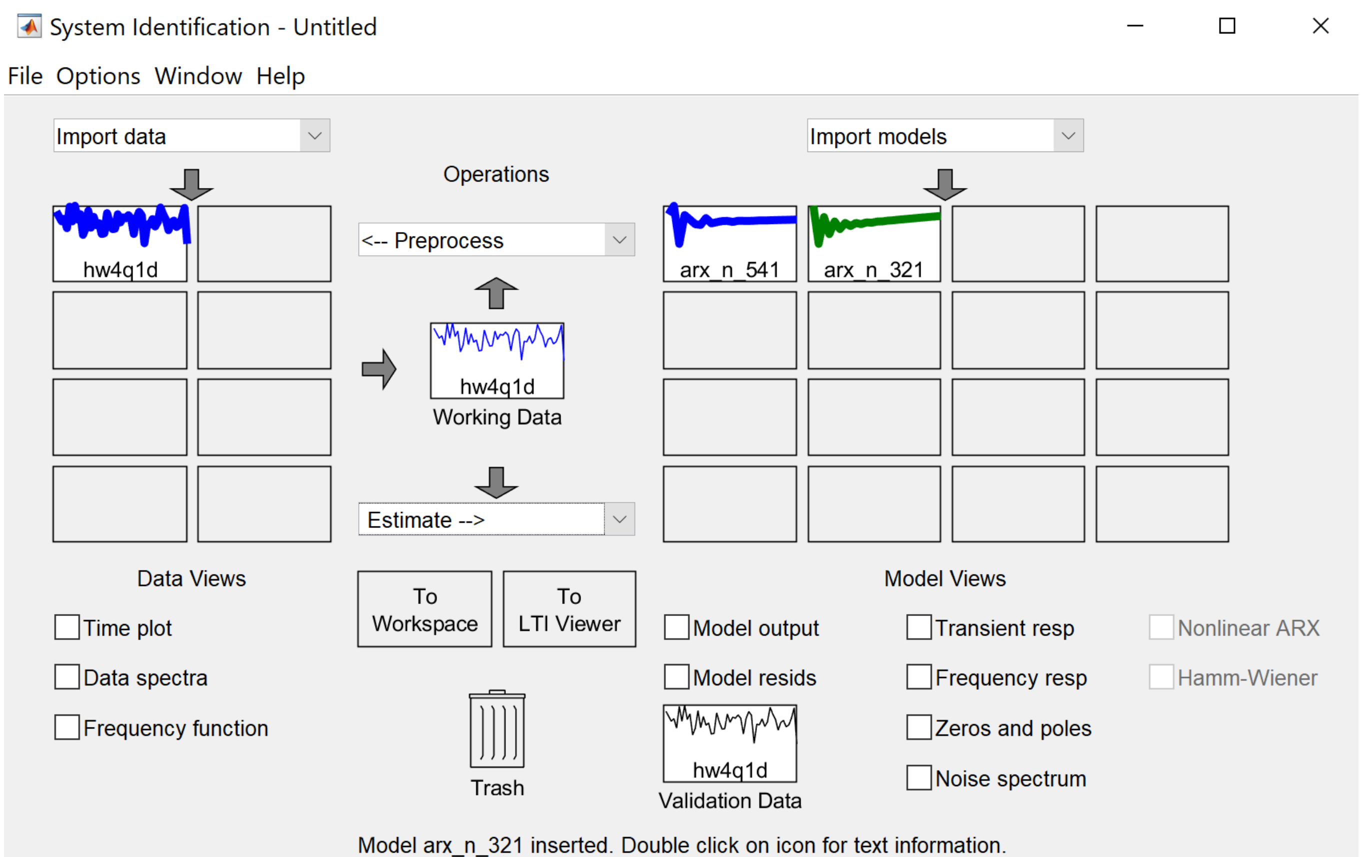


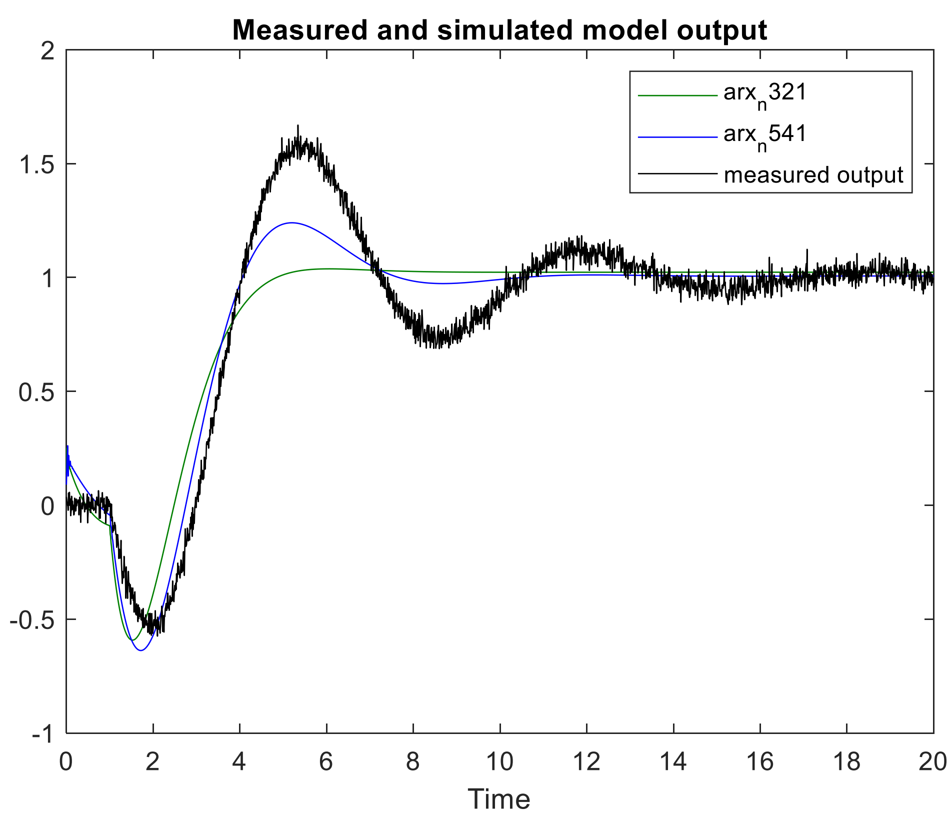


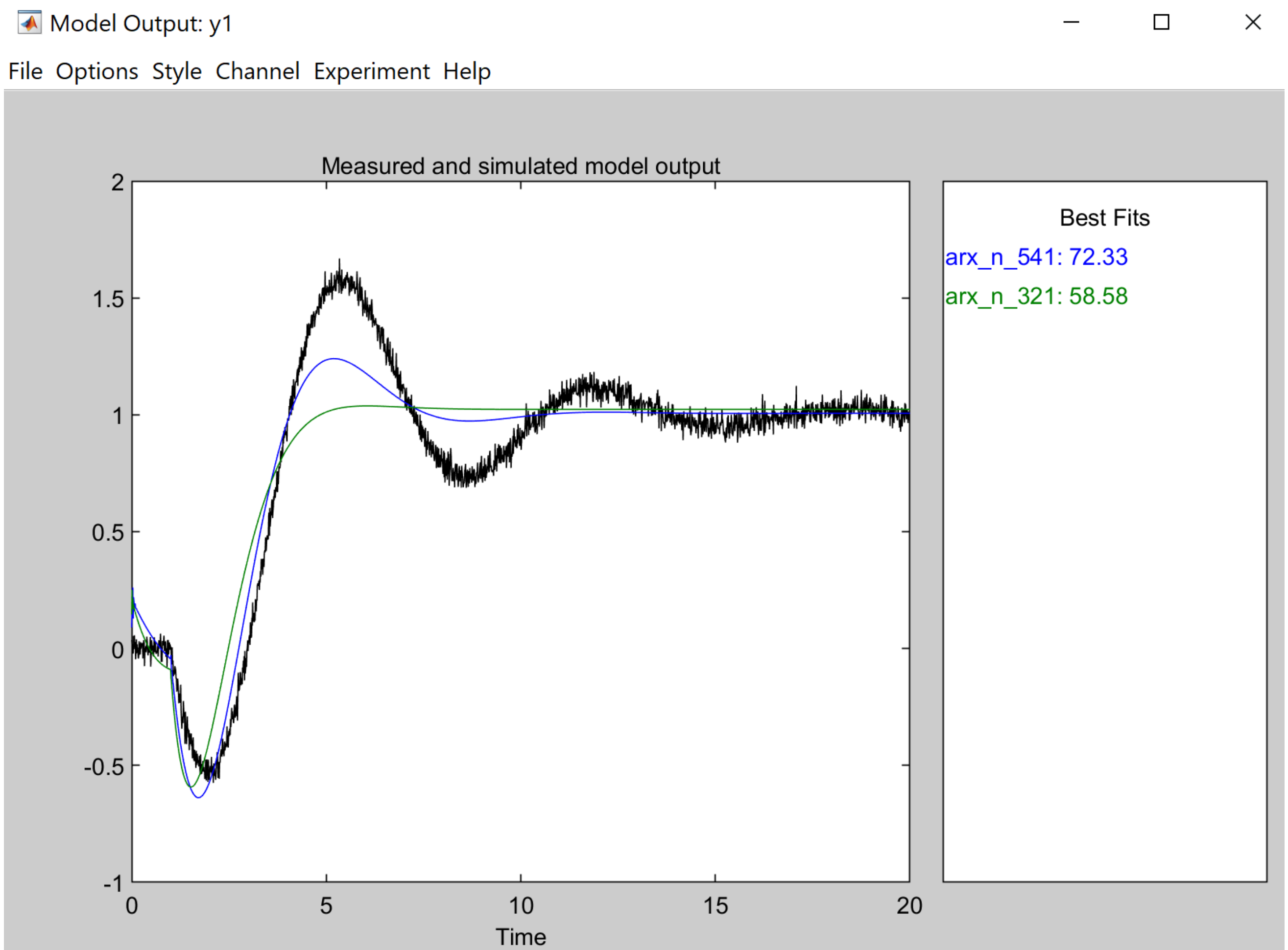
d.

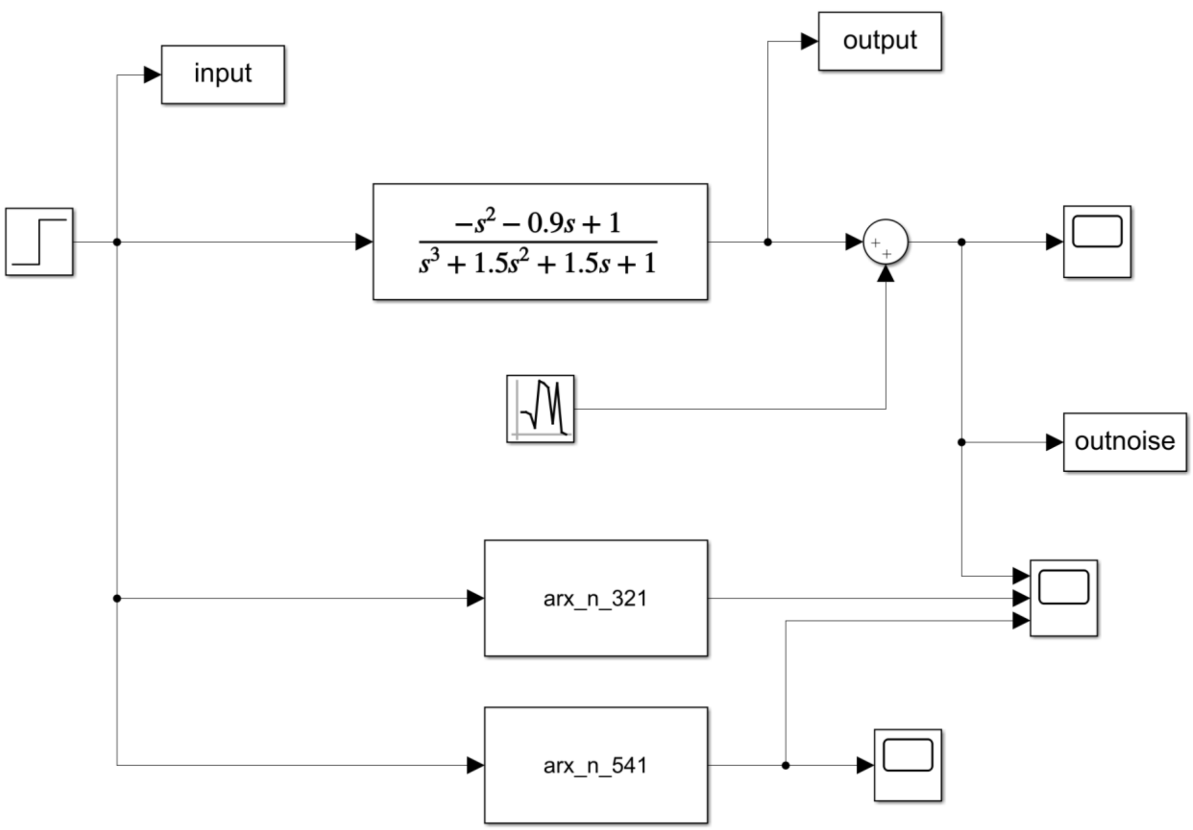


Importing data with noise and estimating polynomial functions

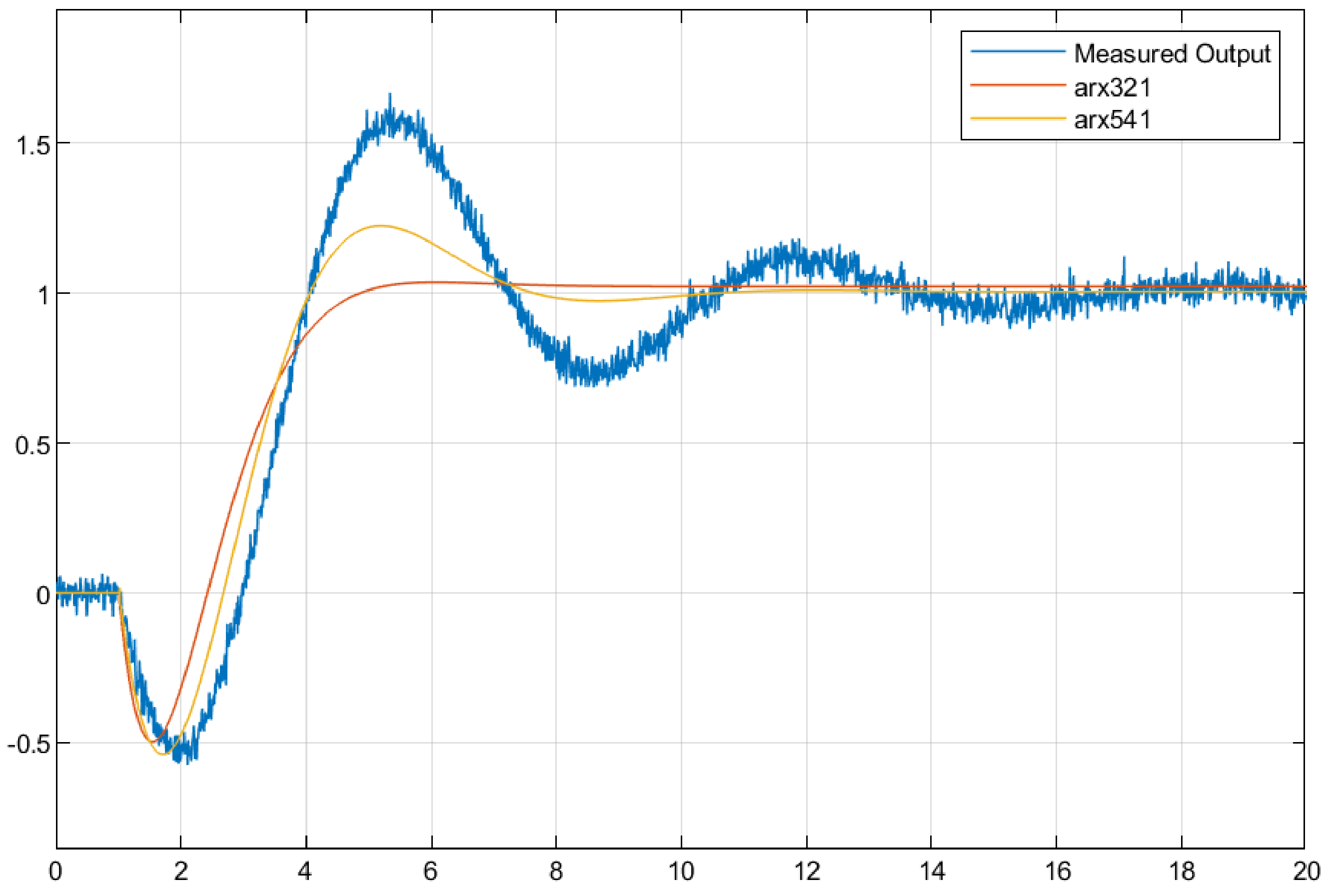

System Identification - Untitled
File Options Window Help
Import data
Import models
Operations
hw4q1d
<-- Preprocess
arx_n_541
arx_n_321
hw4q1d
Working Data
Estimate -->
Data Views
Model Views
Time plot
Data spectra
Frequency function
To Workspace
To LTI Viewer
Trash
Model output
Model resids
hw4q1d
Validation Data
Transient resp
Frequency resp
Zeros and poles
Noise spectrum
Nonlinear ARX
Hamm-Wiener
Model arx_n_321 inserted. Double click on icon for text information.

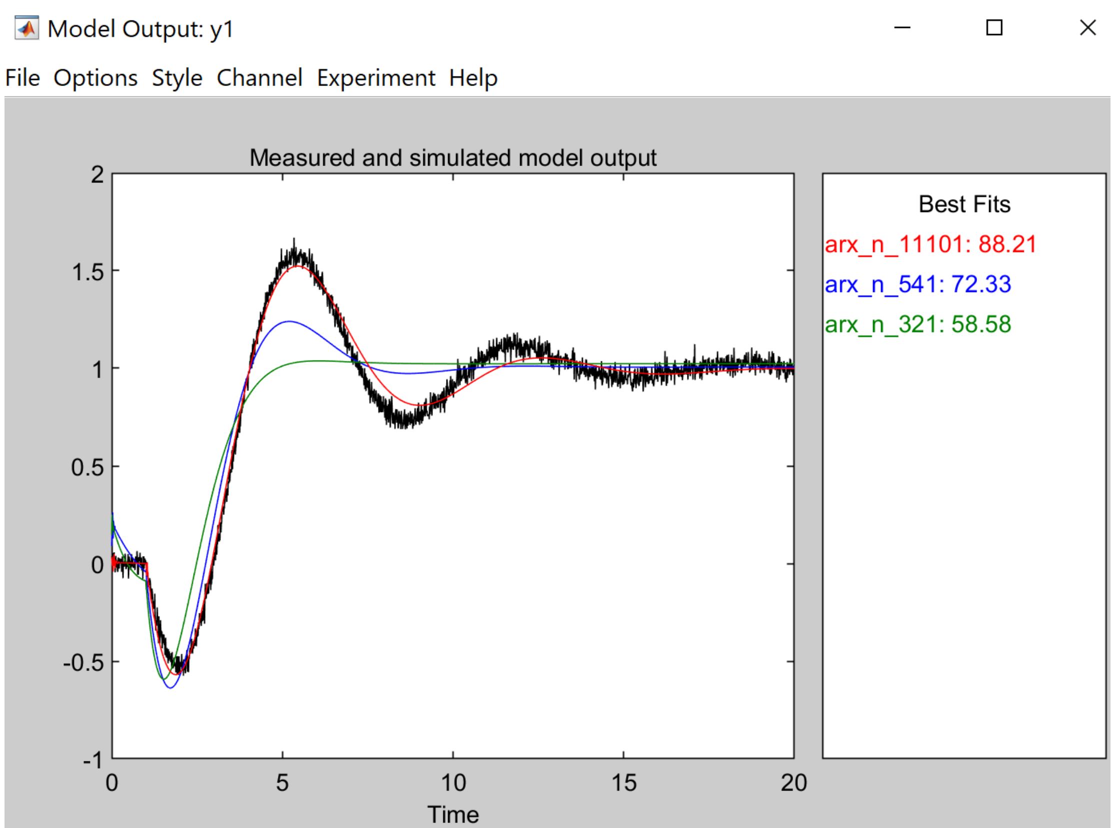

Measured and simulated model output
arx_n321
arx_n541
measured output
2
1.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
16
18
20
Time



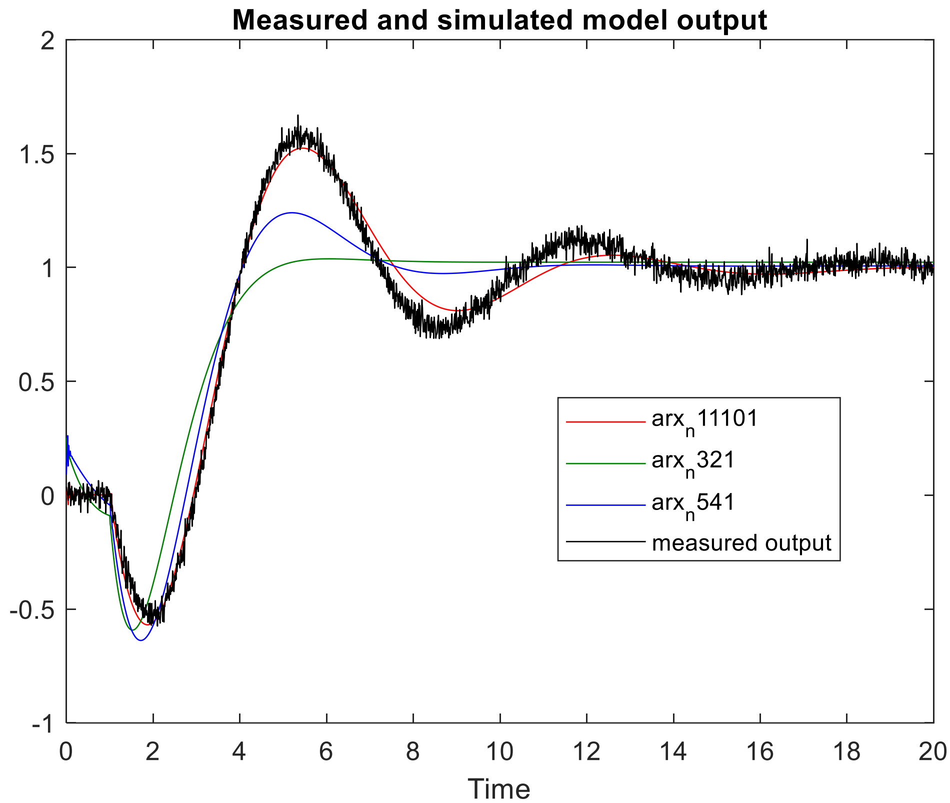


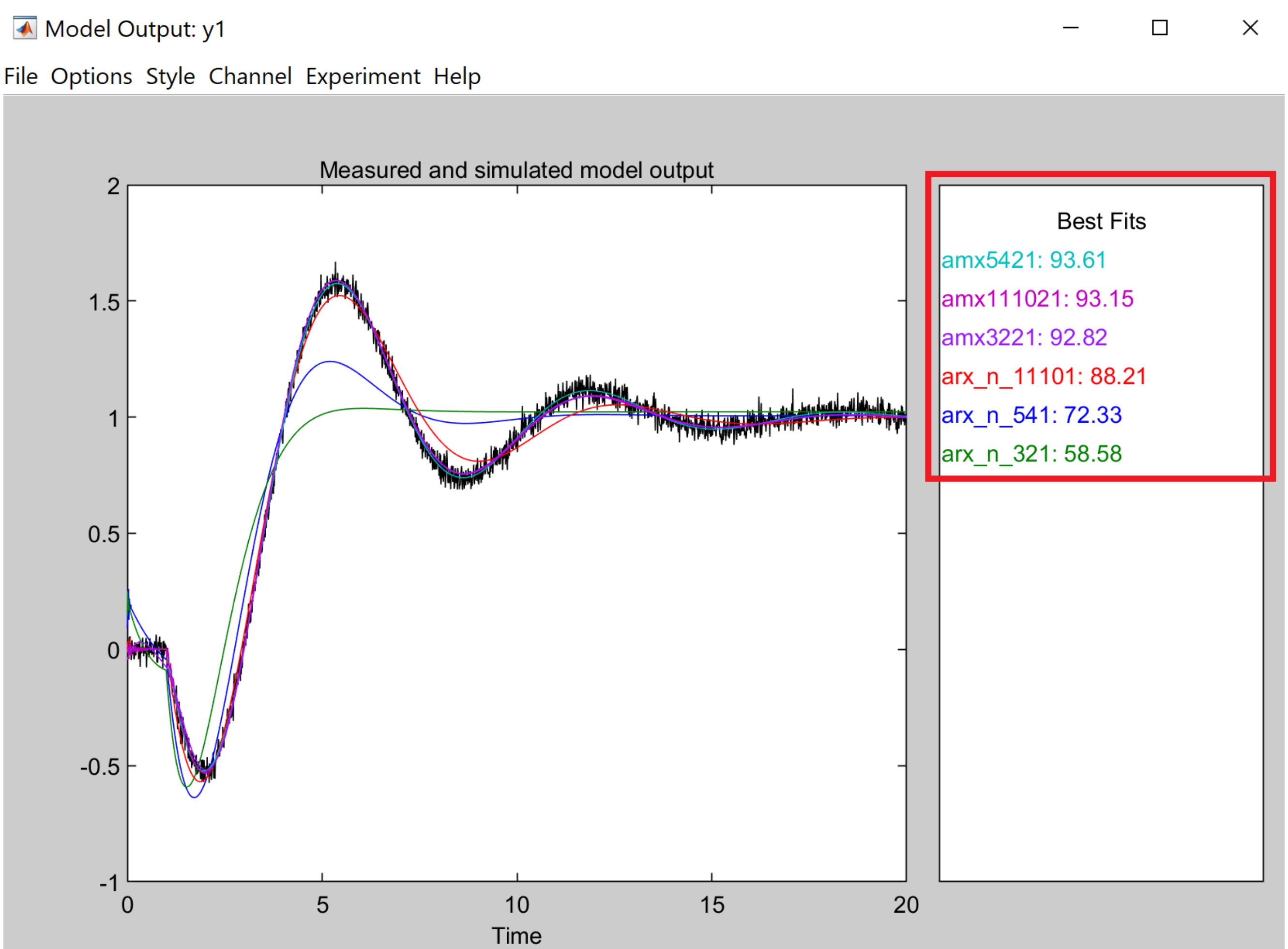
Conclusion - Noise significantly reduces the fitting capability of the ARX model

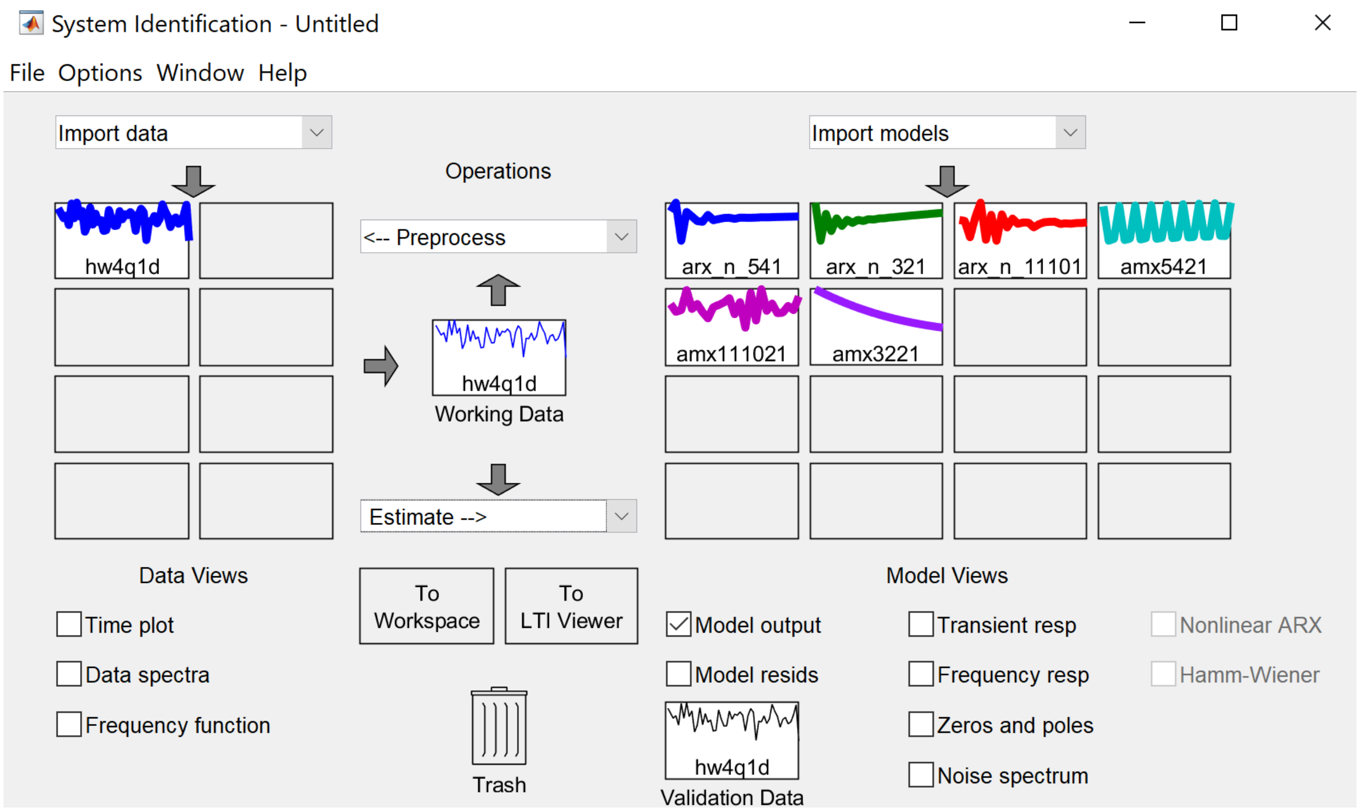




It is observed that using higher orders for estimation in the ARX model allow for better fits in case of data with noise.


Measured and simulated model output
arx_n11101
arx_n321
arx_n541
measured output
Time
2
1.5
1
0.5
0
-0.5
-1
0 2 4 6 8 10 12 14 16 18 20

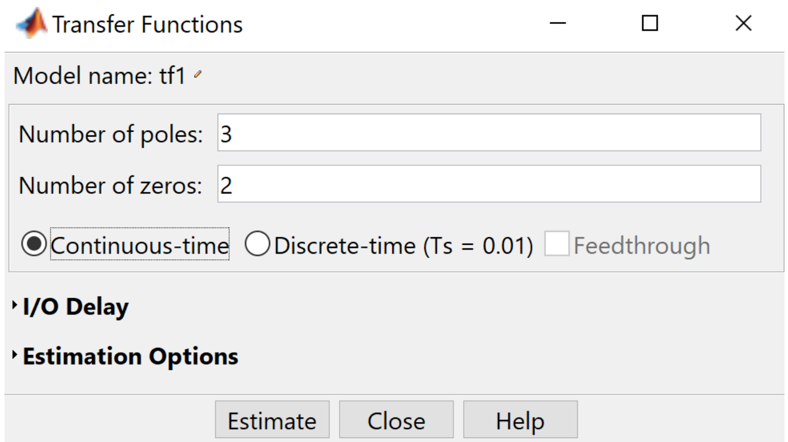

Model Output: y1
File Options Style Channel Experiment Help
Measured and simulated model output
Best Fits
amx5421: 93.61
amx111021: 93.15
amx3221: 92.82
arx_n_11101: 88.21
arx_n_541: 72.33
arx_n_321: 58.58
Time
0 5 10 15 20



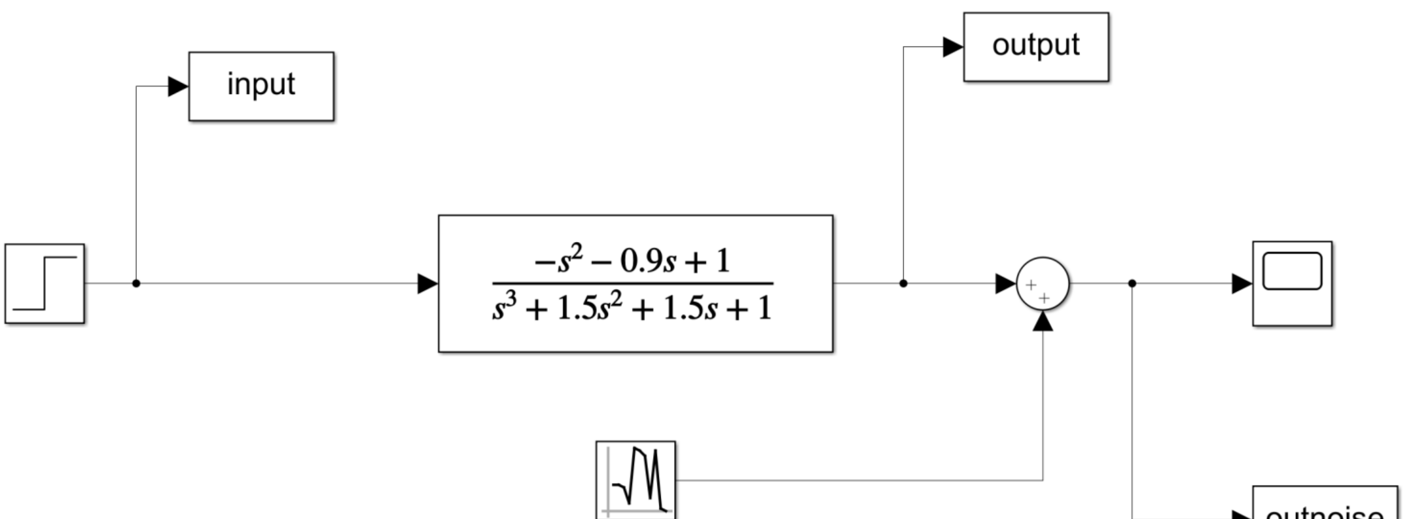
Conclusion: ARMAX model is better in prediction than AMX model. The ARMAX model has the best fit at amx5421, it probably starts chasing noise for higher orders of estimation which results in a lesser fit.

Problem – 2

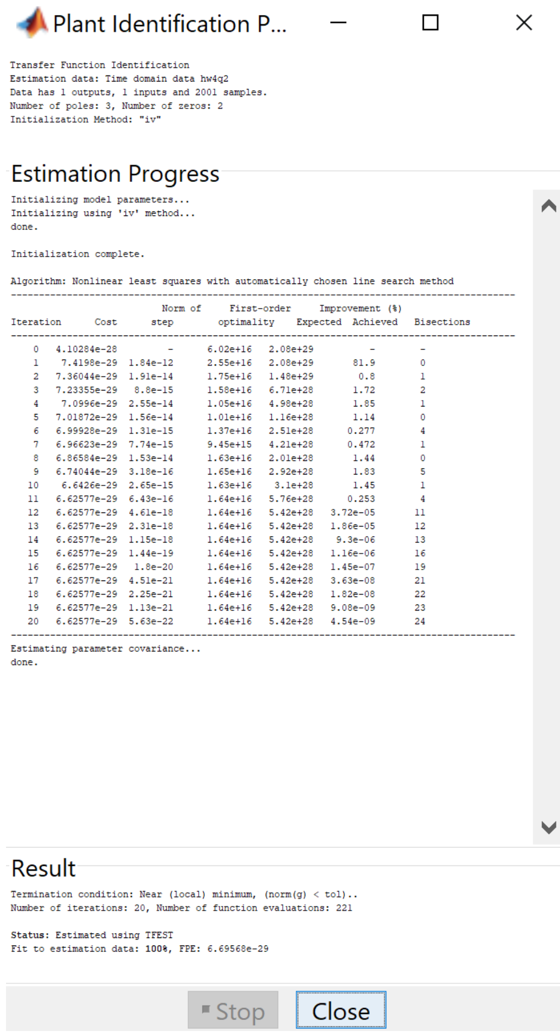
a.



Setting the initial values for estimation of transfer function after importing data



Simulink Block Diagram



Plant Identification P...

```
Transfer Function Identification
Estimation data: Time domain data hw4q2
Data has 1 outputs, 1 inputs and 2001 samples.
Number of poles: 3, Number of zeros: 2
Initialization Method: "iv"
```

Estimation Progress

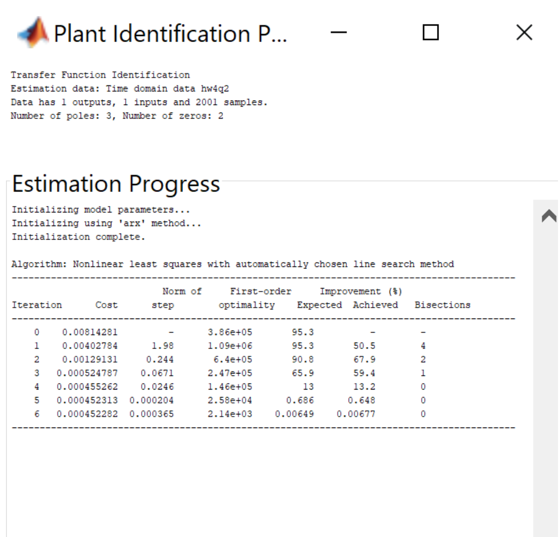
```
Initializing model parameters...
Initializing using 'iv' method...
done.

Initialization complete.

Algorithm: Nonlinear least squares with automatically chosen line search method
---------------------------------------------------------------------------------------------
                            Norm of      First-order     Improvement (%)
Iteration       Cost          step        optimality    Expected  Achieved    Bisections
---------------------------------------------------------------------------------------------
     0   4.10284e-28           -       6.02e+16   2.08e+29          -          -
     1    7.4198e-29  1.84e-12      2.55e+16   2.08e+29       81.9          0
     2   7.36044e-29  1.91e-14      1.75e+16   1.48e+29        0.8          1
     3   7.23355e-29   8.8e-15      1.58e+16   6.71e+28       1.72          2
     4    7.0996e-29  2.55e-14      1.05e+16   4.98e+28       1.85          1
     5   7.01872e-29  1.56e-14      1.01e+16   1.16e+28       1.14          0
     6   6.99928e-29  1.31e-15      1.37e+16   2.51e+28      0.277          4
     7   6.96623e-29  7.74e-15      9.45e+15   4.21e+28      0.472          1
     8   6.86584e-29  1.53e-14      1.63e+16   2.01e+28       1.44          0
     9   6.74044e-29  3.18e-16      1.65e+16   2.92e+28       1.83          5
    10    6.6426e-29  2.65e-15      1.63e+16    3.1e+28       1.45          1
    11   6.62577e-29  6.43e-16      1.64e+16   5.76e+28      0.253          4
    12   6.62577e-29  4.61e-18      1.64e+16   5.42e+28   3.72e-05         11
    13   6.62577e-29  2.31e-18      1.64e+16   5.42e+28   1.86e-05         12
    14   6.62577e-29  1.15e-18      1.64e+16   5.42e+28    9.3e-06         13
    15   6.62577e-29  1.44e-19      1.64e+16   5.42e+28   1.16e-06         16
    16   6.62577e-29   1.8e-20      1.64e+16   5.42e+28   1.45e-07         19
    17   6.62577e-29  4.51e-21      1.64e+16   5.42e+28   3.63e-08         21
    18   6.62577e-29  2.25e-21      1.64e+16   5.42e+28   1.82e-08         22
    19   6.62577e-29  1.13e-21      1.64e+16   5.42e+28   9.08e-09         23
    20   6.62577e-29  5.63e-22      1.64e+16   5.42e+28   4.54e-09         24
---------------------------------------------------------------------------------------------
Estimating parameter covariance...
done.
```

Result

```
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 20, Number of function evaluations: 221

Status: Estimated using TFEST
Fit to estimation data: 100%, FPE: 6.69568e-29
```

Stop Close

Estimation for Continuous transfer function



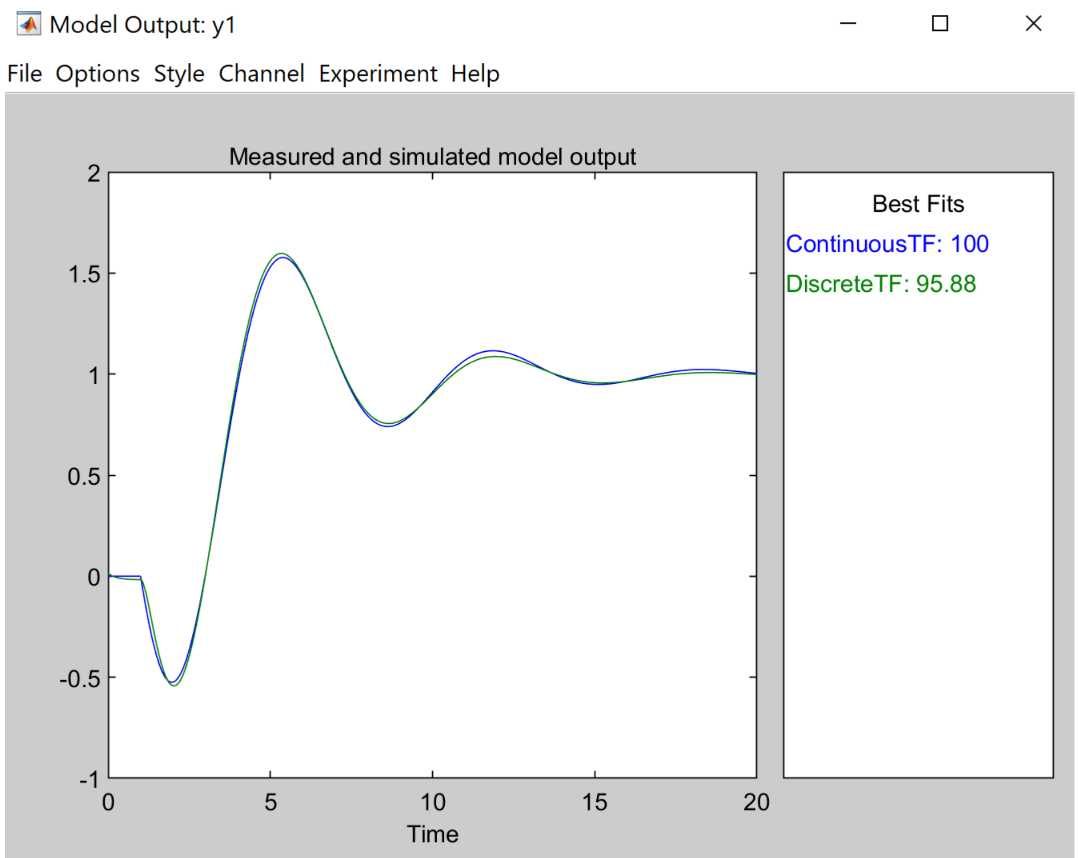
Plant Identification P...

```
Transfer Function Identification
Estimation data: Time domain data hw4q2
Data has 1 outputs, 1 inputs and 2001 samples.
Number of poles: 3, Number of zeros: 2
```

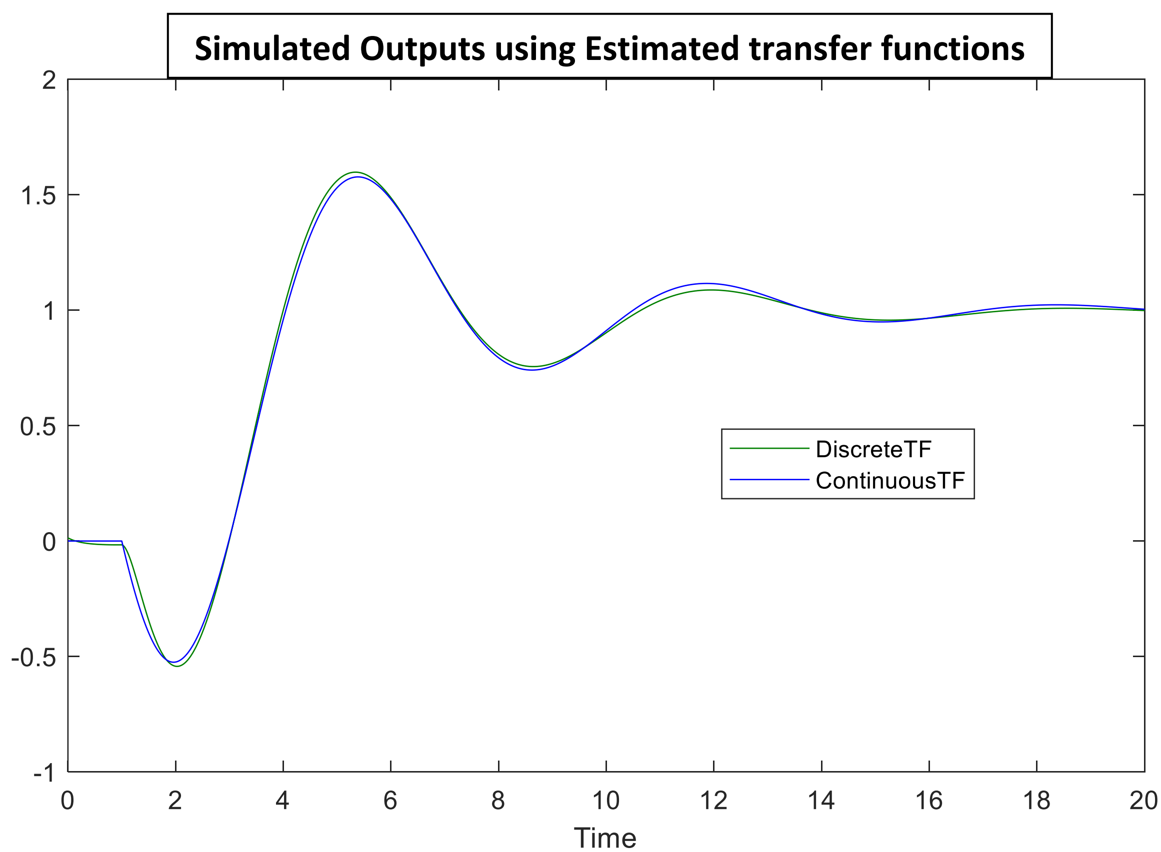
Estimation Progress

```
Initializing model parameters...
Initializing using 'arx' method...
Initialization complete.

Algorithm: Nonlinear least squares with automatically chosen line search method
---------------------------------------------------------------------------------------------
                            Norm of      First-order     Improvement (%)
Iteration       Cost          step        optimality    Expected  Achieved    Bisections
---------------------------------------------------------------------------------------------
     0    0.00814281           -       3.86e+05       95.3          -          -
     1    0.00402784        1.98      1.09e+06       95.3       50.5          4
     2    0.00129131       0.244       6.4e+05       90.8       67.9          2
     3   0.000524787      0.0671      2.47e+05       65.9       59.4          1
     4   0.000455262      0.0246      1.46e+05         13       13.2          0
     5   0.000452313    0.000204      2.58e+04      0.686      0.648          0
     6   0.000452282    0.000365      2.14e+03    0.00649    0.00677          0
---------------------------------------------------------------------------------------------
```

Estimation for discrete transfer function



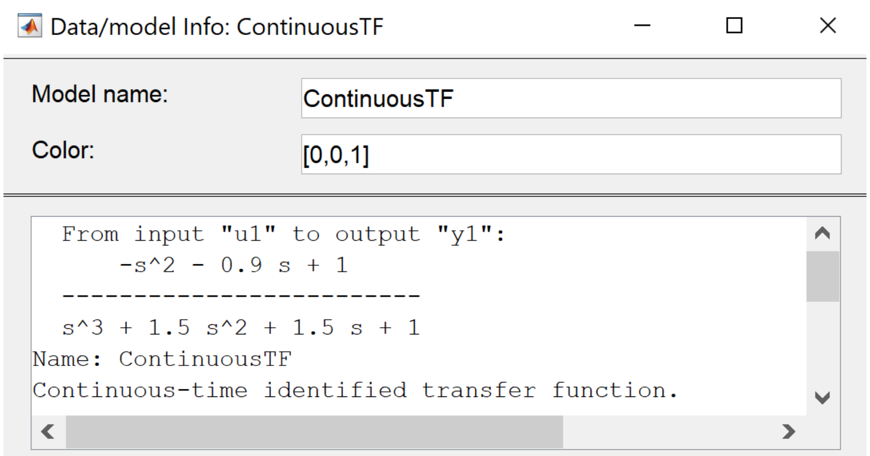
Model Output: y1
File Options Style Channel Experiment Help
Measured and simulated model output
Best Fits
ContinuousTF: 100
DiscreteTF: 95.88
Time



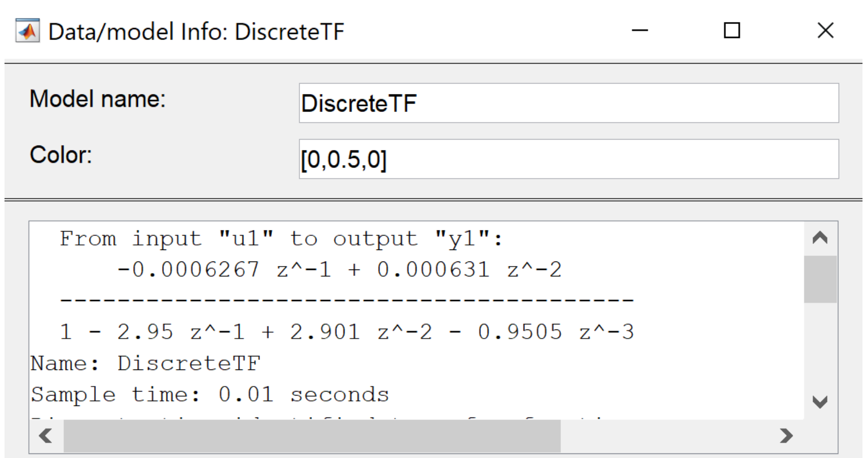
Simulated Outputs using Estimated transfer functions
DiscreteTF
ContinuousTF
Time

b. MATLAB Commands for converting to discrete form for comparison

```
>> C=tf([-1 -0.9 1],[1 1.5 1.5 1])

C =

      -s^2 - 0.9 s + 1
  -------------------------
  s^3 + 1.5 s^2 + 1.5 s + 1
Continuous-time transfer function.

>> D=c2d(C,0.01)

D =

  -0.00997 z^2 + 0.01985 z - 0.00988
  ----------------------------------
  z^3 - 2.985 z^2 + 2.97 z - 0.9851

Sample time: 0.01 seconds
Discrete-time transfer function.
```

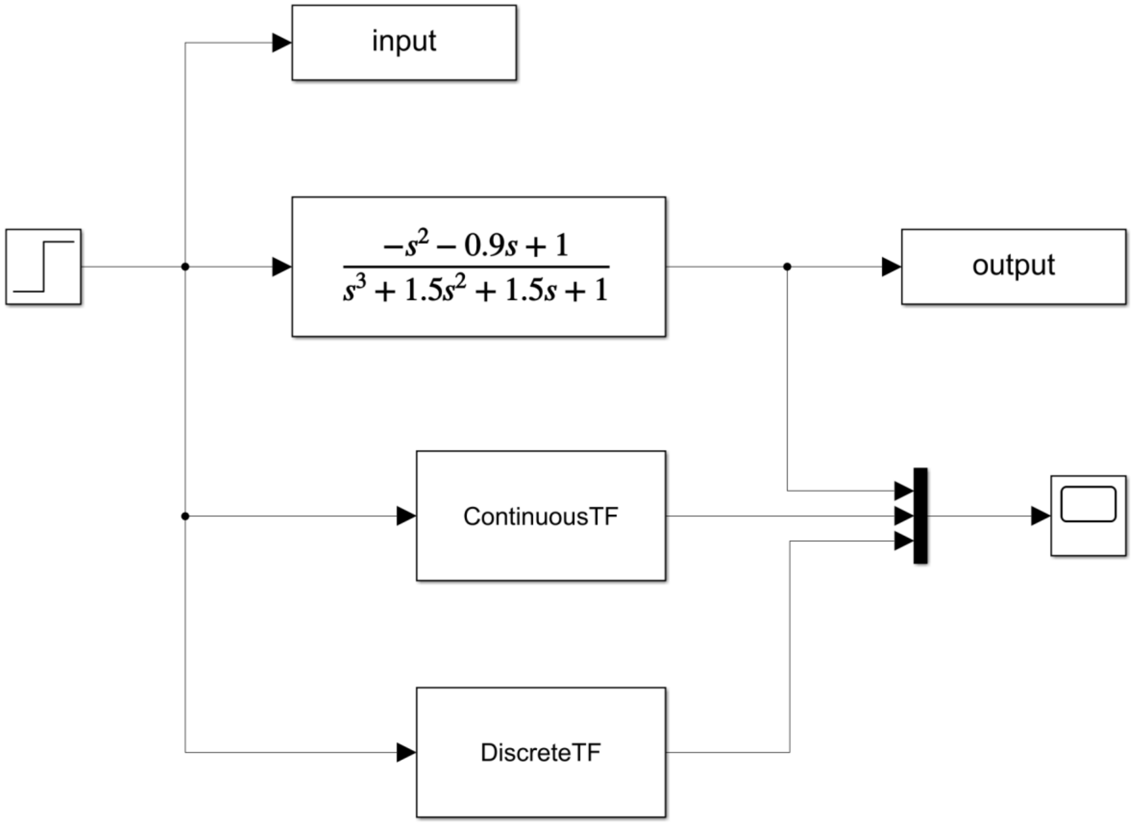


Continuous Transfer Function

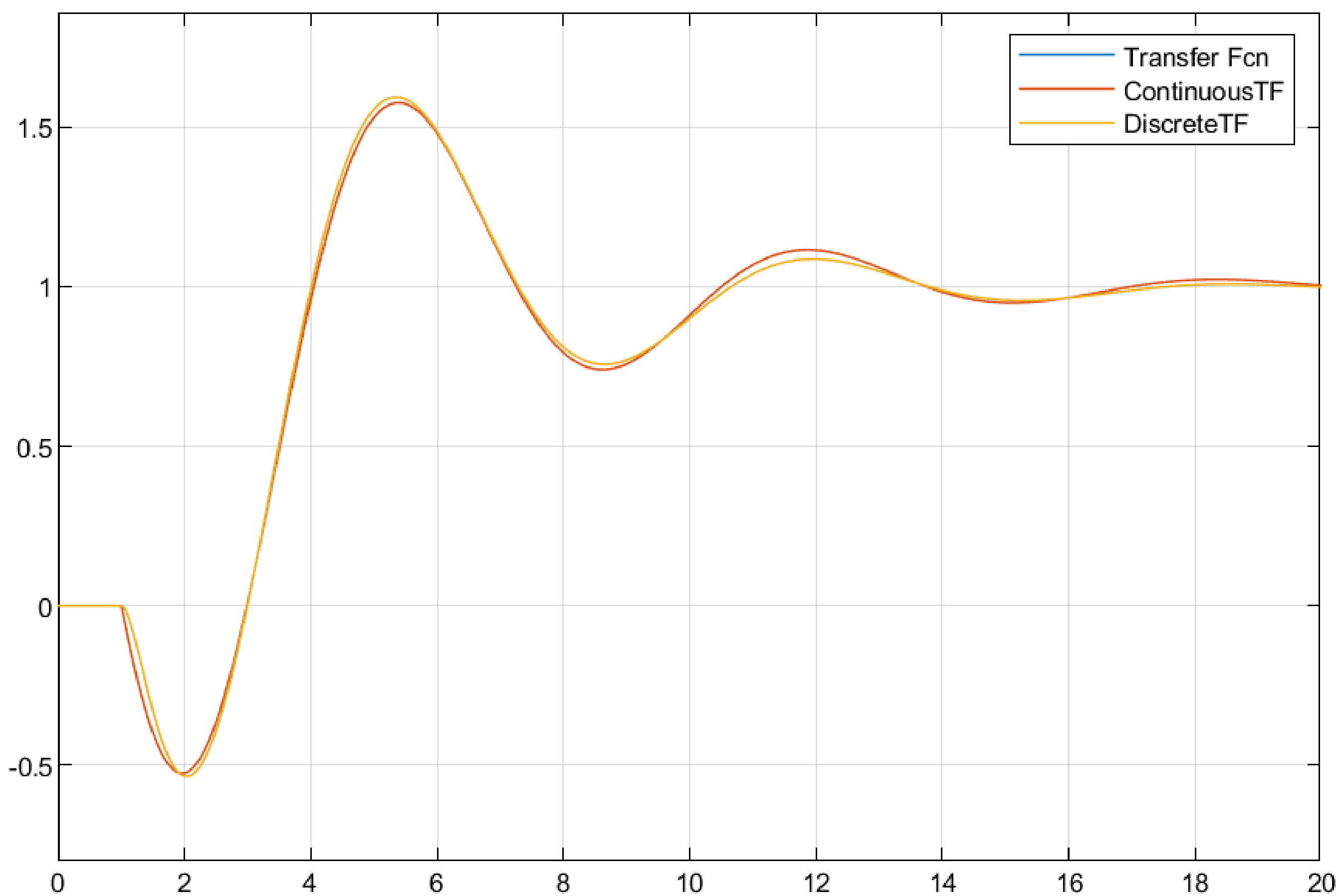


Discrete Transfer function

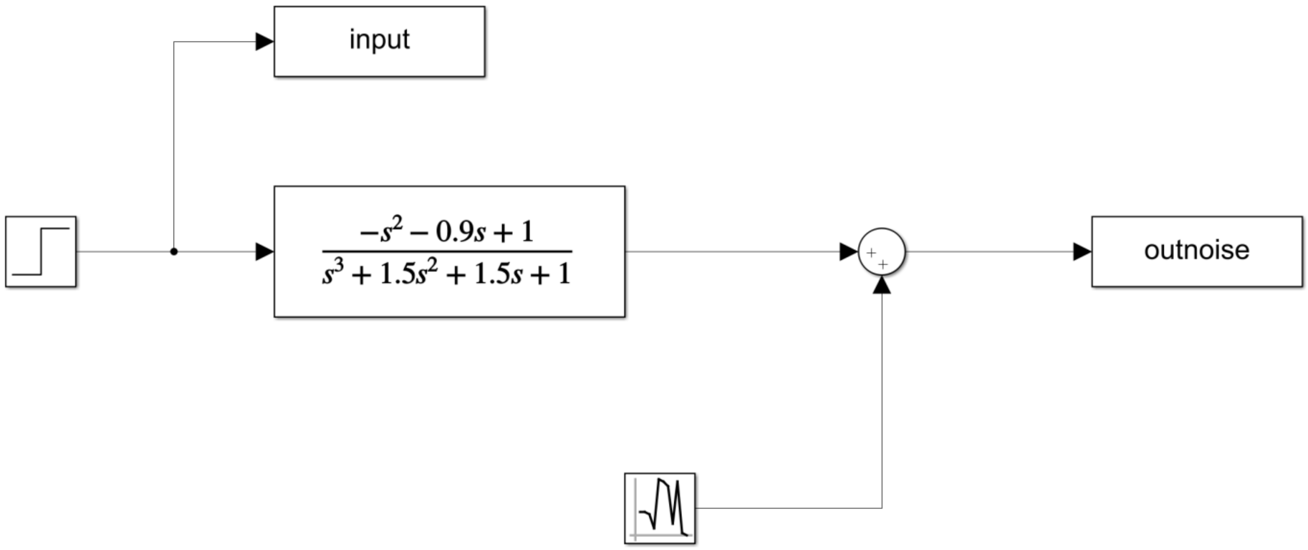
Conclusion - The estimated continuous transfer function is exactly the same as the original system values i.e., poles and zeros are same. Denominator for the estimated discrete transfer function is very similar to A(z) but the numerator is different.



c.

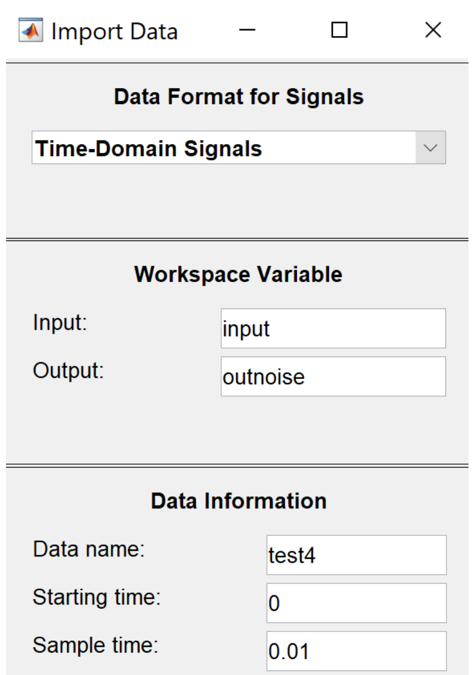


Comparison plots derived from Scope

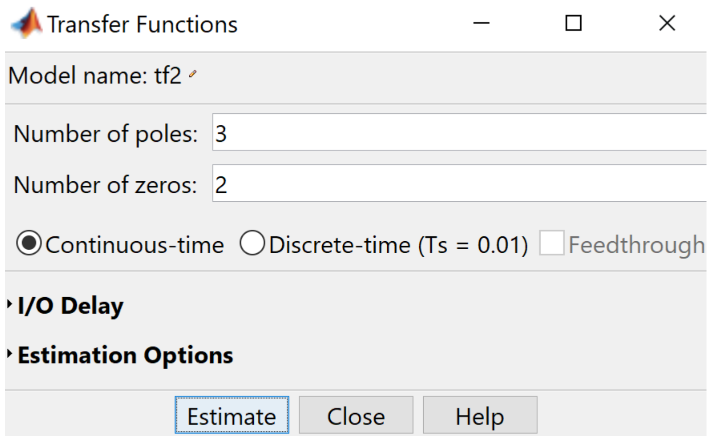


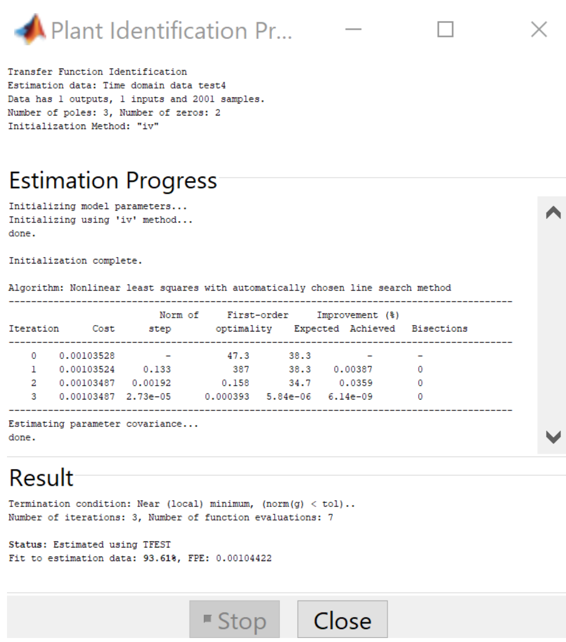
d.

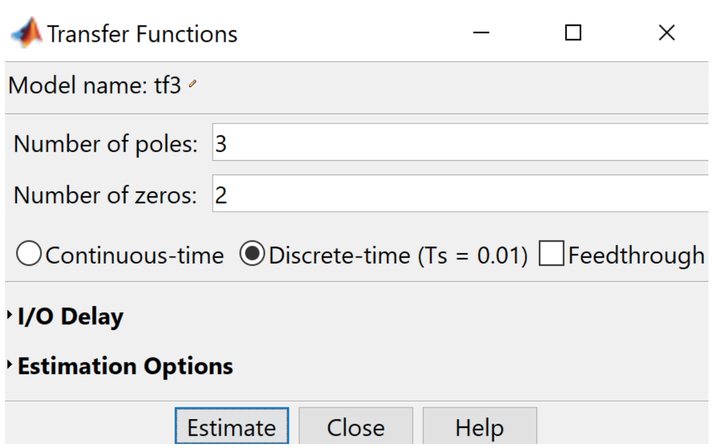
Simulink Block Diagram with Noise

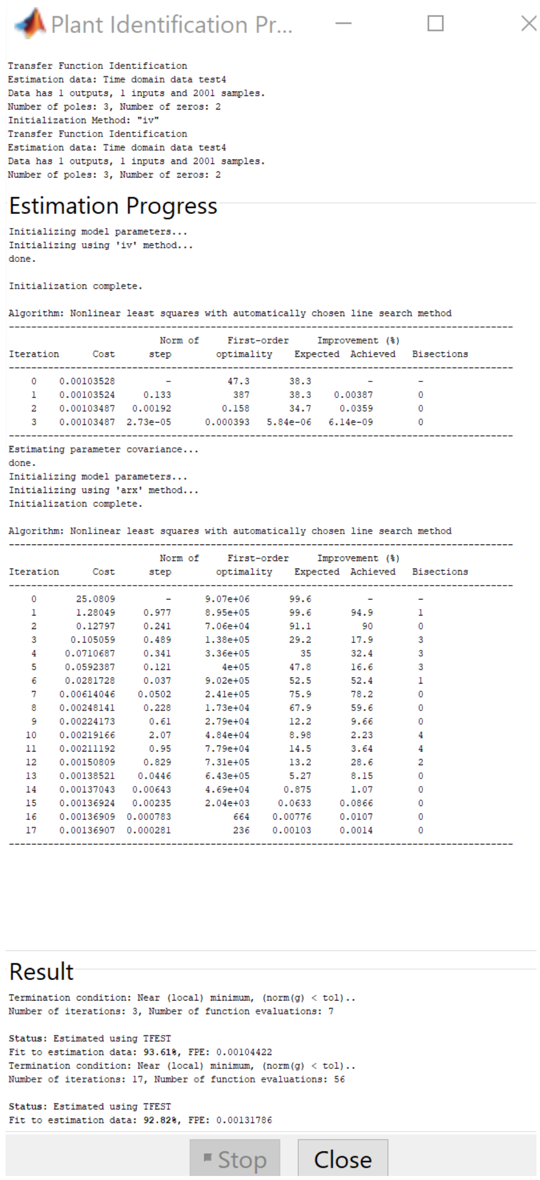


Importing data with noise from workspace


Transfer Functions
Model name: tf2
Number of poles: 3
Number of zeros: 2
Continuous-time
Discrete-time (Ts = 0.01)
Feedthrough
I/O Delay
Estimation Options
Estimate
Close
Help


Plant Identification Pr...
Transfer Function Identification
Estimation data: Time domain data test4
Data has 1 outputs, 1 inputs and 2001 samples.
Number of poles: 3, Number of zeros: 2
Initialization Method: "iv"
Estimation Progress
Initializing model parameters...
Initializing using 'iv' method...
done.
Initialization complete.
Algorithm: Nonlinear least squares with automatically chosen line search method
Norm of First-order Improvement (%)
Iteration Cost step optimality Expected Achieved Bisections
0 0.00103528 - 47.3 38.3 - -
1 0.00103524 0.133 387 38.3 0.00387 0
2 0.00103487 0.00192 0.158 34.7 0.0359 0
3 0.00103487 2.73e-05 0.000393 5.84e-06 6.14e-09 0
Estimating parameter covariance...
done.
Result
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 3, Number of function evaluations: 7
Status: Estimated using TFEST
Fit to estimation data: 93.61%, FPE: 0.00104422
Stop
Close


Transfer Functions
Model name: tf3
Number of poles: 3
Number of zeros: 2
Continuous-time
Discrete-time (Ts = 0.01)
Feedthrough
I/O Delay
Estimation Options
Estimate
Close
Help


Plant Identification Pr...

```
Transfer Function Identification
Estimation data: Time domain data test4
Data has 1 outputs, 1 inputs and 2001 samples.
Number of poles: 3, Number of zeros: 2
Initialization Method: "iv"
Transfer Function Identification
Estimation data: Time domain data test4
Data has 1 outputs, 1 inputs and 2001 samples.
Number of poles: 3, Number of zeros: 2
```

Estimation Progress

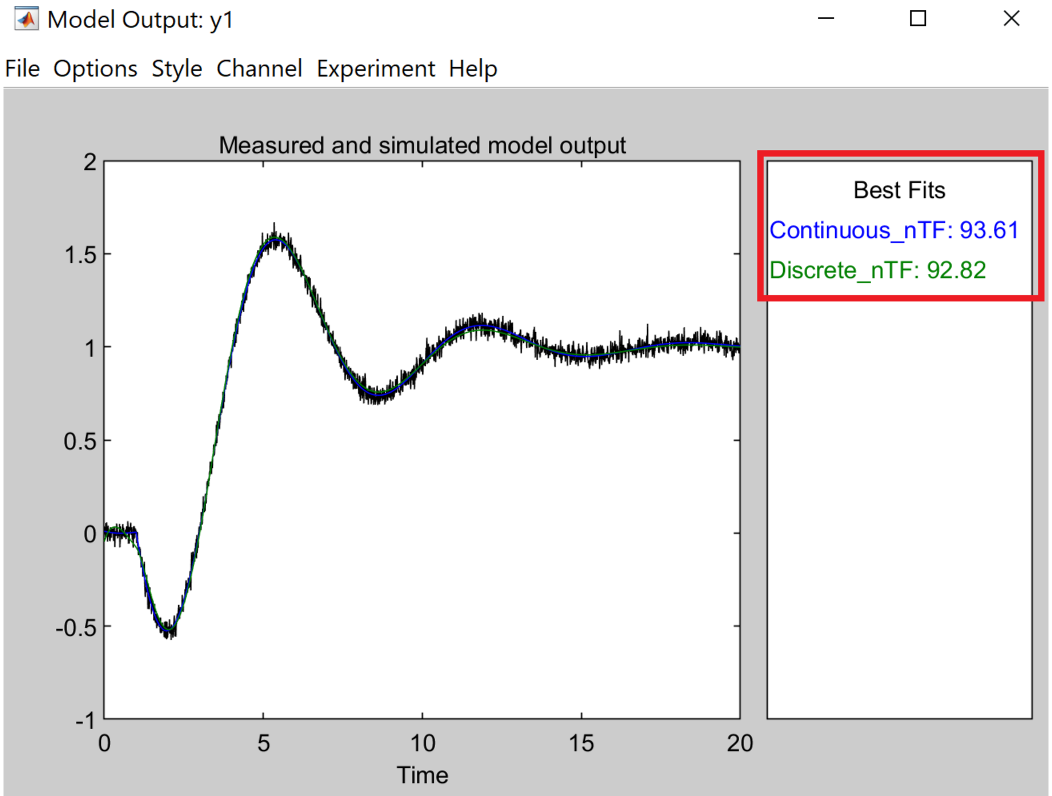
```
Initializing model parameters...
Initializing using 'iv' method...
done.

Initialization complete.

Algorithm: Nonlinear least squares with automatically chosen line search method
---------------------------------------------------------------------------------
                          Norm of     First-order     Improvement (%)
Iteration      Cost       step        optimality    Expected  Achieved   Bisections
---------------------------------------------------------------------------------
    0    0.00103528        -          47.3       38.3          -          -
    1    0.00103524     0.133          387       38.3    0.00387          0
    2    0.00103487   0.00192        0.158       34.7     0.0359          0
    3    0.00103487  2.73e-05     0.000393   5.84e-06   6.14e-09          0
---------------------------------------------------------------------------------
Estimating parameter covariance...
done.
Initializing model parameters...
Initializing using 'arx' method...
Initialization complete.

Algorithm: Nonlinear least squares with automatically chosen line search method
---------------------------------------------------------------------------------
                          Norm of     First-order     Improvement (%)
Iteration      Cost       step        optimality    Expected  Achieved   Bisections
---------------------------------------------------------------------------------
    0       25.0809        -      9.07e+06       99.6          -          -
    1       1.28049     0.977     8.95e+05       99.6       94.9          1
    2       0.12797     0.241     7.06e+04       91.1         90          0
    3      0.105059     0.489     1.38e+05       29.2       17.9          3
    4     0.0710687     0.341     3.36e+05         35       32.4          3
    5     0.0592387     0.121        4e+05       47.8       16.6          3
    6     0.0281728     0.037     9.02e+05       52.5       52.4          1
    7    0.00614046    0.0502     2.41e+05       75.9       78.2          0
    8    0.00248141     0.228     1.73e+04       67.9       59.6          0
    9    0.00224173      0.61     2.79e+04       12.2       9.66          0
   10    0.00219166      2.07     4.84e+04       8.98       2.23          4
   11    0.00211192      0.95     7.79e+04       14.5       3.64          4
   12    0.00150809     0.829     7.31e+05       13.2       28.6          2
   13    0.00138521    0.0446     6.43e+05       5.27       8.15          0
   14    0.00137043   0.00643     4.69e+04      0.875       1.07          0
   15    0.00136924   0.00235     2.04e+03     0.0633     0.0866          0
   16    0.00136909  0.000783          664    0.00776     0.0107          0
   17    0.00136907  0.000281          236    0.00103     0.0014          0
---------------------------------------------------------------------------------
```

Result

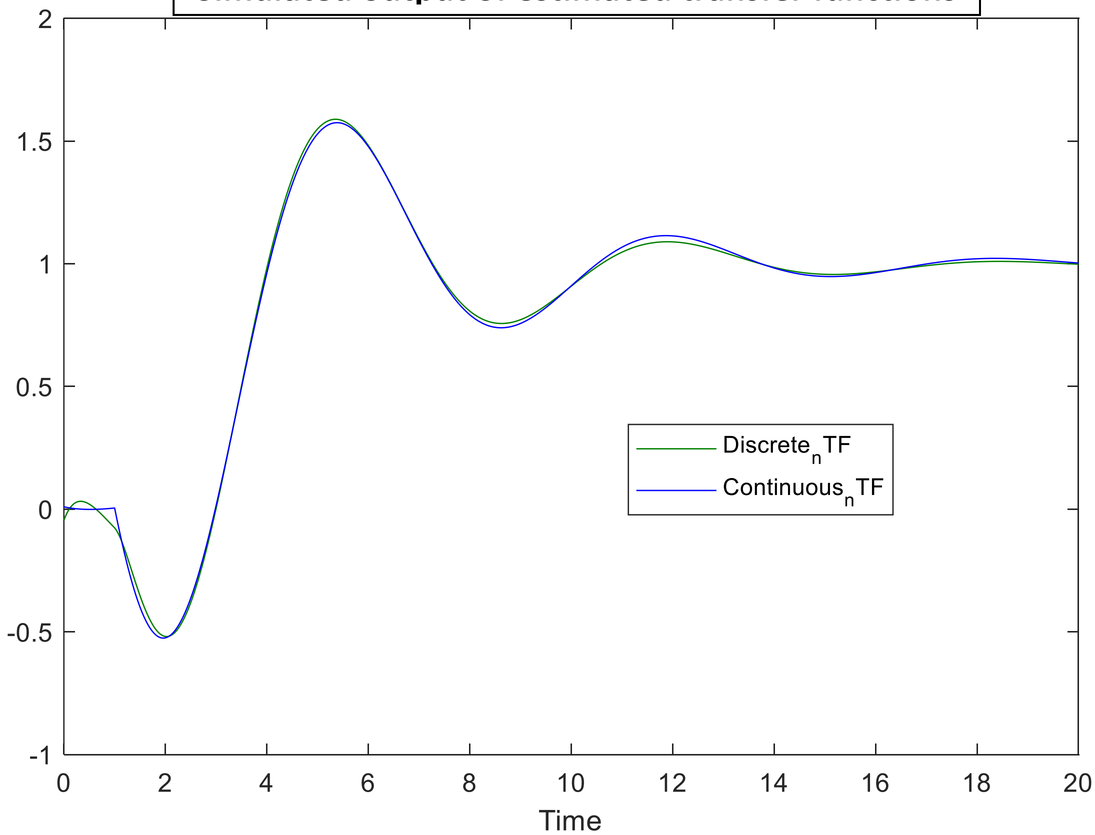
```
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 3, Number of function evaluations: 7

Status: Estimated using TFEST
Fit to estimation data: 93.61%, FPE: 0.00104422
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 17, Number of function evaluations: 56

Status: Estimated using TFEST
Fit to estimation data: 92.82%, FPE: 0.00131786
```

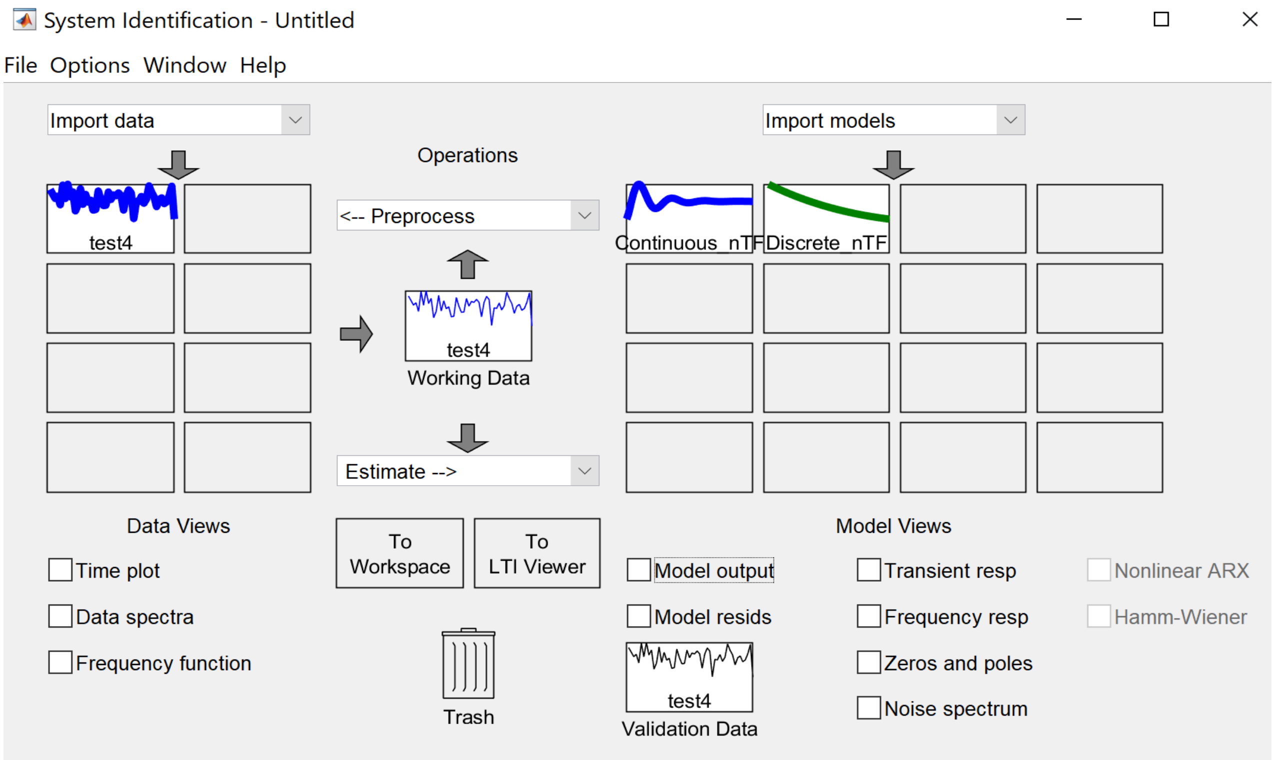
Stop Close

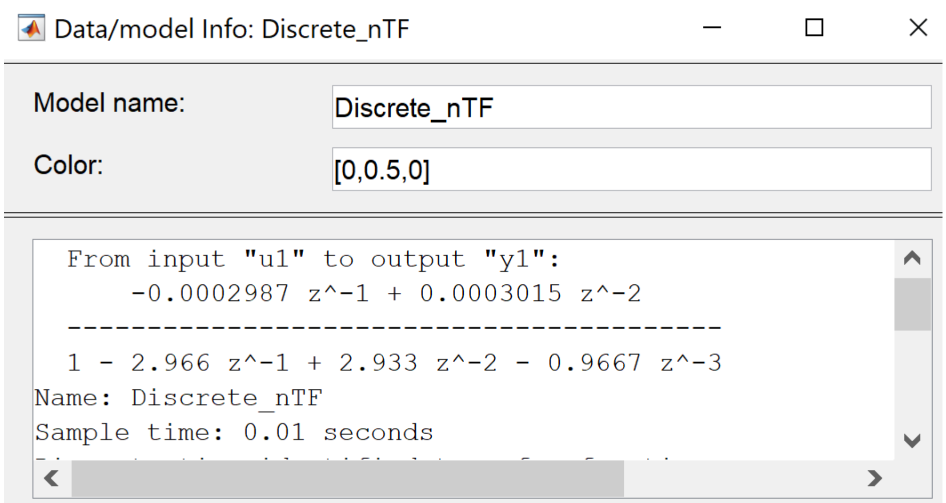
Setting parameters for estimation and estimation process data

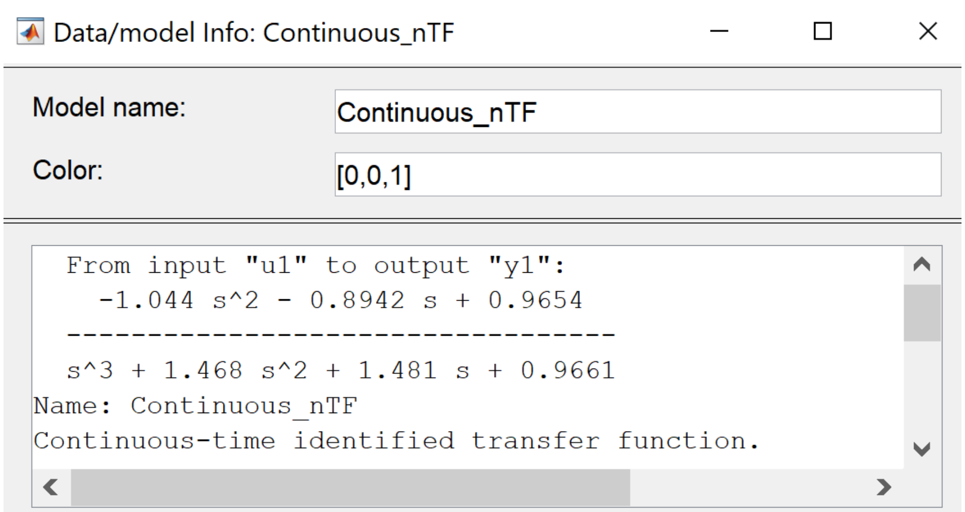


Simulated output of estimated transfer functions

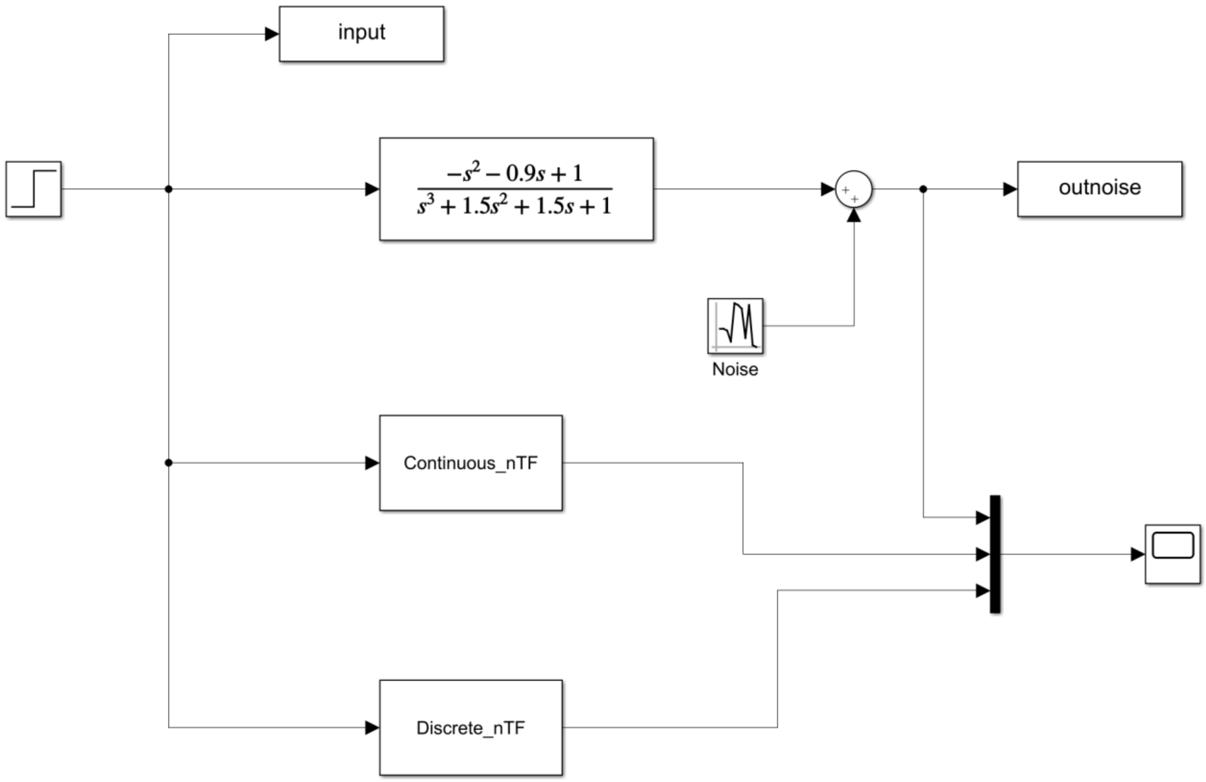




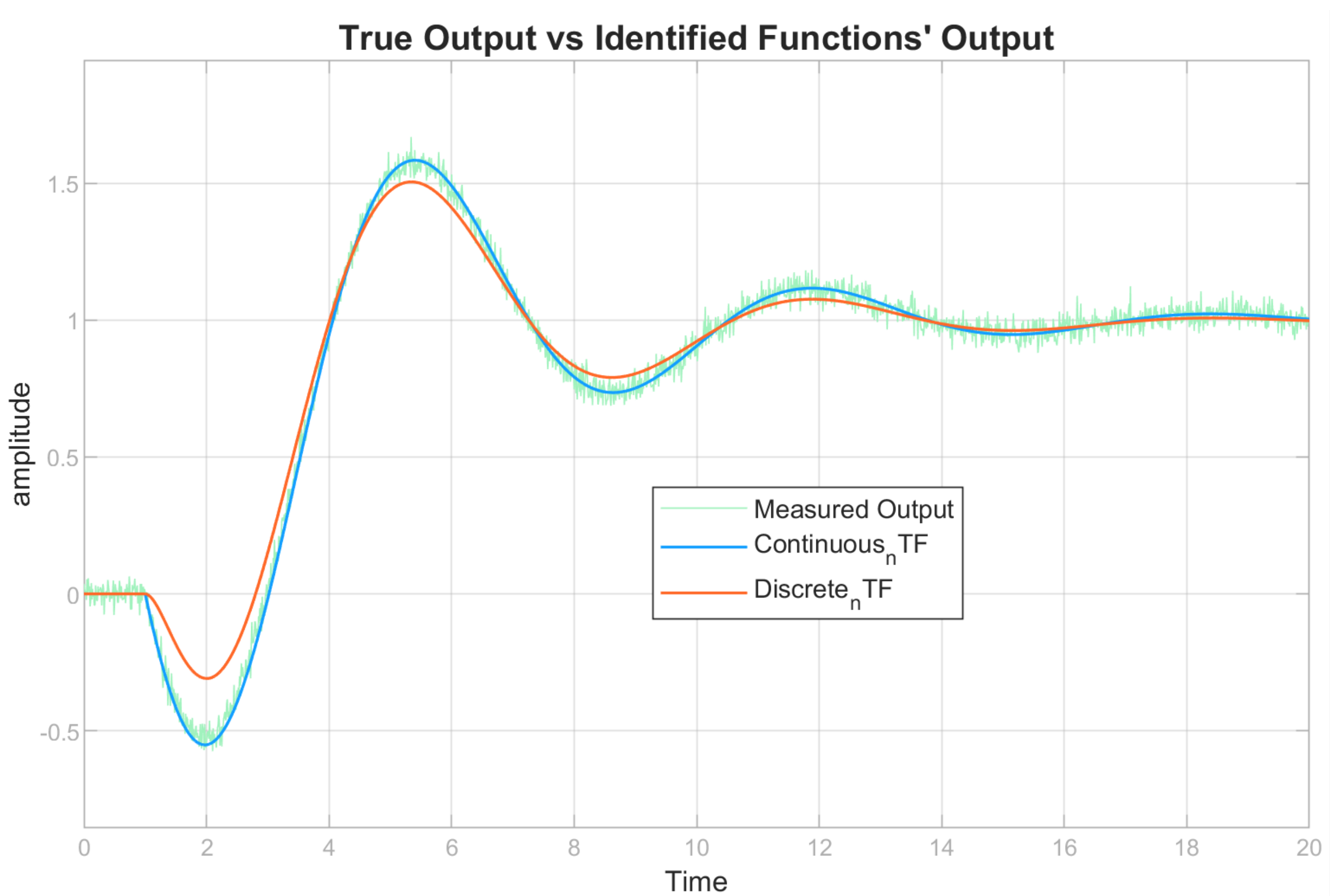


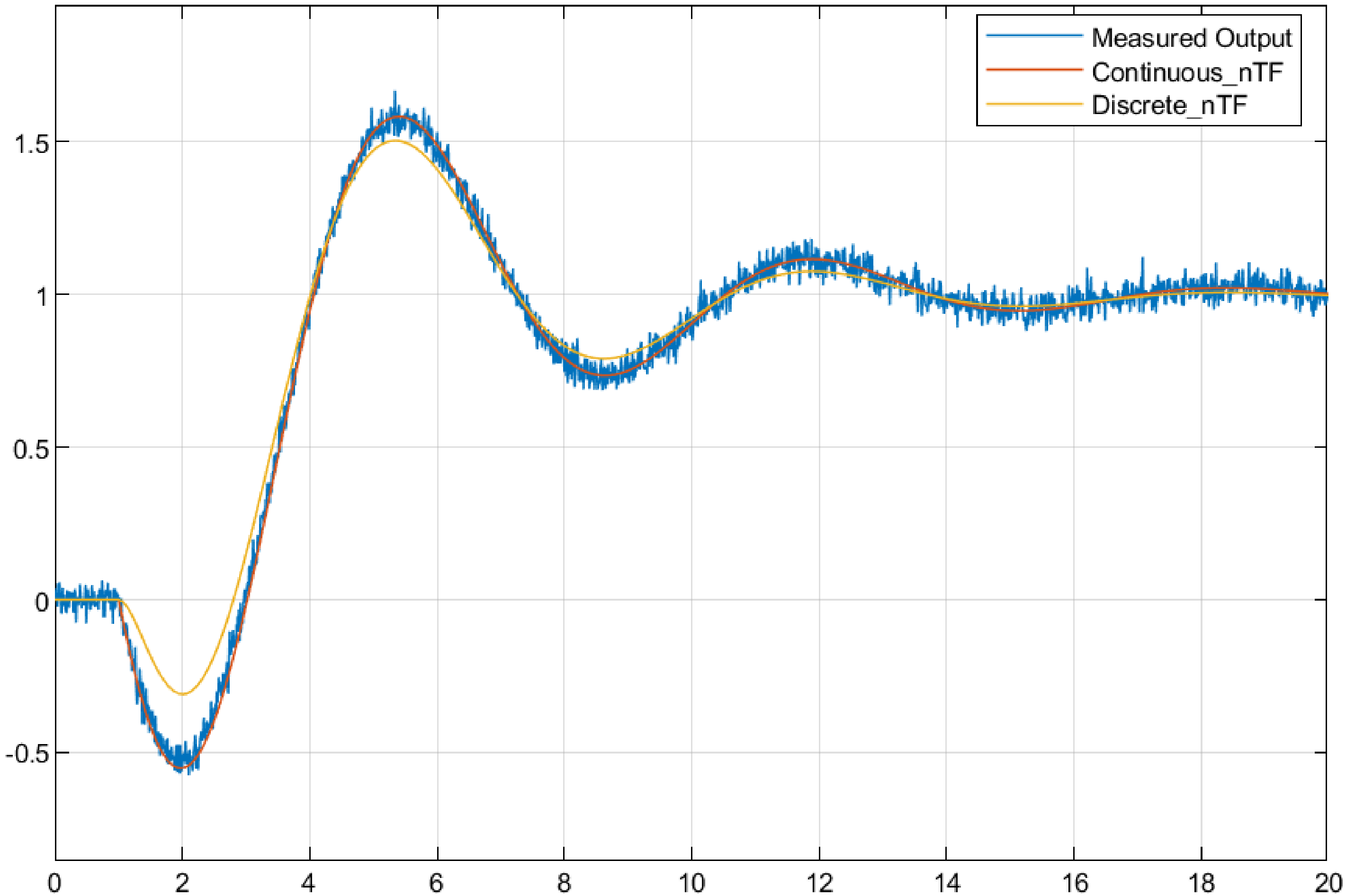


Comparison of poles and zeros of the identified transfer function with the true values



Simulink Block Diagram

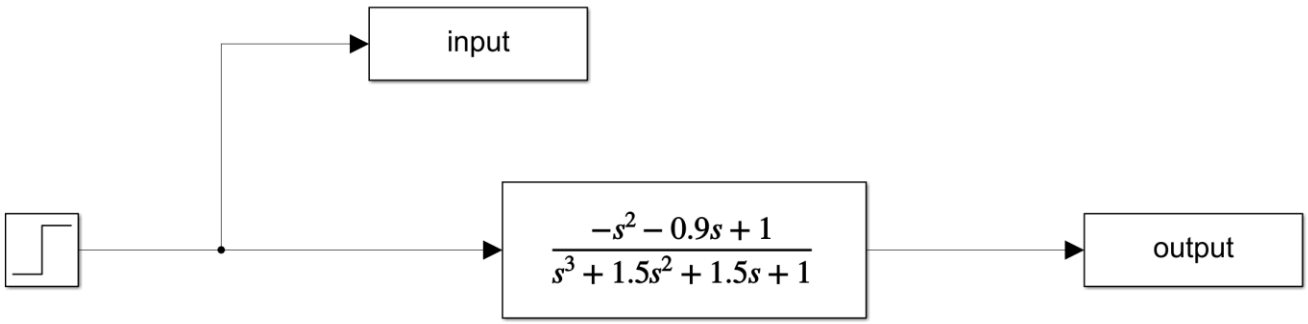




Conclusion – The continuous transfer function is similar to the original transfer function i.e., the poles and zeros are similar which can be confirmed using different commands such as damp or pole. The denominator of the discrete transfer function derived from the original transfer function is also similar to A(z) as observed in the previous parts.

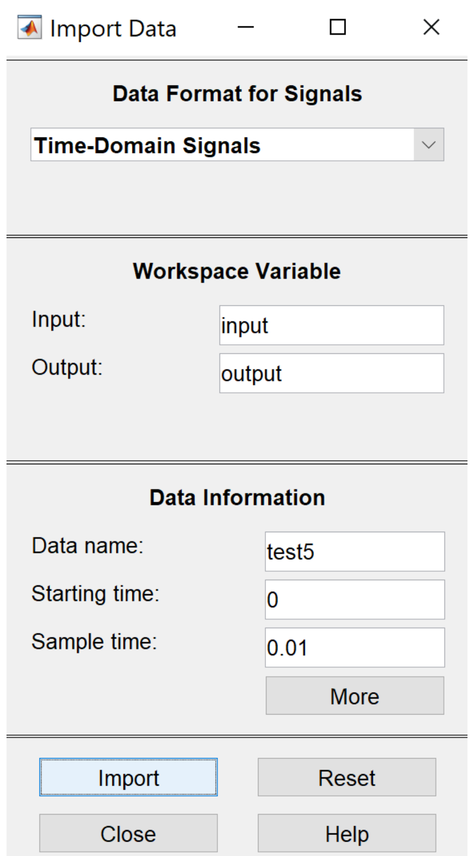
The system identification toolbox does a good job at calculating the transfer functions despite noise. It has an efficiency similar to the ARMAX model used in the previous problem. The continuous transfer function and discrete transfer function denominator are very similar to the original system.

Problem – 3

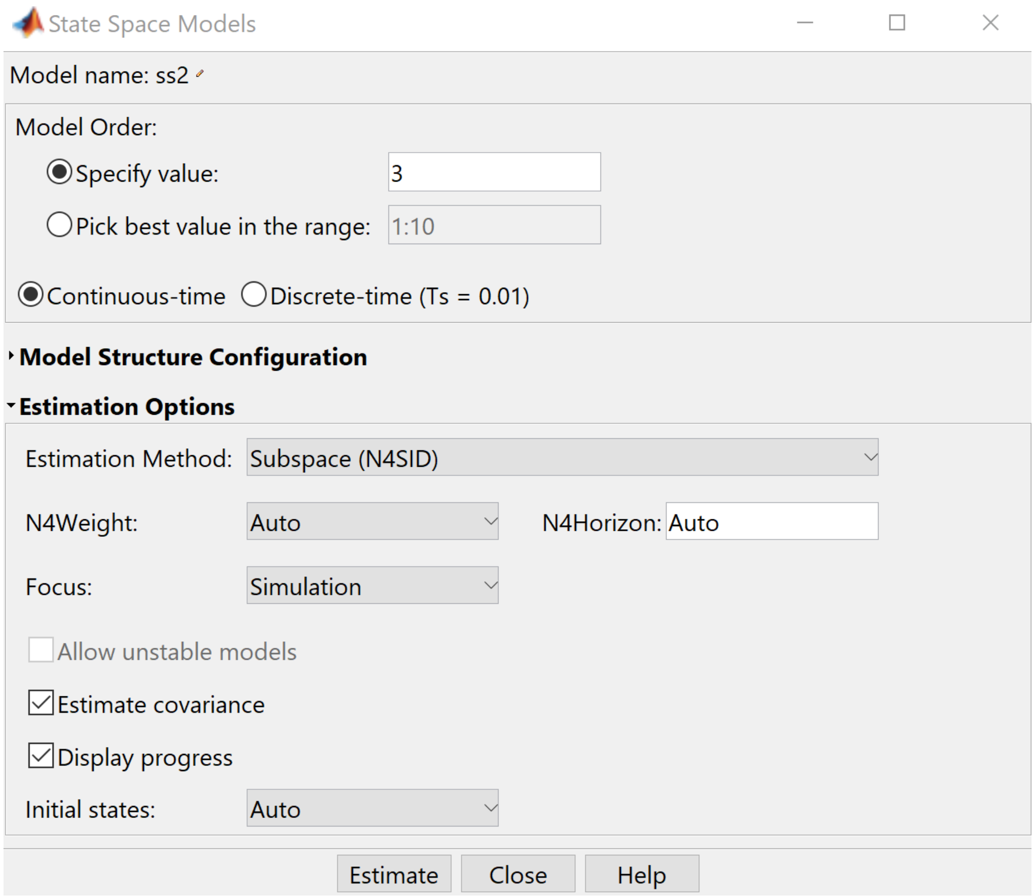


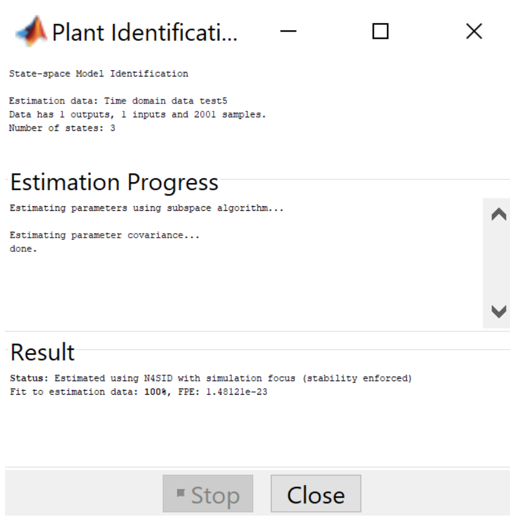
a.

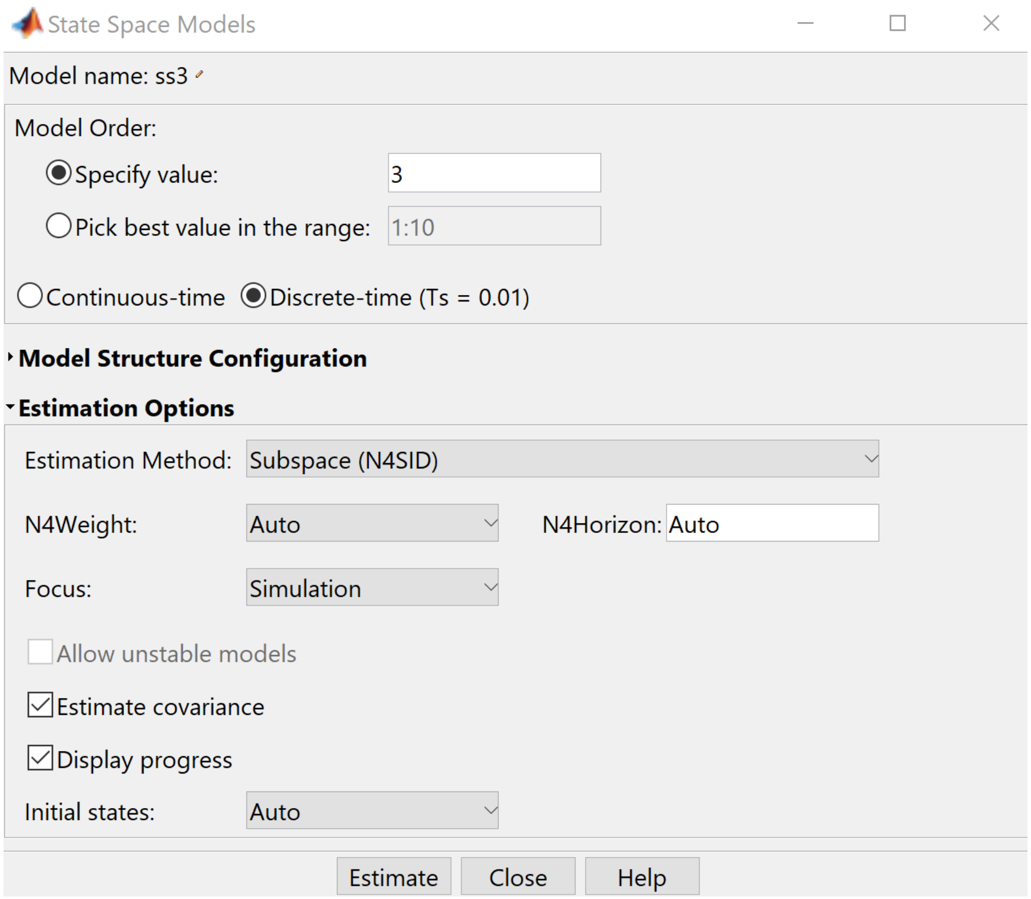
Simulink Block Diagram

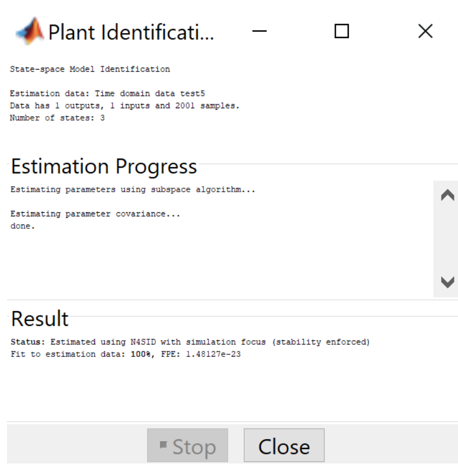


Importing data

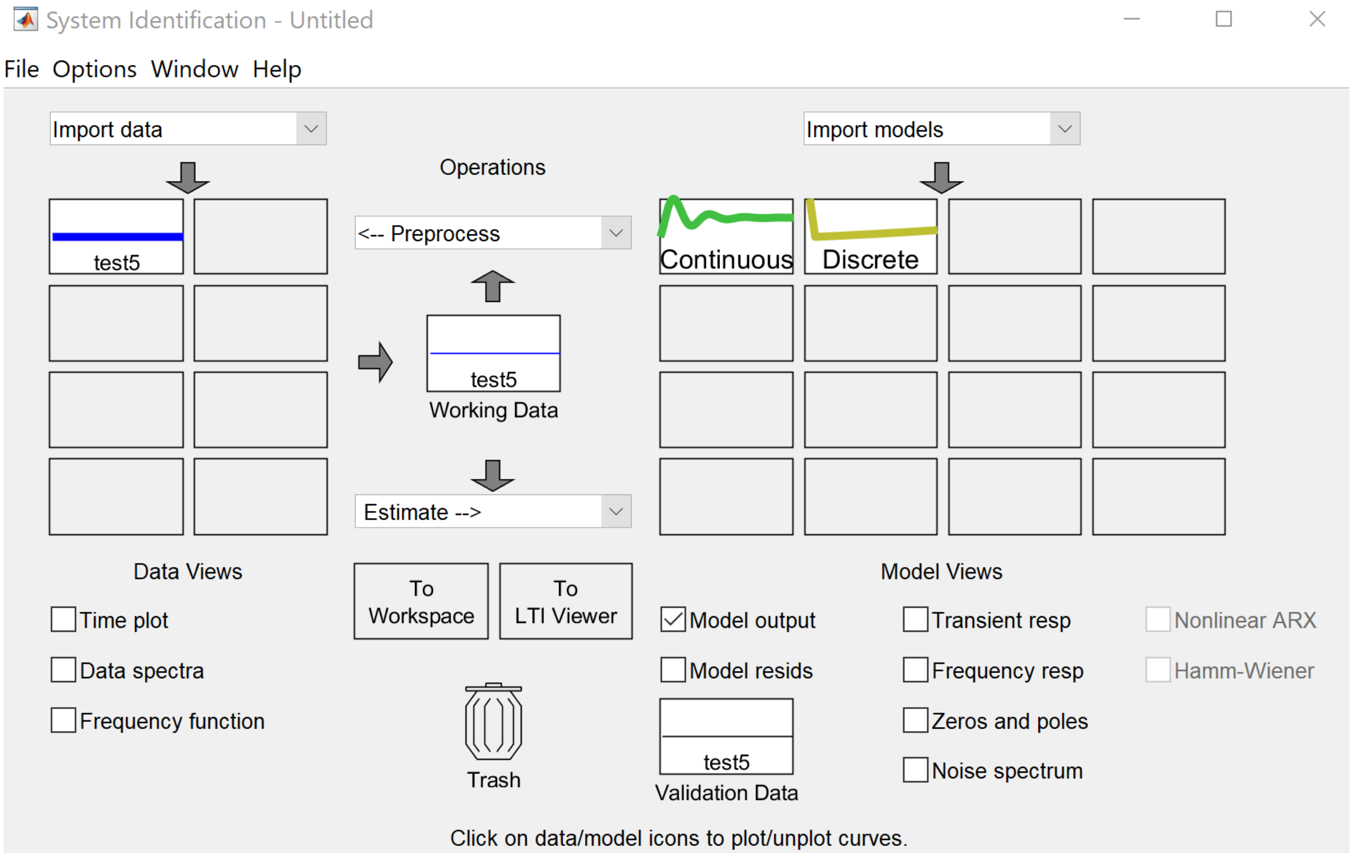

State Space Models
Model name: ss2
Model Order:
Specify value:
3
Pick best value in the range:
1:10
Continuous-time
Discrete-time (Ts = 0.01)
Model Structure Configuration
Estimation Options
Estimation Method:
Subspace (N4SID)
N4Weight:
Auto
N4Horizon:
Auto
Focus:
Simulation
Allow unstable models
Estimate covariance
Display progress
Initial states:
Auto
Estimate
Close
Help

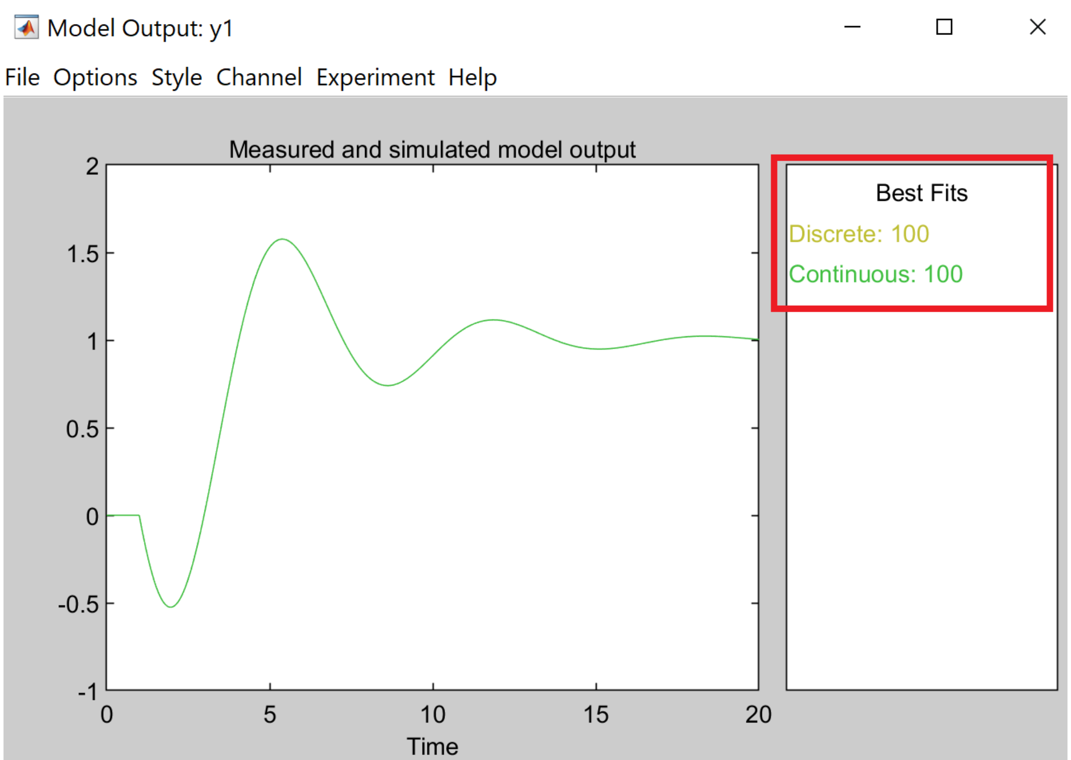

Plant Identificati...
State-space Model Identification
Estimation data: Time domain data test5
Data has 1 outputs, 1 inputs and 2001 samples.
Number of states: 3
Estimation Progress
Estimating parameters using subspace algorithm...
Estimating parameter covariance...
done.
Result
Status: Estimated using N4SID with simulation focus (stability enforced)
Fit to estimation data: 100%, FPE: 1.48121e-23
Stop
Close





Estimating continuous and discrete time models with 3rd order state space functions



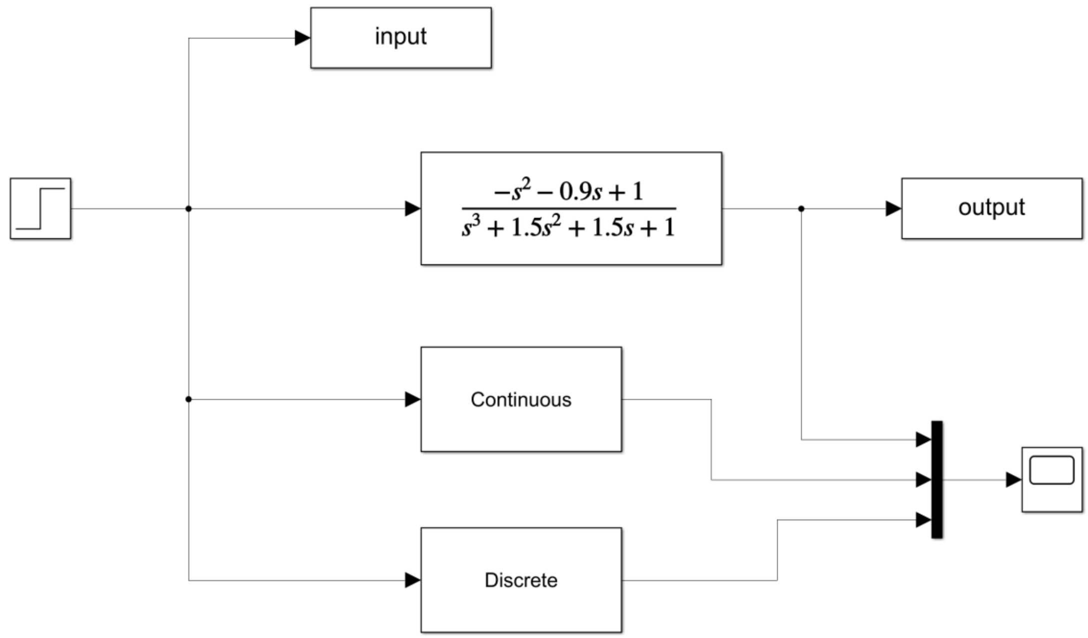


Both discrete and continuous estimations provide perfect results with a 100 percent fit.

b. MATLAB commands to compare transfer functions

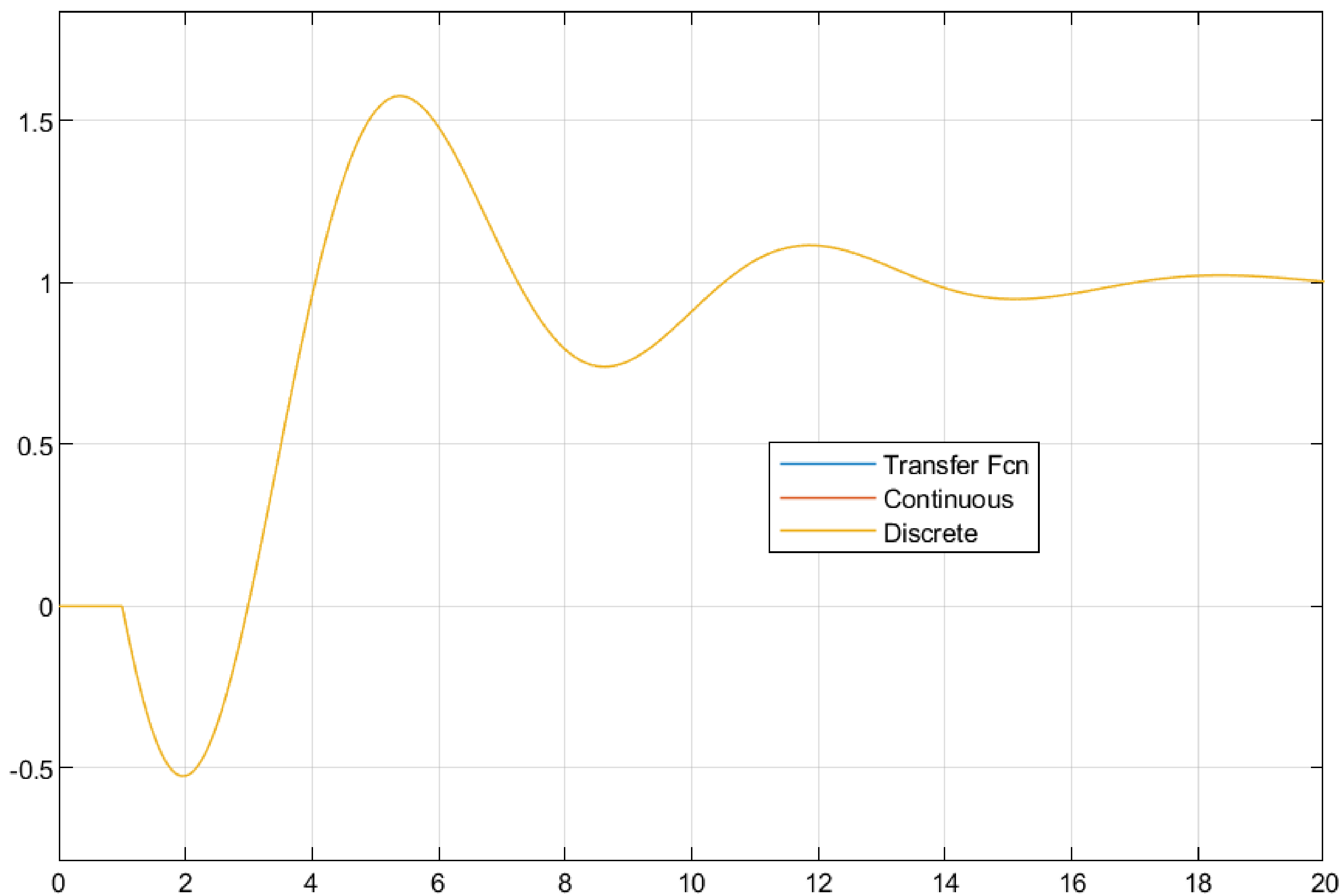
```
AC=[-1.499 -0.8319 -0.001454; 0.9955 -0.004427 0.8192; -0.001322 -0.8184 0.003508]

BC=[0.2601;-0.1717;-0.007281]

CC=[5.836 15.21 -12.98]

DC=0

[NC,DC]=ss2tf(AC,BC,CC,DC)

CTF=tf(NC,DC)

CTF =

  -0.9991 s^2 - 0.9003 s + 1
  --------------------------
  s^3 + 1.5 s^2 + 1.5 s + 1

Continuous-time transfer function.


AD=[0.9851 -0.008257 -4.833e-05;0.009881 0.9999 0.008191;-5.366e-05 -0.008184 1]

BD=[0.002589;-0.001704;-6.583e-05]

CD=[5.836 15.21 -12.98]

DD=0

[ND,DD]=ss2tf(AD,BD,CD,DD)

DTF=tf(ND,DD)

DiTF=c2d(DTF,0.01)

DiTF =

  -0.0001 z^2 + 0.0002021 z - 0.000102
  ------------------------------------
     z^3 - 3.03 z^2 + 3.06 z - 1.03

Sample time: 0.01 seconds

Discrete-time transfer function.
```

Conclusion – The continuous transfer function obtained from the estimated state space model is very similar to original transfer function i.e., the poles and zeros are also similar. The discrete transfer function obtained from the estimated discrete state space model has an A(z) very similar to the discrete transfer function obtained from the original transfer function.



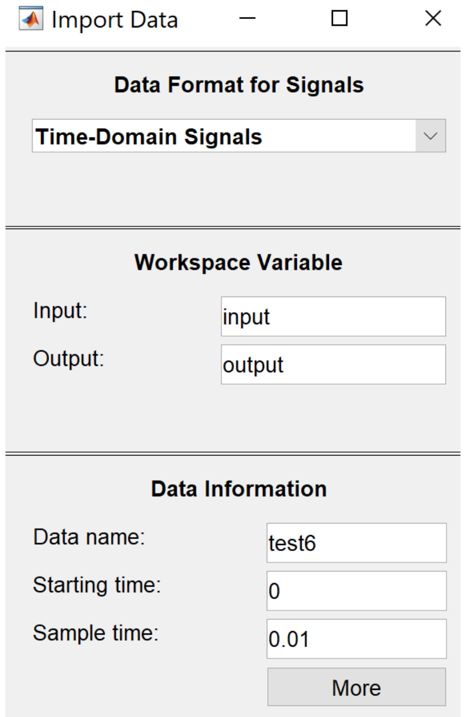
input
$\frac{-s^2 - 0.9s + 1}{s^3 + 1.5s^2 + 1.5s + 1}$
output
Continuous
Discrete

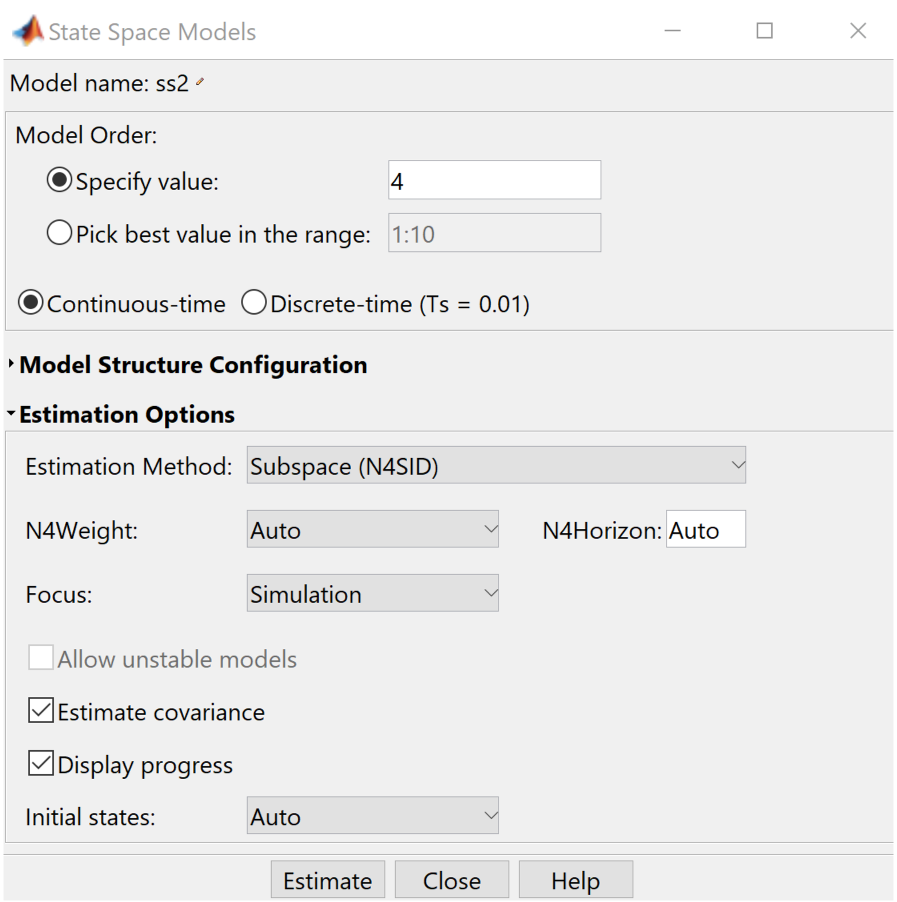
c.

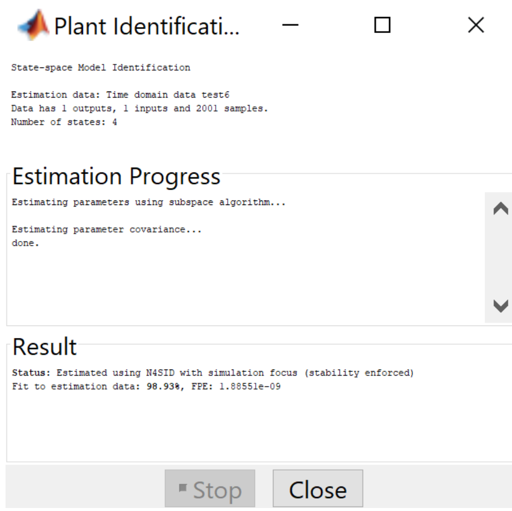


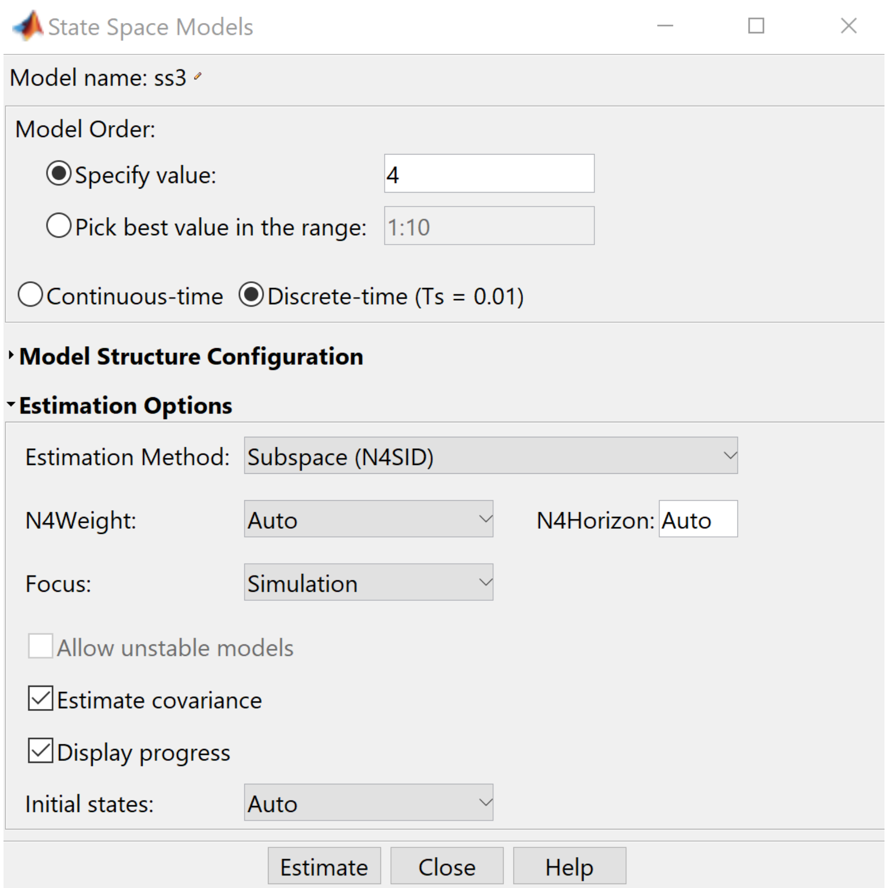
1.5
1
0.5
0
-0.5
0
2
4
6
8
10
12
14
16
18
20
Transfer Fcn
Continuous
Discrete

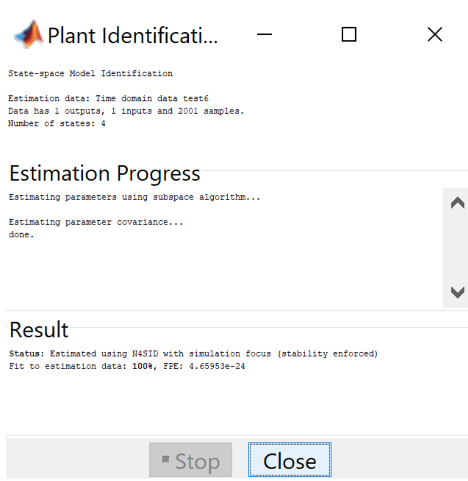
Now estimating with 4^{th} order

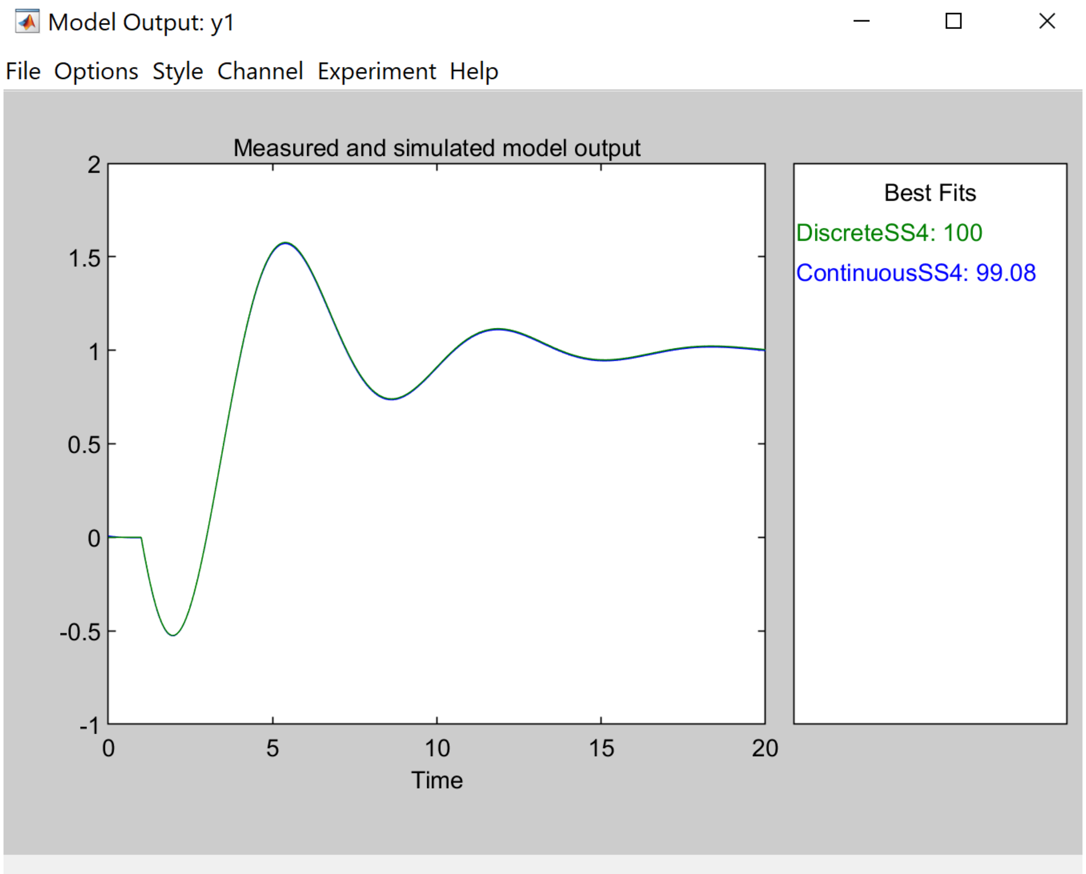




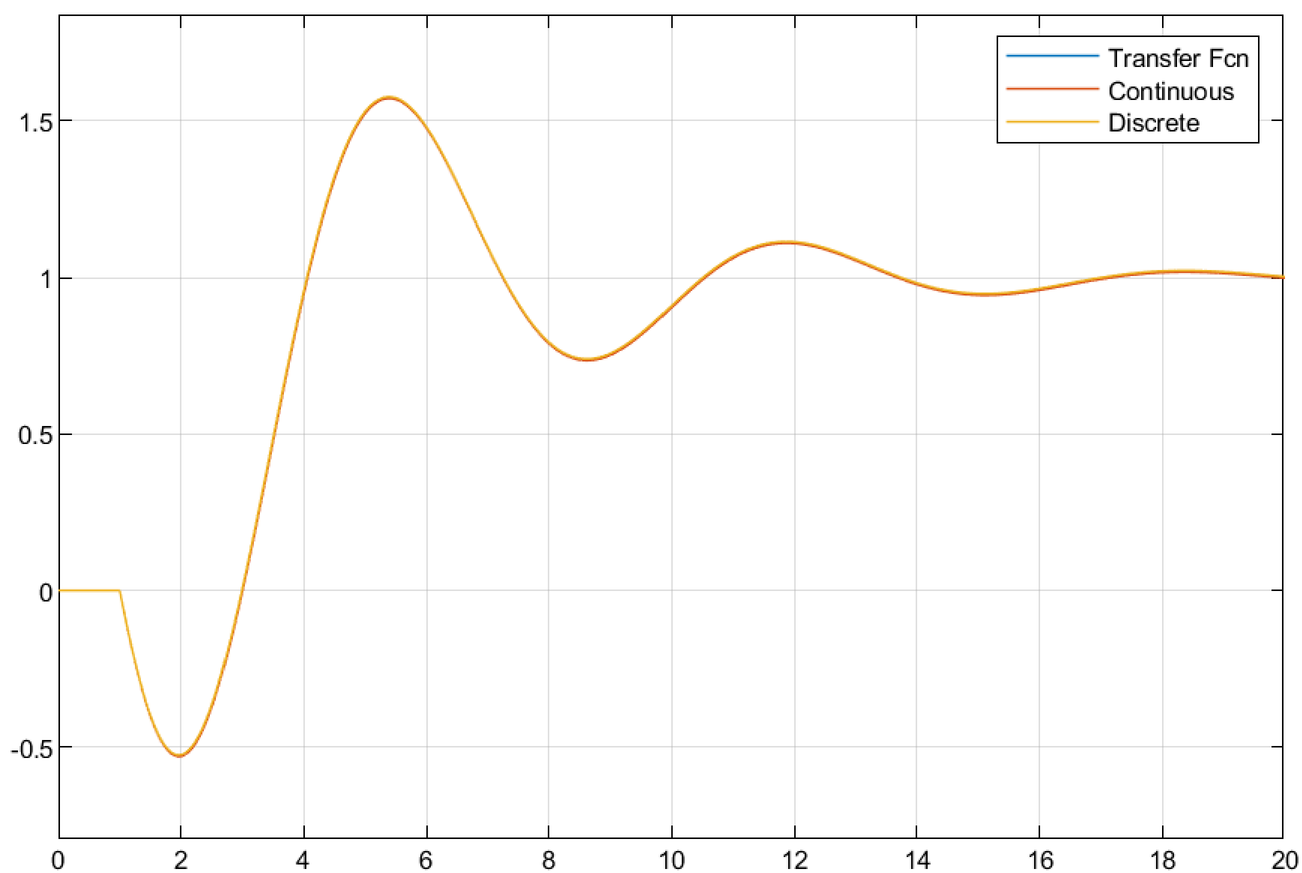

Plant Identificati...
State-space Model Identification
Estimation data: Time domain data test6
Data has 1 outputs, 1 inputs and 2001 samples.
Number of states: 4
Estimation Progress
Estimating parameters using subspace algorithm...
Estimating parameter covariance...
done.
Result
Status: Estimated using N4SID with simulation focus (stability enforced)
Fit to estimation data: 98.93%, FPE: 1.88551e-09
Stop
Close


State Space Models
Model name: ss3
Model Order:
Specify value:
4
Pick best value in the range:
1:10
Continuous-time
Discrete-time (Ts = 0.01)
Model Structure Configuration
Estimation Options
Estimation Method:
Subspace (N4SID)
N4Weight:
Auto
N4Horizon:
Auto
Focus:
Simulation
Allow unstable models
Estimate covariance
Display progress
Initial states:
Auto
Estimate
Close
Help

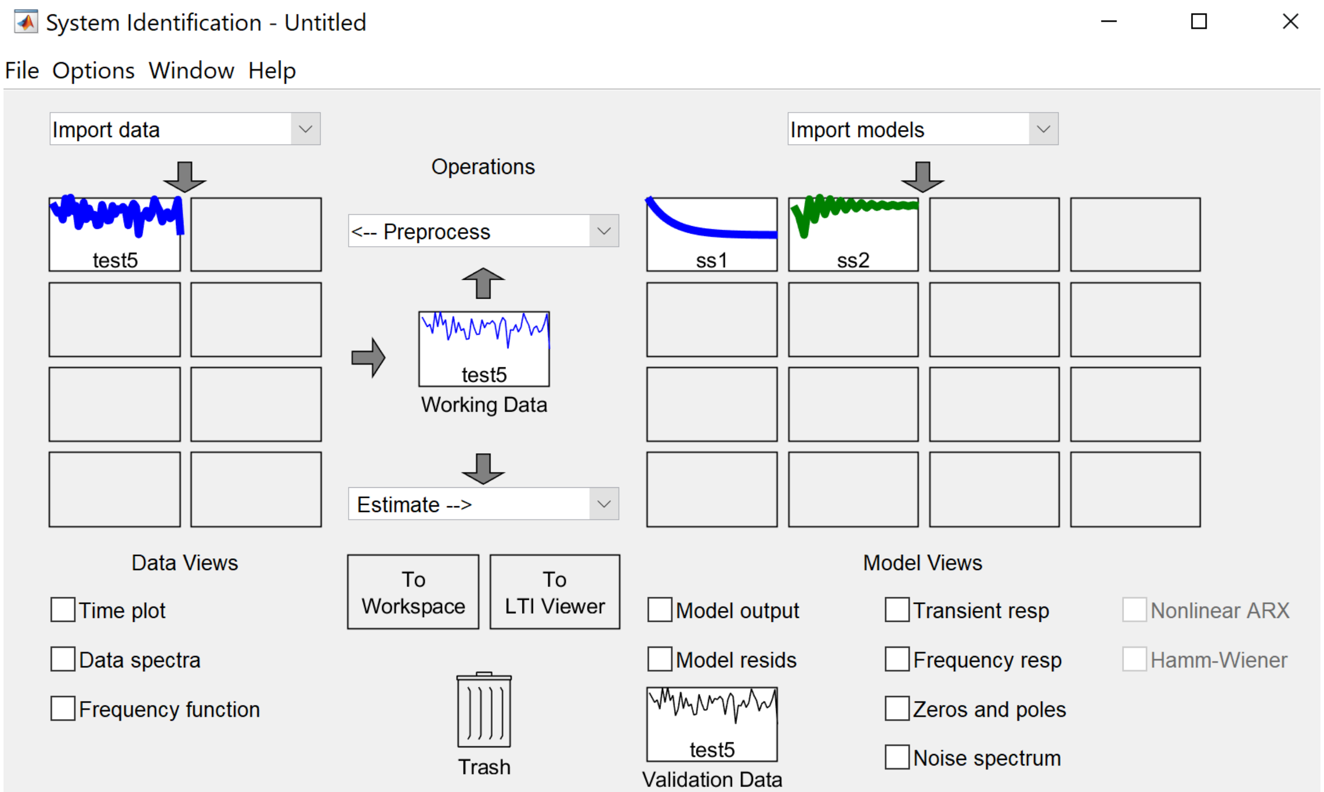

Plant Identificati...
State-space Model Identification
Estimation data: Time domain data test6
Data has 1 outputs, 1 inputs and 2001 samples.
Number of states: 4
Estimation Progress
Estimating parameters using subspace algorithm...
Estimating parameter covariance...
done.
Result
Status: Estimated using N4SID with simulation focus (stability enforced)
Fit to estimation data: 100%, FPE: 4.65953e-24
Stop
Close

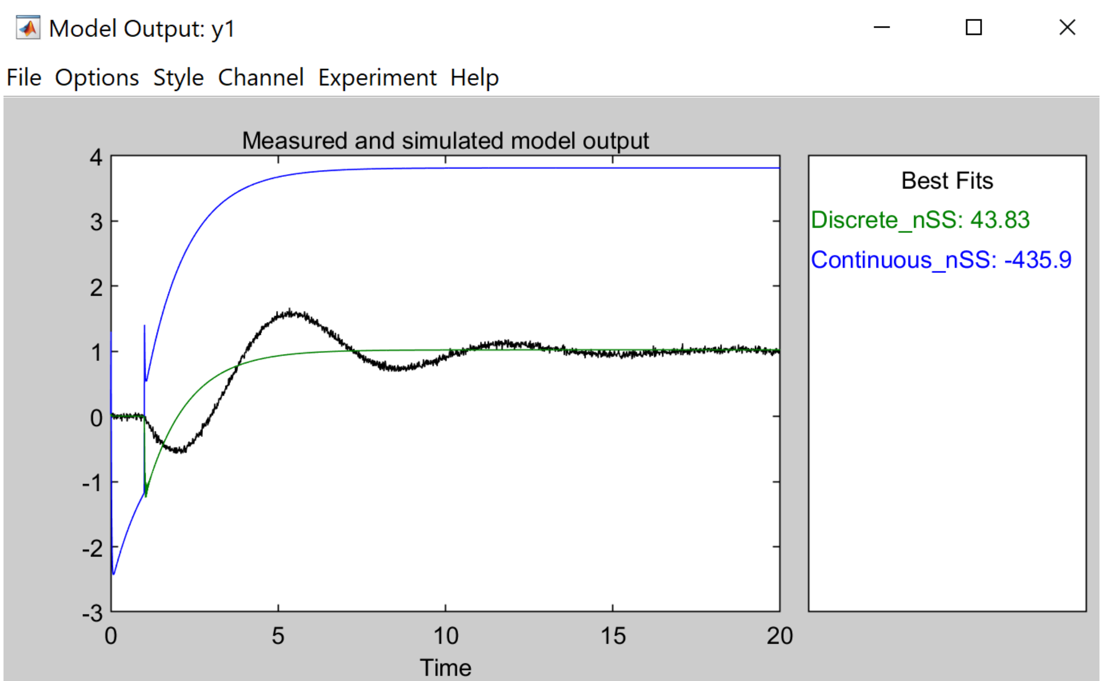


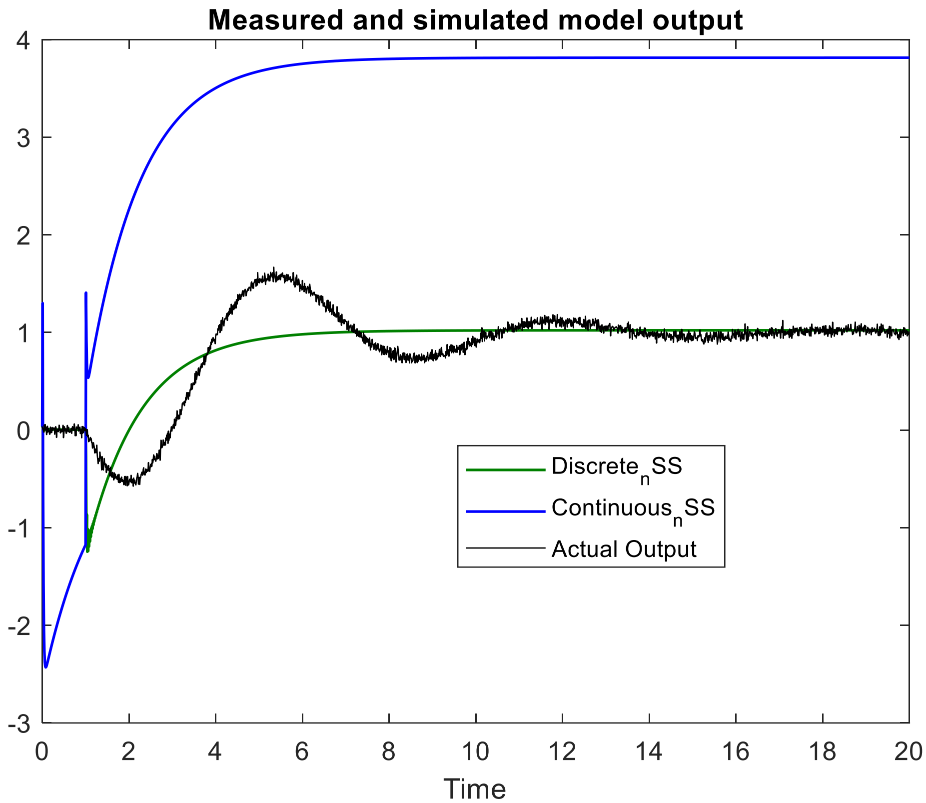
It’s observed that the fit for continuous state space model is slightly lower than before



d.


System Identification - Untitled
File Options Window Help
Import data
Import models
Operations
test5
<-- Preprocess
ss1
ss2
test5
Working Data
Estimate -->
Data Views
Time plot
Data spectra
Frequency function
To Workspace
To LTI Viewer
Trash
Model Views
Model output
Model resids
test5
Validation Data
Transient resp
Frequency resp
Zeros and poles
Noise spectrum
Nonlinear ARX
Hamm-Wiener


Model Output: y1
File Options Style Channel Experiment Help
Measured and simulated model output
Best Fits
Discrete_nSS: 43.83
Continuous_nSS: -435.9
Time



Noise completely throws off the state space estimation, therefore, I used a range.

MATLAB Code for getting the transfer functions

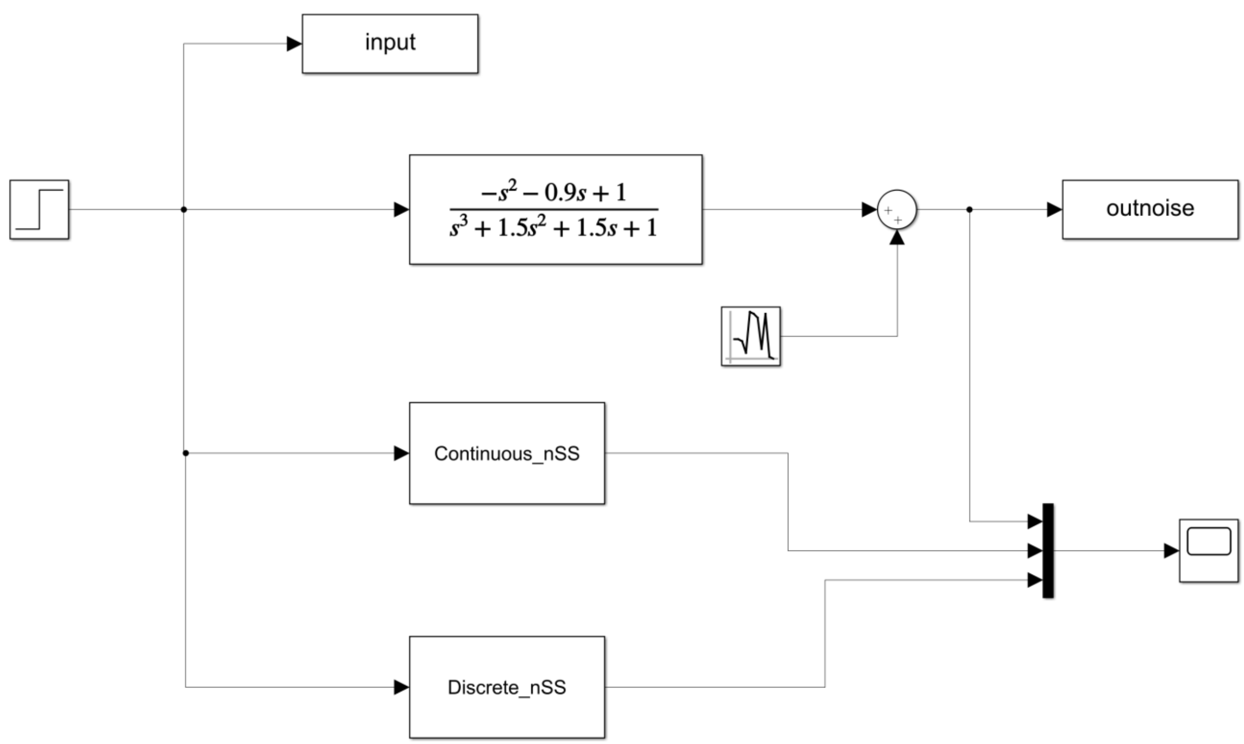
```
ACN=[-0.5403 109.8 2.181;-31.24 -2531 506.7;-1.027 -320.2 -1.48]

BCN=[-15.7;744.5;49.96]

CCN=[18.84 11.55 7.518]

DCN=0

[NCN,DCN]=ss2tf(ACN,BCN,CCN,DCN)

CTFN=tf(NCN,DCN)

CTFN =

     8679 s^2 + 2.673e05 s + 5.245e05
  --------------------------------------
  s^3 + 2533 s^2 + 1.708e05 s + 1.358e05

Continuous-time transfer function.
```

Note – DCN is repeated but by the time it is repeated, the first value becomes irrelevant

```
ADN=[0.9844 0.05965 0.1715;-0.01898 -0.8875 0.2828;0.02005 -0.1792 0.5349]

BDN=[0.1091;0.491;-0.2851]

CDN=[18.66 0.5385 9.019]

DDN=0
```

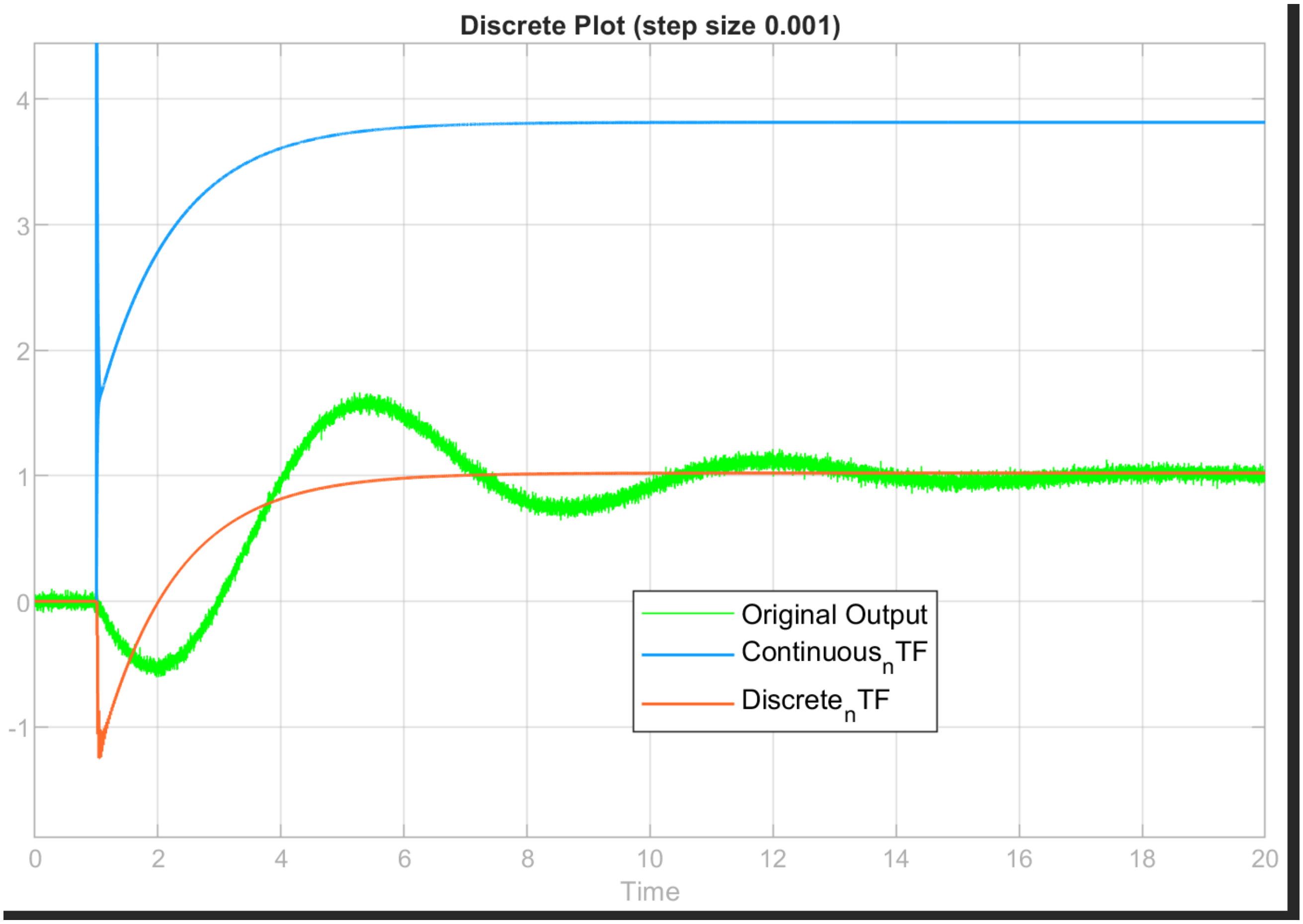
```
[NTDN,DTDN]=ss2tf(ADN,BDN,CDN,DDN)

DTFN=tf(NTDN,DTDN)

DTFNf=c2d(DTFN,0.01)

DTFNf =

  -0.002751 z^2 + 0.00544 z - 0.002688
  ------------------------------------
   z^3 - 3.006 z^2 + 3.013 z - 1.006

Sample time: 0.01 seconds

Discrete-time transfer function.
```

Conclusion – The continuous transfer function is completely different from the original as expected from the fit values i.e., the poles and zeros are different. The denominator of the discrete transfer function remains similar to the discrete transfer function values derived from the original transfer function. However, despite that, it is only able to attain a fit value of 43.83

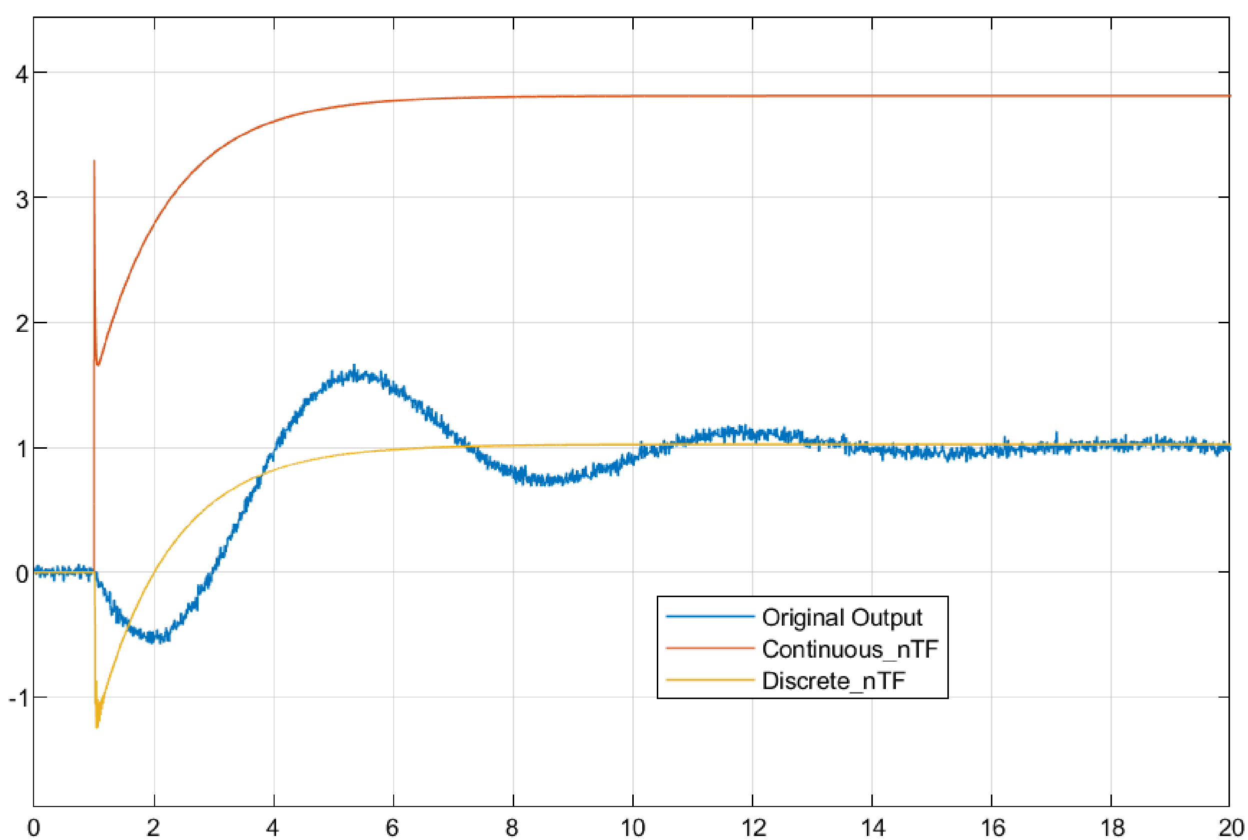


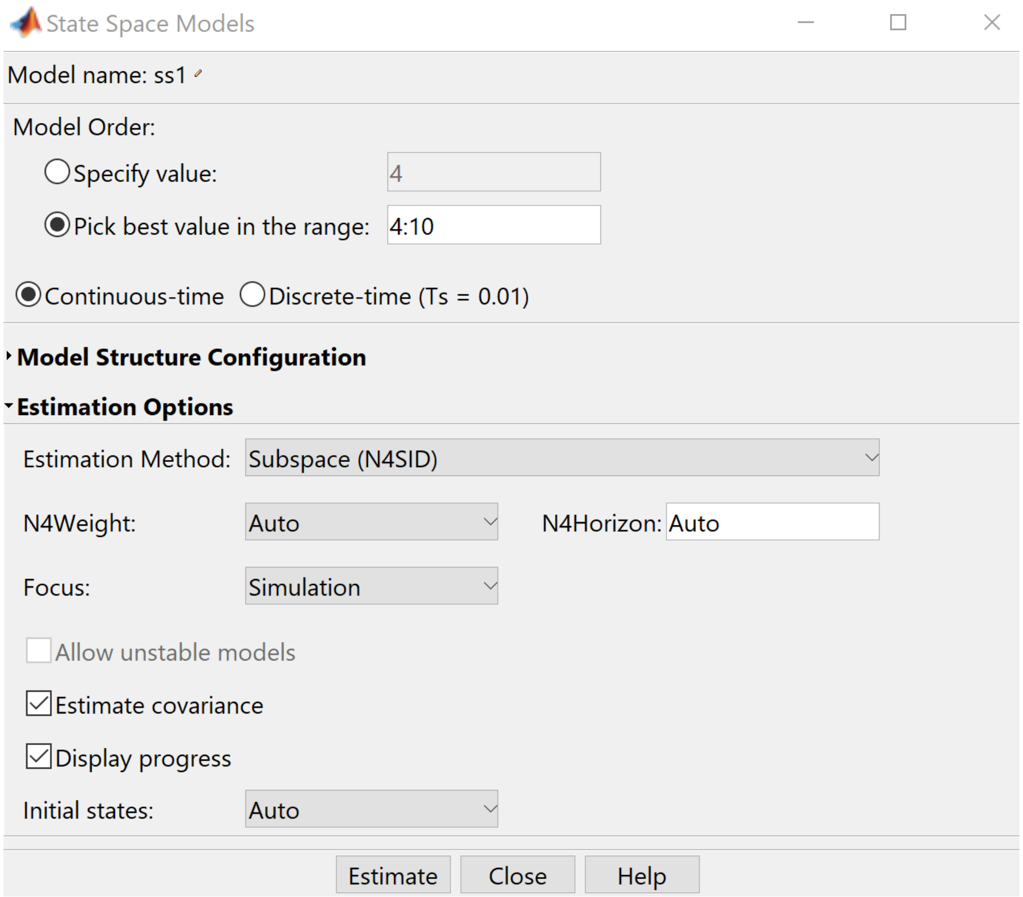
Simulink Block Diagram

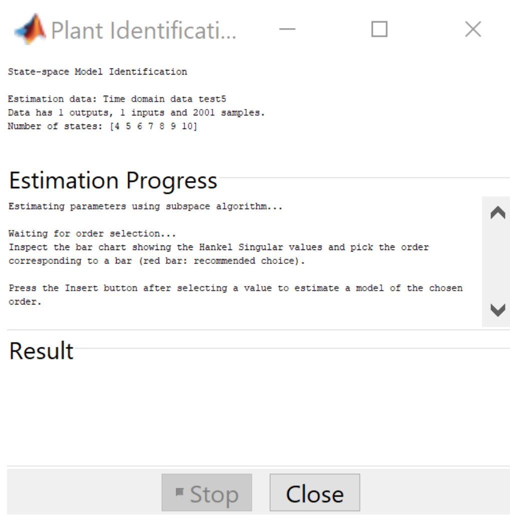
The simulation failed to run using fixed step size of 0.01 and requested me to either reduce the step size or reduce error tolerances. I changed it to variable step and received the outputs. Then I also reduced the step size to 0.001 and received the same outputs on the scope plots. Both are shown below.

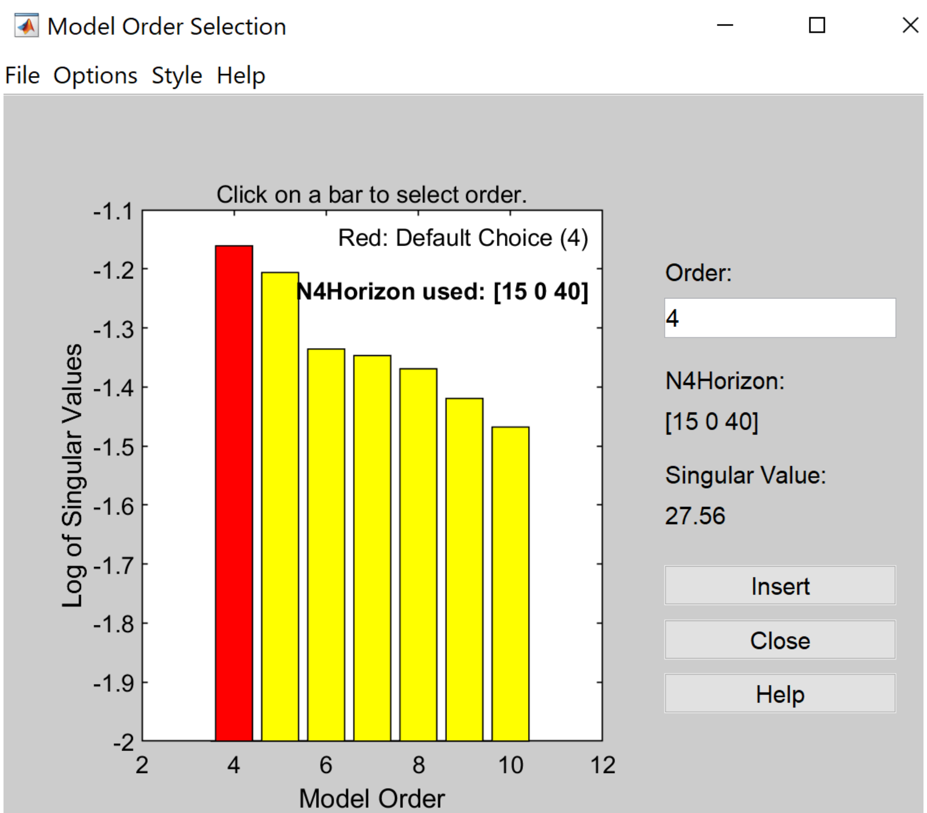


Plot using variable time step settings in Simulink

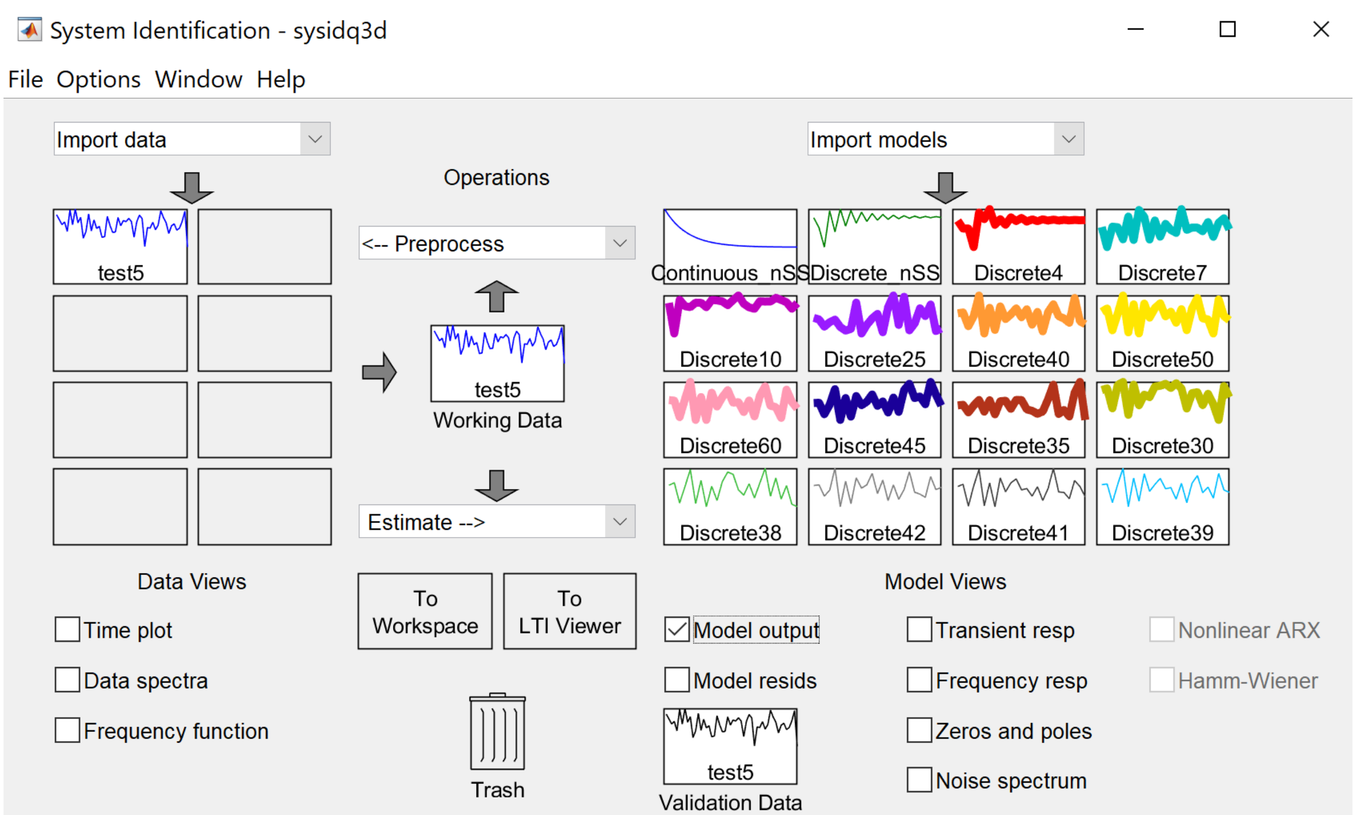


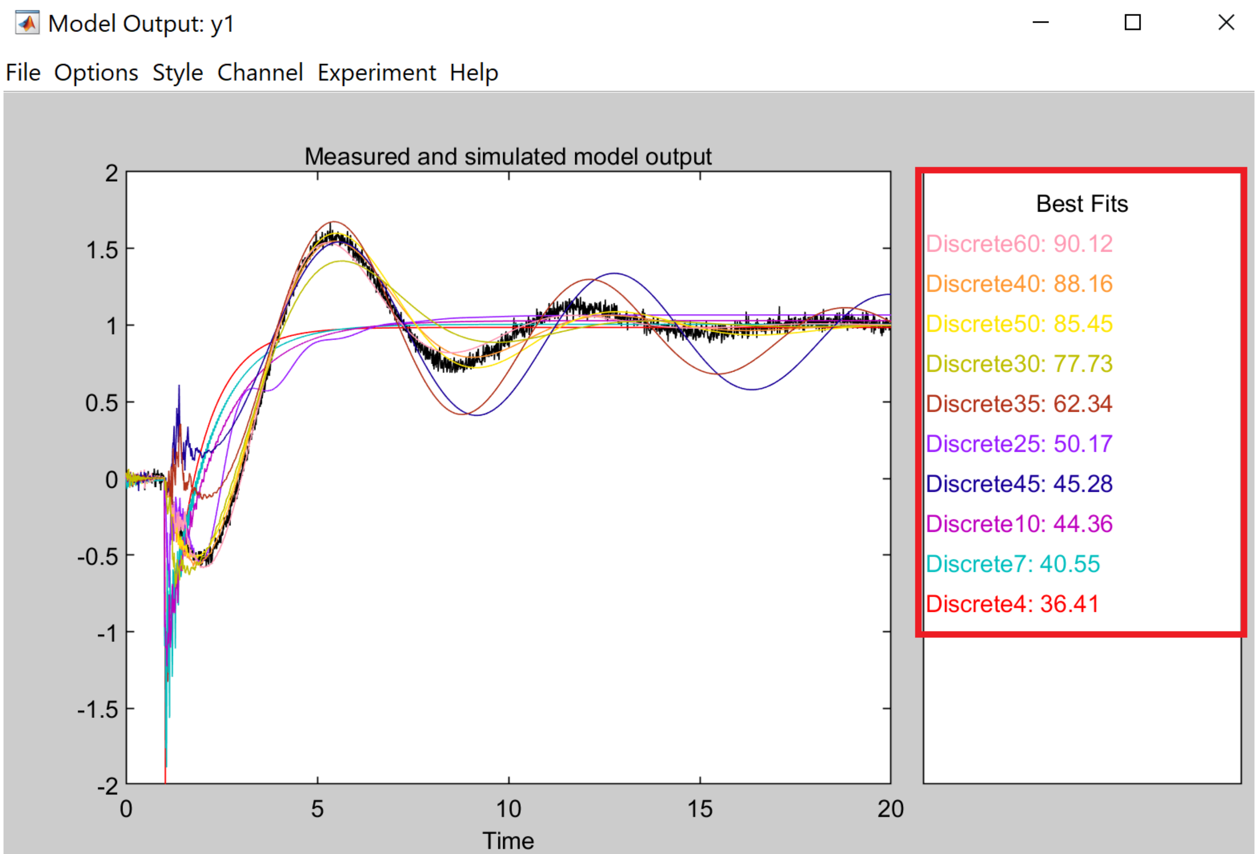

State Space Models
Model name: ss1
Model Order:
Specify value:
4
Pick best value in the range:
4:10
Continuous-time
Discrete-time (Ts = 0.01)
Model Structure Configuration
Estimation Options
Estimation Method:
Subspace (N4SID)
N4Weight:
Auto
N4Horizon:
Auto
Focus:
Simulation
Allow unstable models
Estimate covariance
Display progress
Initial states:
Auto
Estimate
Close
Help

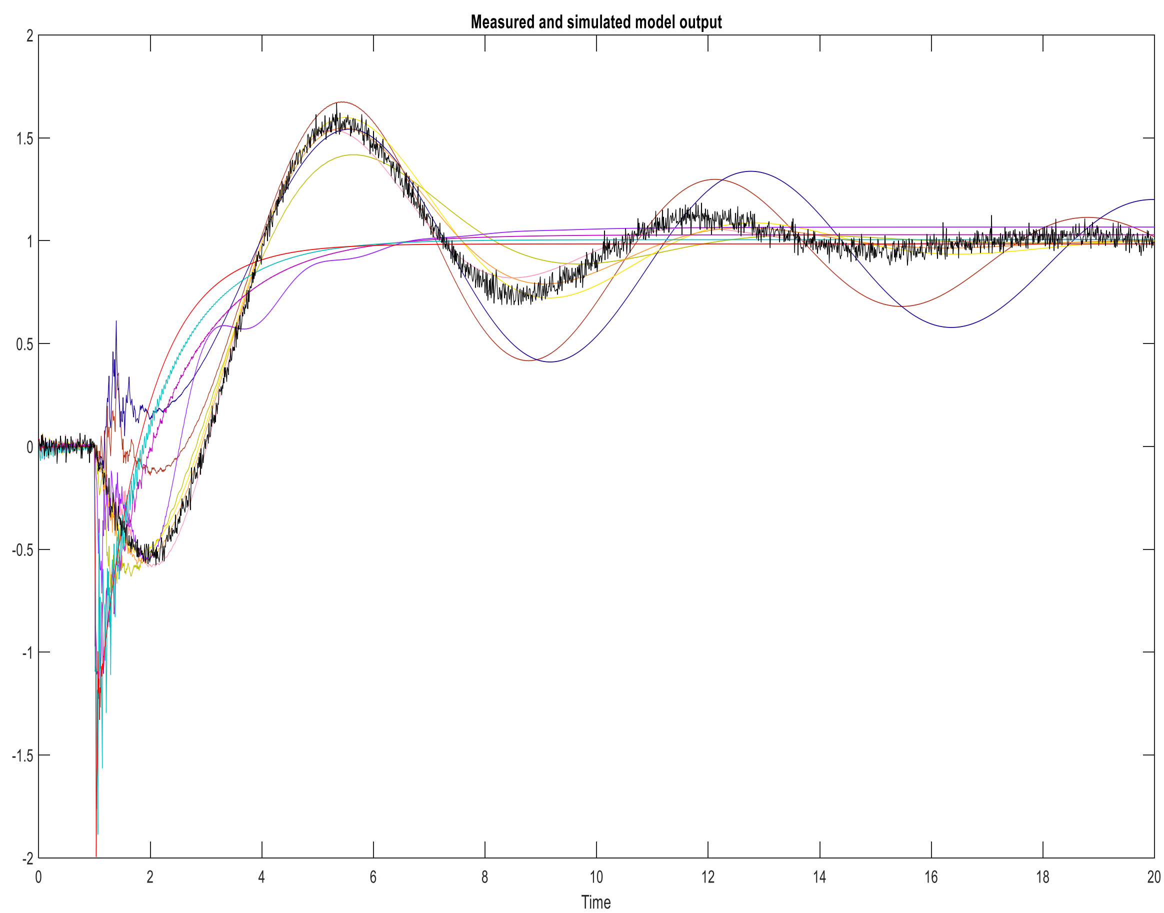

Plant Identificati...
State-space Model Identification
Estimation data: Time domain data test5
Data has 1 outputs, 1 inputs and 2001 samples.
Number of states: [4 5 6 7 8 9 10]
Estimation Progress
Estimating parameters using subspace algorithm...
Waiting for order selection...
Inspect the bar chart showing the Hankel Singular values and pick the order corresponding to a bar (red bar: recommended choice).
Press the Insert button after selecting a value to estimate a model of the chosen order.
Result
Stop
Close

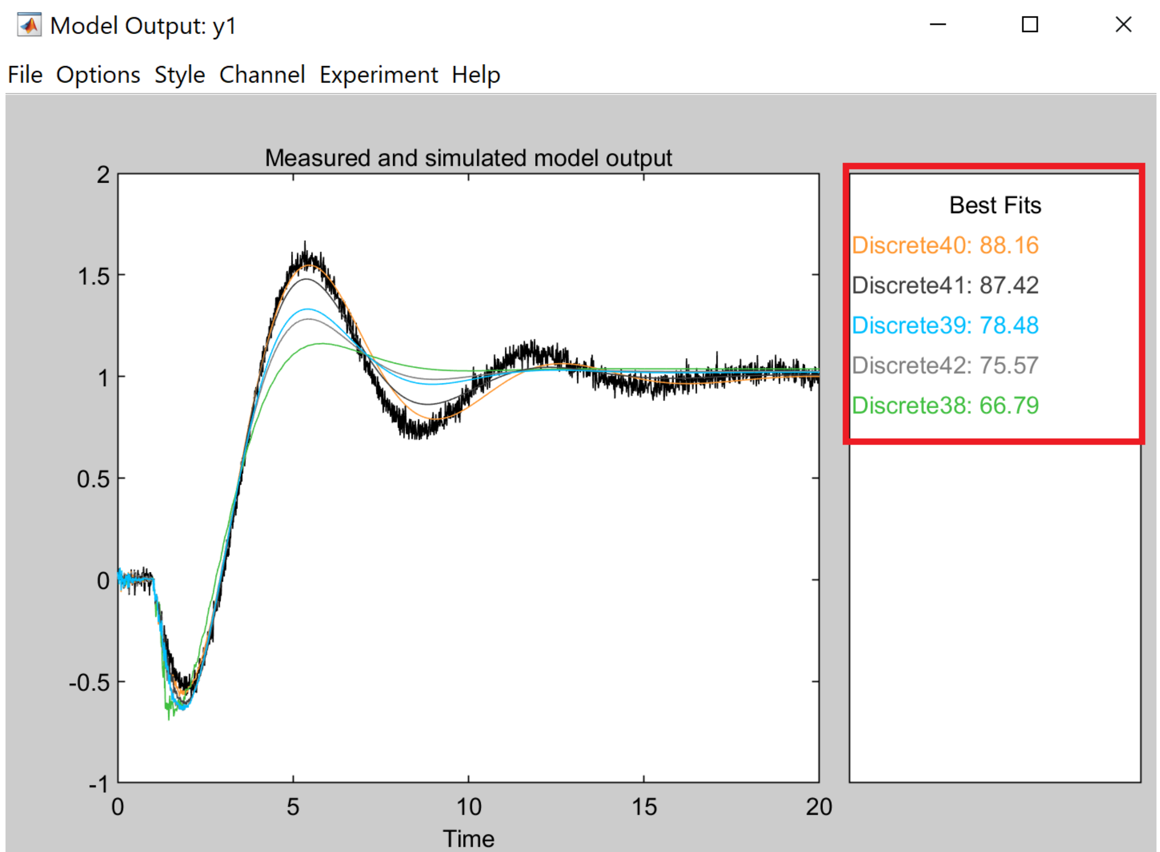


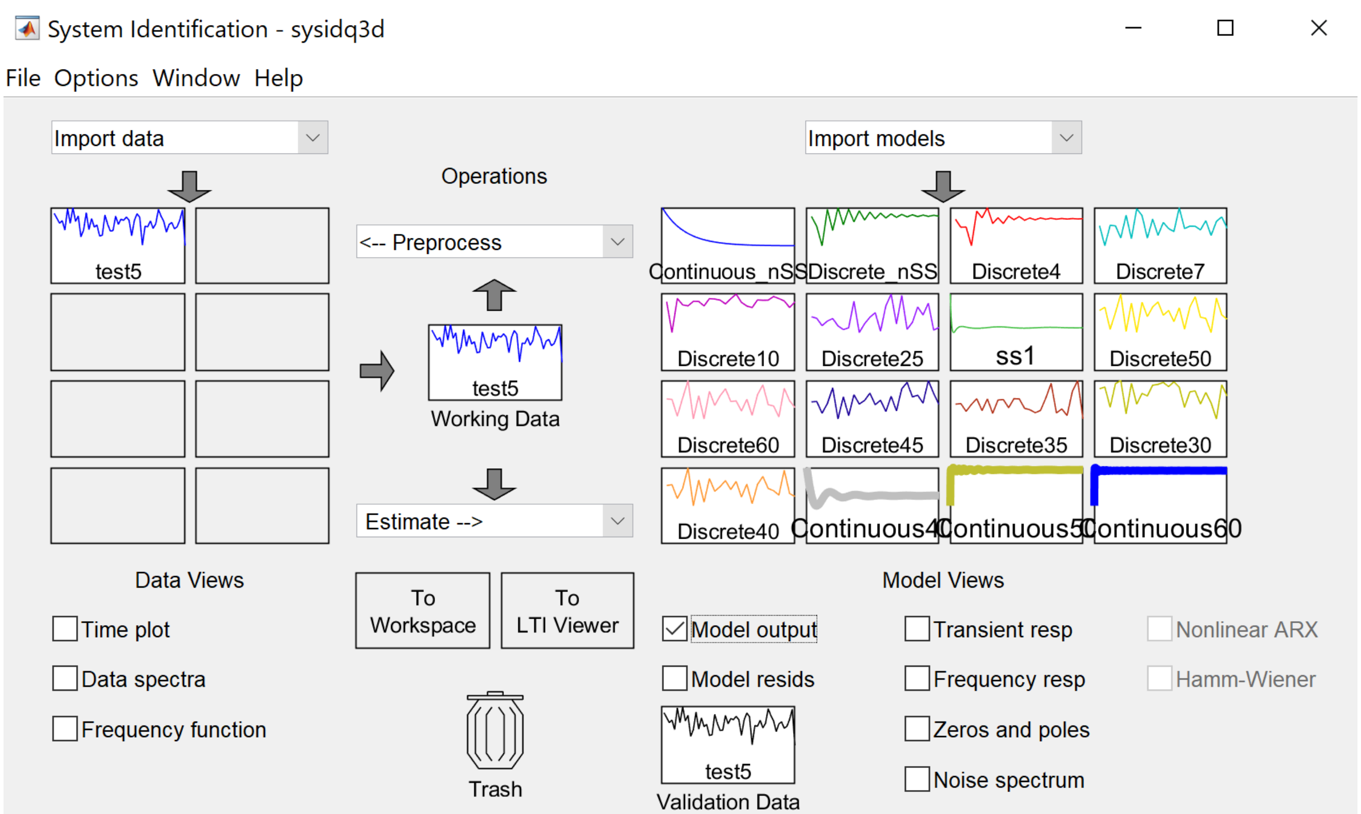
Using the range, the suggested order was 4 which lowered the fit value to 36.41 so I reverted back to increasing the order of the estimated function to see the effects on estimated output. I didn’t experiment much with the continuous estimation as it took too much time for estimation process.

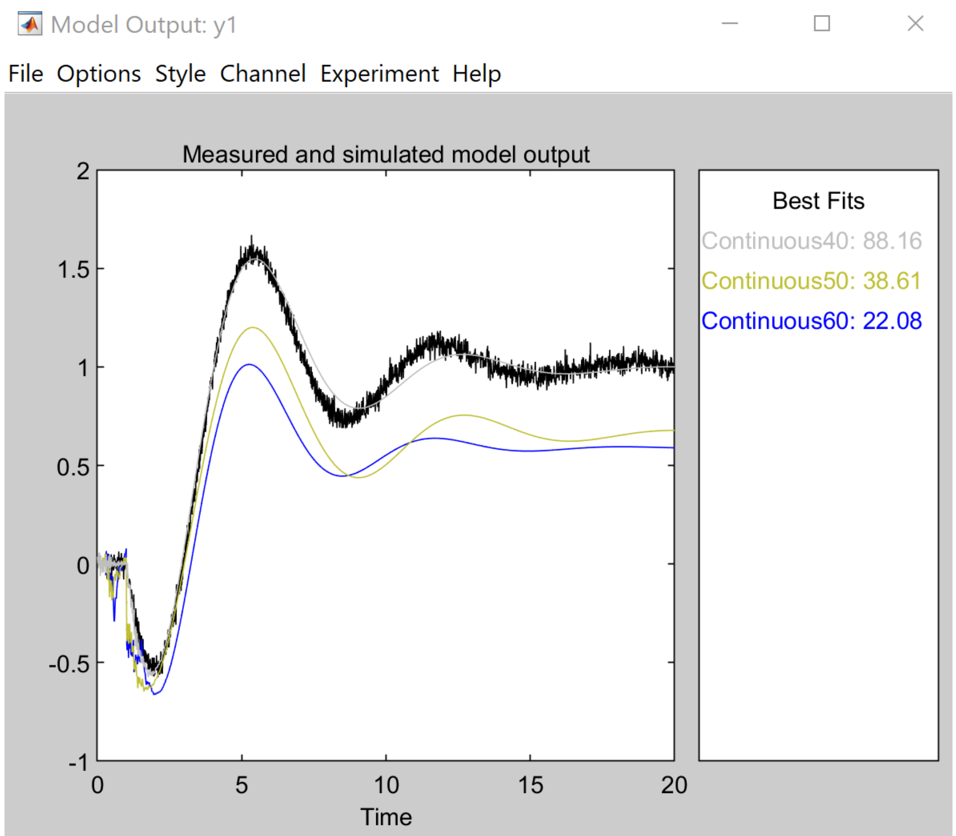



Model Output: y1
File Options Style Channel Experiment Help
Measured and simulated model output
Time
Best Fits
Discrete60: 90.12
Discrete40: 88.16
Discrete50: 85.45
Discrete30: 77.73
Discrete35: 62.34
Discrete25: 50.17
Discrete45: 45.28
Discrete10: 44.36
Discrete7: 40.55
Discrete4: 36.41


Measured and simulated model output
Time


Model Output: y1
File Options Style Channel Experiment Help
Measured and simulated model output
Best Fits
Discrete40: 88.16
Discrete41: 87.42
Discrete39: 78.48
Discrete42: 75.57
Discrete38: 66.79
Time
2
1.5
1
0.5
0
-0.5
-1
0
5
10
15
20


System Identification - sysidq3d
File Options Window Help
Import data
Operations
Import models
test5
<-- Preprocess
Continuous_nSS
Discrete_nSS
Discrete4
Discrete7
Discrete10
Discrete25
ss1
Discrete50
test5
Working Data
Discrete60
Discrete45
Discrete35
Discrete30
Estimate -->
Discrete40
Continuous40
Continuous50
Continuous60
Data Views
To Workspace
To LTI Viewer
Model Views
Time plot
Model output
Transient resp
Nonlinear ARX
Data spectra
Model resids
Frequency resp
Hamm-Wiener
Frequency function
Zeros and poles
Trash
test5
Validation Data
Noise spectrum



Conclusion – Using the hit and trial method, I observed two things for discrete estimation. Firstly, it takes a model of order 60 to take the fit value above 90. Secondly, there is a peak in fit value for the order 40 which might be due to the way the system identification toolbox is set up in MATLAB as explained by Dr. Niestroy during class.

The hit and trial method took a long time with continuous estimation once the order reached values of 20. However, the peak was observed at order 40 similar to what was seen in the case of discrete estimation. But the fits don't improve as the order is increased till 60.

Note: There were many other orders I ran the continuous function with but the documentation started getting messy so I only kept the ones that were important.

e. Noise definitely hurts the estimation as the fits are reduced. For the state space model estimation, noise is highly detrimental as it takes an order of 60 to get the fit value over 90 for the discrete estimation model

Problem – 4

For the noise free data, **all the continuous estimation models produced a fit of 100 whereas for the discrete estimation, the state space model produced the best fit at 100** while the other two were around the value of 95.

For the noisy data, **the best fits were produced by the transfer function method and the ARMAX method at fit values around 92.**

To develop a model from noisy data, the best approach is definitely to try different methods. However, I would first try the **ARMAX and transfer function methods** as they appear to fit the data in the best way.