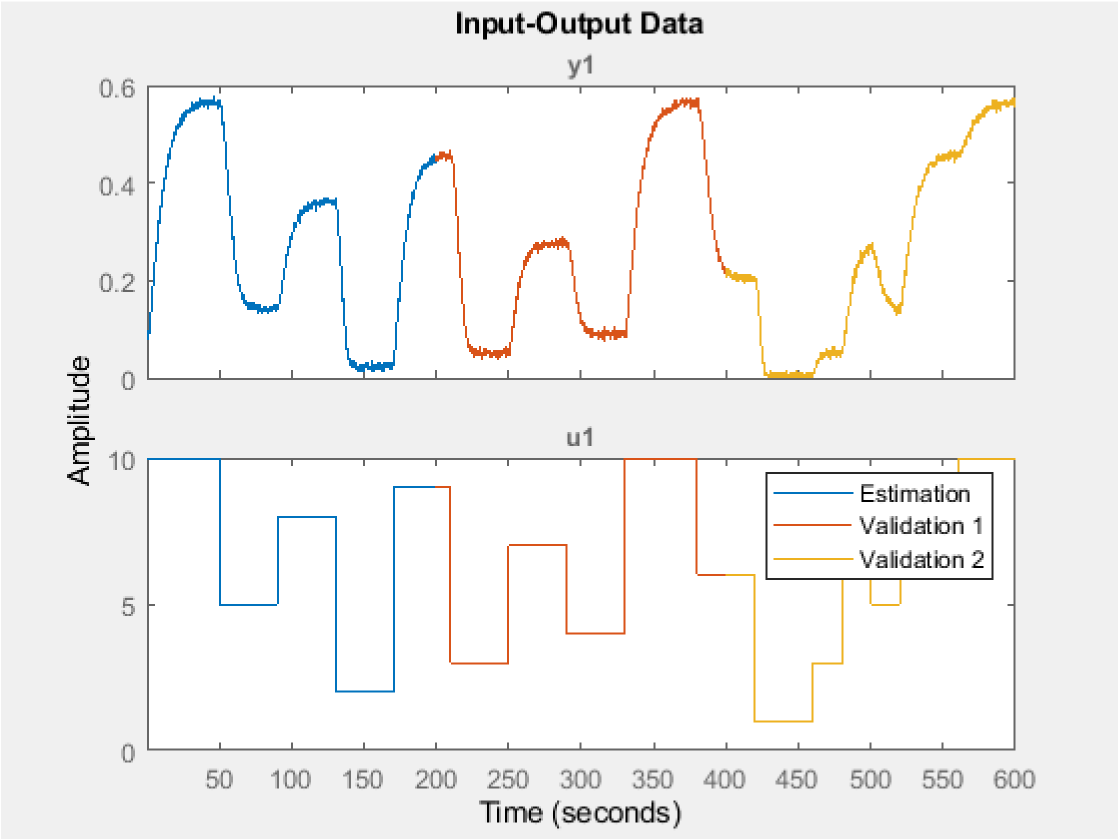
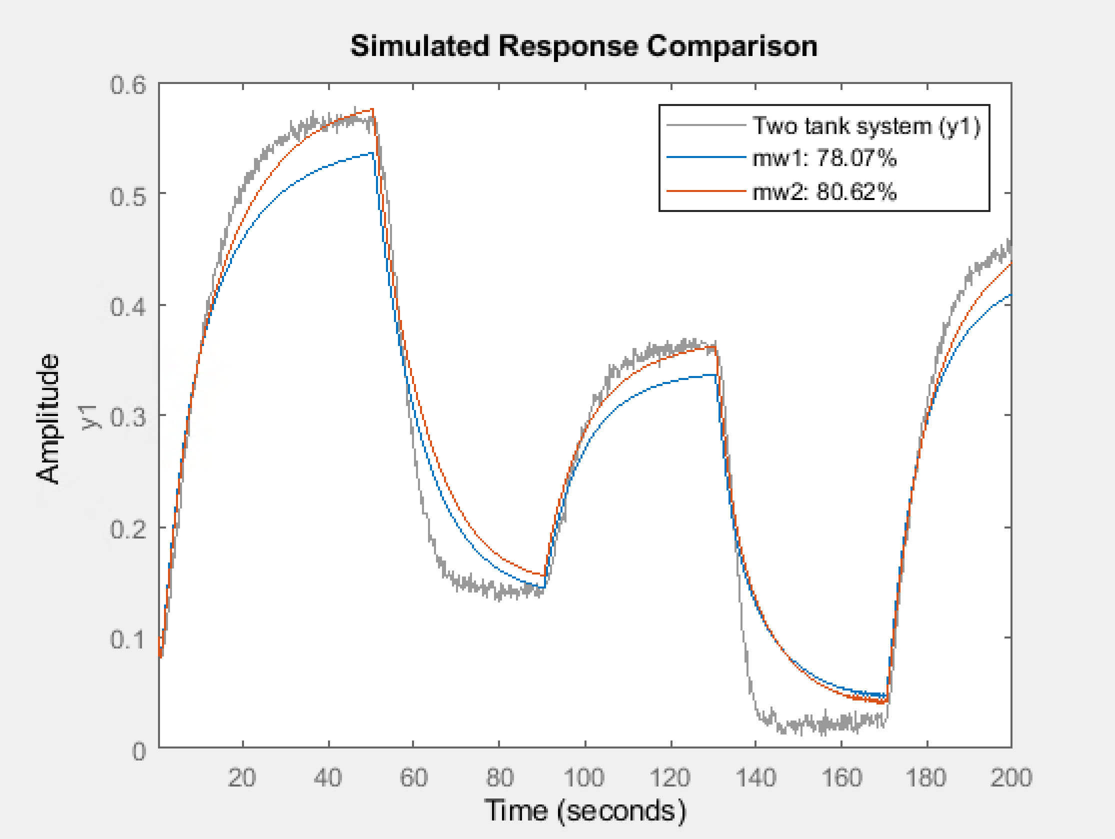
Assignment 10

Task 10.2




```
nn =

     5     1     3
```




```
mw2 =

Nonlinear ARX model with 1 output and 1 input
  Inputs: u1
  Outputs: y1

Regressors:
  Linear regressors in variables y1, u1
  List of all regressors

Output function: Wavelet Network with 8 units

Sample time: 0.2 seconds

Status:
Estimated using NLARX on time domain data "Two tank system".
Fit to estimation data: 96.74% (prediction focus)
FPE: 3.509e-05, MSE: 3.388e-05
```

```
NLFcn =

Wavelet Network
Inputs: y1(t-1), y1(t-2), y1(t-3), y1(t-4), y1(t-5), u1(t-3)
Output: y1

 Nonlinear Function: Wavelet network with 3 units.
 Linear Function: initialized to [0.14 0.111 -0.00352 -0.000372 0.000148 0.000193]
 Output Offset: initialized to 0.288

           Input: [1×1 idpack.Channel]
          Output: [1×1 idpack.Channel]
       LinearFcn: [1×1 nlident.internal.UseProjectedLinearFcn]
    NonlinearFcn: [1×1 nlident.internal.WavenetFcn]
          Offset: [1×1 nlident.internal.ChooseableOffset]
```

```
ans =

  1×1 cell array

    {'u1'}
```

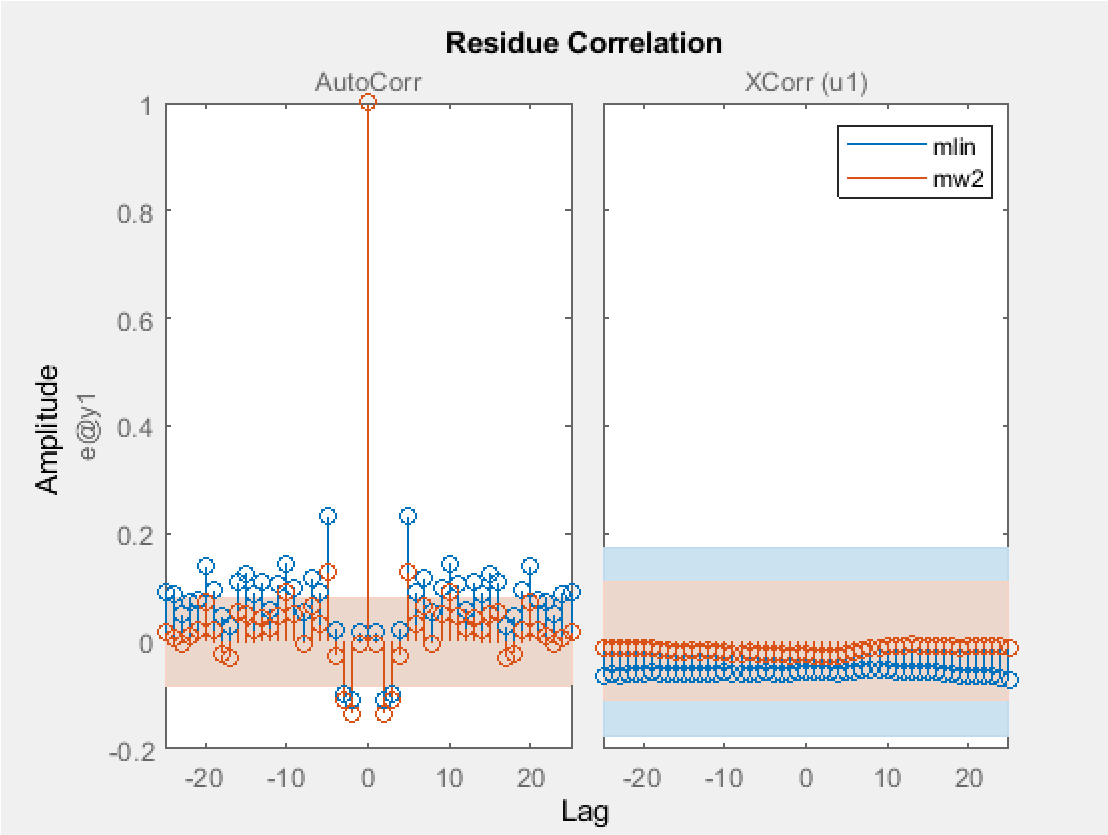
Here, ans is mw2.InputName

```
ans =

  1×1 cell array

    {'y1'}
```

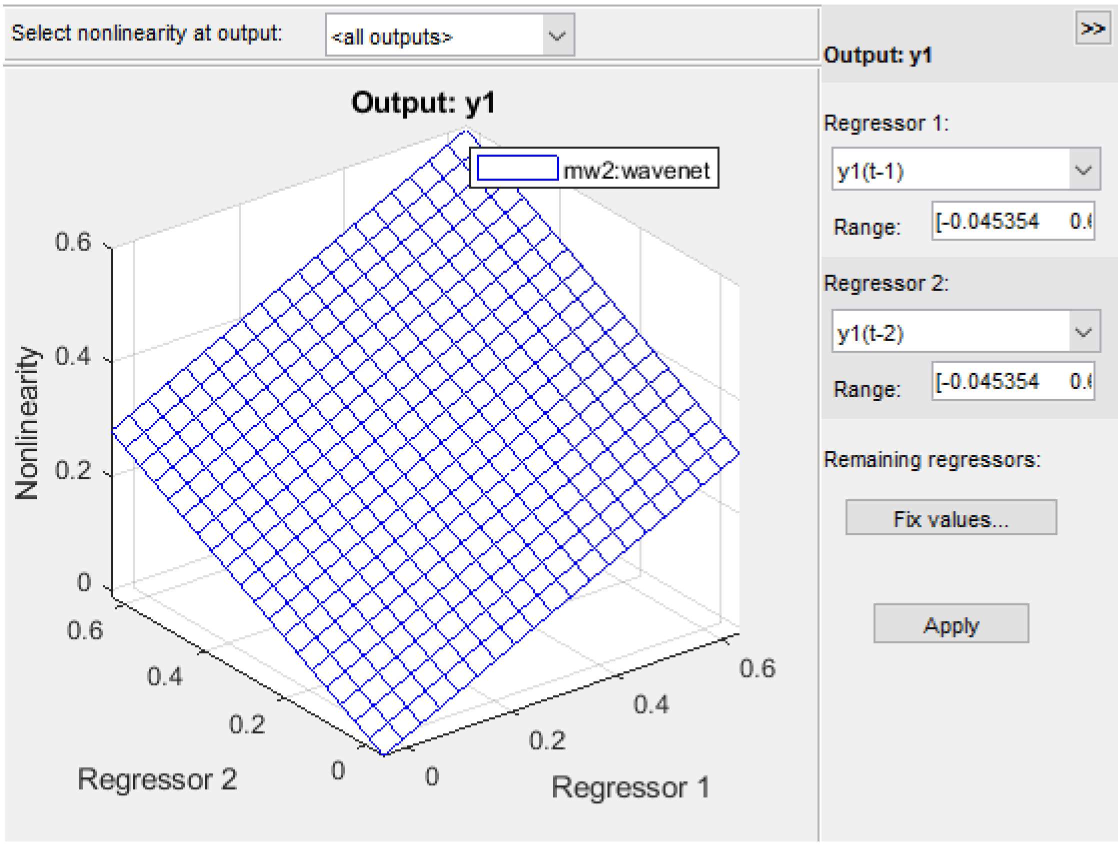
Here, ans is mw2.OutputName

```
ans =

  6×1 cell array

    {'y1(t-1)'}
    {'y1(t-2)'}
    {'y1(t-3)'}
    {'y1(t-4)'}
    {'y1(t-5)'}
    {'u1(t-3)'}
```

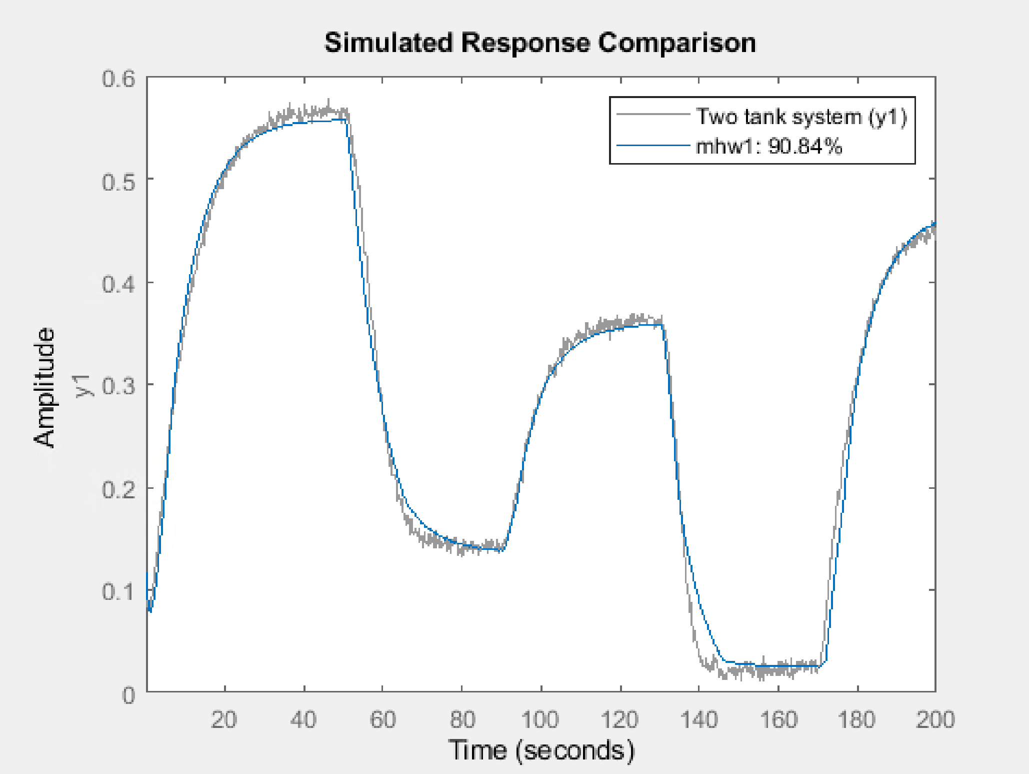
Here, ans is getreg(mw2)



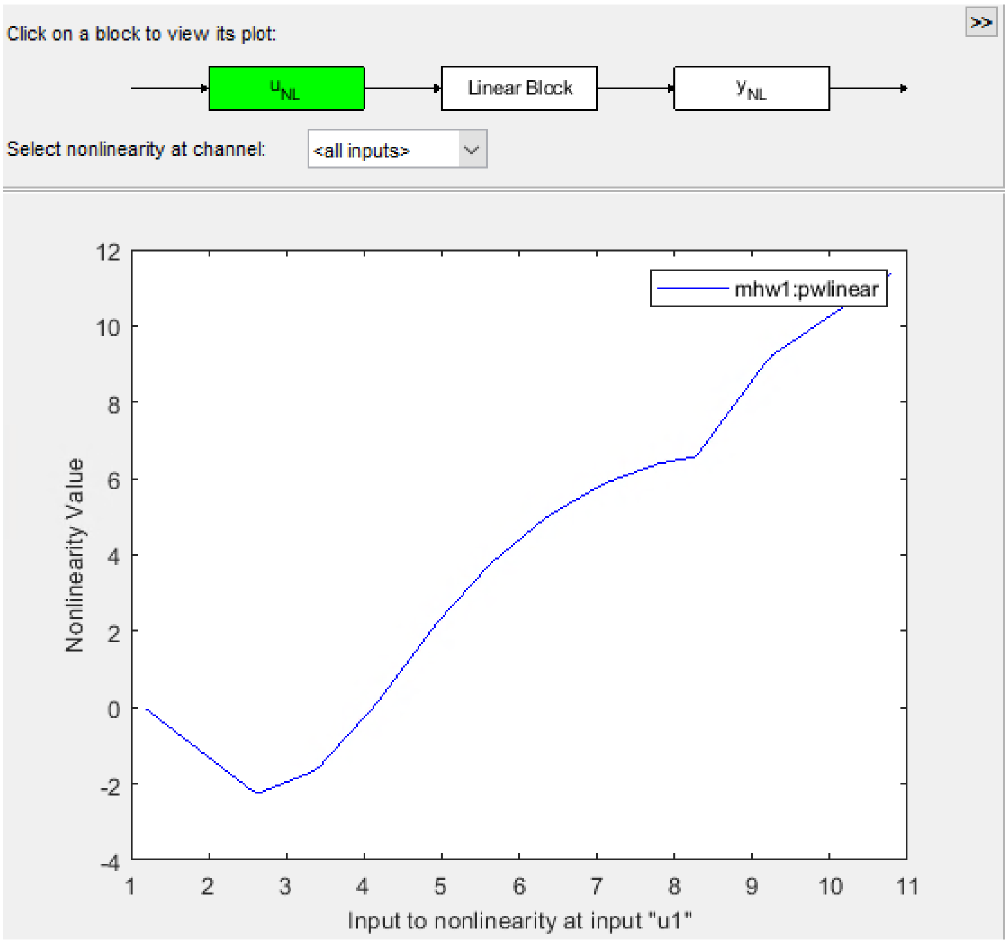
Residue Correlation
AutoCorr
XCorr (u1)
mlin
mw2
Amplitude
e@y1
1
0.8
0.6
0.4
0.2
0
-0.2
-20
-10
0
10
20
Lag



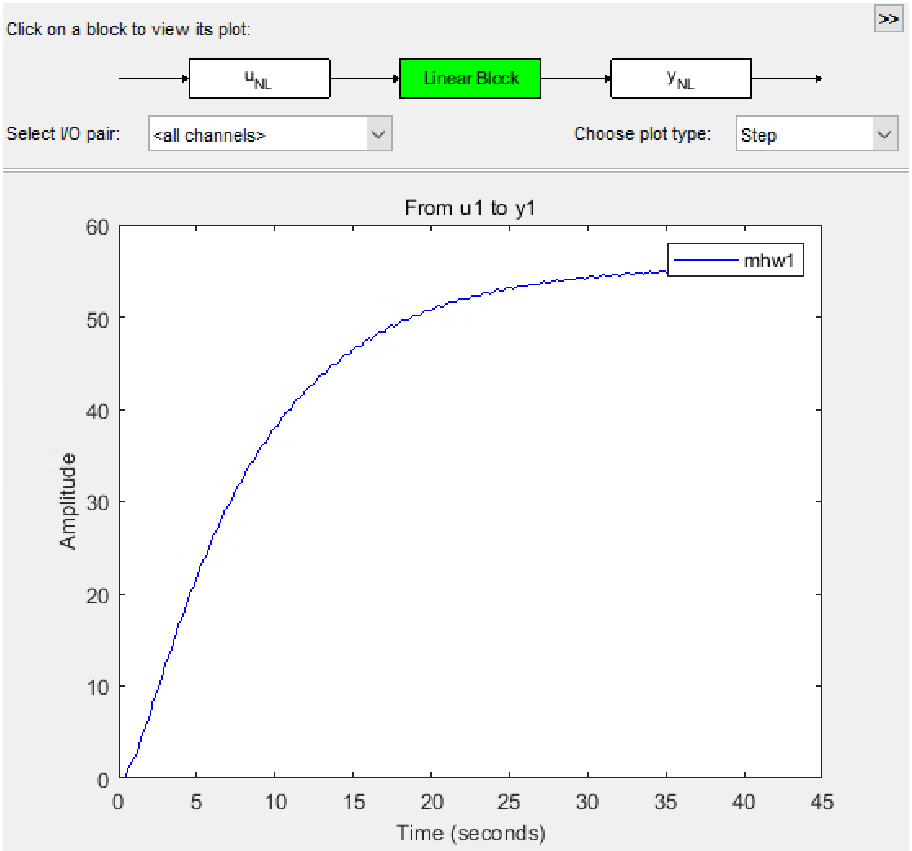
Select nonlinearity at output:
<all outputs>
>>
Output: y1
Output: y1
mw2:wavenet
Nonlinearity
0.6
0.4
0.2
0
Regressor 2
Regressor 1
Regressor 1:
y1(t-1)
Range:
[-0.045354
Regressor 2:
y1(t-2)
Range:
[-0.045354
Remaining regressors:
Fix values...
Apply

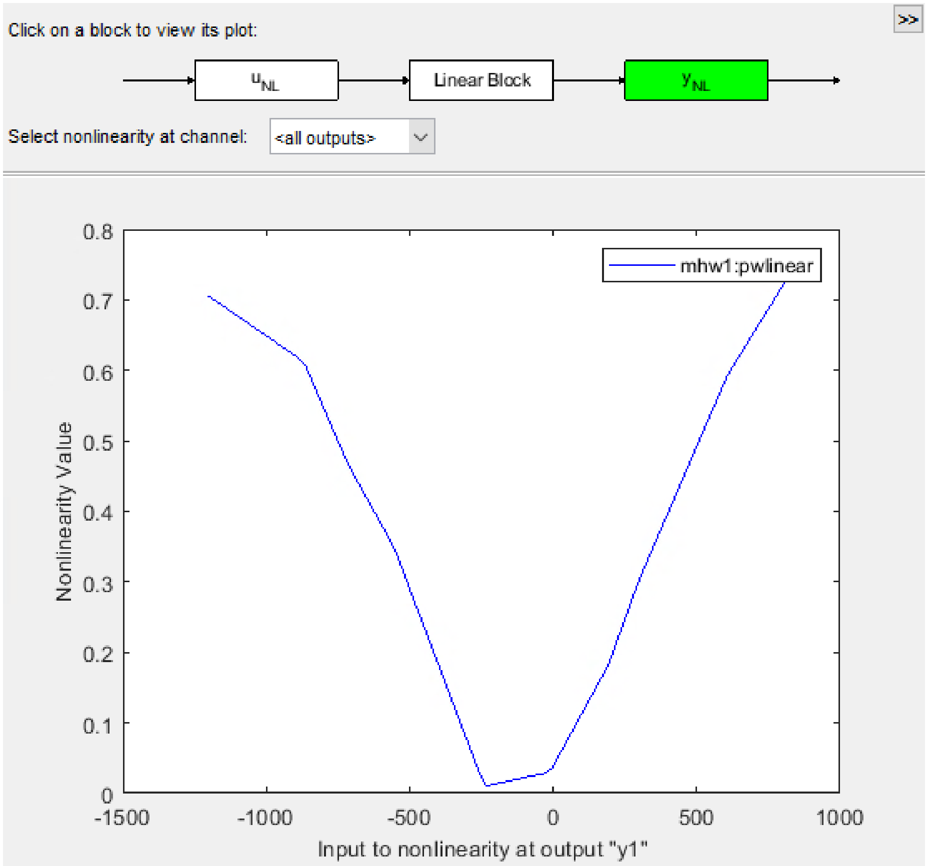


Simulated Response Comparison
Two tank system (y1)
mhw1: 90.84%
Amplitude
y1
Time (seconds)



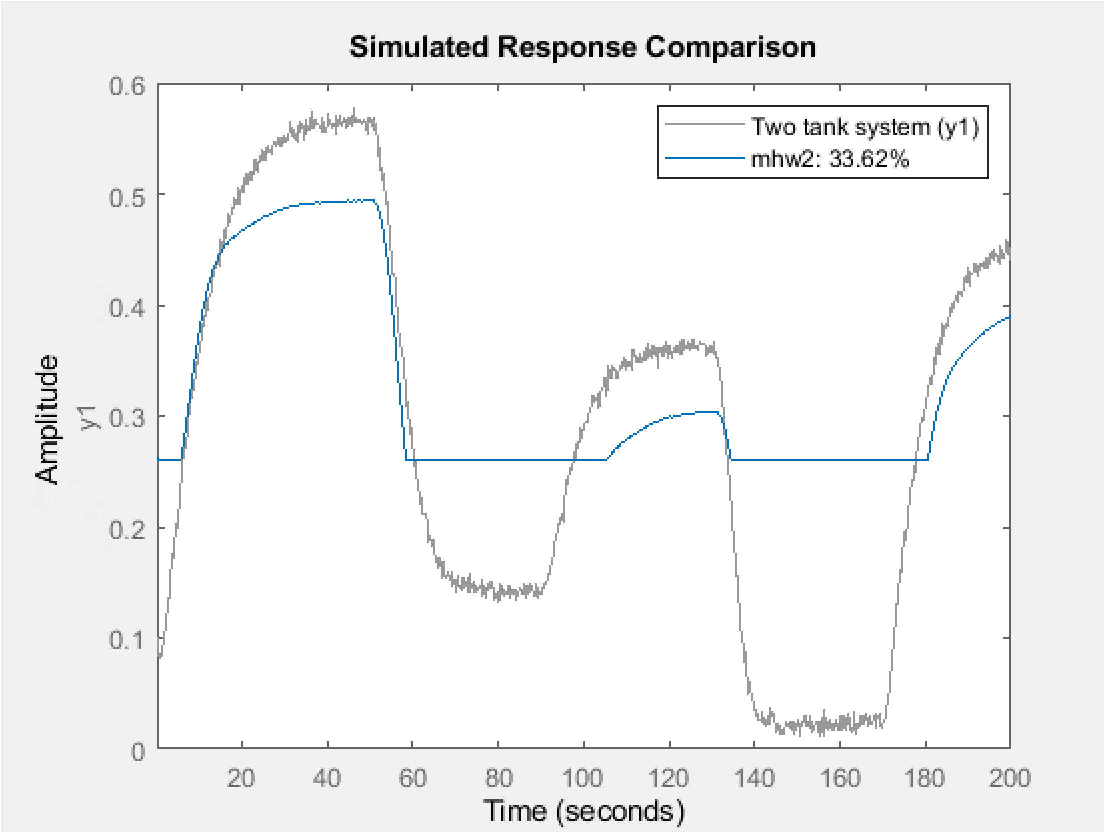
Click on a block to view its plot:
u_NL
Linear Block
y_NL
Select nonlinearity at channel:
<all inputs>
mhw1:pwlinear
Nonlinearity Value
Input to nonlinearity at input "u1"

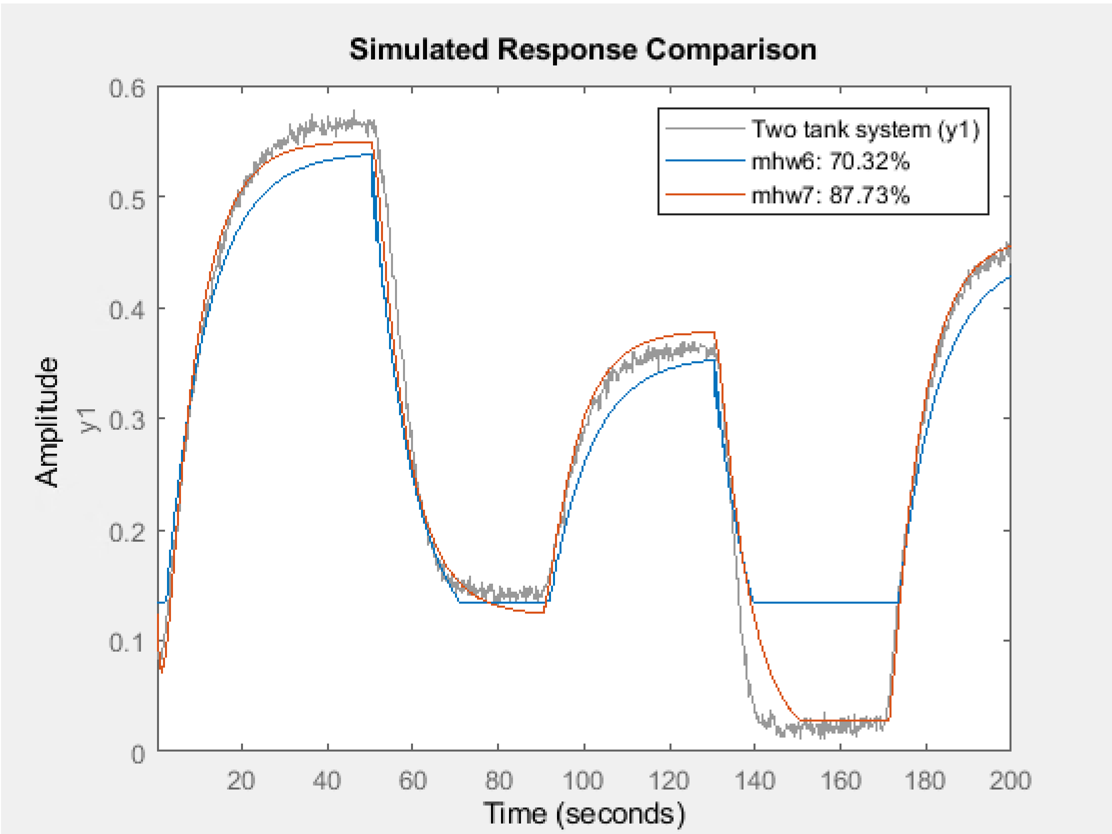

Click on a block to view its plot:
u_{NL}
Linear Block
y_{NL}
Select I/O pair:
<all channels>
Choose plot type:
Step
From u1 to y1
mhw1
Amplitude
Time (seconds)


Click on a block to view its plot:
u_{NL}
Linear Block
y_{NL}
Select nonlinearity at channel:
<all outputs>
mhw1:pwlinear
Nonlinearity Value
Input to nonlinearity at output "y1"

```
ms1 =

Nonlinear ARX model with 1 output and 1 input
  Inputs: u1
  Outputs: y1

Regressors:
  Linear regressors in variables y1, u1
  List of all regressors

Output function: Sigmoid Network with 8 units

Sample time: 0.2 seconds

Status:
Estimated using NLARX on time domain data "Two tank system".
Fit to estimation data: 96.8% (prediction focus)
FPE: 3.777e-05, MSE: 3.258e-05

mt1 =

Nonlinear ARX model with 1 output and 1 input
  Inputs: u1
  Outputs: y1

Regressors:
  Linear regressors in variables y1, u1
  List of all regressors

Output function: Tree Partition with 127 units

Sample time: 0.2 seconds

Status:
Estimated using NLARX on time domain data "Two tank system".
Fit to estimation data: 96.7% (prediction focus)
MSE: 3.473e-05

mhw1 =
Hammerstein-Wiener model with 1 output and 1 input
 Linear transfer function corresponding to the orders nb = 1, nf = 5, nk = 3
 Input nonlinearity: Piecewise Linear
 Output nonlinearity: Piecewise Linear
Sample time: 0.2 seconds

Status:
Estimated using NLHW on time domain data "Two tank system".
Fit to estimation data: 90.46%
FPE: 0.0003199, MSE: 0.00029

ans =

    2.6179    3.3895    4.1611    4.9327    5.6297    6.3573    7.0850    7.8127    8.2741    9.2257
   -2.2681   -1.6345    0.1033    2.2067    3.7696    4.9984    5.8644    6.4075    6.5725    9.1867
```

Here, ans is mhw1.InputNonlineratity.BreakPoints, i.e. the piecewise affine input function in the screenshot above.

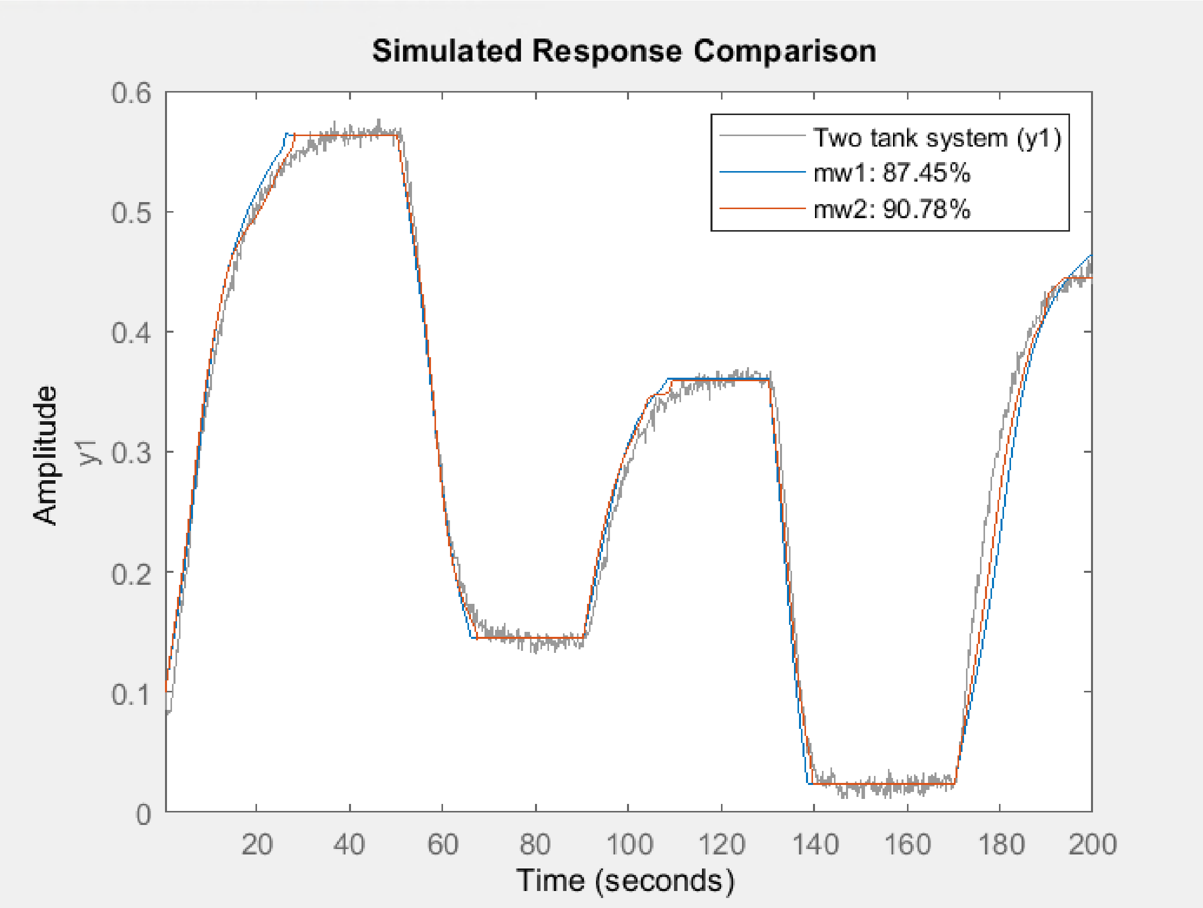

Simulated Response Comparison
Two tank system (y1)
mhw2: 33.62%
Amplitude
y1
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120
140
160
180
200
Time (seconds)


Simulated Response Comparison
Two tank system (y1)
mhw6: 70.32%
mhw7: 87.73%
Amplitude
y1
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120
140
160
180
200
Time (seconds)

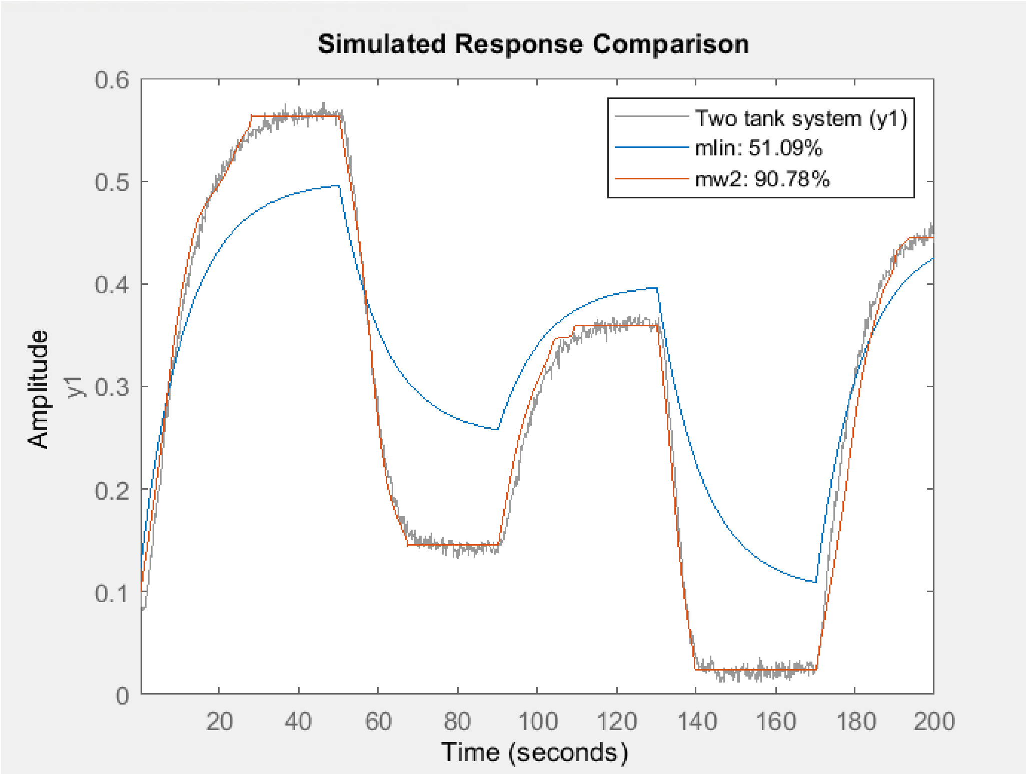
Task 10.3

Using nn = [1 1 1] gives the regressor set {y(t-1), u(t-1)}, so only the most recent input and most recent output is used in the model.

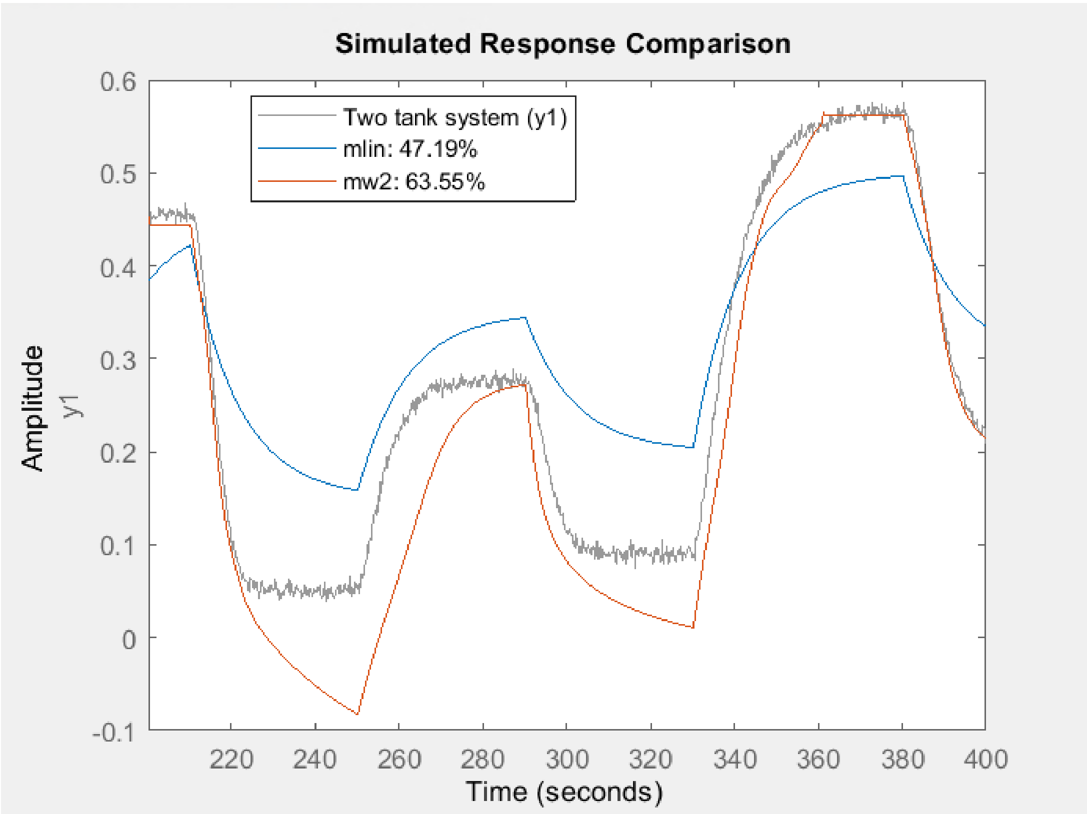
Here is using default wavenet and wavenet with 20 units



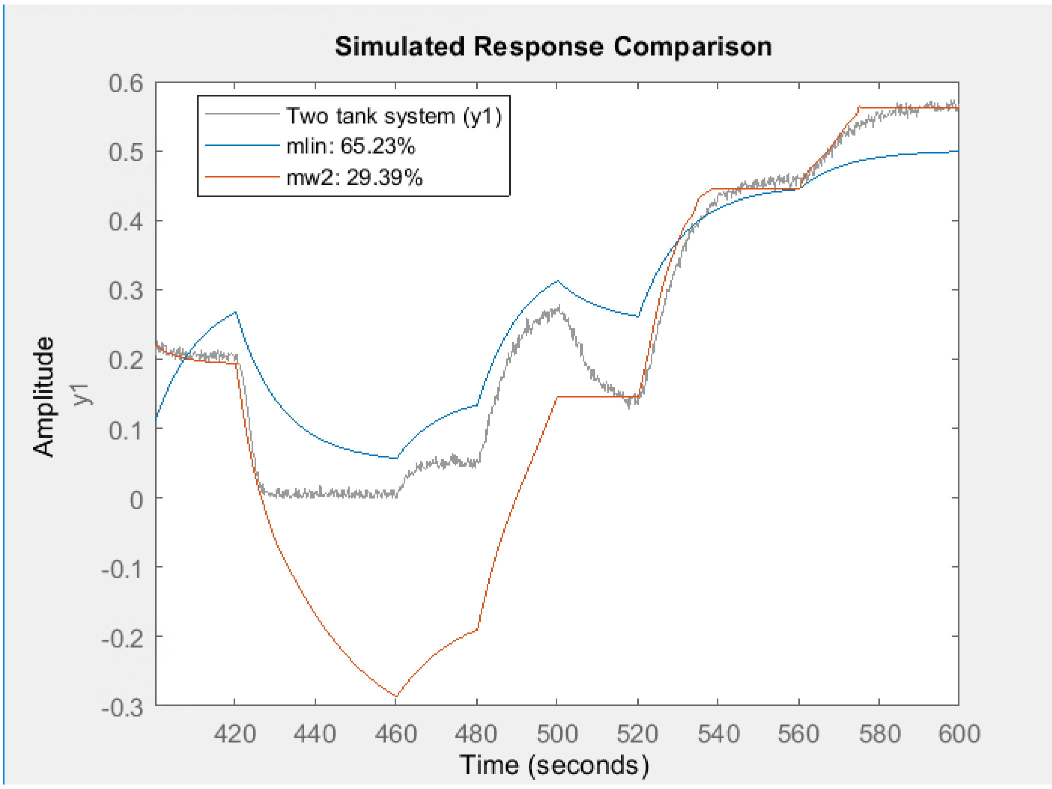
Both are almost equal, and they seem to follow the input data better than above.



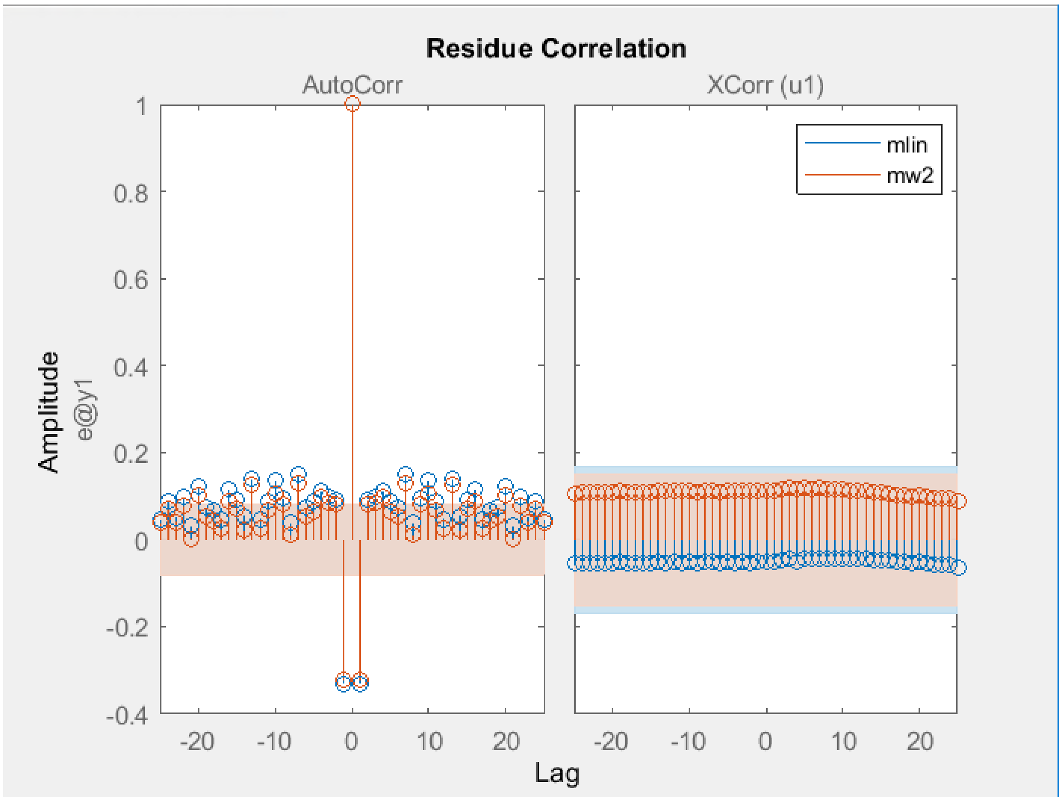
Wavelet function does well on training (z1), linear does okay, but not too well.



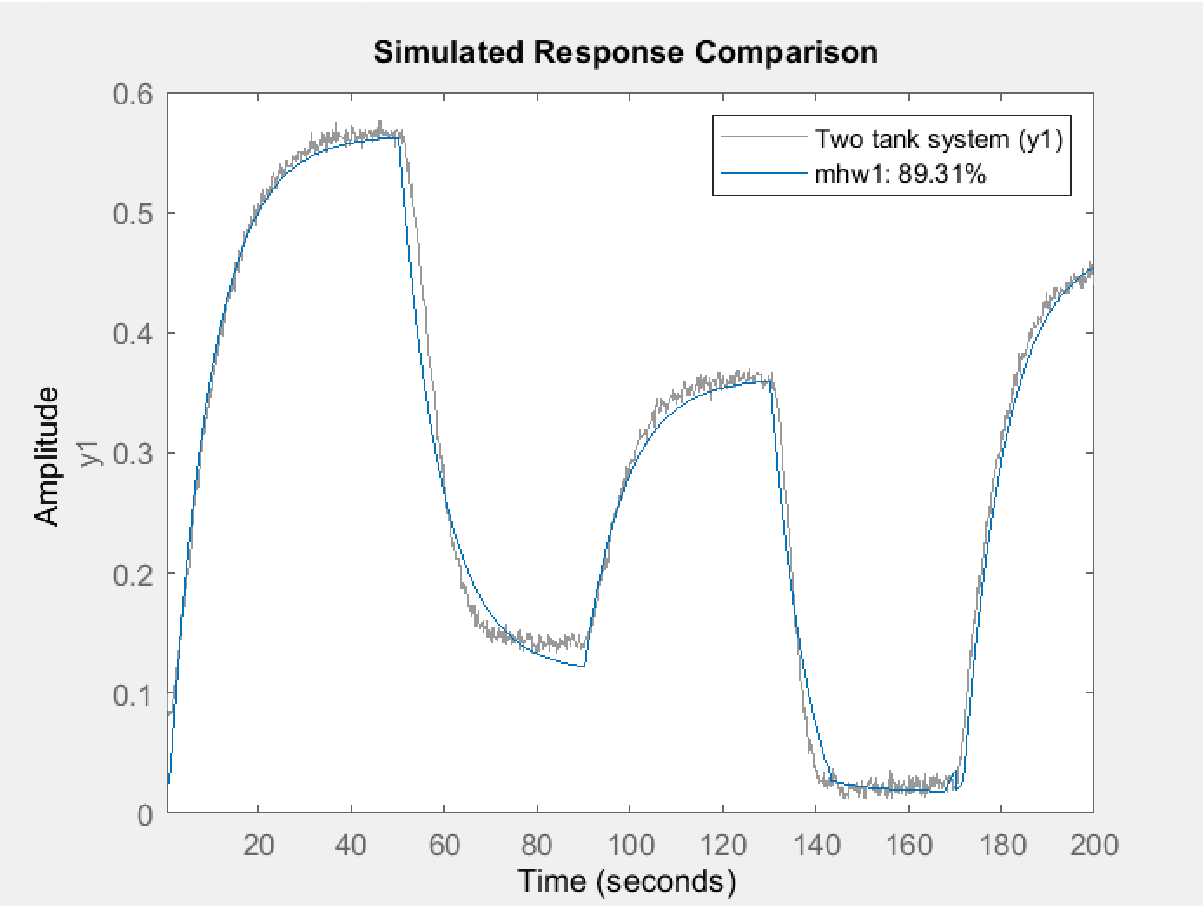
On validation (z2) both models perform not too well even though the data looks similar to the training data.



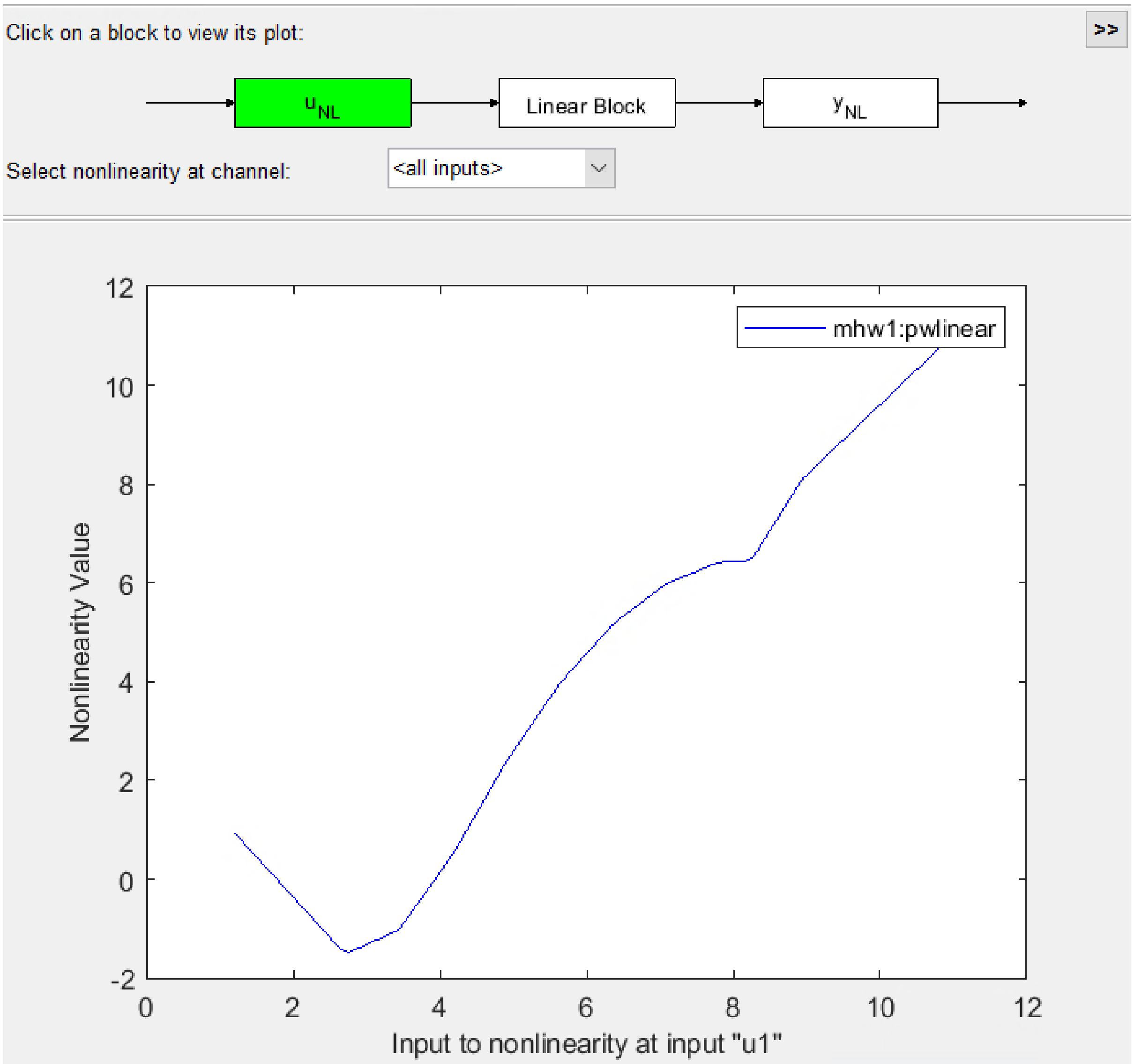
On second validation (z3) mw2 wavenet model does terribly. Linear also doesn't do too well, but it performs better.



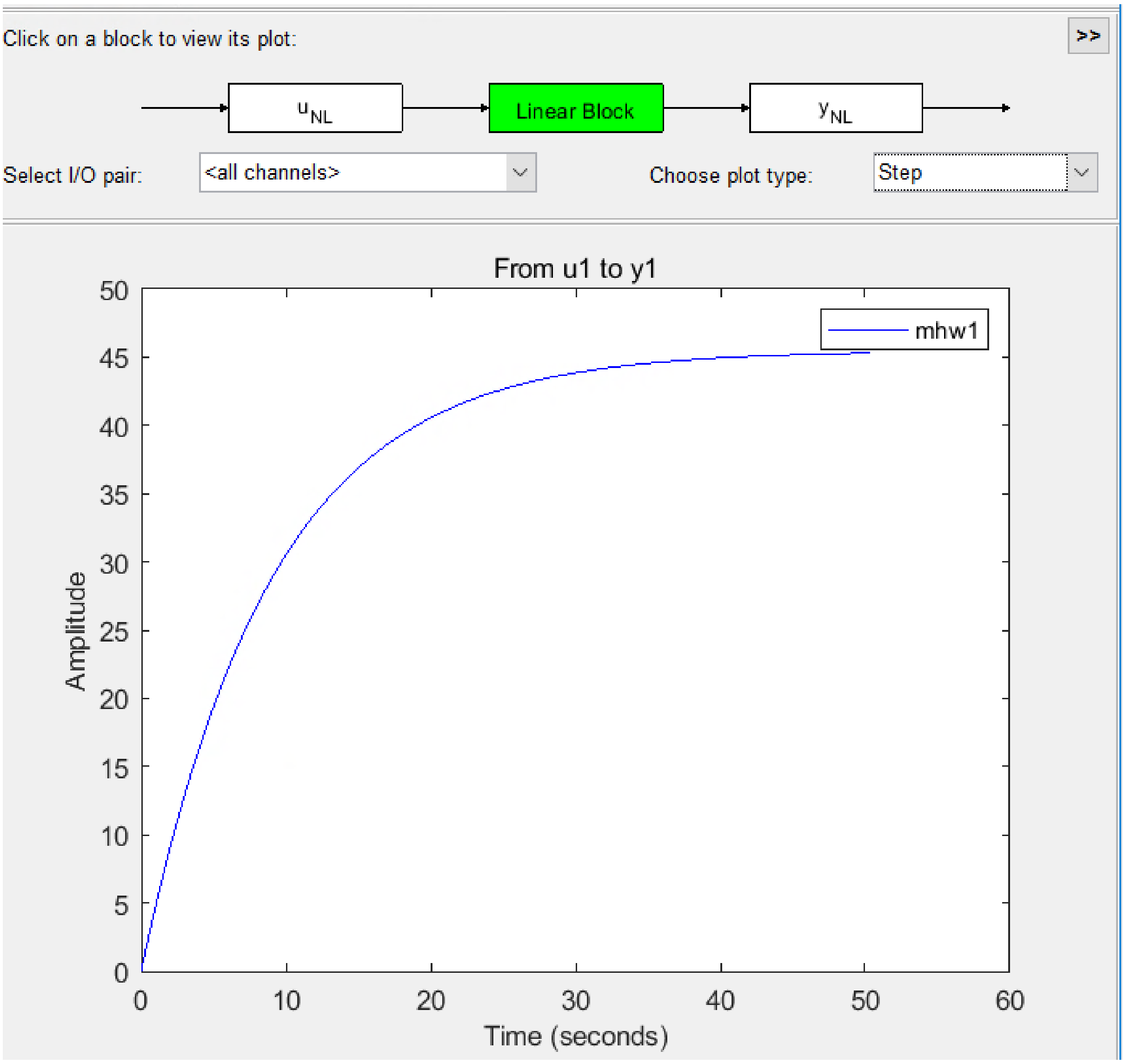
The correlation analysis shows that both models are only autocorrelated at lag 0 and +-1, which makes sense given which regressors we use. They are also both uncorrelated with the inputs. Compared to earlier there is less autocorrelation, but is likely just due to the fact that we're using fewer regressors.



Hammerstein-wiener model does very similar to earlier on the training set (z1).

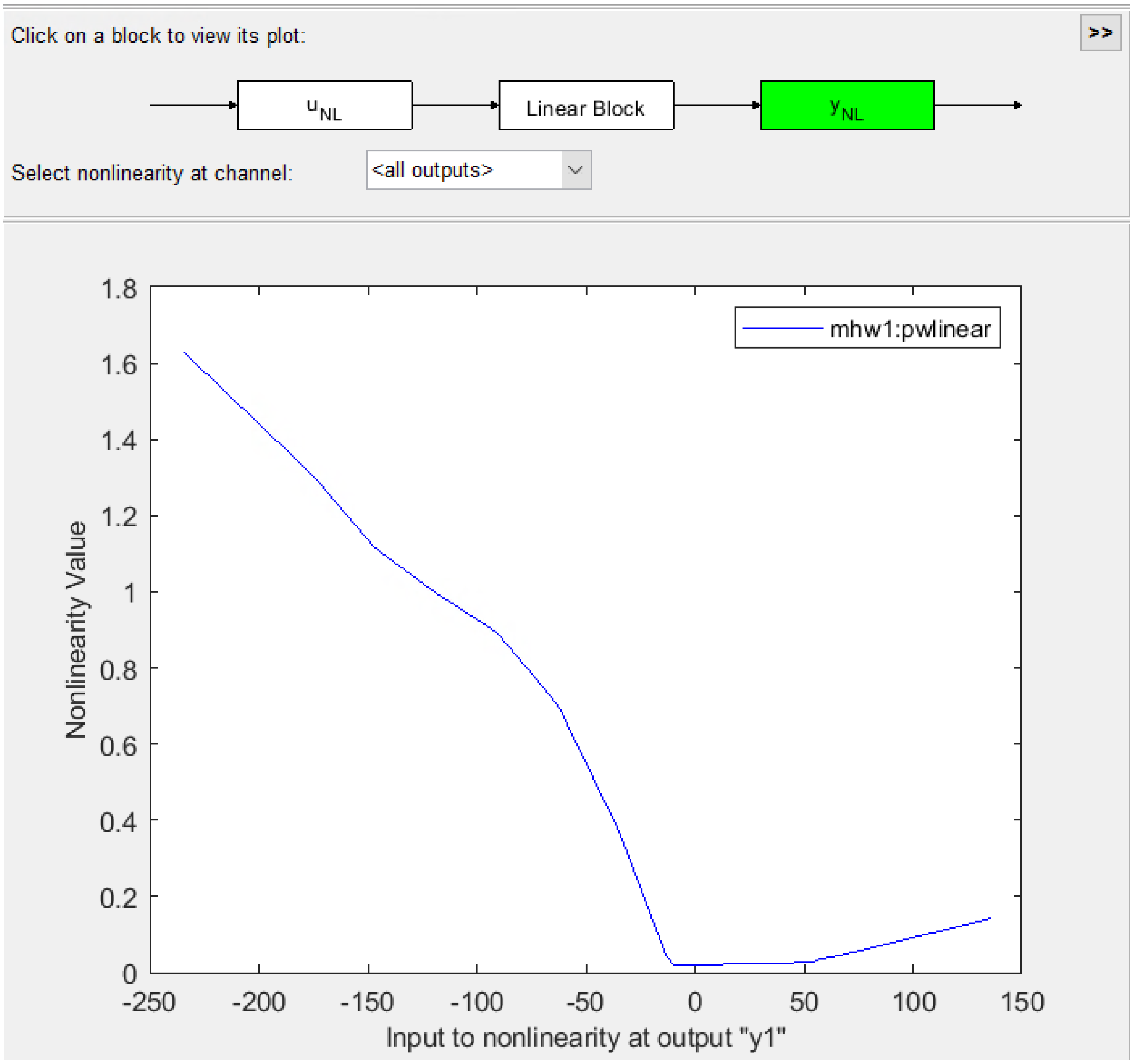


The input non-linearity is strikingly similar in shape, amplitude and input values as earlier. The main difference is in the beginning, it starts at 1 instead of 0. The similarity in input is expected as the same input is used. It is also not too unexpected that it has similar shape and amplitude since the same input is used.



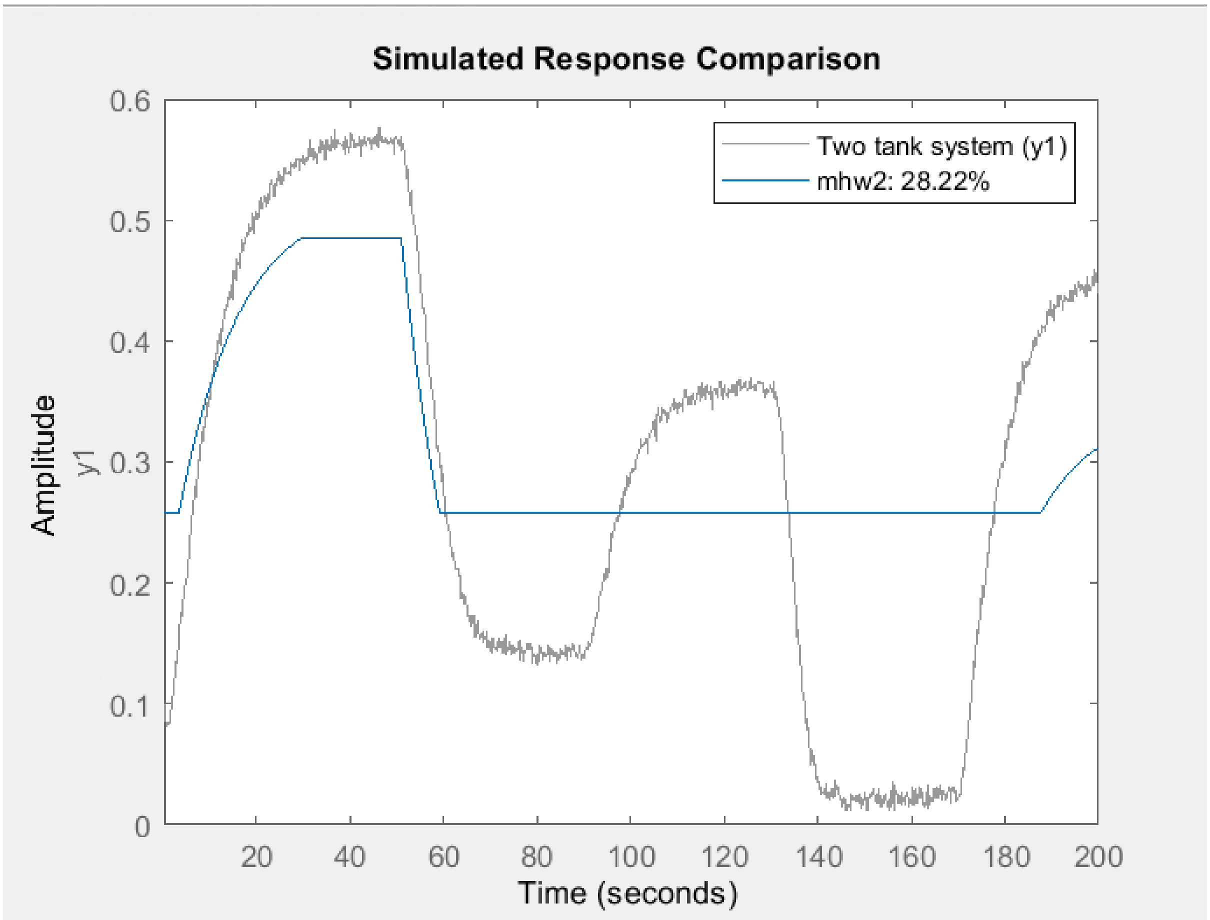
The linear block is also very similar, but doesn’t achieve the same amplitude as earlier. It also seems smoother, but this can also just be a rendering artifact. Again, the similarities is not too unexpected as the same input is used, and the input non-linearity is similar to earlier.

The earlier regressors for input is just one input regressor with lag 3 instead of 1, and this indicates that there is much similarity between lags in the input, which is confirmed by looking at the input (first image) where we see that it is a square wave, and there is only change a few times.

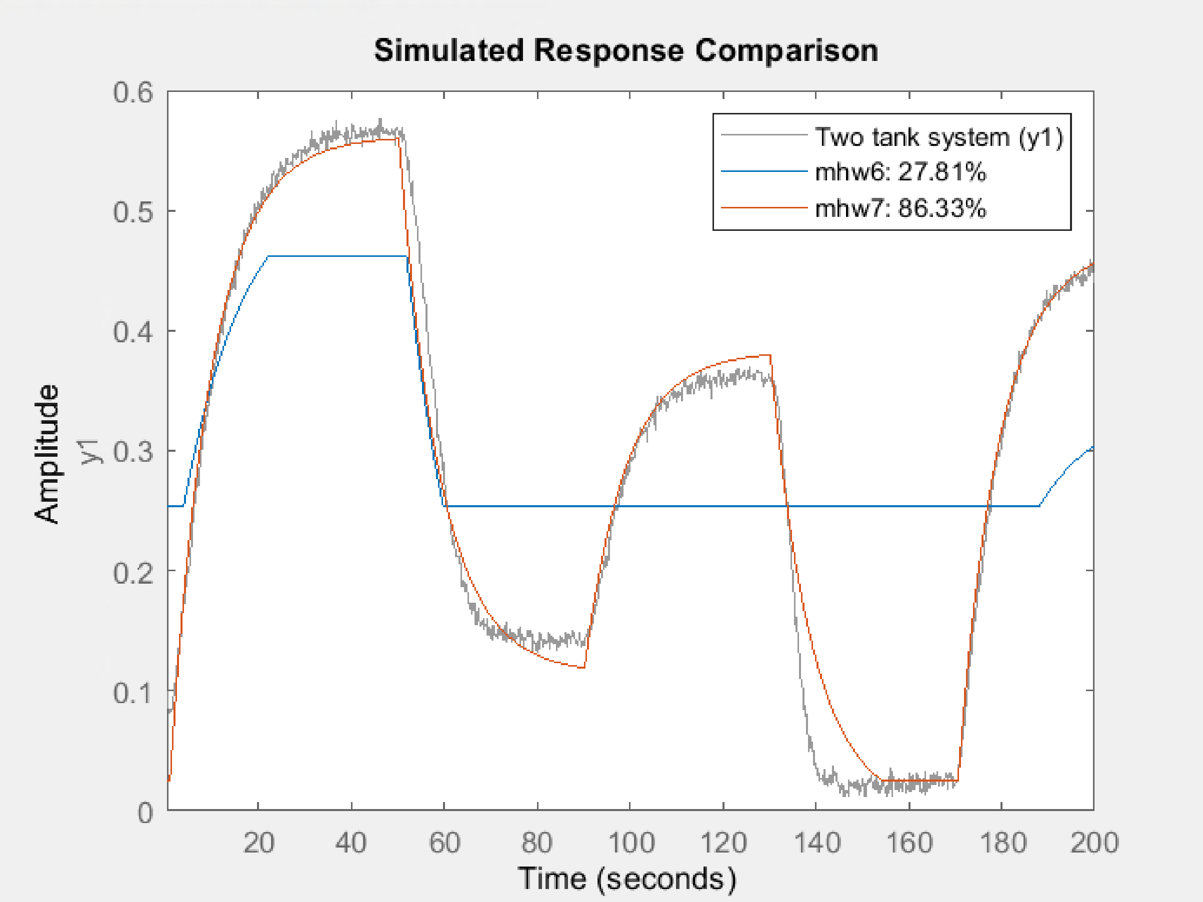


It's in the output non-linearity that we see the biggest difference in both output range (x-axis), amplitude (y-axis) and shape.

A reason for this can be that the small dissimilarities from the two earlier blocks have now compounded so the output is very different.



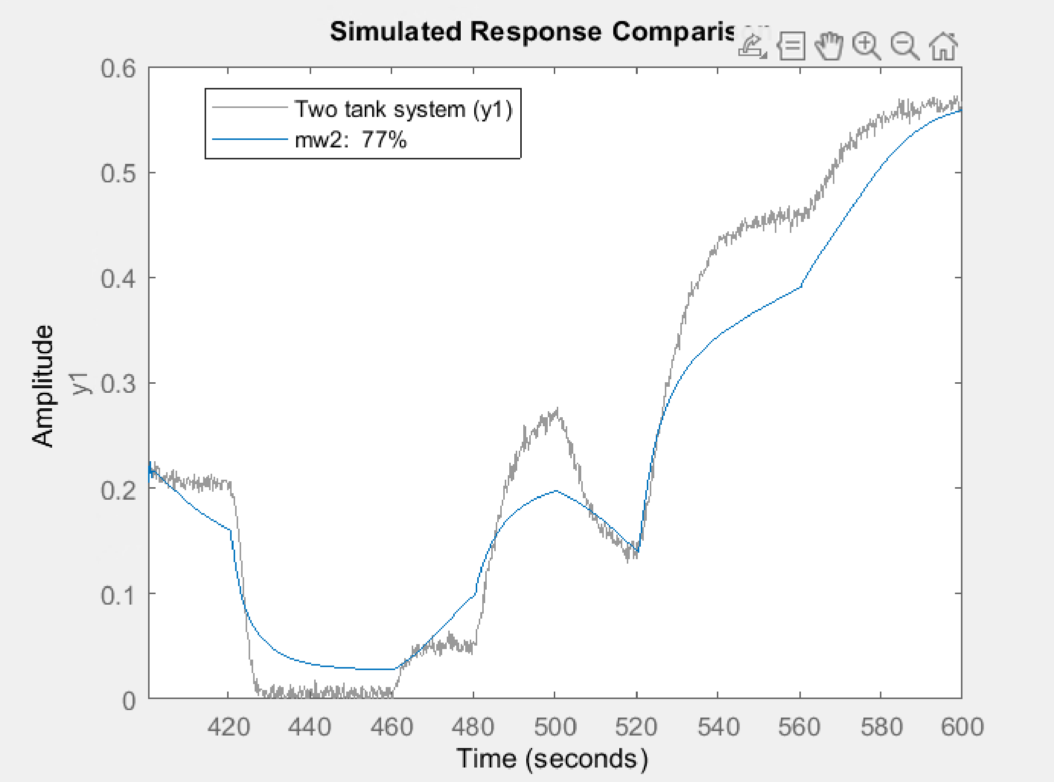
The deadzone non-linearity affects this model with fewer regressors more, where it cannot “detect”/model the small jump at around 100-140 seconds, while the previous model did manage to capture this.



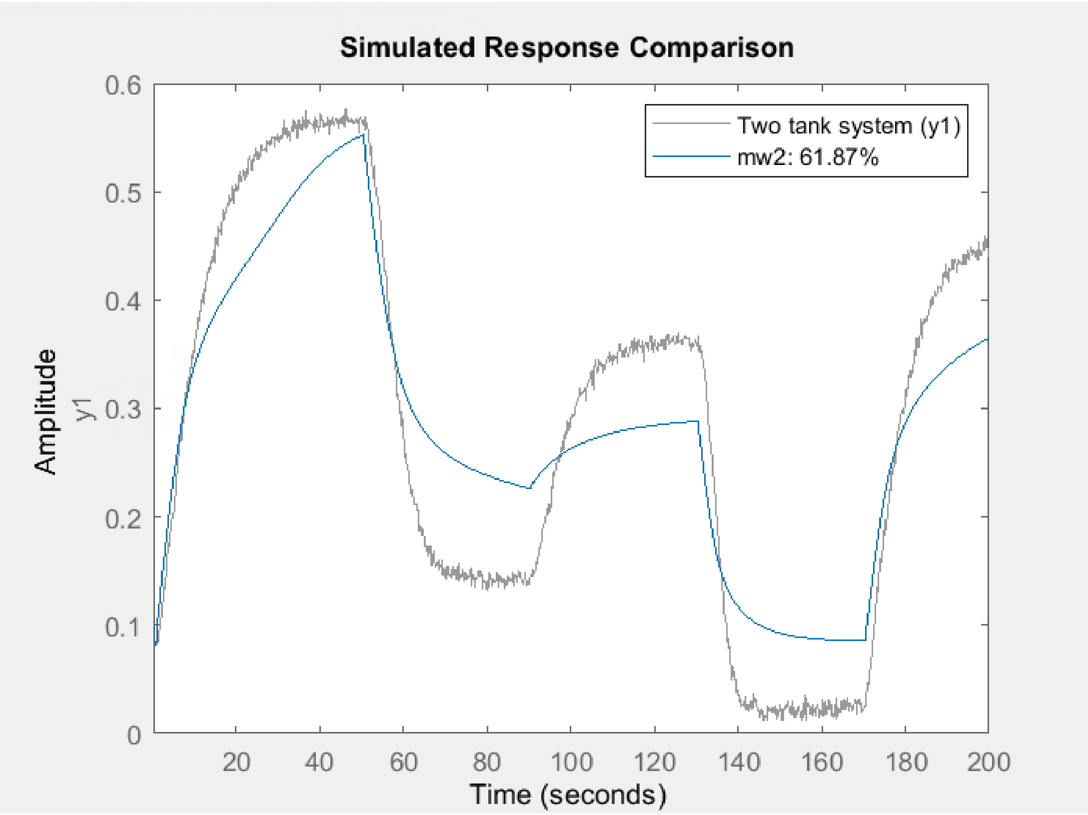
Mhw6 uses only 7 iterations, while mhw7 uses 30 iterations, and both uses the same deadzone and saturation parameters. This shows that using more iterations yields better model (which is not surprising as it has more iterations to adapt to the data). One weird thing is that the deadzone for mhw7 is not as pronounced as earlier, but I'm not sure why…

Task 10.4

First wavenet model with 8 units training on z3

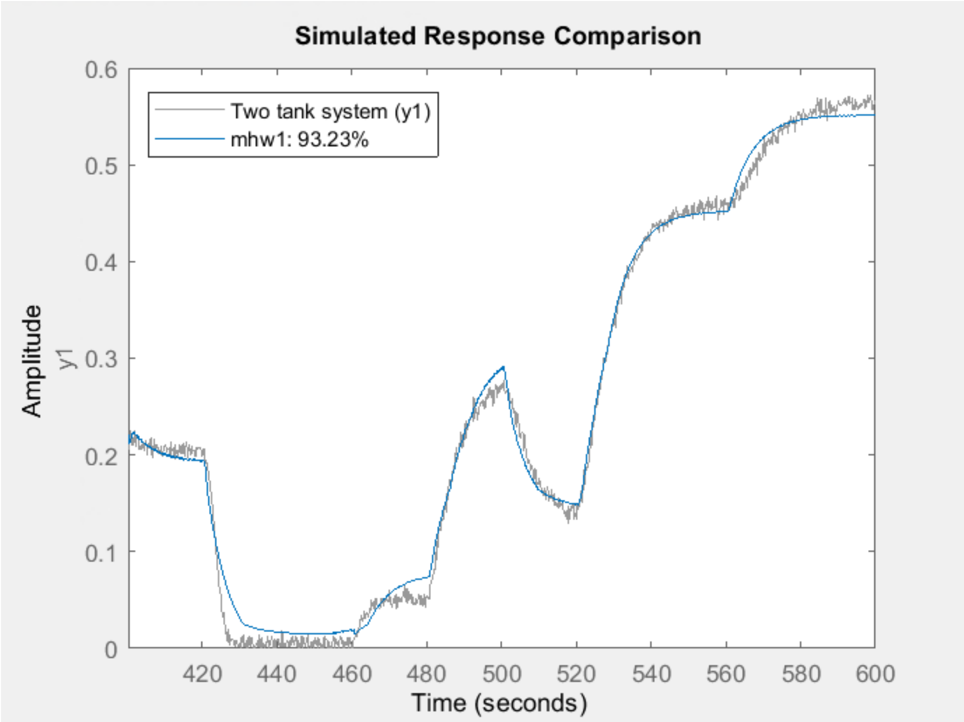


Validating on z1

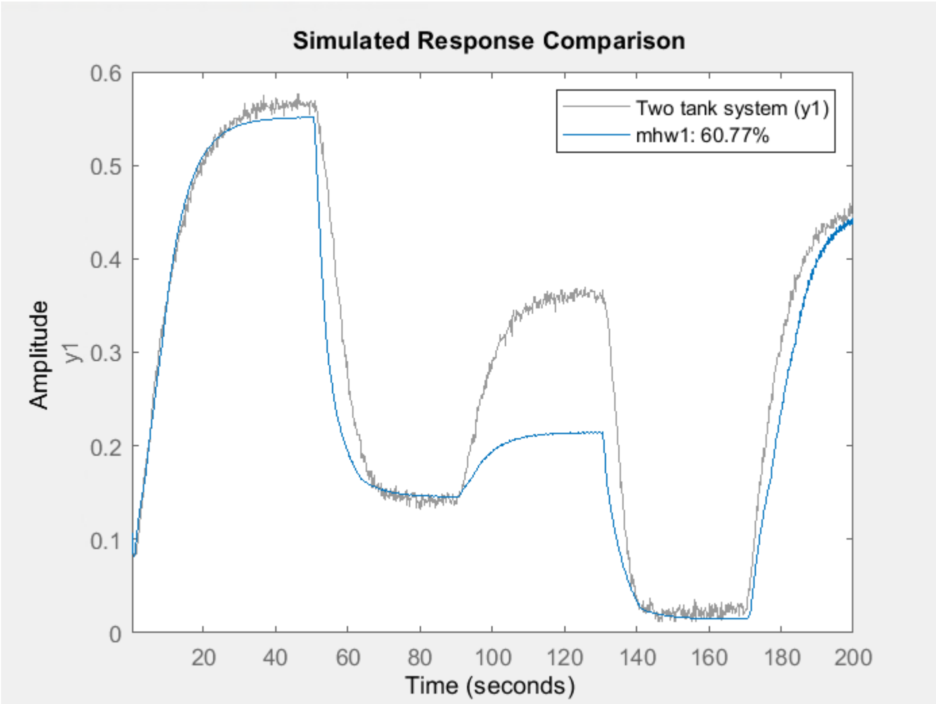


It does okay, but it seems like z3 is much harder to model than z1, which can make sense as it has more and faster variation.

The first Hammerstein wiener model with piecewise non-linearities does well modeling z3



On validation on z1 it also does okay, but struggles with the smaller one in the middle.



The change in performance might be due to different levels of excitation in the different datasets.