TTK 4260 - Multivariat - Assignment 10

Petter Skau-Nilsen
Siri Stueland Nysted
Ivar Tesdal Galtung

May 2021

1 Task 2

In this task, we finished the tutorial as it is presented on the Matlab-web page. Figure 1a compares the linear algorithm with mw2, where mw2 is a nonlinear ARX-model with a fixed number of units in the wavenet function, in this case 8. We see that here, the linear ARX model is not very good for estimating the system output. mw2 is significantly better.

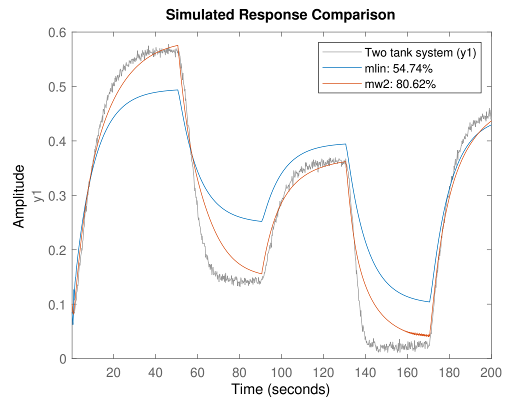
Figure 1b shows about the same relationship between the linear model, mw2 and the system data as Figure 1a.

On the second validation set, the linear model is still significantly worse than mw2, as shown in Figure 1c. However, both of these models perform better than they do in both the test set and the first validation set. The reason that the linear model performed better is probably because the shape of the validation set 2 is overall an inclination, Meaning that as long as the linear model is increasing at the same rate, it is going to have a high accuracy to the validation set 2.

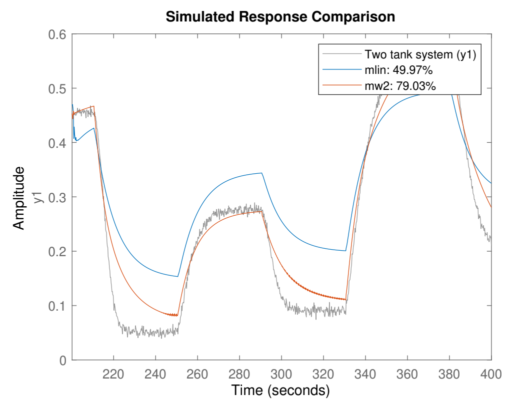
Figure 2 compares the model mw2 to mw1, where mw1 is a nonlinear ARX-model with a number of wavenet model units that is determined by the estimation algorithm, here the number is 3. The two models are relatively similar, although mw1 performs a little bit worse than mw2.

Figure 3 shows the models based on decision trees and sigmoid functions. Both of these performed very well on the test set, and the sigmoid model had a comparable accuracy on both of the validation sets. We are not entirely sure why these models performed so much better, but guess that it has something to to with the sigmoids model being able to take in more parameters without overfitting.

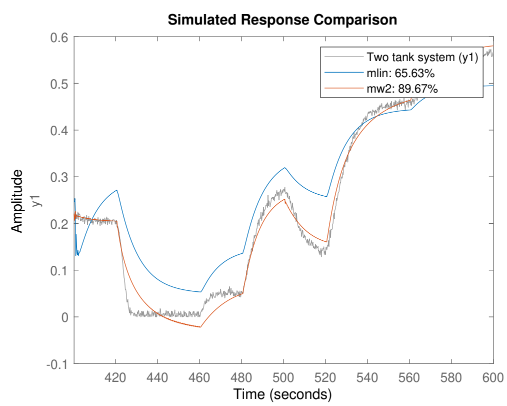
In the tutorial, a Hammerstein-Wiener model is made of the system, and its performance on the training set it shown in Figure 4a. This is also a very good model, comparable to the sigmoid model and the decision tree based model from Figure 3. In Figure 4b, both a deadzone (actuator does not respond to very small changes in input), and a saturation (output cannot be above a certain threshold) are introduced to the Hammerstein-Wiener model. We can see that this causes the model to "stop short" of the peaks, and noes not go below a set level. In figure Figure 4c, hw3 has only deadzone implemented, while hw4 has only saturaton implemented. The deadzone model seems to accumulate some error over time, while the saturation-only model struggles to reach the peaks in the data.



(a) lin compared to mw2 on test set.



(b) lin compared to mw2 on validation set 1.



(c) lin compared to mw2 on validation set 2.

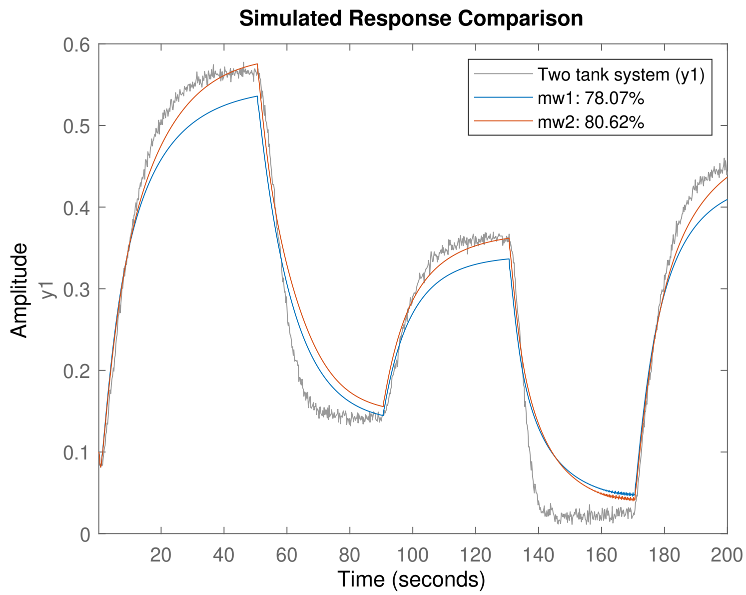


Figure 2: mw1 compared to mw2.

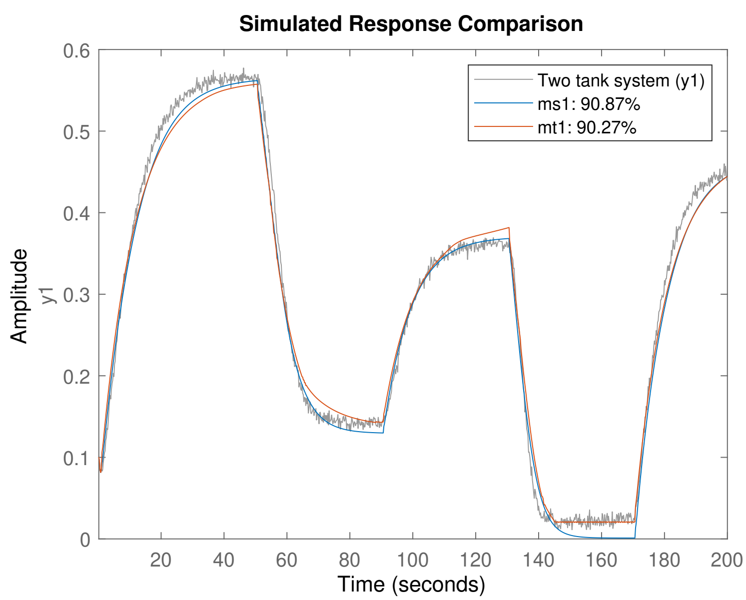
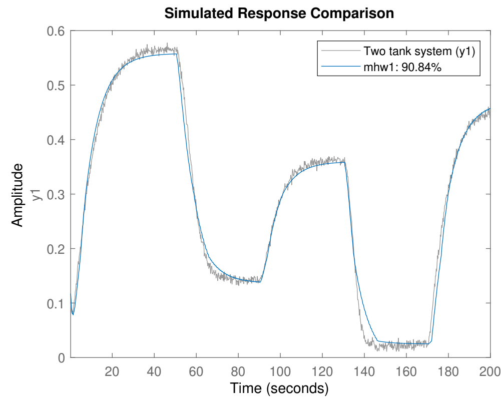
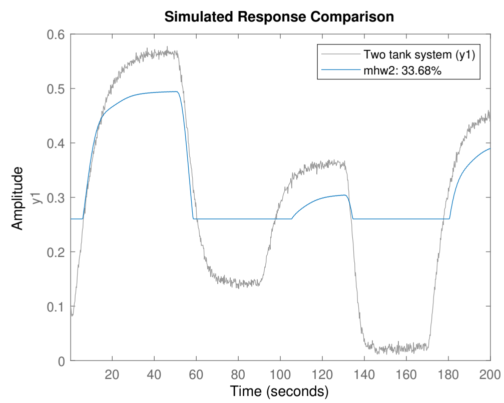


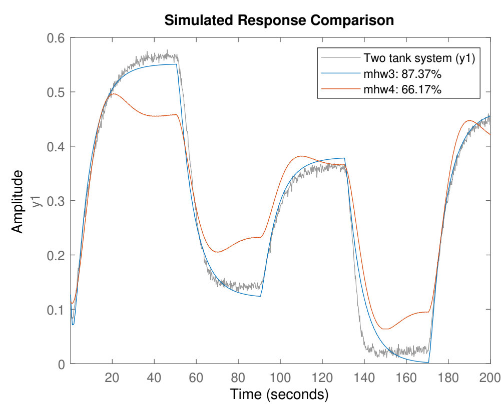
Figure 3: tree partition compared to sigmoid models on the test set.



(a) The Hammerstein-Wiener model on the training set.



(b) The Hammerstein-Wiener model on the training set with deadzone and saturation.



(c) The Hammerstein-Wiener model on the training set. Comparison between model with only saturation and model with only deadzone.

2 Task 3

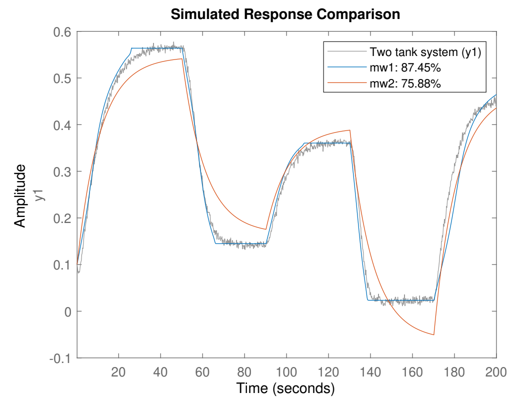
Here, the tutorial was re-done with the regressor selection given by `nn = 1 1 1` rather than `nn = 5 1 3`. This means that the regressors are simply $y(t-1)$ and $u(t-1)$, which gives a less complex model.

As the models in this task are less complex than the ones in task 2, we expect them to give poorer estimates for the output of the system. As seen from Figure 5, this is indeed the case. Here, as before, mw2 performs better than both the linear model and mw1. However, mw2 performs a lot better on the training set than any of the validation sets, which can be an indication of overfitting.

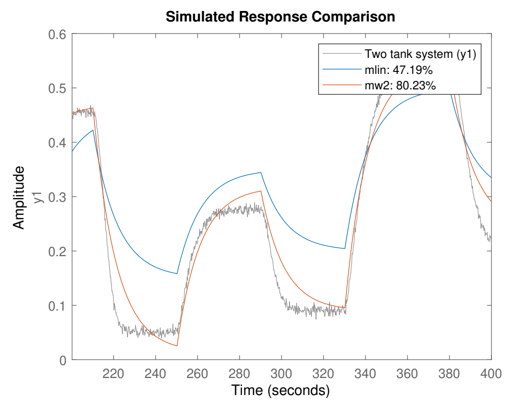
Figure 6 shows the performance of the simpler models when the number of wavenet parameters is set to 30. Here, both mw1 and mw2 have a very high accuracy when applied to the training set (Figure 6a). However, for both of the validation sets, mw2 becomes unstable and goes towards infinity! This shows that this model is clearly very overfitted and is completely useless for other data than the training set.

The ovefitting that we see in this task may be caused by the simplicity of the models. If they are not complicated enough to accurately represent the behaviour of the system, we cannot expect to get good models. This causes the models to "memorize" the training data rather than create a model that accurately represents the system.

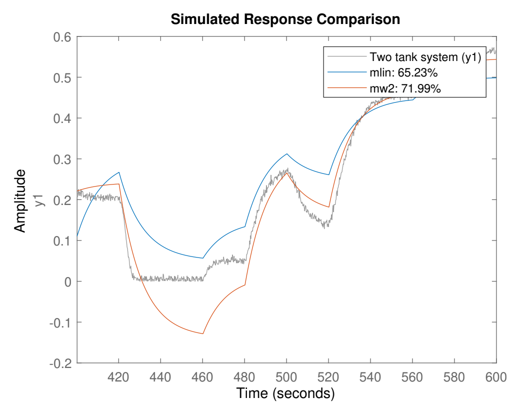
Figure 7 shows the simpler desicion tree model and the sigmoid model used on the training set. Here, the desicion tree is the only one of the models in this task that does not perform significantly poorer than the corresponding model in task 2.



(a) mw1 compared to mw2 on test set.

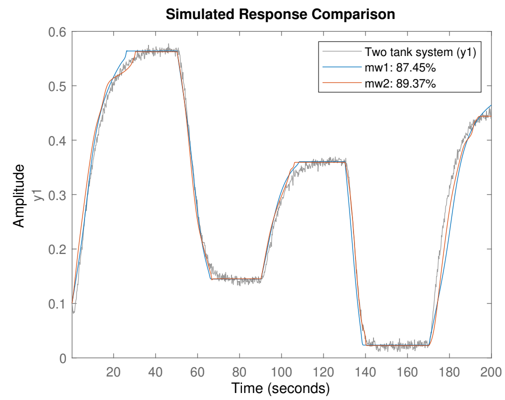


(b) lin compared to mw2 on first validation set.

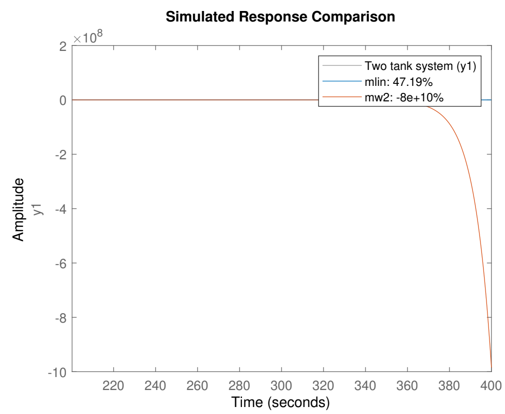


(c) lin compared to mw2 on first validation set.

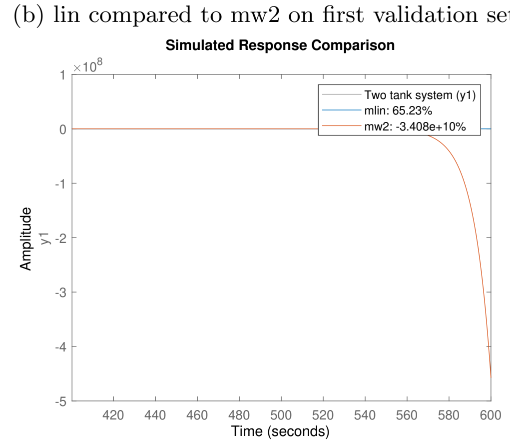
Figure 5: Number of wavenet parameters = 0.



(a) mw1 compared to mw2 on test set.



(b) lin compared to mw2 on first validation set.



(c) lin compared to mw2 on first validation set.

Figure 6: Number of wavenet parameters = 30.

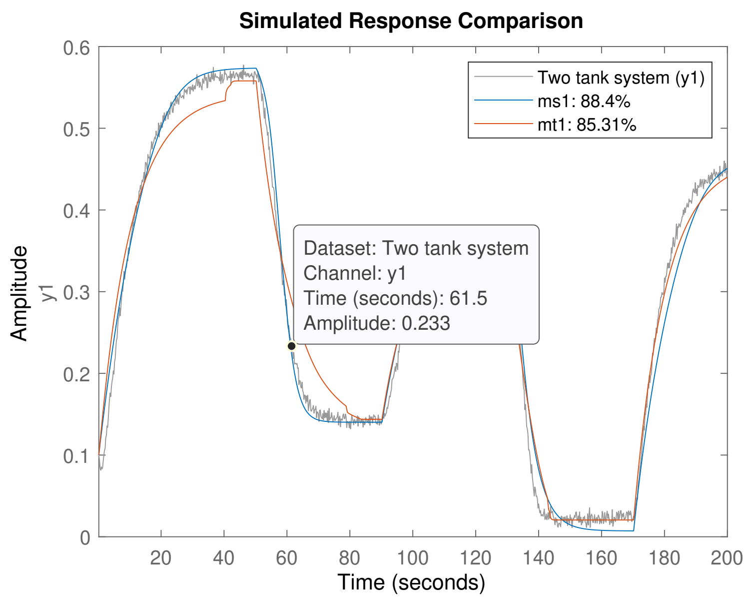


Figure 7: Decision tree based model compared to sigmoid model on the training set.

3 Task 4

In this part of the assignment, the regressor selection was again set to `nn = 5 1 3`, but the second validation set (last 1000 samples) was used for training, and the two validation sets were the first 2000 samples.

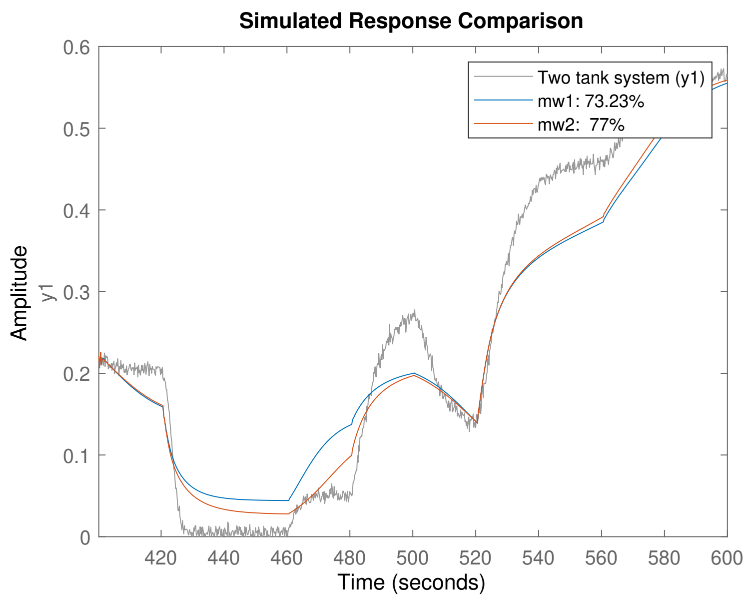
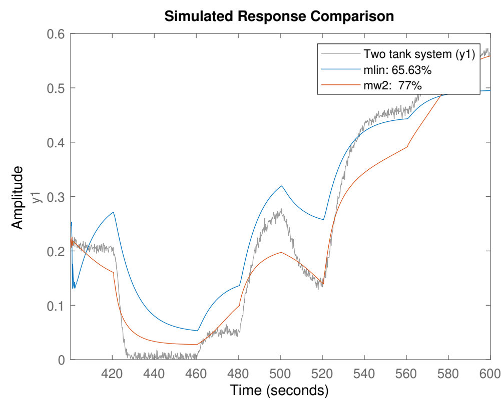


Figure 8: The models mw1 and mw2 on the new training set.

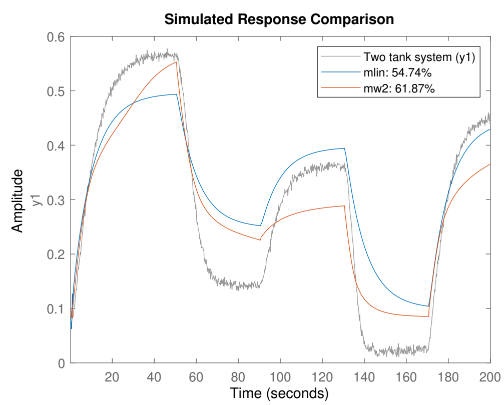
If Figure 8 is compared to Figure 2, we can see that both mw1 and mw2 perform worse on the new training set than they did on the training set that was used i task 2. As shown in Figure 2, the models both change values a little bit slower than the actual system. The system value changes more often in the new training set, and the changes are less regular, which can be the reason why the models are doing worse with this new training set.

As one might expect, the models that have been trained on the last 1000 samples where the system behaves differently than in the first 2000 samples, does not perform very well on the validation sets. The accuracy for the linear and base model are much worse in Figure 9 than in task 2, where the same models were trained on the first 1000 samples.

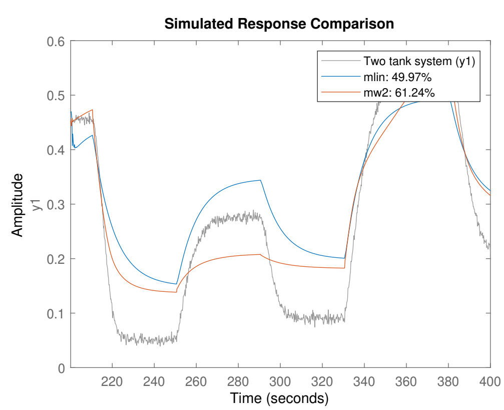
Figure 10 shows a comparison between the residual plots of mw2 and the linear model for the three datasets. From Figure 10a, we observe that there is little correlation between the residue of our model and u1 when running on the test set. On the two validation sets however, this seems to change. We observe that the crosscorrelation between our model output and u1 increase with higher



(a) Training

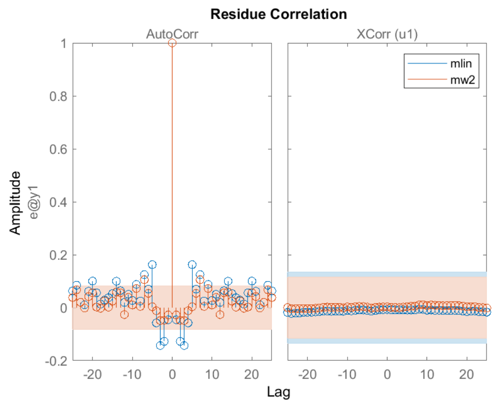


(b) Validation 1

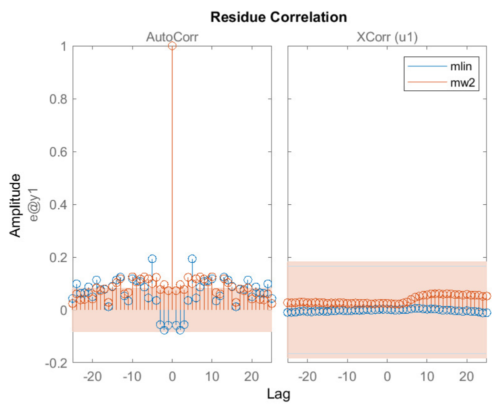


(c) Validation 2

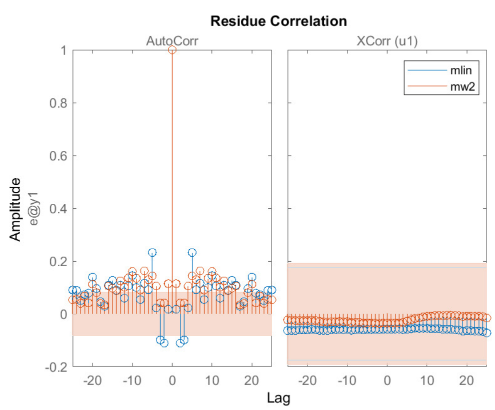
Figure 9: A linear model created with the new training set plotted against the base model for the three datasets.



(a) Training



(b) Validation 1



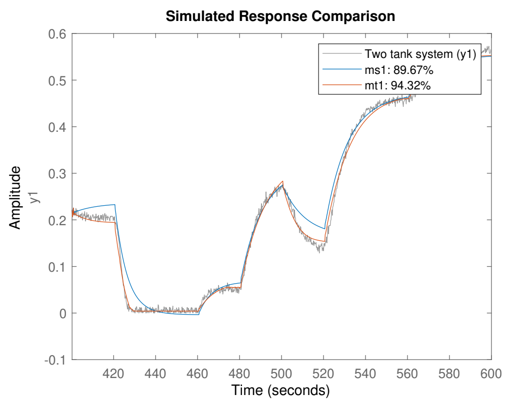
(c) Validation 2

Figure 10: Residual plots for the comparison between base model and linear model for the three datasets.

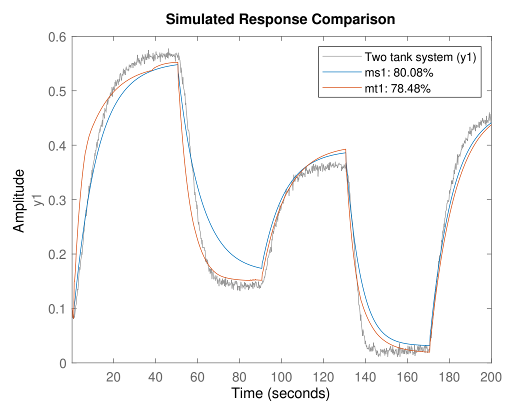
lags, for the two validation sets. The correlation is positive, meaning that the residues increase for increasing input strength. It seems that with higher lag, the model performs worse for larger input values.
Figure 11 shows that, as before, the sigmoid-based model and the Treepartition-based models both performs much better than the linear and base models. However, we can still see that they are a bit overfitted to the training set, and perform significantly worse on the validation sets than on the training set.

As before in this task, we see from Figure 12 that the Hammerstein-Wiener model is overfitted to the training data that is very different from the validation data. This is the clearest example of this overfitting that we have. The model follows the system values very closely on the training set, but has a very low accuracy on both of the validation sets.
Figure 13 shows that the models using saturation and deadzone are not very well suited for modelling the training set. Particularly the model including both of these (Figure 13b) does not correspond with the physical system.

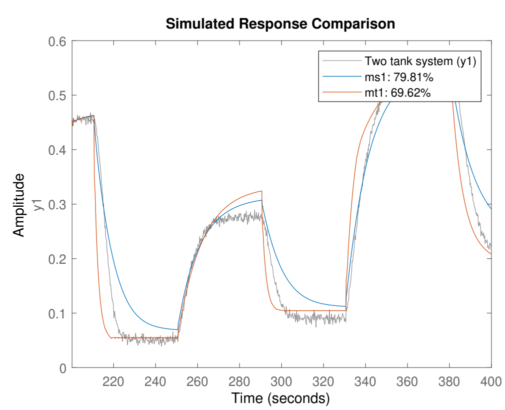
As seen throughout this ask, the models that are trained on the last 1000 samples are not as good as the ones made i task 1 where we used the 1000 samples for training. This is because in the last 1000 samples, the system behaves differently than it does in the 2000 first, causing the trained model to be less accurate for the periodically differing signals that we have in the first 2000 samples. The last 1000 samples are also less “structured” than the earlier ones, and require a more complex model. However, this seems to encourage some overfitting.



(a) Training

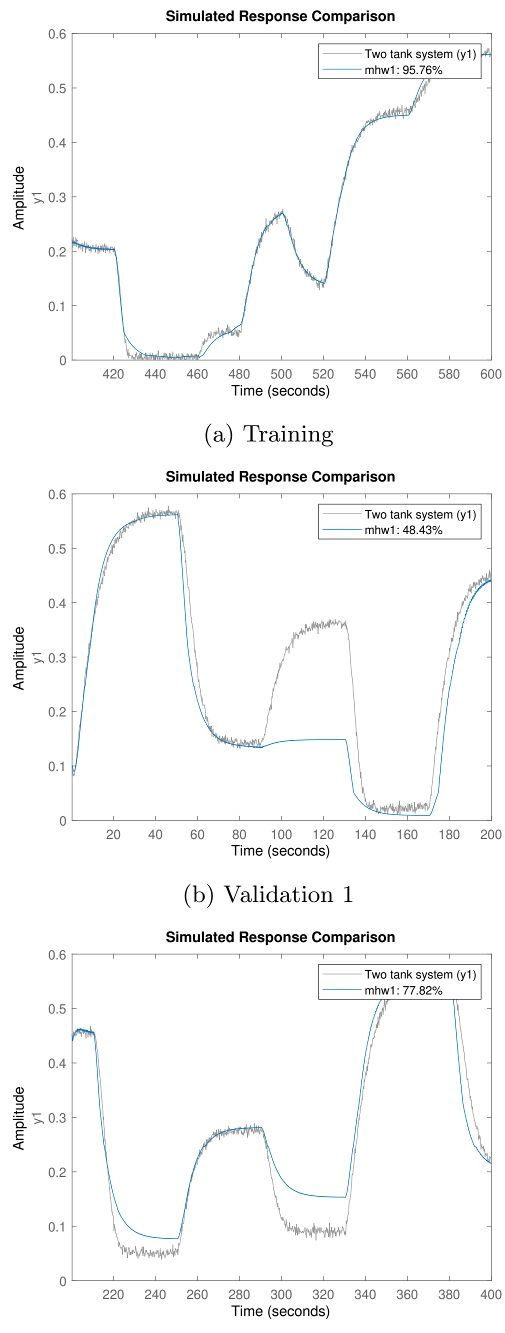


(b) Validation 1



(c) Validation 2

Figure 11: A sigmoid-based model and a Treepartition-based model created with the new training set plotted against eachother for the three datasets.

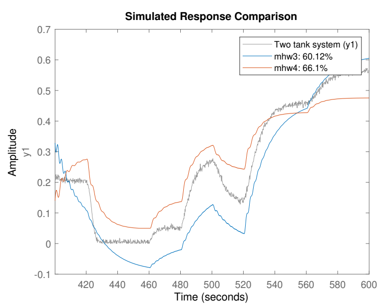


(a) Training

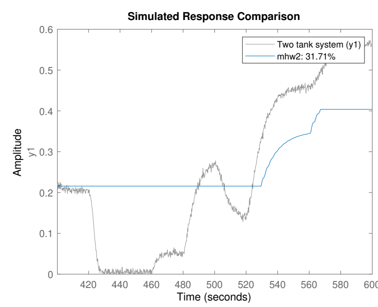
(b) Validation 1

(c) Validation 2

Figure 12: A Hammerstein-Wiener model created with the new training set plotted for the three datasets. The model uses a piece-wise linear function for both the non-linear blocks.



(a) mhw3 uses deadzone as input and unit gain as output, mhw4 uses unit-gain as input and saturation as output.



(b) HW-model with both deadzone and saturation. The first nonlinear block is deadzone and the second is saturation.

Figure 13: Hammerstein-Wiener models with saturation and deadzone.