American University of Beirut

Maroun Semaan Faculty of Engineering and Architecture

Mechanical Engineering Department

MECH 656: System Identification

Final Report

Spring 2019-2020

System Identification of a Liquid-Saturated Steam Heat Exchanger

Theodor Chakhachiro 201700593

Omar Farhat 201701185

Table of Contents

List of Figures

List of Tables

1 PROBLEM DESCRIPTION AND LITERATURE REVIEW

Heat Exchangers are widely used in the industrial space in order to deliver adequate desired temperatures for different usage. Applications of Heat Exchangers range from household temperature maintenance to industrial machinery operations. The fundamental basis of the Heat Exchanger is the heat transfer law between two or more fluids for cooling and heating purposes. The importance of a Heat Exchanger thus lies in the control of the temperature of the outlet fluid that needs to reach a specific temperature in order not to damage the process it is fed to.

The problem of modeling the heat exchanger was tackled by many papers. The complexity of the modeling is due to the nonlinearities within a heat transfer process between two or more fluids. Notable work such as [1] and [2] resort to a neural network approach to be able to obtain a good estimate of the parameters of the designed nonlinear model that best represents the actual physical process. The nonlinear model presented in [1] is a modified Auto Regressive Moving Average with Exogenous Input that includes a nonlinear component of the input in order to minimize the error between the simulated model and the physical process. Other work [3] modeled the heat exchanger system using the distributed parameter model represented by partial differential equations to accommodate for the state variation that takes place in time as well as space. The system was modeled using the following linear models: ARX, ARMAX, OE, and BJ. Because of the non-linear property of the system, the Hammerstein-Wiener model was also used as a model structure. After data samples that met the predefined standard criterion were collected, the model structure was assumed. Then, a parameter estimation algorithm, namely the Prediction error method, was used. Afterwards, an extensive simulation and validation process was carried out using different validation indices, which include the final prediction error, goodness-of-FIT index, and the mean square error. The results showed that the OE model was the best among the linear models. However, the Hammerstein-Wiener model provided the best FIT percentage which was higher than that of all the linear models used.

This problem takes a look at the process happening inside a basic Heat Exchanger where water is heated by pressurized saturated steam through a copper tube [1]. A good system identification of the process parameters is crucial in order to generate a standard two degrees of freedom control law that provides good output tracking of the desired fluid outlet temperature as well as proper stability of the input fluid velocity. The importance of the estimates in the system identification process is thus valued.

2 DATA OVERVIEW

To understand the nature of the data, we will go back to the purpose of the use of a heat exchanger for a heating process. The ultimate result of a heating process is to achieve the desired temperature of the output fluid while minimizing the energy used, that includes the fluid velocity of the heating fluid which is the source of heat for the system.Thus choosing the input to be the velocity of the heating fluid and the output to be the outlet temperature of the second fluid is reasonable for understanding the Input/Output behavior and achieving a good control law based of the study of the IO relationship. In figure 1, the Heat Exchange has two inlets w two outlets for the two fluids. The water enters the coil of the Heat Exchanger at a temperature $\theta_{li}(t)$ and heat transfer starts to occur between the water and the hot pressurized steam entering from the top of the cross flow Heat Exchanger at a temperature $\theta_s(t)$. The heated water will exit at the outlet at a temperature of $\theta_{lo}(t)$. The copper coil of the Heat Exchange is modeled as a continuous long tube with no heat dissipation, the temperature $\theta_l(t,x)$ of the water and the temperature $\theta_m(t,x)$ of the surface of the copper tube are assumed to be a function of time and distance along the tube since heat exchange is occuring along the whole surface and inside the tube. The temperature $\theta_s(t)$ of the steam is only a function of time since the steam reservoir and heat capacity is very large which allows it to take in heat from the water with minimal change of its own temperature along the coils of the cross flow Heat Exchanger [1]. Now that we know the input and output of the system, we need to choose the data accordingly. Since

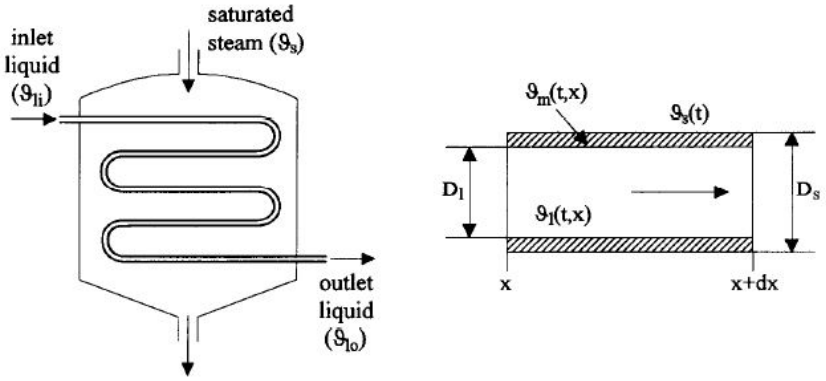

Figure 1: Heat Exchanger physical model

the heat exchanger operates to achieve a high outlet temperature, it is logical to choose an input that yields a high temperature output. Thus a studied choice of the input presented in [1] is depicted as follow.The nominal operating temperature point is chosen to be at $98.765°C$ which corresponds to $0.3\,m/s$. The input is constituted of 4000 samples at a sampling rate of $1\,Hz$, the samples are distributed as follow [1]:

- 100 steady-state samples at $98.765°C$
- 100 Gaussian distributed samples centred at $0.3\,m/s$
- 600 beta-distributed samples with a = 2.142, b = 1.415, weighted towards low speed values
- 1200 beta-distributed samples with a = 1.621, b = 3.829, which privilege high speeds
- 2000 uniformly distributed samples

The above data was obtained from [4] The input/output data thus is not periodic and does not have a zero mean, the mean can be computed from the samples obtained. Moreover, data to be tested at different temperature operating points is missing and thus validation of the proposed model is not as strong.

3 DATA ANALYSIS

3.1 Initial Data

Figures 2 and 3 depict the distribution of the input and output data over all the collected samples. We notice a high peak at the nominal operating point, as expected.

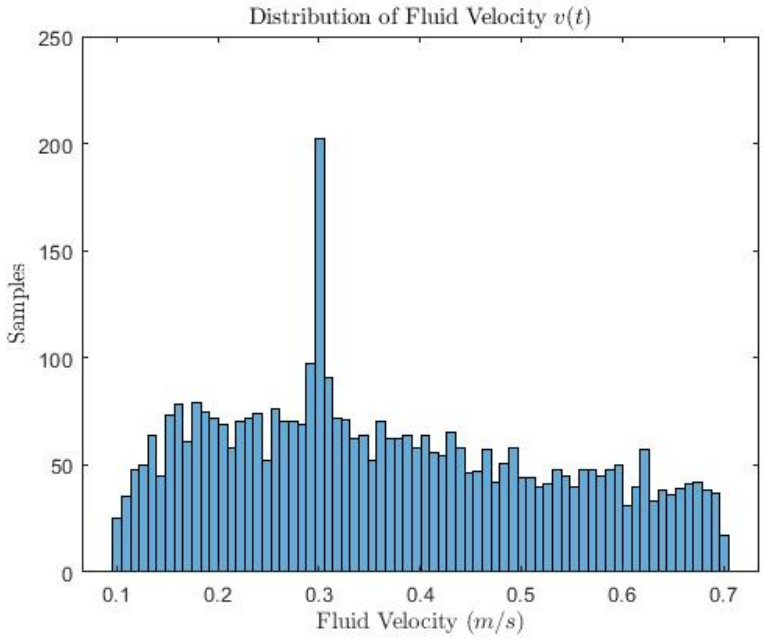

Figure 2: Distribution of Input Fluid Velocity v(t)

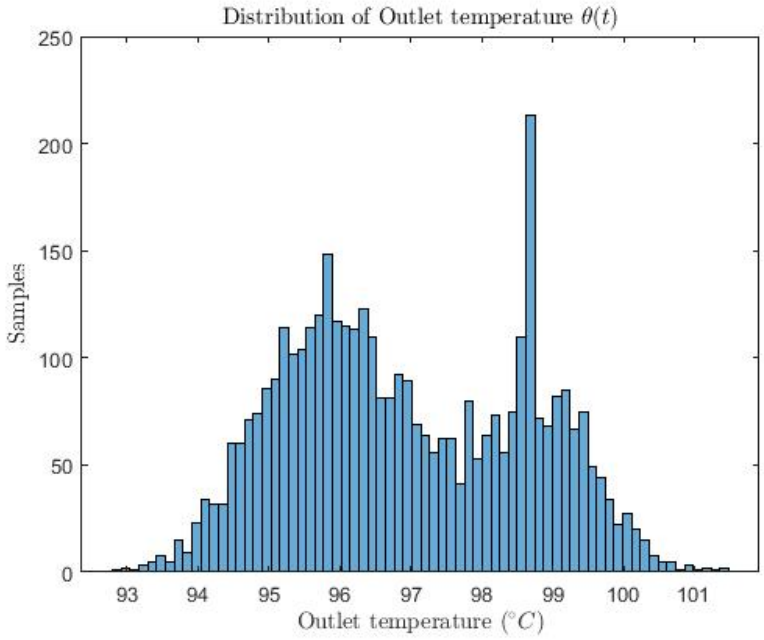

Figure 3: Distribution of Output Outlet Temperature $\theta(t)$

Figure 4 shows the input-output relationship of the system. As seen from the plot, the results are not valuable since the input constitutes of batches of samples, as stated before. Moreover, continuous data is missing from the obtained dataset. Obviously, as stated before, the input-output relationship of the proposed heat exchanger is non-linear and is modeled as a non-linear time-invariant system.

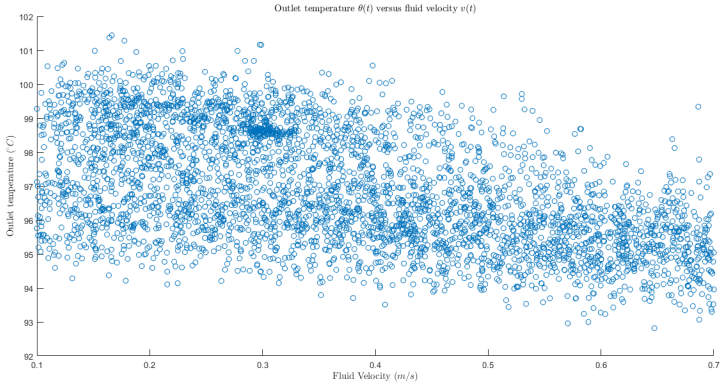

Figure 4: Outlet Temperature $\theta(t)$ versus Fluid Velocity v(t)

3.2 Data Preprocessing

Before proceeding with our analysis, we have taken out the mean of our data for a clearer view of the plots. This is done by removing the mean of each of the five batches mentioned above. Figures 5 and 6 depict the distribution of the input and output data over all the collected samples, after accounting for the mean.

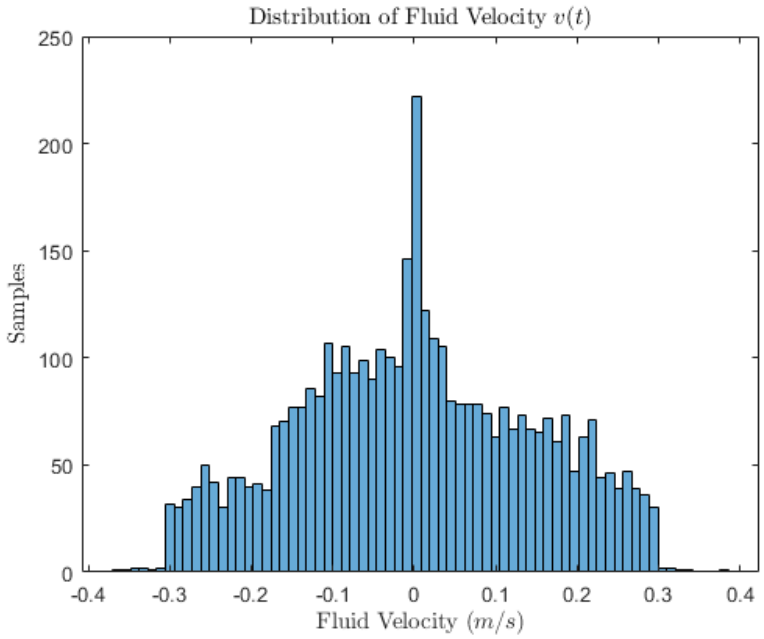

Figure 5: Distribution of Input Fluid Velocity v(t)

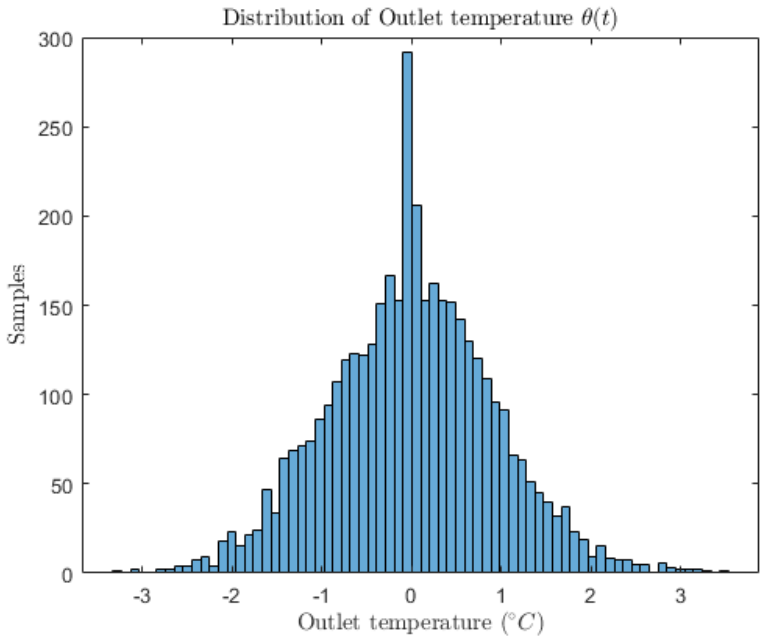

Figure 6: Distribution of Output Outlet Temperature $\theta(t)$

Figure 7 shows the input-output relationship of the system, after accounting for the mean. In fact, figure 7 shows the I/O relationship for each of the five batches as well as the I/O relationship for the whole data.

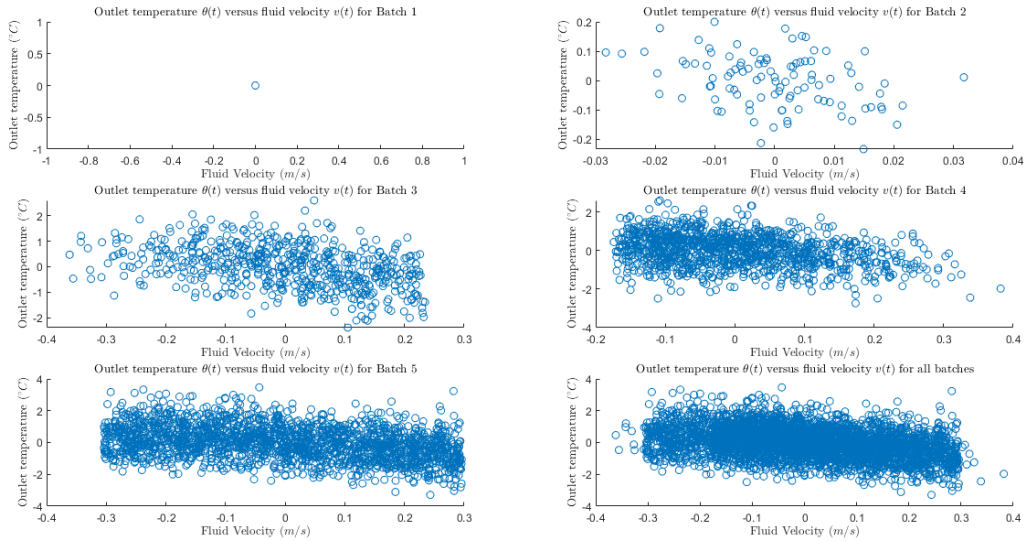

Figure 7: Outlet Temperature $\theta(t)$ versus Fluid Velocity v(t)

To study the feasibility of modeling the input-output relationship of our data using a linear time-invariant system, we have plotted the coherence for the whole data, as well as for each of the five input batches. From figure 8, we can see that the value of coherence is greater than 0.5 for a very wide range of frequencies. Therefore, a linear system can be sufficient to model the I/O relationship. However, for some batches, frequencies between 50 and 70 rad/s admit a lower coherence compared to higher and lower frequencies. Hence, a good choice would be to model the process using an LTI system for high frequencies and an NLTI system for frequencies lying between 50 and 70 rad/s.

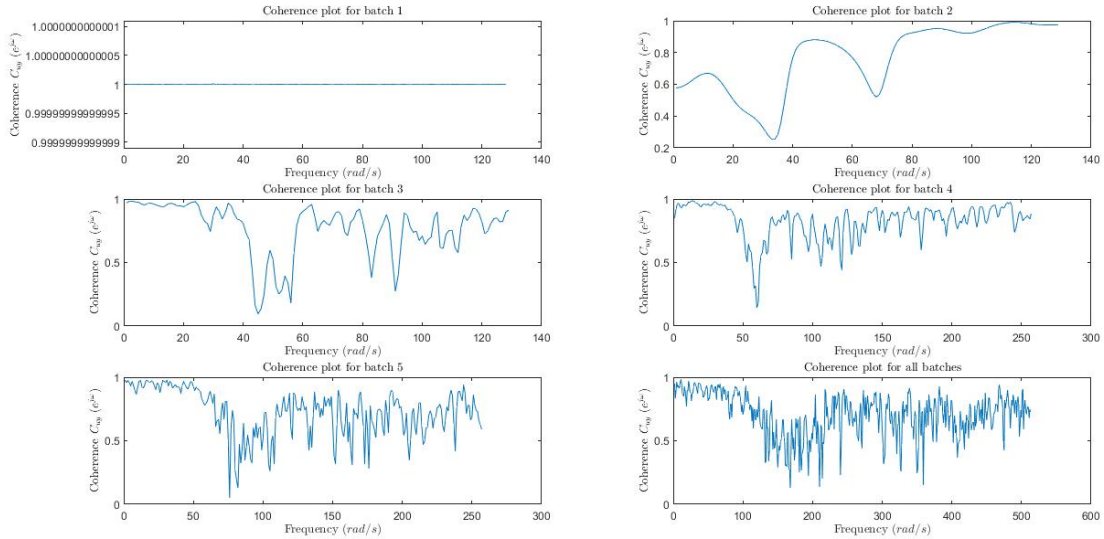

Figure 8: Coherence

3.3 Overview of the Batches

As mentioned above, the processed data is divided into five batches. In what follows, we have plotted the input and output versus time, input and output spectral density, input and output autocorrelation, input and output autocovariance, and input-output crosscorrelation and crosscovariance for each of the five batches. These plots are shown in figures 9 to 13, which correspond to batches 1 to 5 respectively.

For batch 1, the power spectral density of the input is maximal at the origin and lower but constant for increasing frequencies. This is expected since our input is constant and thus has a persistence of excitation order of zero. The same applies for the output power spectral density. From the input and output autocorrelation plots, we can see that the autocorrelation function is slowly varying over the lags since the magnitude is of order 10^{-20}. This is expected also since we have a constant input and output which means that we have equally correlated inputs/outputs over lags. This result can be further justified by looking at the autocovariance plots of the input and the output. We can see that the autocovariance function is equally distributed over all lags with a mean of zero. Moreover, the crosscorolation and crosscovariance plots further justify this result since the crosscorrelation is slowly varying over lags (with a magnitude of power 10^{-27}) and the crosscovariance expresses the same result as the autocovariance plots.

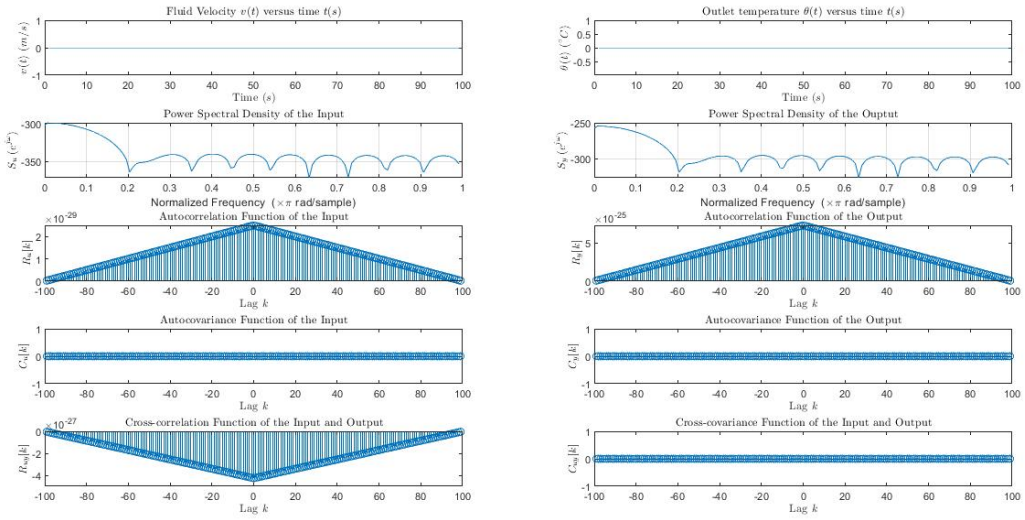

Figure 9: Batch 1

The second batch includes 100 Gaussian distributed samples around 0.3 m/s. From figure 10, we can see that the power spectrum of the input is overall equally distributed across all frequencies as expected (inaccuracies are due to the lack of data). The power spectrum of the output slowly decays with increasing frequencies. The autocorrelation of the input and output, the autocovariance of the input and output as well as the crosscovariance and crosscorelation all show a maximum at the origin and have equal distribution for all other lags. This result is expected for a Gaussian distribution.

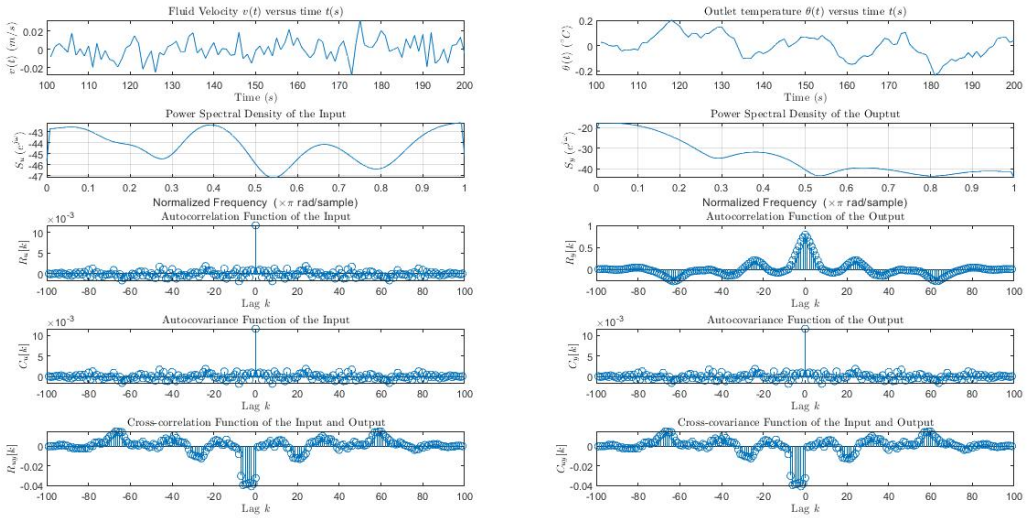

Figure 10: Batch 2

The third and fourth batch are $\beta-$Distributed samples where the third batch is positively skewed and the fourth batch is negatively skewed. This distribution resembles the Gaussian distribution and the results displayed in figures 11 and 12 justify this statement. The justification of the results is the same as the results in batch 2.

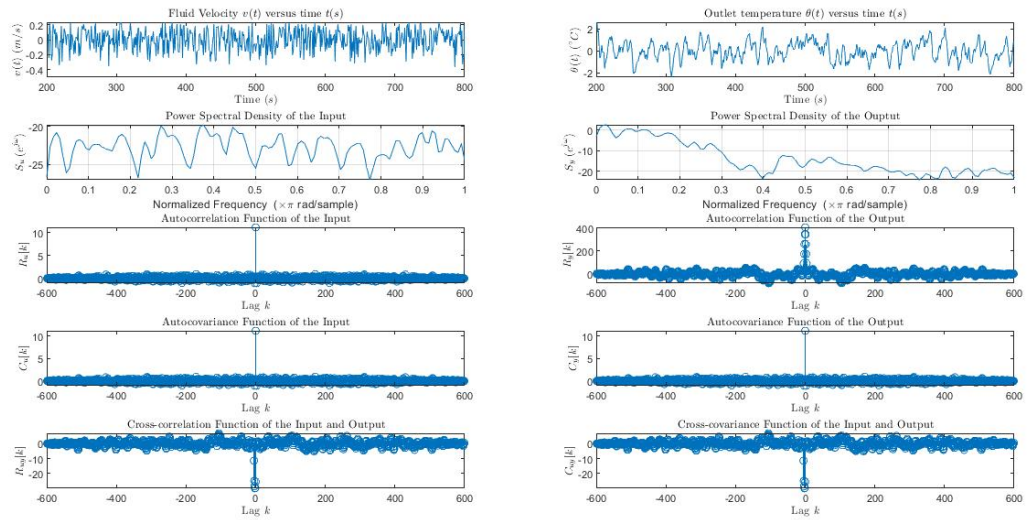

Figure 11: Batch 3

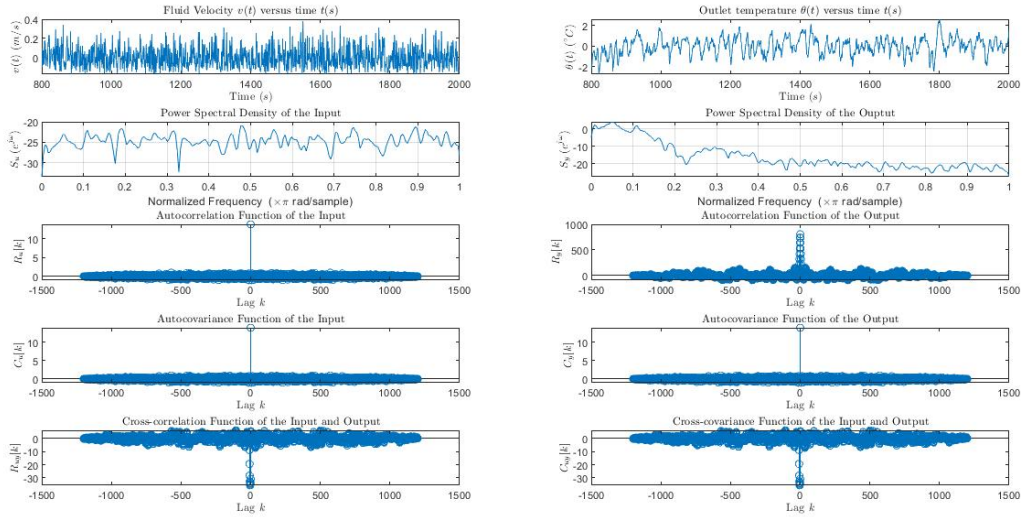

Figure 12: Batch 4

The last batch contains 2000 normally distributed samples. The spectrum of the input is equally distributed across all frequencies as expected. The output spectrum is maximum at the origin and slowly decays with increasing frequencies. Finally, the autocorrelation of the input and output, the autocovariance of the input and output as well as the crosscovariance and crosscorelation all show a maximum at the origin and have equal distribution for all other lags. This results highly resembles that of batches two, three and four due to the common properties of the Gaussian distribution and the $\beta-$Distribution.

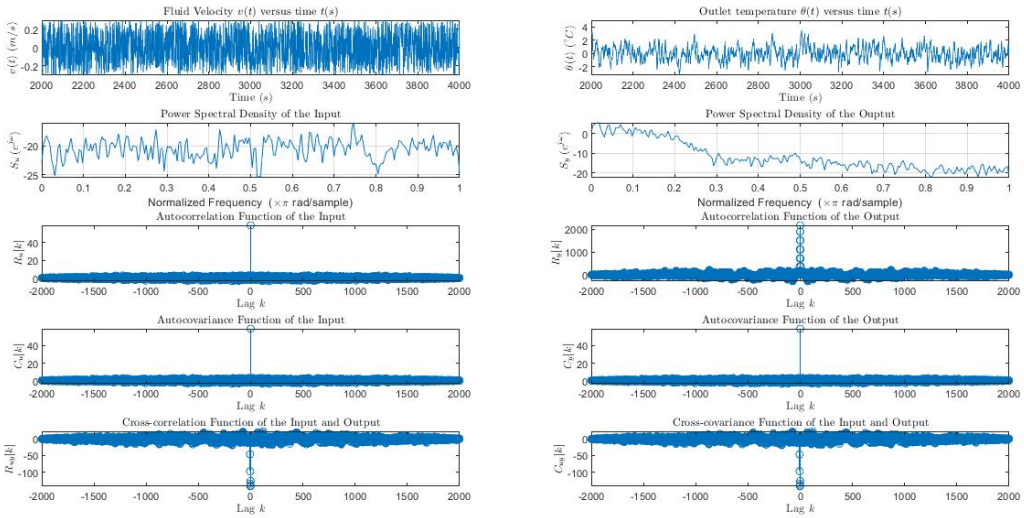

Figure 13: Batch 5

As a side note, we can see that the crosscorrelation and crosscovariance functions of all batches express negative magnitude at low lags. A physical interpretation can explain this phenomenon. In fact, at low lags, the outlet temperature $\theta(t)$ highly depends on the input velocity $v(t)$. However, this dependency includes a negative sign which can be interpreted by the fact that the outlet temperature begins to fight off the incoming effect of the most recent input velocity. The fighting off ability is due to the previously stored energy of the system which is non minimum phase. Thus, with a stored energy, the system can fight off the initial perturbation of the input velocity but finally succumbs to it and follows its effect in a similar fashion.

4 MODELING

We have decided to use the fifth batch, which constitutes of 2000 (out of 4000) data points, for the modeling part and training each model. Afterwards, we are going to test our models using the fourth batch which includes 1200 samples. The main reason behind this choice is by looking at the coherence plots and seeing that our system can be represented by and LTI model at high frequencies which is the case with the Gaussian distribution as well as the $\beta-$Distribution. Since our system is clearly non minimum phase and since the output depends highly on the input at low lags, an appropriate model choice decision would be to discard the Auto-regressive as well as the Finite Impulse Response model. The model choices that will be presented include the Auto-Regressive with Exogenous input model, the Auto-Regressive Moving Average with Exogenous input model, the Output Error model, and the Box Jenkins model. Finally, we will also model our data using the Non-linear Hammerstein-Wiener model since the behavior of our system cannot be described as linear over all the range of frequencies.

4.1 ARX model

Starting with model, we are assuming that the noise as well as the input express similar dynamics when introduced to the system. The ARX model can be represented by the following model:

$$A(z^{-1})y(k) = B(z^{-1})u(k - n_k) + e(k)$$

Where:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{n_a} z^{-n_a}$$
$$B(z^{-1}) = 1 + b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_b} z^{-n_b}$$

For this model, we decided to vary the orders of the polynomials $A(z^{-1})$ and $B(z^{-1})$. Thus, we tried various combinations of orders where $n_a = deg(A(z^{-1}))$ and $n_b = deg(B(z^{-1}))$ ranged each from 1 to 10. Also, we set $n_k = 0$ since our system showed high correlation and covariance for low lags.

4.2 Output Error model

Assuming that the noise is introduced at the output and that the initial output is noise free, the Output Error model can be represented by the following model:

$$y(k) = \frac{B(z^{-1})}{F(z^{-1})} u(k - n_k) + e(k)$$

Where:

$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_b} z^{-n_b}$$
$$F(z^{-1}) = f_1 + f_2 z^{-1} + ... + f_{n_f} z^{-n_f}$$

Similarly to the ARX model, we decided to vary the orders of the polynomials $B(z^{-1})$ and $F(z^{-1})$. Thus, we tried various combinations of orders where $deg(B(z^{-1}))$ and $deg(F(z^{-1}))$ ranged each from 1 to 10. Also, we set $n_k = 0$ since our system showed high correlation and covariance for low lags.

4.3 ARMAX model

Assuming the noise introduced is colored, the ARMAX model can be represented by the following model:

$$A(z^{-1})y(k) = B(z^{-1})u(k - n_k) + C(z^{-1})e(k)$$

Where:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{n_a} z^{-n_a}$$
$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_b} z^{-n_b}$$
$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + ... + c_{n_c} z^{-n_c}$$

Similarly to the previous model, we decided to vary the orders of the polynomials $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$. Thus, we tried various combinations of orders where $n_a = deg(A(z^{-1}))$,$n_b = deg(B(z^{-1}))$ and $n_c = deg(C(z^{-1}))$ ranged each from 1 to 10. Also, we set $n_k = 0$ since our system showed high correlation and covariance for low lags.

4.4 Box-Jenkins model

Assuming the noise and input express different dynamics when introduced to the system and that the noise introduced is in fact colored, we obtain the following Box-Jenkins model:

$$y(k) = \frac{B(z^{-1})}{F(z^{-1})}u(k - n_k) + \frac{C(z^{-1})}{D(z^{-1})}e(k)$$

Where:

$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_b} z^{-n_b}$$
$$F(z^{-1}) = 1 + f_1 z^{-1} + f_2 z^{-2} + ... + f_{n_f} z^{-n_f}$$
$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + ... + c_{n_c} z^{-n_c}$$
$$D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} + ... + d_{n_d} z^{-n_d}$$

Similarly to the previous model, we decided to vary the orders of the polynomials $B(z^{-1})$, $F(z^{-1})$, $C(z^{-1})$ and $D(z^{-1})$. Thus, we tried various combinations of orders where $n_b = deg(B(z^{-1}))$,$n_f = deg(F(z^{-1}))$, $n_c = deg(C(z^{-1}))$ and $n_d = deg(D(z^{-1}))$ ranged each from 1 to 10. Also, we set $n_k = 0$ since our system showed high correlation and covariance for low lags.

4.5 Non-Linear Hammerstein-Wiener model

Finally, the None-Linear Hammerstein-Wiener model is considered since our coherence plot showed that our system cannot be linearized for frequencies between 50 and 70 rad/s The Hammerstein-Wiener is a block-oriented model that is a combination of two static non-linear elements and linear dynamic element [3]. It can be expressed as follows:

$$y(k) = f_2\left(\frac{B_1(z^{-1})}{A_1(z^{-1})}f_1(u(k))\right) + \epsilon(k)$$

Where:

$$A_1(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{n_a} z^{-n_a}$$
$$B_1(z^{-1}) = 1 + b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_b} z^{-n_b}$$

and f_1 and f_2 are non-linear functions. In our case, f_1 and f_2 are piecewise-linear functions.

5 RESULTS AND DISCUSSION

After running the simulation for the various orders of each model listed in the previous section, the final 7 models picked are displayed in table 1. The choice of these models were based on the normalized Akaike's Information Criterion as well as the FIT percentage. On a side note, the table also includes the Final Prediction Error (FPE) and the Mean Square Error (MSE) corresponding to each model. We can notice that all the seven chosen models admit very close values of the FPE and MSE, except for the Output Error model which has a relatively higher MSE and FPE values.
Since the results showed that the models based on the normalized AIC criterion provided similar FIT percentage to the models based on the latter, we decided to discard the models based on the FIT percentage and only study the behavior of the models based on the normalized AIC criterion. The number of models is reduced to five: Models 1,2,3,4 and 5.

Table 1: Results of the Models

Models	Type of Model	Polynomial Orders	FIT %	Normalized AIC	FPE	MSE
Model 1	ARMAX	$[n_a, n_b, n_c]$=[3,8,9]	78.1062	-2.9245	0.0537	0.0521
Model 2	Output Error	$[n_b, n_f]$=[9,3]	70.727	-2.3535	0.095	0.0932
Model 3	ARX	$[n_a, n_b]$=[3,8]	77.9615	-2.9233	0.0538	0.0528
Model 4	Box-Jenkins	$[n_b, n_c, n_d, n_f]$=[2,10,10,8]	78.3628	-2.9291	0.0534	0.0509
Model 5	Nonlinear Hammerstein-Wiener	$[n_b, n_f]$=[8,3]	78.1388	-2.9034	0.0548	0.052
Model 6	ARMAX	$[n_a, n_b, n_c]$=[4,10,10]	78.1434	-2.9239	0.05373	0.05193
Model 7	ARX	$[n_a, n_b]$=[10,10]	78.031	-2.9177	0.0541	0.0525

To validate the selection of the models obtained, we plotted the normalized autocorrelation of the residual error as well as the normalized crosscorrelation of the input and the residual error for all five models previously chosen. Figure 14 shows the normalized autocorrelation function of the residual error using the training data. The red lines correspond to the 95% confidence interval for $N = 2000$ which corresponds to:

$$|x_\tau| \leq \frac{1.96}{\sqrt{(N)}}$$

Where:

$$x_\tau = \frac{R_\varepsilon[\tau]}{R_\varepsilon[0]}$$

We can see that all models express the same behaviour. In fact, for low lags the autocorrelation is high but it decreases and stays within he 95% confidence interval bounds. Models 1 and 4 have the best results by a slow margin compared to the other models. This result implies that the residual error is indeed zero mean white noise and that it has symmetric properties.

Figure 15 shows the normalized crosscorelation function between the input and the residual error using the training data to further validate the model choice under the 95% confidence interval criterion where:

$$|x_\tau| \leq \frac{1.96}{\sqrt{(N)}}$$

Where:

$$x_\tau = \frac{R_{u\varepsilon}[\tau]}{\sqrt{R_u[0]R_\varepsilon[0]}}$$

From the plot, we can see that all models express the same behaviour again. The normalized crosscorrelation is equally distributed along all the lags. Its mean value is 0 and only fluctuates a little around this mean. This allows us to confirm that the residual error is independent of passed/all inputs and that there is no more information to be extracted from the input. Models 1 and 4 also provided the best results by a slow margin compared to the other models.

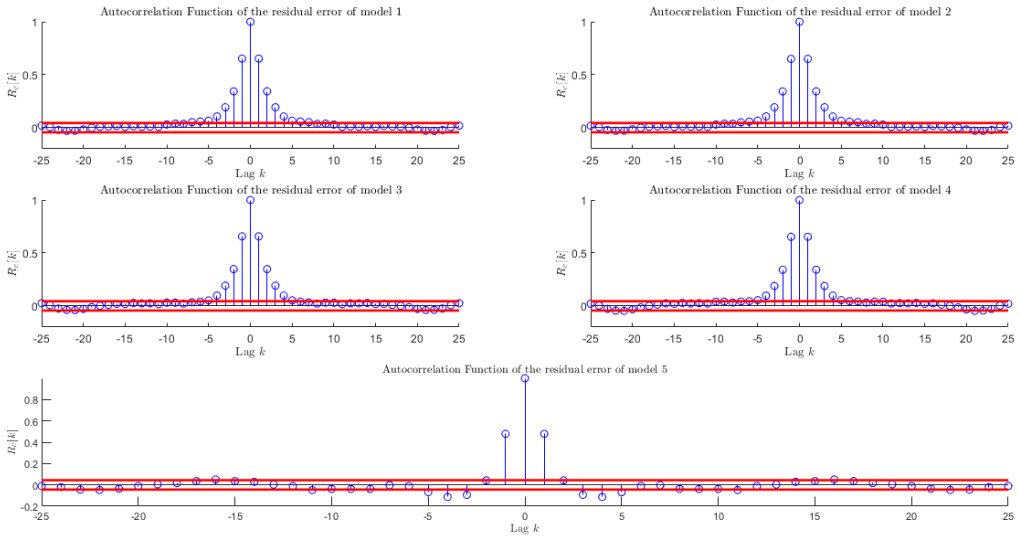

Figure 14: Normalized Autocorrelation Function of the Residual Error using Training Data

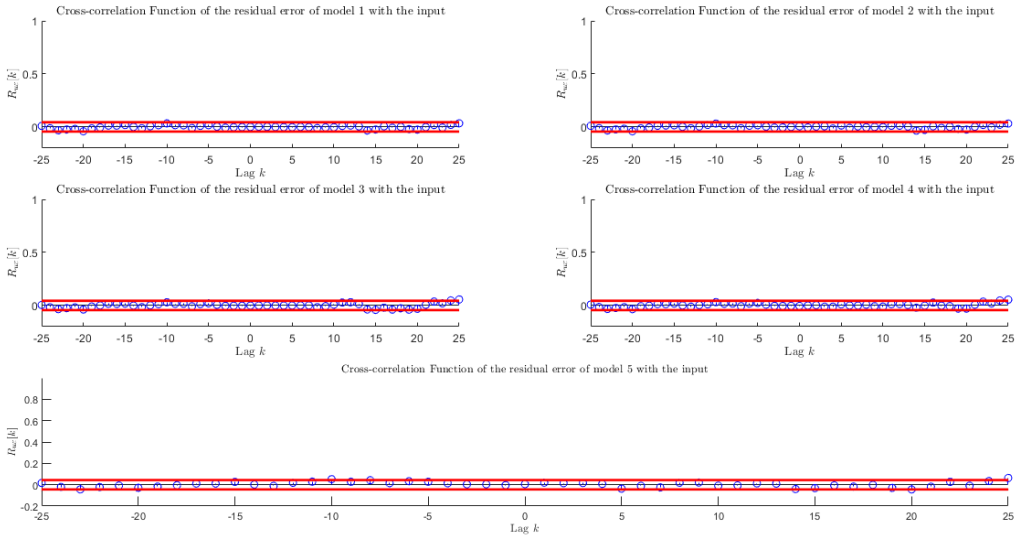

Figure 15: Normalized Crosscorelation Function between the Input and the Residual Error using Training Data

Figure 16 shows the residual error for each model using the training data. We can see that the residual error highly fluctuates between -0.2 and 1 for all models except model 5 which has very high residual error.

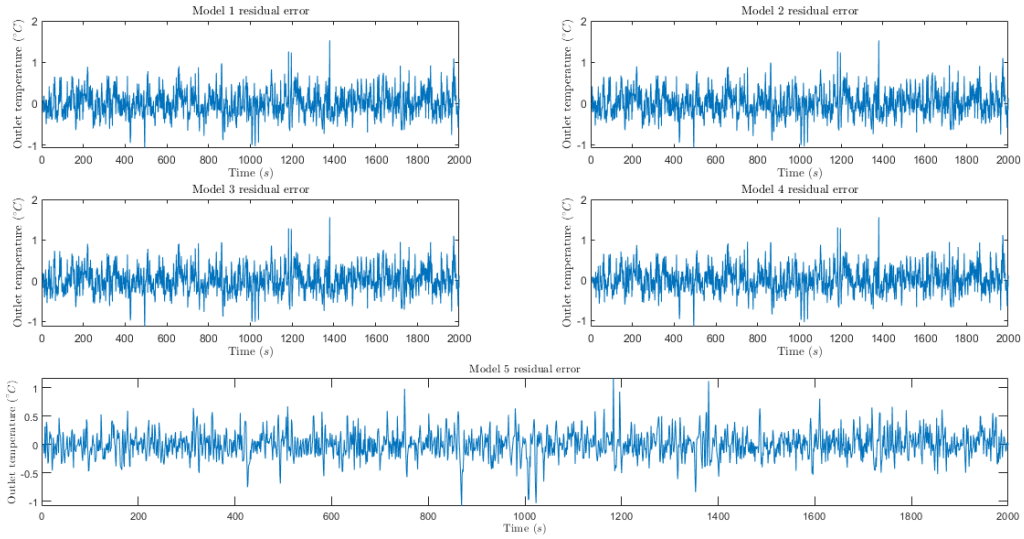

Figure 16: Residual Error for all models using Training Data

Figures 17 and 18 show the accuracy of the models for the training data and the testing data respectively. We can see that in general, models 1 and 4 expressed the best results while model 5 had the lowest testing results.

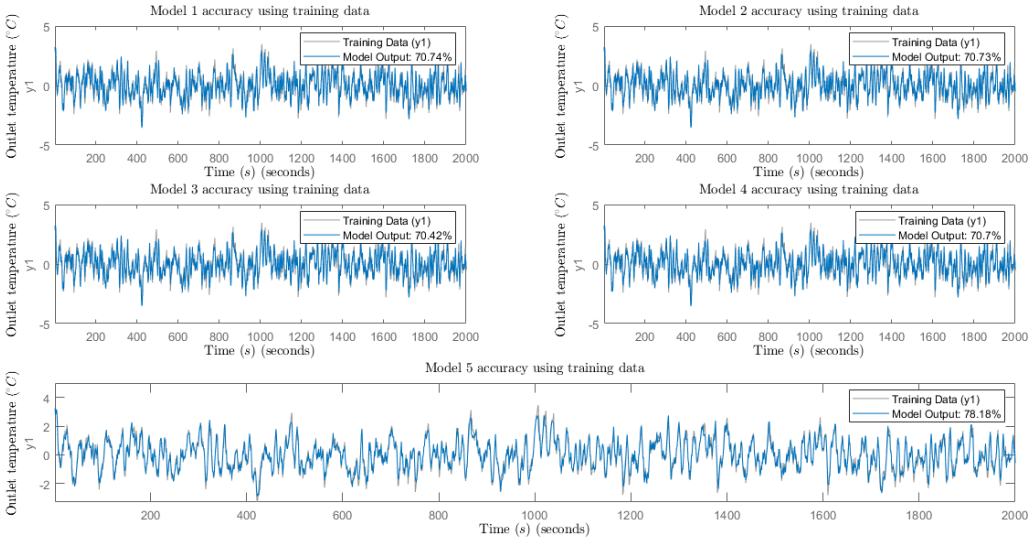

Figure 17: Models Accuracy using Training Data

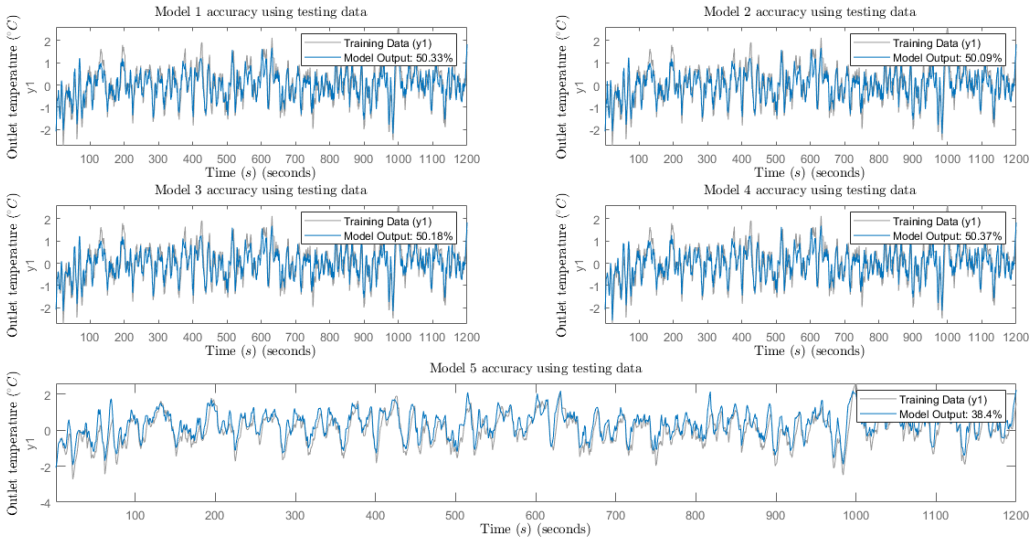

Figure 18: Models Accuracy using Testing Data

Based on the results previously discussed, the final two models chosen are model 1 and model 4 which correspond to the ARMAX model and the Box-Jenkins model respectively. The structures of these two models are displayed below:

Model 1:

$$A(z^{-1})y(k) = B(z^{-1})u(k - n_k) + C(z^{-1})e(k)$$

Where:

$$A(z^{-1}) = 1 - 2.0787z^{-1} + 1.542z^{-2} - 0.3162z^{-3}$$
$$B(z^{-1}) = -2.293 + 2.218z^{-1} - 0.837z^{-2} - 0.3843z^{-3} - 0.2895z^{-4} + 0.08653z^{-5} - 0.03416z^{-6} - 0.4578z^{-7} - 0.3354z^{-8}$$
$$C(z^{-1}) = 1 - 1.328z^{-1} + 0.3595z^{-2} + 0.2699z^{-3} + 0.01106z^{-4} + 0.03118z^{-5} + 0.03369z^{-6} + 0.01117z^{-7} - 0.04165z^{-8} + 0.03251z^{-9}$$

Model 4:

$$y(k) = \frac{B(z^{-1})}{F(z^{-1})}u(k - n_k) + \frac{C(z^{-1})}{D(z^{-1})}e(k)$$

Where:

$$B(z^{-1}) = -2.298 + 2.112z^{-1} - 1.784z^{-2}$$
$$F(z^{-1}) = 1 + -2.028z^{-1} + 1.898z^{-2} - 0.9453z^{-3} + 0.1221z^{-4} + 0.167z^{-5} + -0.07976z^{-6} - 0.04968z^{-7} + 0.04164z^{-8}$$
$$C(z^{-1}) = 1 - 0.5178z^{-1} - 0.1165z^{-2} - 0.1546z^{-3} + 0.3028z^{-4} - 0.07065z^{-5} + 0.1002z^{-6} - 0.7752z^{-7} + 0.6368z^{-8} + 0.3851z^{-9} - 0.0109z^{-10}$$
$$D(z^{-1}) = 1 - 1.276z^{-1} + 0.4638z^{-2} - 0.2332z^{-3} + 0.4584z^{-4} - 0.345z^{-5} + 0.2329z^{-6} - 0.8906z^{-7} + 1.275z^{-8} - 0.3232z^{-9} + -0.07813z^{-10}$$

Side Note:
Another simulation was done where we used all the data we had, i.e. all five batches, to train both models 1 and 4. The results showed that model 1 had a 87.48 % FIT with an FPE of 0.04441 and MSE of 0.04375. Model 4 showed an 87.4 % fit with an FPE of 0.04541 and an MSE of 0.04431. The results are depicted in figure 19.

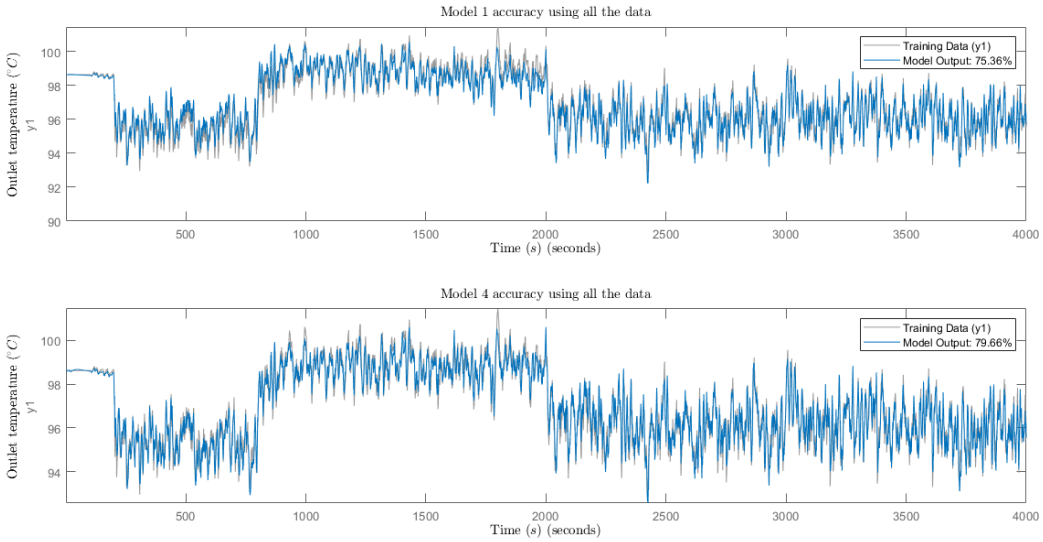

Figure 19: Models 1 and 4 Accuracy Using all the Dataset

6 CONCLUSION

Based on our discussion in the previous section, we can see that models 1 and 4 best describe our data. They both have almost equal values for the different criteria considered: FIT percentage, AIC, FPE, and MSE. Moreover, they have resulted in equally accurate models using both the training and testing data. However, we have decided that Model 1 is our choice because it has a lower order of 9 compared to Model 4 which has an order of 10. Also, Model 1 has only 20 parameters which is much less than that of model 4 that has 30 parameters. Therefore, we can say that our system can be correctly represented by an ARMAX model of order 9 which means that the noise introduced is colored and that it has the same model dynamics as the input when introduced to the system.
Regarding the data, the number of samples used to train was low which caused a lower FIT percentage compared to training the full dataset. Therefore, future work can take into account this lack of data by augmenting the dataset and testing/training on the expanded dataset.

7 REFERENCES

[1] S. Bittanti and L. Piroddi, “Nonlinear identification and control of a heat exchanger: A neural network approach,” *Journal of the Franklin Institute*, vol. 334, no. 1, p. 135–153, 1997.

[2] T. R. Biyanto, M. Ramasamy, and H. Zabiri, “Modeling heat exchanger using neural networks,” *2007 International Conference on Intelligent and Advanced Systems*, 2007.

[3] S. Gupta, R. Gupta, and S. Padhee, “Parametric system identification and robust controller design for liquid–liquid heat exchanger system,” *IET Control Theory Applications*, vol. 12, no. 10, p. 1474–1482, Mar 2018.

[4] B. De Moor, “Liquid-saturated steam heat exchanger, daisy: Database for the identification of systems, department of electrical engineering, esat/stadius, ku leuven, belgium,” 1995. [Online]. Available: https://homes.esat.kuleuven.be/~smc/daisy/daisydata.html