2024

CHE 662

Process Identification

QUADRUPLE TANK LEVEL SYSTEM DATA DRIVEN PROCESS IDENTIFICATION

Lab Instructor:

Hossein Mohammadghasemi

Course Instructor:

Dr. Junyao Xie

Author

Abhi Chauhan

1811065

Abstract:

This project investigated identifying a MIMO (Multiple Inputs Multiple Output) quadruple tank level system using different parametric and non-parametric method. A step test is designed Random Binary Sequences (RBS) for system excitation. The MIMO system was separate out into two MISO processes for individual tank level modeling. Spectral analysis revealed similar frequency responses with a phase shift in the 2nd input at higher frequencies whereas the Impulse responses gives us the time delay. Different types of parametric model are used for model identification. Two different: Best model and Optimal model is identified for each MISO process. Then the result is compared with nonlinear ARX model, state space model as well as the LSTM model (enrichment part). LSTM network was implemented to identify the MISO processes. The LSTM network aimed to capture the system dynamics, potentially including non-linear effects, long-term dependencies, and compared with previously explored models. However, necessity of lager data set to train the LSTM models causes the poor performance for model identification. Hence, BJ (Box-Jenkins) model is identified as the best model having 92.75% and 83.65% MISO data fitting and based on parsimony's principal, OE (Output Error) model is identified as optimal model having least AIC of -11.7 & -9.55 for MISO models.

Table of Contents

List of Tables

List of Figures

1 Introduction:

As shown in Figure 1 four water tanks used which are interconnected to each other as per configuration. 2 no. of pump (identified as left pump and right pump) with remotely adjustable flowrate used for the water circulation into the system. Each pump outlet connected to respected top tank and bottom tank (from the cross side) as shown in figure. Therefore, bottom tank will receive the flow rate from pump as well as the upper tank. Our area of interest is the two inputs which are left pump flowrate (U1) and right pump flowrate (U2) whereas the output of the system is the bottom tank 1 and 2, mark as outputs Y1 and Y2. Therefore, system has two inputs and two output which makes it as MIMO system. To obtain the basic characteristic such as the time delay, gain and time constant we design the step input as per the Table 1. Then the data is used to design the perfect input for the system: RBS (Random Binary Signal). For each input i.e. left and right pump, two RBS signal is created. Then RBS signal sends to the system to generate the output data. Over period of time data is generated and collected. Further, data preprocessing, model identification and advance method such as LSTM is performed and discuss in this report.

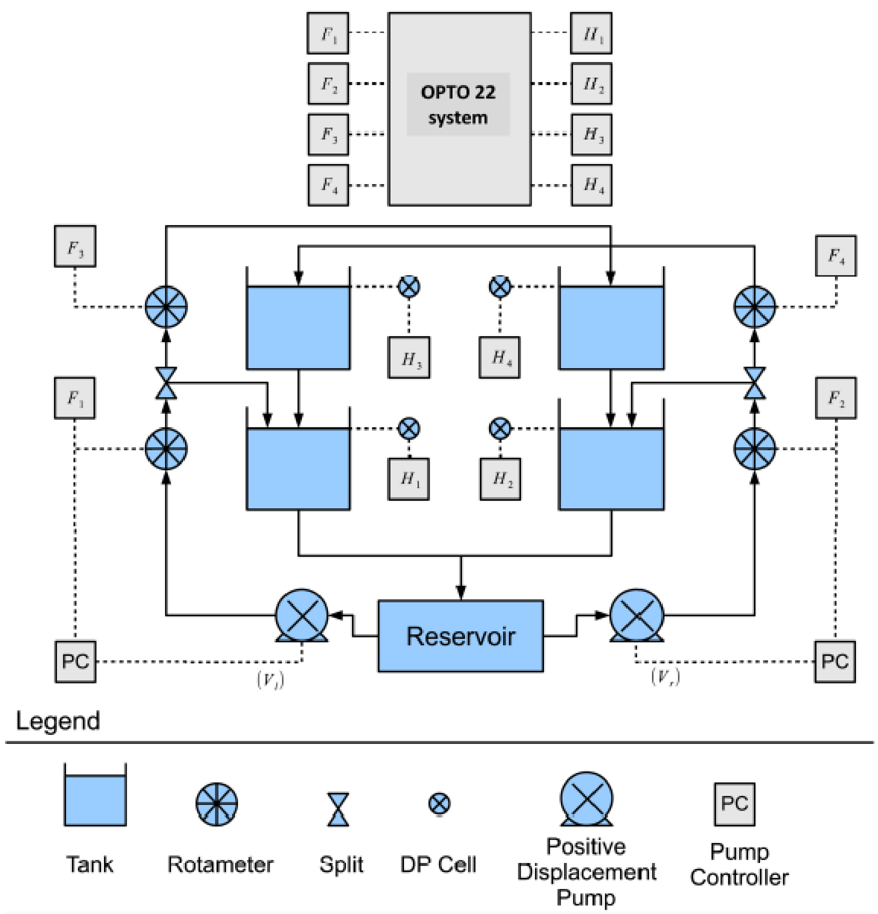


Figure 1 Schematic diagram of Four-tank system

2 Methods

2.1 Data extraction from Step Test

To identify the First Order Plus Time Delay (FOPTD) model of the system, first we perform the step test from steady state flow rate 12 L/min to 14 L/min from each pump. The results from the step response are shown in Appendix (Data Sheet: -5.1).

2.2 Design of RBS signal: -

As experiments have two inputs (Left and right pump flow rates), resultant 2 RBS signals are created by using MATLAB function ***'idinput'***. To create the RBS following parameters are taken into consideration.

Table 1 RBS signal parameters selections

Parameters	values	Criteria
Sampling time (T_s)	14 seconds	0.15 times of time constant (α=0.15)
Frequency Bandwidth (ω)	0.1452	k =3
Step size	[-1.5 1.5]	0.75 time of step input (0.75 x 2 =1.5)

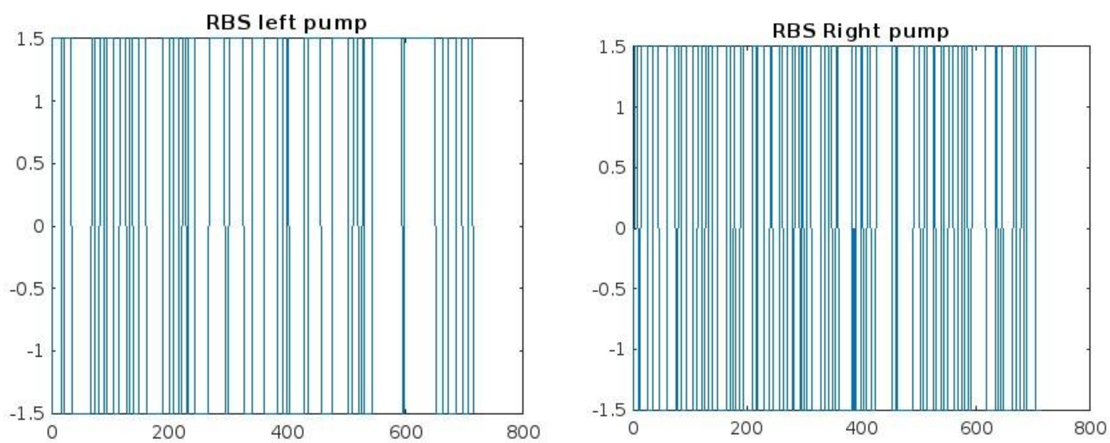


Figure 2 RBS signal generated for Left pump and Right pump

3 Result and Discussion

In this project we are focusing on the two MISO (Multiple Inputs Single Output) models: Z_L which is bottom left level as output (Y1) with respect to two input as left flowrate (U1) and right flowrate (U2). Z_R which is bottom right level as output (Y2) with respect to two input as left.

3.1 Data Preprocessing

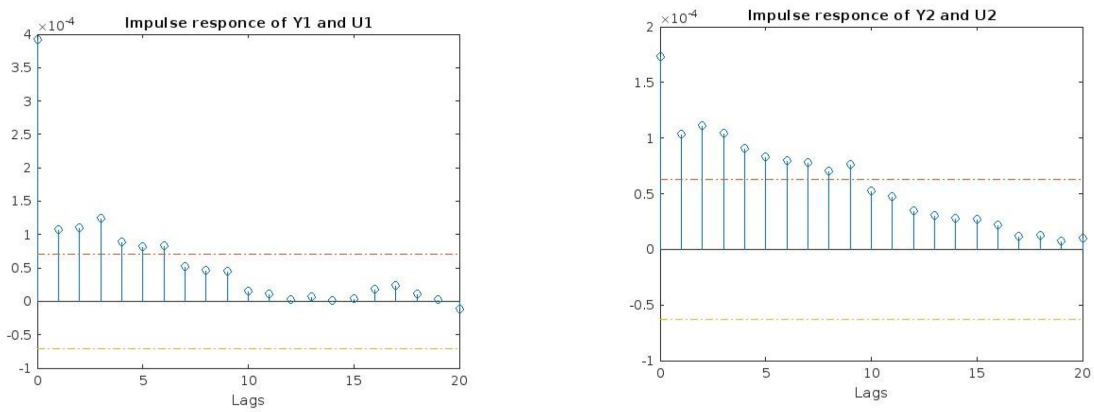
To detect the outlier from the 2-model dataset 3rd order ARX model is used. Based on the residual and MSE (Mean square error) from the actual and predicted based on ARX model outliers are detected. Conventional 3σ rules is used for the outlier detection and it was relaced by the predicted values.

$$-3\sigma \leq di \geq \ 3\sigma$$

$$where, di = \frac{Ei}{\sqrt{MSE}}$$

From the Figure 10[1] (in Appendix 5.2) it can see that there is total **7** and **16** outlier in Z_L and Z_R respectively. Which was replaced by predicted values. For further treatment of the data, we use MATLAB function ‘***dtrend’*** which deal with any nonzero means from the dataset.

3.2 Impulse Response of the system



To determine the impulse response of the system,

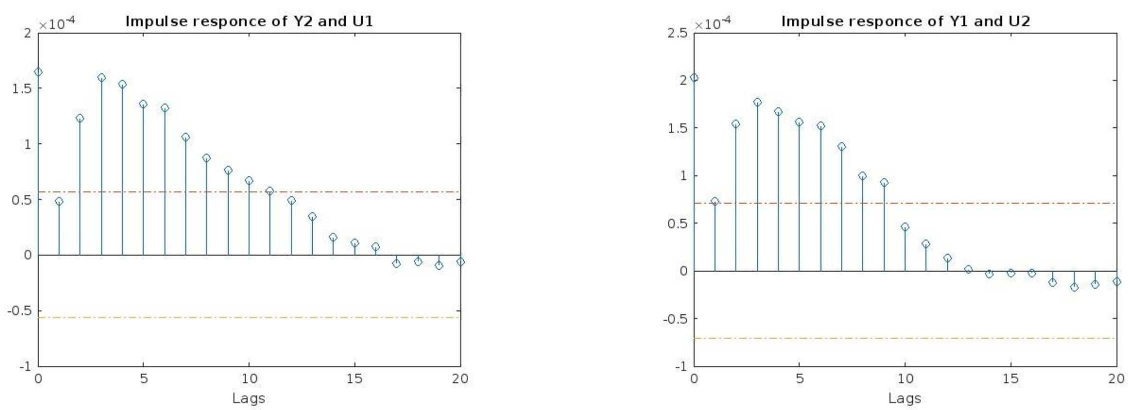


Figure 3 Impulse response of input (Y1& Y2) output(U1&U2)

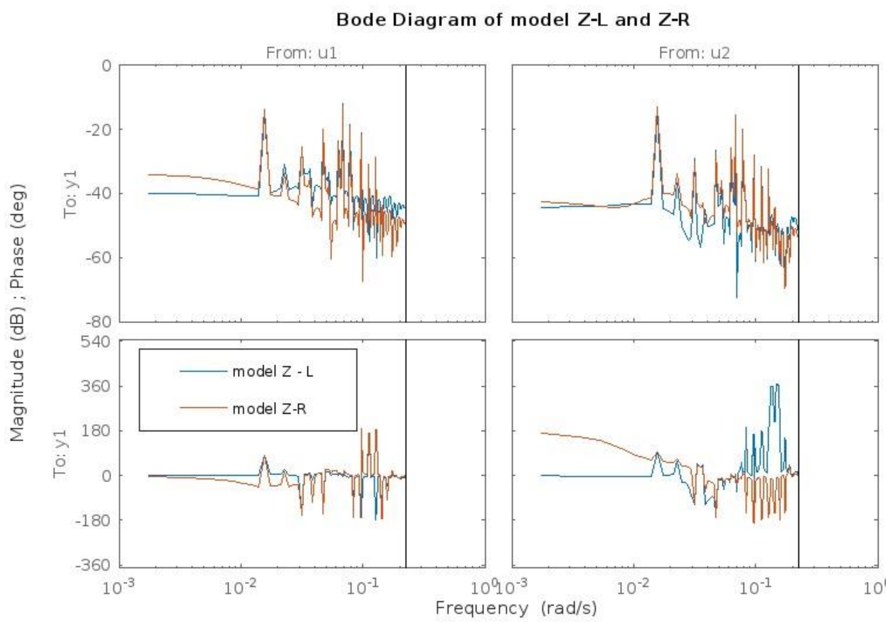
MATLAB function ***‘cra’*** is used. Figure 3 shows that impulse response of the Y1 and Y2 with

[1] Due to lack of space plots plots are shown in Appendix

respect to U1 and U2. To determine the time delay of the model we also used the MATLAB function ***'delayest'***. That gives time delay of the system model Z_L is [1 2] and for the model Z_R is [1 1]. Later same time delay has been used for model identification.

3.3 Frequency Response Analysis of the system

Frequency response analysis of the system will provide magnitudes and phase change with respect to the different frequencies. For that MATLAN has ***'SPA'*** function, which enable the frequency analysis of the discreate data models. Later we use MATLAB function ***'bode'*** to plot the same. From the Figure 4, it can be seen that magnitude of the both MISO model shown similar trends at higher frequencies whereas the phase of the system is significantly differing from each other.



3.4 Model Identification, validation and comparison

Figure 4 Frequency Response of MISO model Z_L and Z_R

In this part of report, parametric modeling is discussed. Different types of the data structures are used for the model fitting. To validate the model correctly, each data set (both Z_T and Z_R) is partitioned into two different portions: training dataset (ZT_1 and ZT_2) and validation dataset (ZV_1 and ZV_2). Dimensions of the dataset as per the Table 2

Table 2 Dimensions of Train and Validation dataset

Dataset	Dimensions	Dataset	Dimensions
Z_T	715 × 2	ZV_1	215 × 2
Z_R	715 × 2	ZT_2	500 × 2
ZT_1	500 × 2	ZV_2	215 × 2

Different model structures such as the ARX (Auto-regressive exogeneous), ARMAX (Auto-regressive Moving-average exogeneous), BJ (Box-Jenskin), OE (Output Error) are used. Furthermore, this result is compared with the Non-Linear Arx model, Subspace equation model and LSTM model.

3.4.1 MISO model Z_L (Left bottom tank Models):

To identify the best fitted model with quad tank level process identification, above mentioned four parametric model, non-linear and subspace model is used. All the model comparison are shown in Table 3. To visualize it more clearly, bode plots and validation (fitting) plots are shown in Figure 5 and Figure 6.

Table 3 Parameters of the MISO model Z_L (Left Bottom Tank)

Model	na	nb $[nb_1\ b_2]$	nc	nd	nf $[nf_1,nf_2]$	nk [u1 u2]	Auto corr.	Cross Corr. u1	u2	Model Fit	MSE (10^{-6})	AIC
ARX	1	[4 3]	-	-	-	[1 2]	Pass	Pass	Pass	90.99%	2.431	-12.9
ARMAX	2	[3 3]	1	-	-	[1 2]	Pass	Pass	Pass	91.81%	2.419	-12.9
OE	-	[2 2]	-	-	[1 1]	[1 2]	Fail	Pass	Pass	92.75%	7.418	-11.7
BJ	-	[4 2]	2	2	[2 2]	[1 2]	Pass	Pass	Pass	91.83%	2.242	-13
Non-Lin.	-	-	-	-	-	-	Pass	Pass	Pass	88.88%	2.431	-12.9
Subspace	-	-	-	-	-	-	Pass	Pass	Pass	82.31%	1.131	-12.9

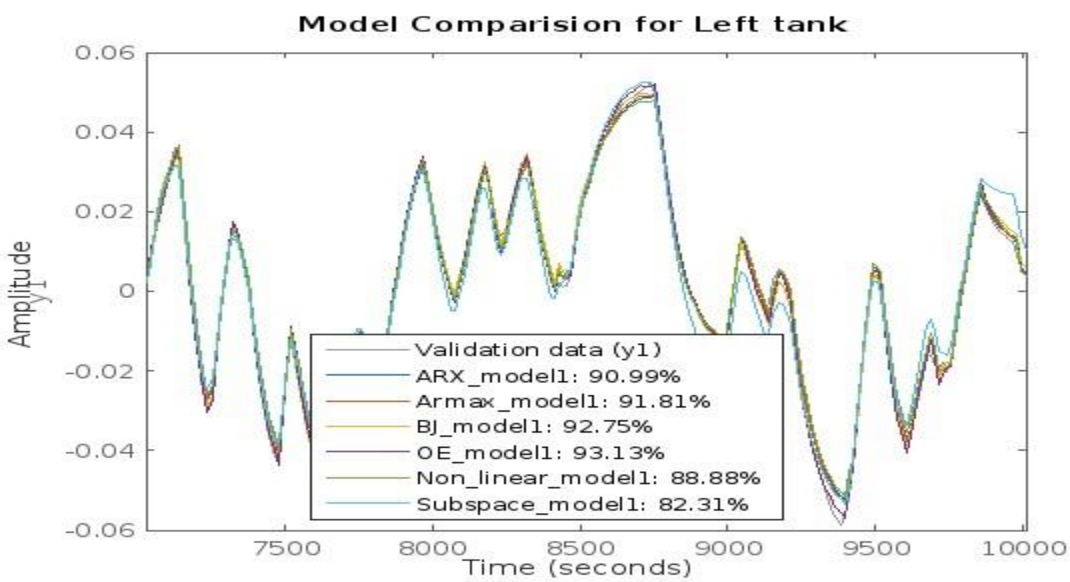


Figure 5 Left Bottom tank (Z_L) model comparison with ARX, ARMAX, BJ, OE, NL and SS models

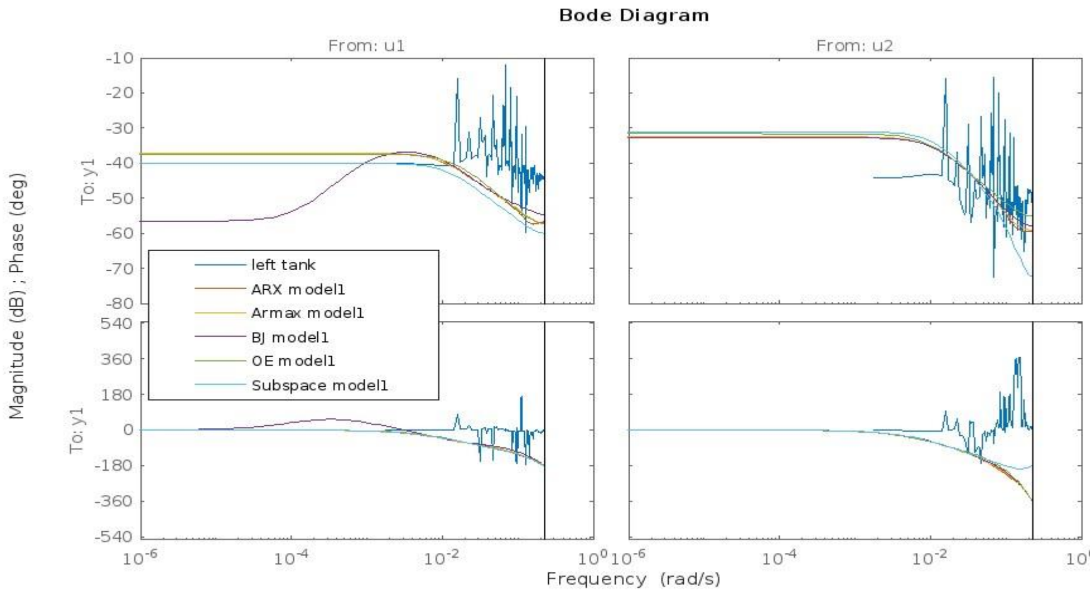


Figure 6 Bode plots comparison of the MISO model Z_L - Left (U1 and U2) with diffrent models

From the Table 3 selection of the model can be done with two distinguish approach:

1) Best Model: Best model can be selected based on the model fitting respect to validation dataset[2]. It is clear that BJ model has highest model fitting with 92.75%. Nonetheless, it also passes both the Autocorrelation as well as the cross-correlation test. Hence, BJ is best model for the MISO dataset of left bottom tank.
2) Optimal Model: In reality best is not always better as best is always comes with cost. Therefore, trade of between the model fitting and model complexity is important. For this report, parsimony principal for model complexity in terms of Akaike's Information Criterion (AIC) was used. MATLAB function ***'aic'*** is used to determine the AIC of the models. From the model complexity perspectives OE model is optimal model which has lowest AIC with (-11.7) as well as good model fitting. However, the key note is that OE model fails the auto-correlation test.

3.4.1.1 <u>Non-linear method:</u>

$Inputs: \ y1(t-1), y1(t-2), y1(t-3), u1(t-1), u1(t-2), u2(t-2), u2(t-3)$

$output: y(t) \ nonlinear$

[2] Model fitting percentage (%) is highly depend on the validation dataset. Hence, changing the validation dataset will affect the fitting percentage

3.4.1.2 Subspace identification Method:

Discrete time identified state-space model can be written as follows. To determine the subspace model, we have used the MATLAB function ***'n4sid'*** is used.

$$x(t+Ts) = A\,x(t) + B\,u(t) + K\,e(t)$$
$$y(t) = C\,x(t) + D\,u(t) + e(t)$$

$$A = \begin{bmatrix} 0.9063 & 0.09998 & -0.006999 & 0.003504 \\ -0.2326 & 0.4413 & 0.2022 & 0.1033 \\ 0.01249 & -0.1679 & 0.1337 & 0.2189 \\ 0.00007 & -0.04419 & -0.5162 & 0.8641 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.003727 & -0.001852 \\ 0.004068 & -0.0284 \\ 0.001189 & -0.007157 \\ 0.002445 & -0.007337 \end{bmatrix}$$

$$C = [-0.4981 \quad -0.01106 \quad -0.000682 \quad -0.00294]$$

$$D = [0 \quad 0]$$

$$K = \begin{bmatrix} -2.12 \\ -4.996 \\ -3.511 \\ -16.36 \end{bmatrix}$$

3.4.2 MISO model Z_R (Right bottom tank Models):

Same as the MISO model Z_L, modeling of MISO model Z_R also has been done with four linear, non-linear and subspace model is used to identify the correct the model. Table 4 summarized the parameters that has been used with the result.

Table 4 Parameters of the MISO model (Z_R) Right Bottom tank models

Model	na	nb [nb_1 b_2]	nc	nd	nf [nf_1nf_2]	nk [u1 u2]	Auto corr.	Cross Corr. u1	Cross Corr. u2	Model Fit	MSE (*10^{-6})	AIC
ARX	2	[3 6]	-	-	-	[1 1]	Pass	Pass	Pass	78.44%	3.006	-12.7
ARMAX	2	[3 3]	1	-	-	[1 1]	Pass	Pass	Pass	80.59%	3.118	-12.7
OE	-	[1 1]	-	-	[1 1]	[1 1]	Fail	Pass	Pass	78.01%	0.825	-9.55
BJ	-	[2 3]	1	1	[2 2]	[1 1]	Pass	Pass	Pass	83.65%	3.148	-12.6
Non-Lin.	-	-	-	-	-	-	Pass	Pass	Pass	80.58%	2.824	-12.9
Subspace	-	-	-	-	-	-	Pass	Pass	Pass	77.17%	2.949	-12.7

Like earlier, from Table 4 selection of the model can be done with two distinguish approach:

1) Best Model: Best model can be selected based on the model fitting respect to validation dataset. It is clear that BJ model has highest model fitting with 83.65%. Nonetheless, it

also passes both the Autocorrelation as well as the cross-correlation test. Hence, BJ is best model for the MISO dataset of left bottom tank.

2) Optimal Model: From the model complexity perspectives OE model is optimal model which has lowest AIC with -9.55 as well as good model fitting. However, the key note is that OE model fails the auto-correlation test.

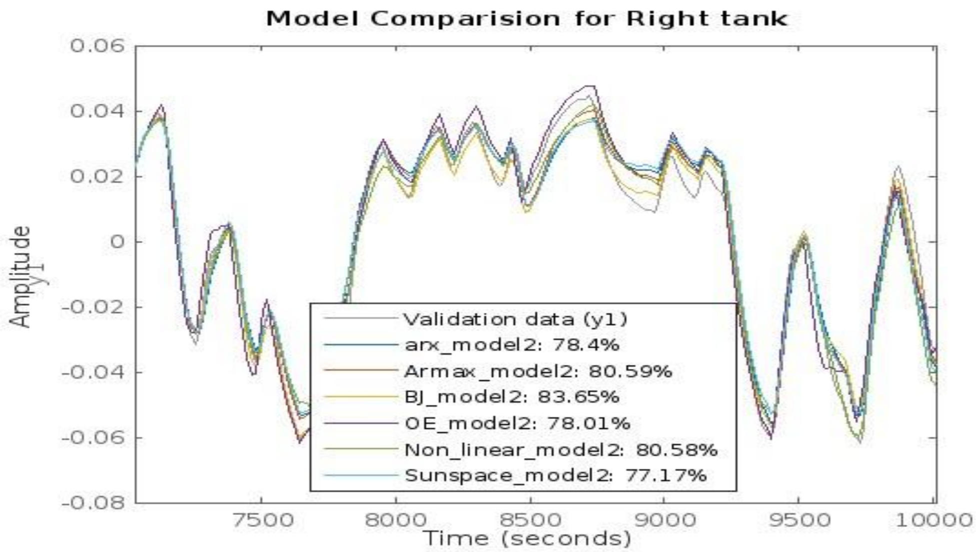


Figure 8 Right Bottom tank (Z_L) model comparison with ARX, ARMAX, BJ, OE, NL and SS models

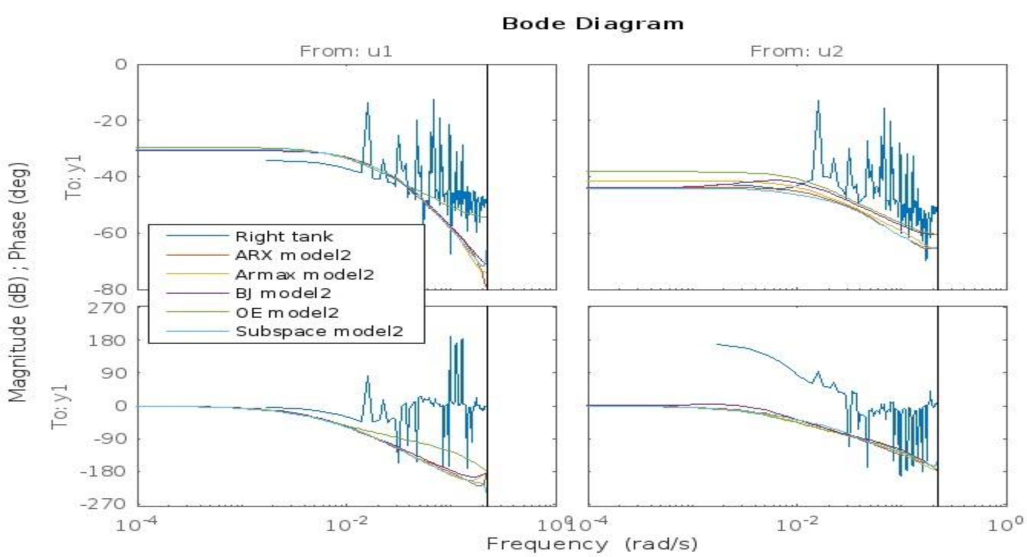


Figure 7 Bode plots comparison of the MISO model Z_R – right (U1 and U2) with different models

3.4.2.1 Non-linear method:

$Inputs: \; y1(t-1), y1(t-2), y1(t-3), u1(t-1), u1(t-2), u2(t-2), u2(t-3)$

$output: y(t) \; nonlinear$

3.4.2.2 <u>Subspace identification Method:</u>

Discrete time identified state-space model can be written as follows. Subspace model of right bottom tank is as follows

$$x(t+Ts) = A\,x(t) + B\,u(t) + K\,e(t)$$
$$y(t) = C\,x(t) + D\,u(t) + e(t)$$

$$A = \begin{bmatrix} 0.9239 & 0.1127 & -0.004036 & 0.00006 \\ -0.2531 & 0.5047 & 0.03309 & 0.4022 \\ 0.000007 & 0.1272 & -0.8463 & 0.3838 \\ 0.00007 & 0.045557 & -0.6172 & 0.5899 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.000090 & 0.00167 \\ 0.0.02331 & -0.00022 \\ 0.01139 & -0.007945 \\ 0.001292 & -0.001132 \end{bmatrix}$$

$$C = [0.6314 \quad 0.2662 \quad 0.0007636 \quad -0.004911]$$

$$D = [0 \quad 0]$$

$$K = \begin{bmatrix} 2.122 \\ 6.101 \\ -3.144 \\ -1.629 \end{bmatrix}$$

3.5 Enrichment

In this report, LSTM (Long-Short term Memory) method is used for the forecasting of the data and then compare with the actual data set. The prediction of MISO model Z_L is 82.12 % and the MISO model Z_R is 60.85%. The predicted result is not up to the mark for the MISO models as compare to parametric methods used earlier. This could be because of the fact that we have only

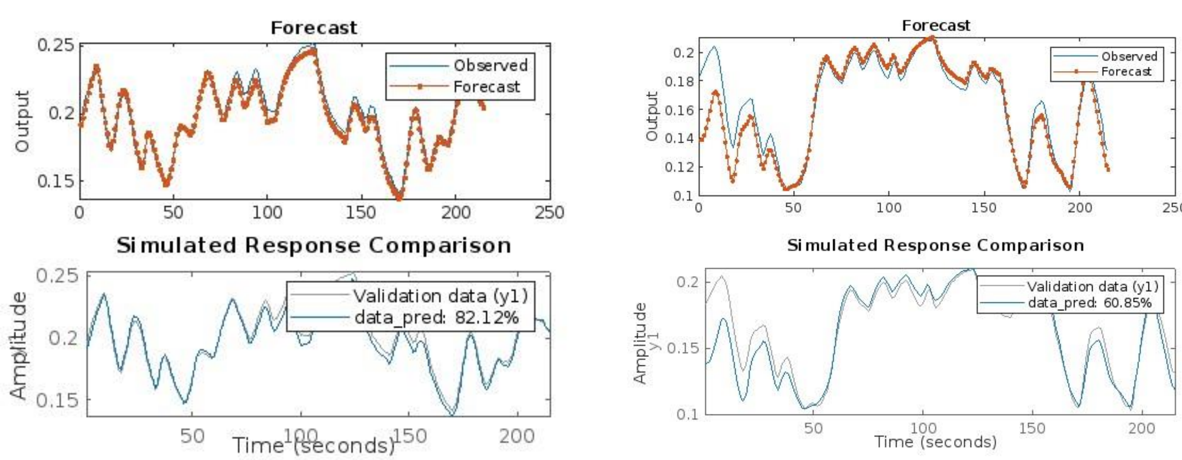


Figure 9 LSTM Forecast and Fit performance (A) Left bottom tank and (B) Right bottom tank

500 data point for the training and 215 data point for the validation. LSTM required the large amount of data pools to train and test. Upon close look of the error stem plot (Appendix 5.3.7) it has been observed that with increase in time steps the MSE is decreasing significantly. Hence, with smaller to medium scale dataset like Z_L and Z_R, LSTM might not be best model.

4 Conclusion

To identify the Quad-Tank MIMO model we have used data-driven approach: Process Identification. For that, FOPTD model is derived and design best input 'RBS' which gives persistence excitation of all the orders. Data is collected digitally and preprocessed. Outlier are identified and replaced with the predicted one. To get basic idea of the system non-parametric method such as the step response and frequency response was used to get parametric estimation. To identify the model, four parametric methods (ARX, ARMAX, OE & BJ model), non-linear method and subspace method was used. Two different approach is used to select the model: Best and Optimal. Best model is identified based on the Model fitting and optimal model based on AIC. Later, LSTM model is used to forecast the output based on the inputs. It provides the good result for the MISO model Z_L (82.12%), but poor result for the MISO model Z_R (60.85%). However, when LSTM result compare to the parametric result then BJ model shows significant better fitting having 92.75% and 83.65% for MISO Z_L and MISO Z_R respectively. This could be the fact that training dataset fed to the LSTM model is small and it perform better when dataset it larger. For the optimal model, AIC is included with model fitting. OE model found to be having least AIC as well as significant good model fitting with AIC of -11.7 & -9.55 respectively.

5 Appendix

5.1 Data Sheet: -

Steady State Values				
γ_L (left)	0.3 (30%)	Left Pump Flow Rate		11.9 L/min
γ_R (right)	0.3 (30%)	Right Pump Flow Rate		12.1 L/min
Steady State Heights for the 4 Tanks				
	Left		Right	
Top	0.1901		0.1901	
Bottom	0.1042		0.1543	
Step Test Information				
	+ Left Pump		+ Right Pump	
Left Tank	y_0	0.1901	y_0	0.1901
	y_∞	0.2217	y_∞	0.2412
	KM	0.0158	KM	0.0255
	$y\tau$	0.2101	$y\tau$	0.2224
	τ	111	τ	128
	θ	0	θ	0
	Left Pump		Right Pump	
Right Tank	y_0	0.1042	y_0	0.1543
	y_∞	0.1995	y_∞	0.1823
	KM	0.0477	KM	0.0140
	$y\tau$	0.1644	$y\tau$	0.1720
	τ	117	τ	82
	θ	8	θ	0
System Identification Information				
T_s	14		k	3
Step Size	1.5		Total Time	10000 s

5.2 Plot of Outlier

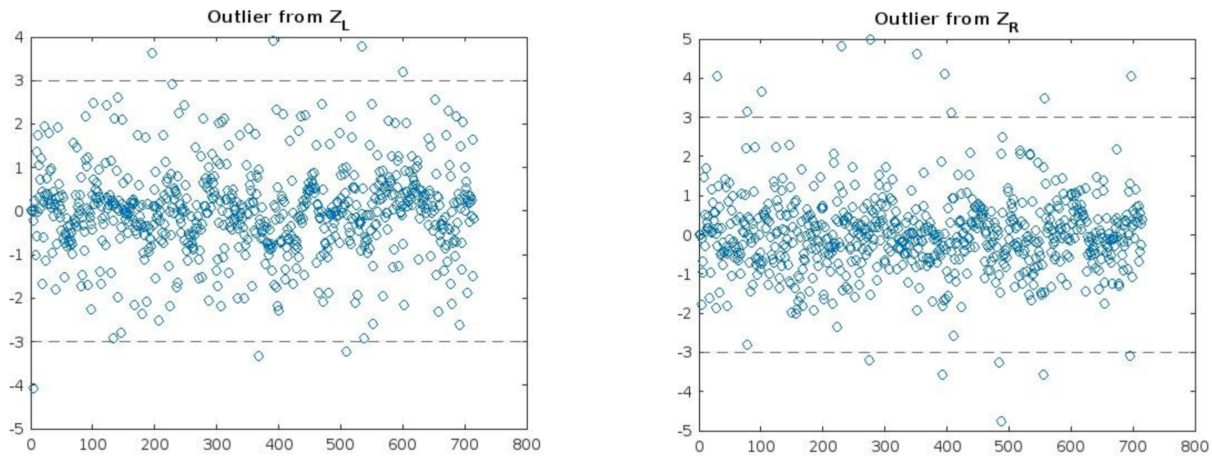
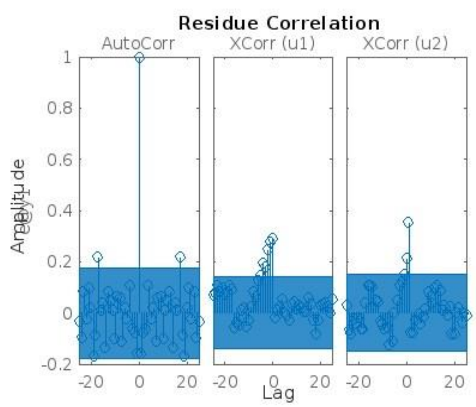
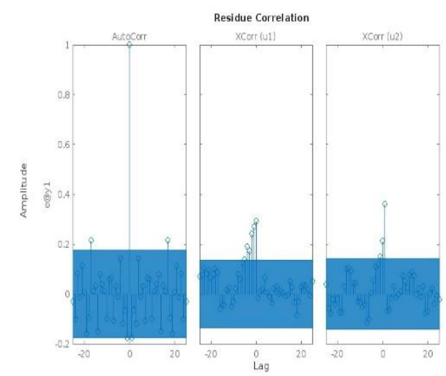


Figure 10 Outlier plot from MISO Z_L and Z_R models

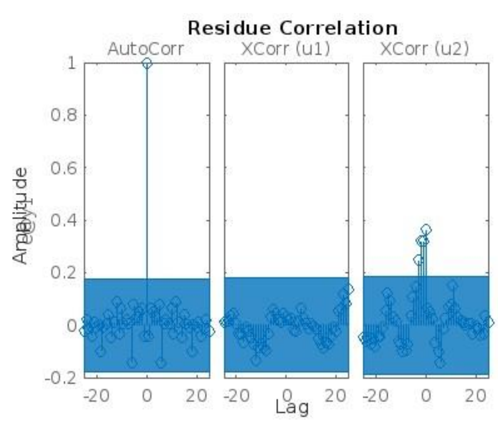
5.3 Residue plot of each model (Left Tank model on left side)

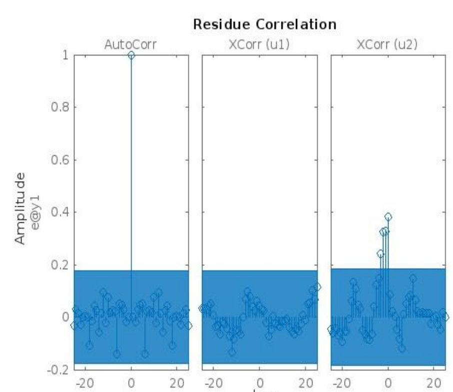
5.3.1 ARX Model Residue plot



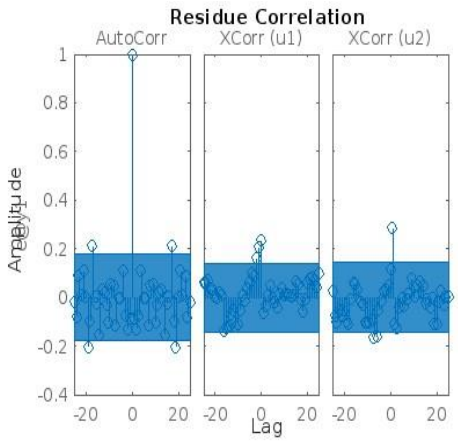


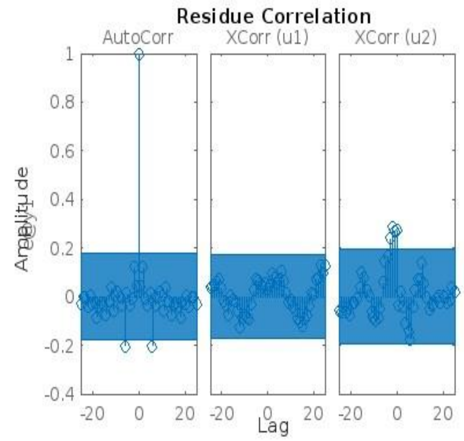
5.3.2 ARMAX model Residue plot



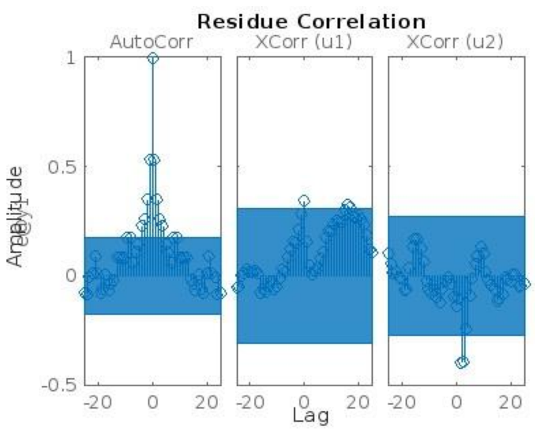


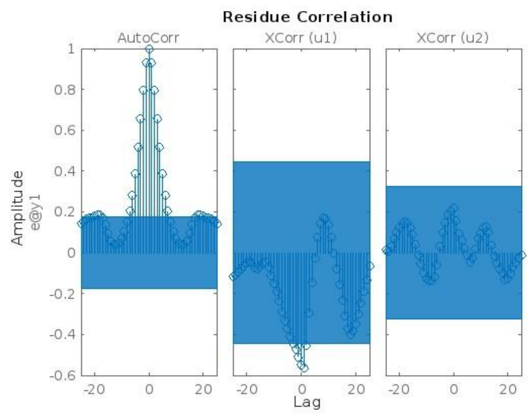
5.3.3 BJ model Residue plot



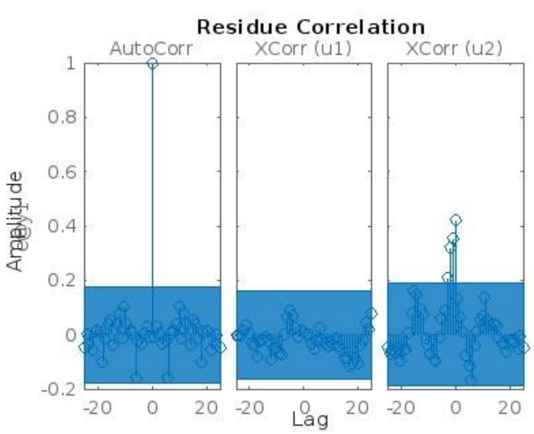


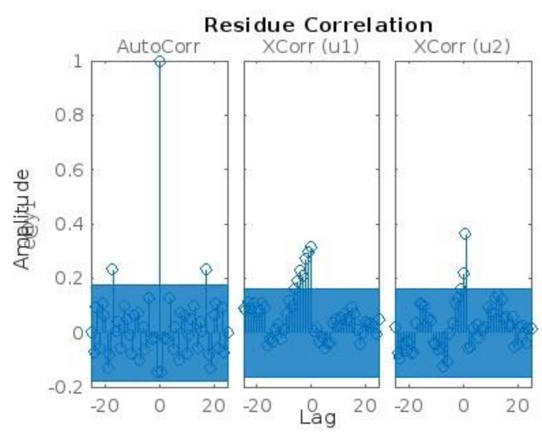
5.3.4 OE model Residue plot



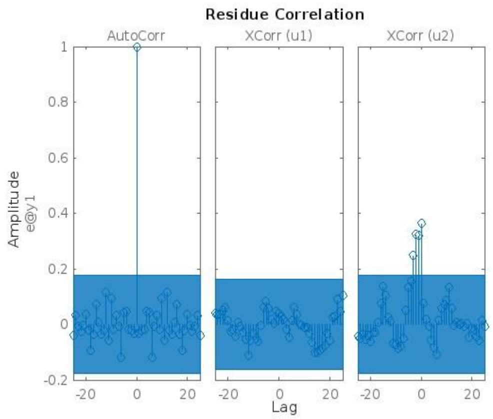


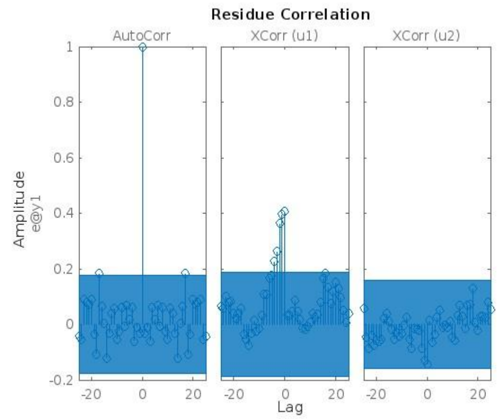
5.3.5 Non-linear Model Residue plot



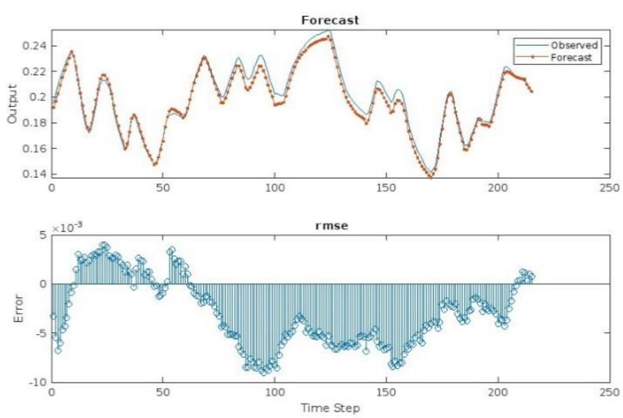


5.3.6 Subspace Model Residue plot





5.3.7 LSTM Residue plot



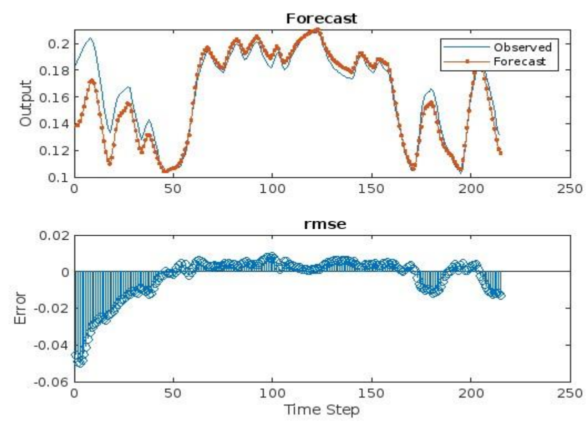


Figure 11 LSTM forecast and MSE stem plots

5.4 Transfer Function of BJ and OE model:

Left Bottom Tank Model

BJ Model

$$\boldsymbol{Y}\ (\boldsymbol{t}) = \frac{0.002888\ \mathrm{z}^{-1} - 0.003613\ \mathrm{z}^{-2} + 0.0005542\mathrm{z}^{-3} + 0.0001738\mathrm{z}^{-4}}{1 - 1.859\ \mathrm{z}^{-1} + 0.8606\ \mathrm{z}^{-2}}\boldsymbol{u1}(\boldsymbol{t}) + \frac{0.002647\mathrm{z}^{-2} - 0.000972\mathrm{z}^{-3}}{1 - 1.376\mathrm{z}^{-1} + 0.4493\ \mathrm{z}^{-2}}\ \boldsymbol{u2}(\boldsymbol{t}) + \frac{1 - 1.488\ \mathrm{z}^{-1} + 0.5047\ \mathrm{z}^{-2}}{1 - 1.942\ \mathrm{z}^{-1} + 0.9439\ \mathrm{z}^{-2}}\ \boldsymbol{e}(\boldsymbol{t})$$

OE Model

$$\boldsymbol{Y}(\boldsymbol{t}) = \frac{0.003157\ \mathrm{z}^{-1} - 0.000838\ \mathrm{z}^{-2}}{1 - 0.8346\ \mathrm{z}^{-1}}\ \boldsymbol{u1}(\boldsymbol{t}) + \frac{0.002767\ \mathrm{z}^{-2} + 0.0006091\mathrm{z}^{-3}}{1 - 0.868\ \mathrm{z}^{-1}}\ \boldsymbol{u2}(\boldsymbol{t}) + \boldsymbol{e}(\boldsymbol{t})$$

Right Bottom Tank Model

BJ Model

$$\boldsymbol{Y}\ (\boldsymbol{t}) = \frac{0.001426\ \mathrm{z}^{-1} - 0.0006327\ \mathrm{z}^{-2}}{1 - 1.396\ \mathrm{z}^{-1} + 0.4665\ \mathrm{z}^{-2}}\boldsymbol{u1}(\boldsymbol{t}) + \frac{0.001627\mathrm{z}^{-1} - 0.001639\mathrm{z}^{-2} - 0.0000687\mathrm{z}^{-3}}{1 - 1.799\mathrm{z}^{-1} + 0.8083\ \mathrm{z}^{-2}}\ \boldsymbol{u2}(\boldsymbol{t}) + \frac{1 + 0.3151\ \mathrm{z}^{-1}}{1 - 0.9739\ \mathrm{z}^{-1}}\ \boldsymbol{e}(\boldsymbol{t})$$

OE Model

$$\boldsymbol{Y}(\boldsymbol{t}) = \frac{0.003623\mathrm{z}^{-1}}{1 - 0.8903\ \mathrm{z}^{-1}}\ \boldsymbol{u1}(\boldsymbol{t}) + \frac{0.001766\ \mathrm{z}^{-1}}{1 - 0.8563\ \mathrm{z}^{-1}}\ \boldsymbol{u2}(\boldsymbol{t}) + \boldsymbol{e}(\boldsymbol{t})$$

5.5 MATLAB Code:

```
%Ts 14
z_L = iddata(y1,[u1 u2],Ts)
z_R = iddata(y2,[u1 u2],Ts)

nk1 = delayest(z_L,3,3,1,100)
nk2 = delayest(z_R,3,3,1,100)

% ARX approxiamtion for outlier detection
ARX1 = arx(z_L,[3 [3 3], nk1])
ARX2 = arx(z_R,[3 [3 3] nk2])
% Defining residual
R1 = predict(ARX1,z_L,1)
es_1 = (y1 - R1.y)
R2 = predict(ARX2,z_R,1)
es_2 = (y2 - R2.y)
N_1 = length(y1);
P = 9;
MSE_1 = es_1'*es_1/(N_1-P)
MSE_2 = es_2'*es_2/(N_1-P)
d1 = es_1./sqrt(MSE_1)
d2 = es_2./sqrt(MSE_2)

figure
plot(d1,'o')
yline(3,'--')
yline(-3,'--')
title('Outlier from Z_L')
figure(2)
plot(d2,'o')
yline(3,'--')
yline(-3,'--')
title('Outlier from Z_R')
outliers_y1 = [];
outliers_y2 = [];
for i = 1:length(y1)
    if d1(i) > 3 || d1(i) < -3
        outliers_y1 = [outliers_y1; i, y1(i)];
    end
    if d2(i) > 3 || d2(i) < -3
        outliers_y2 = [outliers_y2; i, y2(i)];
    end
end
disp('Outliers in y1:')
disp(outliers_y1) %7
disp('Outliers in y2:')
disp(outliers_y2) %16
z_L = iddata(y1,[u1 u2],Ts)
```

```
z_R = iddata(y2,[u1 u2],Ts)
delayest(z_L)
delayest(z_R)

ZT_1 = z_L(1:ceil(0.7*length(y1))); % Extract identification data 1st output
ZT_2 = z_R(1:ceil(0.7*length(y2))); % Extract identification data 2nd output
ZV_1 = z_L(ceil(0.7*length(y1))+1:end); % Valid identification data 1st output
ZV_2 = z_R(ceil(0.7*length(y2))+1:end); % Valid identification data 1st output

ZT_1 = dtrend(ZT_1); % Final testing data for y1
ZT_2 = dtrend(ZT_2); % Final testing data for y2
ZV_1 = dtrend(ZV_1); % Final validation data for y1
ZV_2 = dtrend(ZV_2); % Final validation data for y2

nk1_Tf = delayest(ZT_1,3,3,1,100)
nk2_Tf = delayest(ZT_2,3,3,1,100)
z_11 = iddata (y1, u1, Ts)
z_12 = iddata (y1, u2, Ts)
z_13 = iddata (y2, u1, Ts)
z_14 = iddata (y2, u2, Ts)

figure
cra (z_11)
title('Impulse responce of Y1 and U1')
figure
bode_11 = spa (z_L)
bode_12 = spa (z_R)

bode(bode_11,bode_12)
title ('Bode Diagram of model Z-L and Z-R ')
legend('model Z - L','model Z-R')

nk1_Tf = delayest(ZT_1,3,3,1,100)
nk2_Tf = delayest(ZT_2,3,3,1,100)

%% Model Identification
%%% ARX %%%
ARX_model1= arx(ZT_1,[1 [4 3] nk1_Tf]);
resid(ZV_1,ARX_model1)
compare(ZV_1,ARX_model1)
compare(ZT_1, ARX_model1)

arx_model2 = arx(ZT_2,[2 [3 4] nk2_Tf]); % 2 3 6
resid(ZV_2,arx_model2)
compare (ZT_2, arx_model2)
```

%%% ARMAX %%%

```
Armax_model1 = armax(ZT_1,[2 [3 3] 1 nk1_Tf]); %2 [2 2] 2
resid(ZV_1,Armax_model1)
compare(ZV_1,Armax_model1)
```

```
Armax_model2 = armax(ZT_2,[2 [3 3] 1 nk2_Tf]); % 2 [3 3] 2
resid(ZV_2,Armax_model2)
compare(ZV_2,Armax_model2)
```

%%% OE %%%

```
OE_model1 = oe(ZT_1,[[2 2] [1 1] nk1_Tf]);
resid(ZV_1,OE_model1)
compare(ZV_1,OE_model1)

OE_model2 = oe(ZT_2,[[1 1] [1 1] nk2_Tf]);
resid(ZV_2,OE_model2)
compare(ZV_2,OE_model2)
```

%%% BJ %%%

```
BJ_model1 = bj(ZT_1,[[4 2] 2 2 [2 2] nk1_Tf]); %[4 2] 2 2 [4 2]
resid(ZV_1,BJ_model1)
compare(ZV_1,BJ_model1)

BJ_model2 = bj(ZT_2,[[2 3] 1 1 [2 2] nk2_Tf]); % [1 1] 1 1 [1 1]
resid(ZV_2,BJ_model2)
compare(ZV_2,BJ_model2)
% Uinput = [u1,u2];
```

%%% non-linear ARX %%%

```
Non_linear_model1 = nlarx(ZT_1,[3 [2 2] nk1_Tf]);
resid(ZV_1,Non_linear_model1)
compare(ZV_1,Non_linear_model1)

figure
Non_linear_model2 = nlarx(ZT_2,[3 [2 2] nk2_Tf]);
resid(ZV_2,Non_linear_model2)
compare(ZV_2,Non_linear_model2)

Non_linear_model1

%%% subspace %%%
Subspace_model1=n4sid(ZT_1,4);
Sunspace_model2=n4sid(ZT_2,4);
resid(ZV_1,Subspace_model1)
resid(ZV_2,Sunspace_model2)
compare(ZV_1,Subspace_model1)
figure
compare(ZV_2,Sunspace_model2)

Subspace_model1
Sunspace_model2

figure
compare (ZV_1,ARX_model1,Armax_model1,BJ_model1,OE_model1,Non_linear_model1,Subspace_model1)
title ('Model Comparision for Left tank ')
```

```
figure
compare (ZV_2,arx_model2,Armax_model2,BJ_model2,OE_model2,Non_linear_model2,Sunspace_model2)
title ('Model Comparision for Right tank')

bode (bode_11, ARX_model1,Armax_model1,BJ_model1,OE_model1,Subspace_model1)
legend ('left tank ', 'ARX model1','Armax model1','BJ model1','OE model1','Subspace model1')

bode (bode_12, arx_model2,Armax_model2,BJ_model2,OE_model2,Sunspace_model2)
legend ('Right tank ', 'ARX model2','Armax model2','BJ model2','OE model2','Subspace model2')

AIC_arx1 = aic(ARX_model1)
Aic_Armax1 =  aic(Armax_model1)
Aic_BJ1  = aic(BJ_model1)
Aic_OE1 = aic(OE_model1)
Aic_nonlinear1 = aic(Non_linear_model1)
Aic_SubSpace1 = aic (Subspace_model1)

aic ( arx_model2,Armax_model2,BJ_model2,OE_model2,Non_linear_model1, Sunspace_model2)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%## LSTM ##%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
U = [u1, u2];
Y = y1;
numTimeStepsTrain = 500;
UTrain = U(1:numTimeStepsTrain, :);
YTrain = Y(1:numTimeStepsTrain);
UTest = U(numTimeStepsTrain+1:end, :);
YTest = Y(numTimeStepsTrain+1:end);
mu_U = mean(UTrain);
sigma_U = std(UTrain);
UTrainStandardized = (UTrain - mu_U) ./ sigma_U;
mu_Y = mean(YTrain);
sigma_Y = std(YTrain);
YTrainStandardized = (YTrain - mu_Y) / sigma_Y;
XTrain = UTrainStandardized;
numFeatures = 2;
numResponses = 1;
numHiddenUnits = 200;
layers = [
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits)
    fullyConnectedLayer(numResponses)
    regressionLayer
         ];
options = trainingOptions('adam', ...
    MaxEpochs=200, ...
    SequencePaddingDirection='left', ...
    Shuffle='every-epoch', ...
    Plots='training-progress', ...
    Verbose=0);
```

```
net = trainNetwork(XTrain', YTrainStandardized', layers, options);
UTestStandardized = (UTest - mu_U) ./ sigma_U;
XTest = UTestStandardized;
net = resetState(net);
YPredStandardized = predict(net, XTest');
YPred = sigma_Y * YPredStandardized + mu_Y;
rmse = sqrt(mean((YPred - YTest).^2));
figure;
plot(YTest);
hold on;
plot(YPred, '.-');
hold off;
legend(["Observed", "Predicted"]);
xlabel("Time Step");
ylabel("Output");
title(sprintf("Forecast with RMSE = %.2f", rmse));
UTestStandardized = (UTest - mu_U) ./ sigma_U;
XTest = UTestStandardized;
net = resetState(net);
YPredStandardized = predict(net, XTest');
YPred = sigma_Y * YPredStandardized + mu_Y;
rmse = sqrt(mean((YPred - YTest).^2));
figure;
subplot(2,1,1);
plot(YTest);
hold on;
plot(YPred, '.-');
hold off;
legend(["Observed", "Forecast"]);
ylabel("Output");
title("Forecast");
subplot(2,1,2);
stem(YPred' - YTest);
xlabel("Time Step");
ylabel("Error");
title(' rmse');
YPred_double = double(YPred);
data_pred = iddata(YPred_double', [], 1);
data_actual = iddata(YTest, [], 1);
compare(data_actual, data_pred);
```