APPENDIX 1.1

Result for i-butane

Neural network prediction of i-butane composition (MIMO model)

Table A1 Important variables for neural network prediction

Inputs		
Variable	Symbol	Description
MV2	mv2 (k)	Manipulated reboiler flow rate
	mv2 (k-1)	Lag MV2
MV3	mv3 (k)	Manipulated reflux flow rate
	mv3 (k-1)	Lag MV3
Temp 5	e (k)	Reboiler outlet temperature to column
	e (k-1)	Lag Temp 5
Component 2	o_top (k)	Top composition i-butane
	o_top (k-1)	Lag composition top
	o_bot (k)	Bottom composition i- butane
	o_bot (k-1)	Lag composition bottom
	o_top (k+1)	Future predictions i- butane top
	o_bot (k+1)	Future predictions i- butane bottom

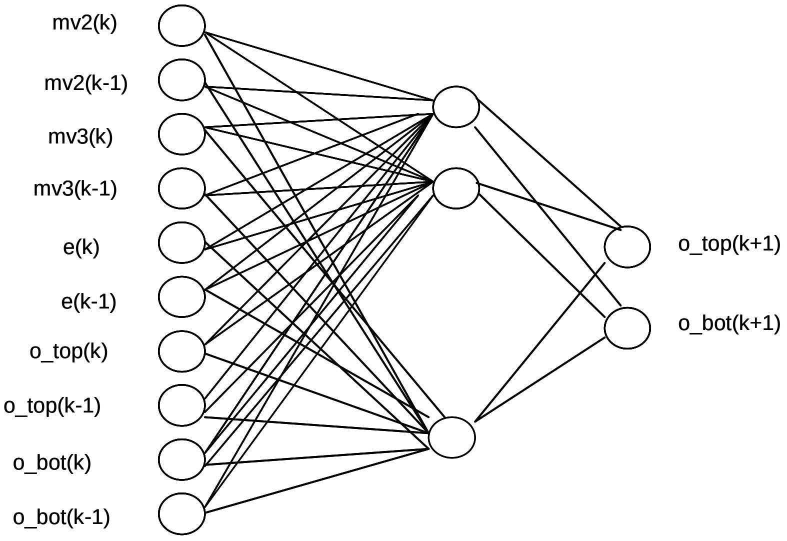


Figure A1 Neural network architecture for i-butane

Table A2. Neural network architecture

Parameters	Description	
Network	NARX series parallel network (newnarxsp)	
Category	With partitioning divided into 2	with partitioning divided into 3
Training function	TRAINLM	TRAINLM
Adaptation learning function	LEARNGDM	LEARNGDM
Performance function	MSE	MSE
Epochs	1000	1000
Goal	1e-6	1e-6
Number of layers	3	3
Layer 1: Number of Neuron Transfer function	10 PURELIN	10 PURELIN
Layer 2: Number of Neuron Transfer function	8 PURELIN	8 PURELIN
Layer 3: Number of Neuron Transfer function	2 PURELIN	2 PURELIN

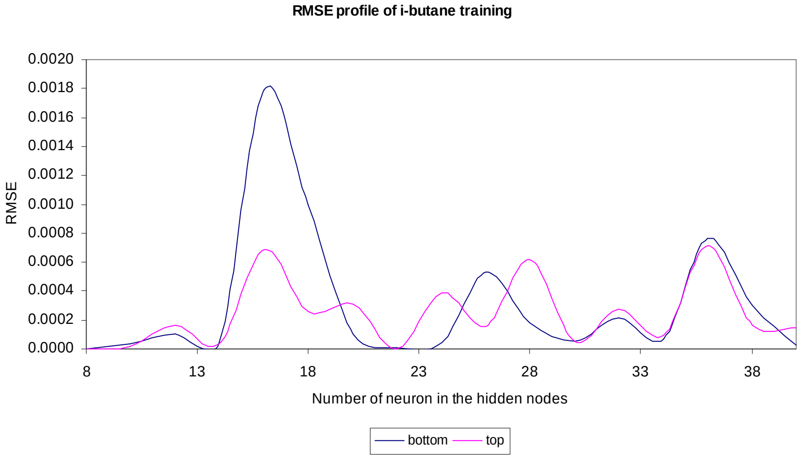


Figure A2 Profile of the RMSE of i-butane training

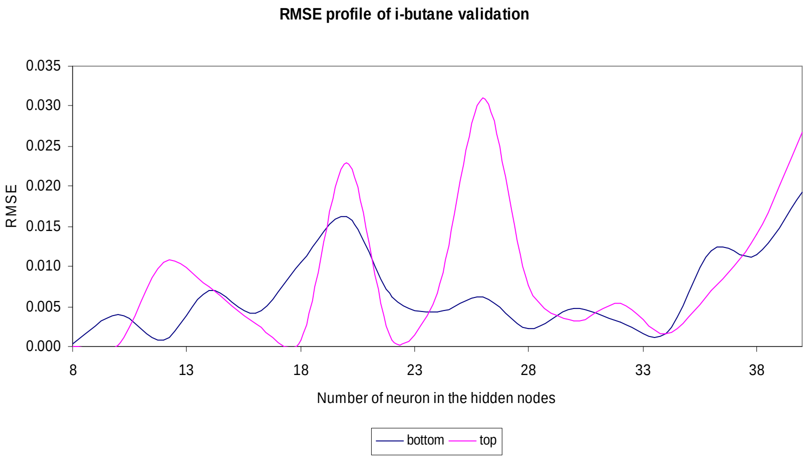


Figure A3 Profile of the RMSE of i-butane validation

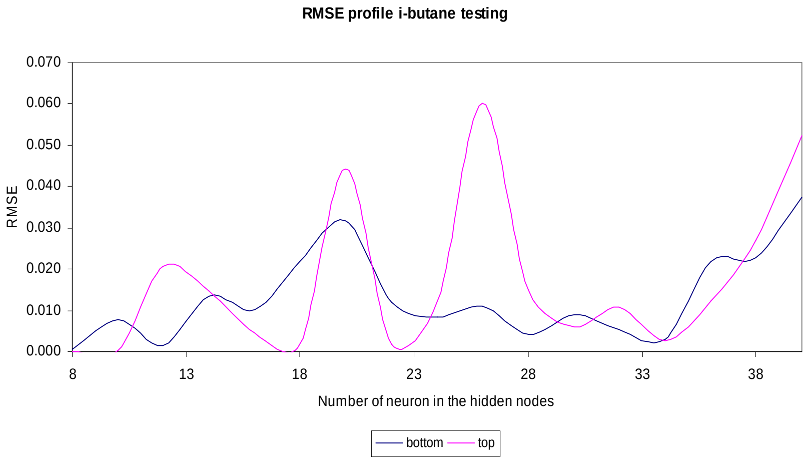


Figure A4 Profile of the RMSE of i-butane testing

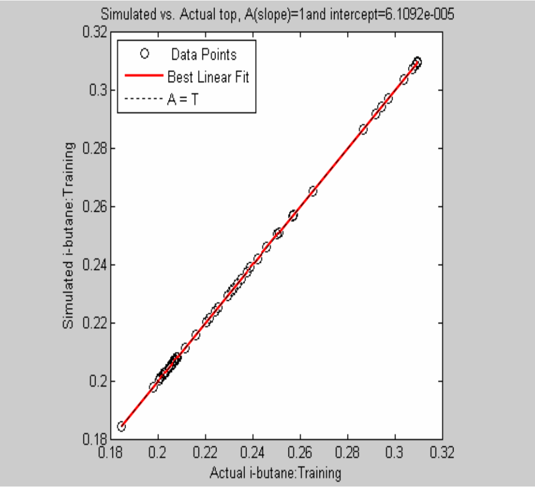


Figure A5 Actual and simulated i-butane top composition training

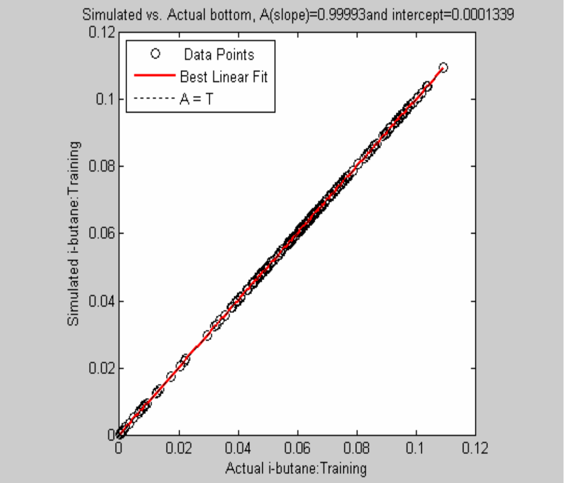


Figure A6 Actual and simulated i-butane bottom composition training

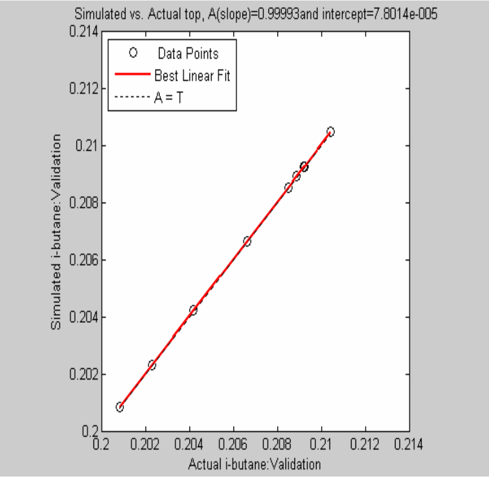


Figure A7 Actual and simulated i-butane top composition validation

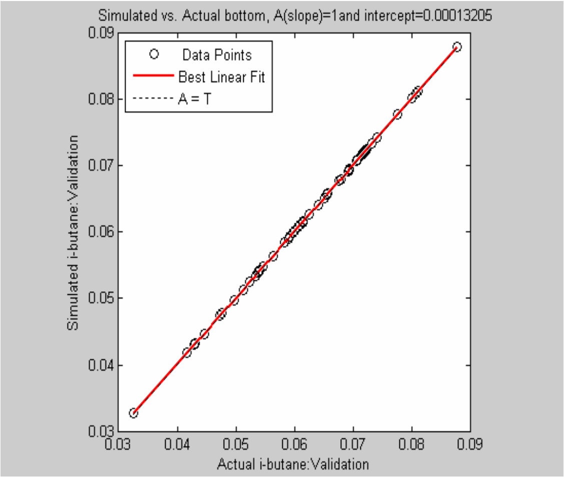
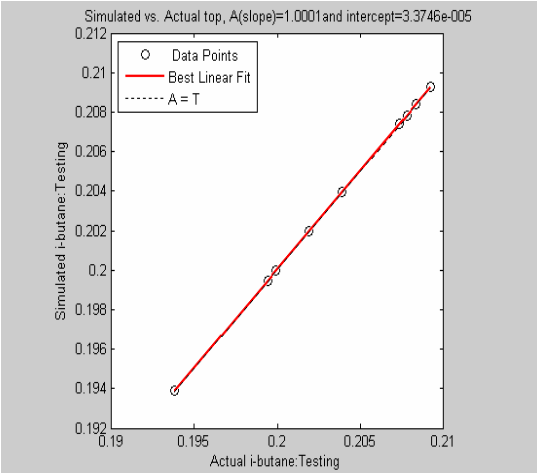


Figure 8A Actual and simulated i-butane bottom composition validation



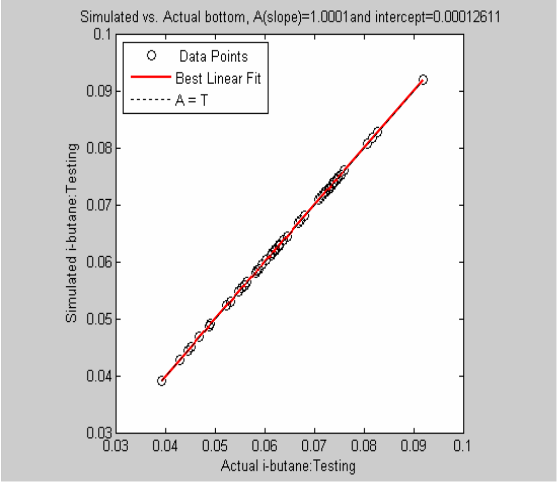
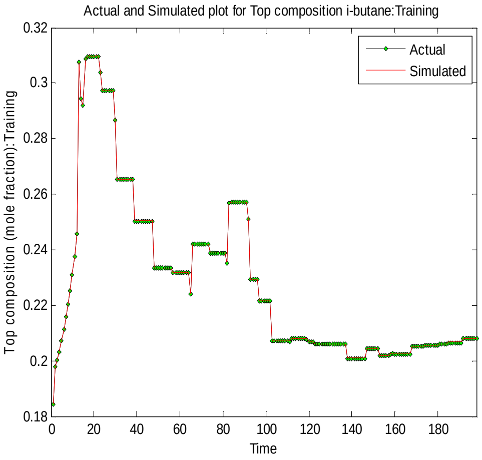


Figure A9 Actual and simulated i-butane top composition testing

Figure A10 Actual and simulated i-butane bottom composition testing



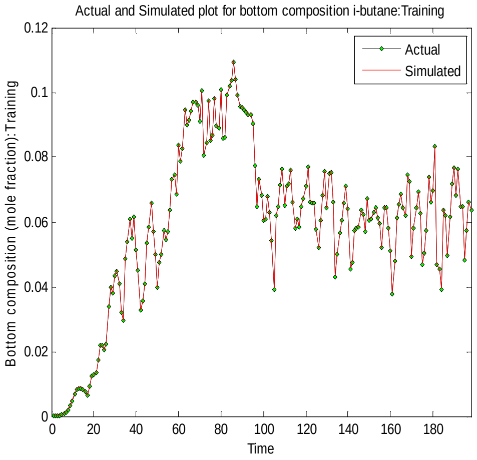
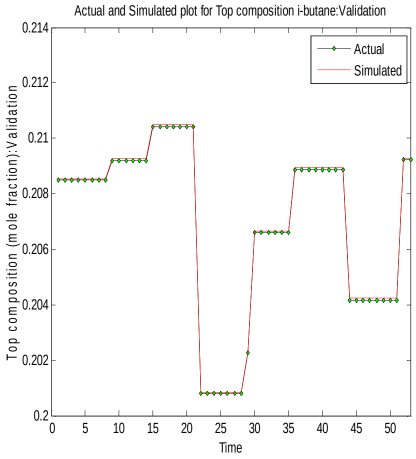


Figure A11 Actual and simulated i-butane top composition line plot training

Figure A12 Actual and simulated i-butane bottom composition line plot training



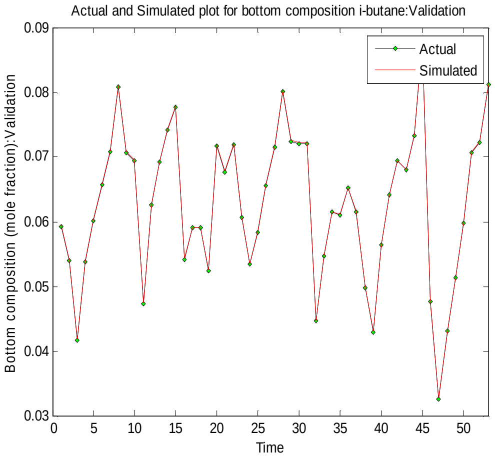
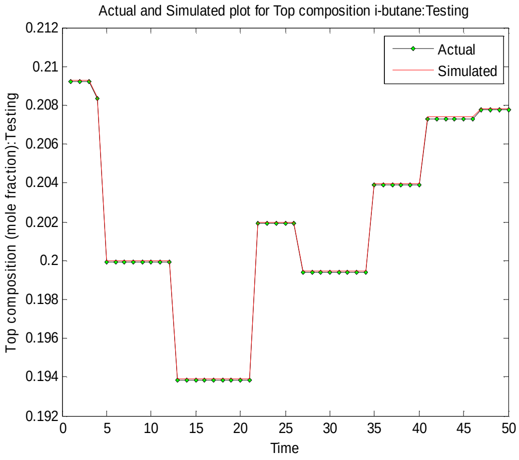


Figure A13 Actual and simulated i-butane top composition line plot validation

Figure A14 Actual and simulated i-butane bottom composition line plot validation



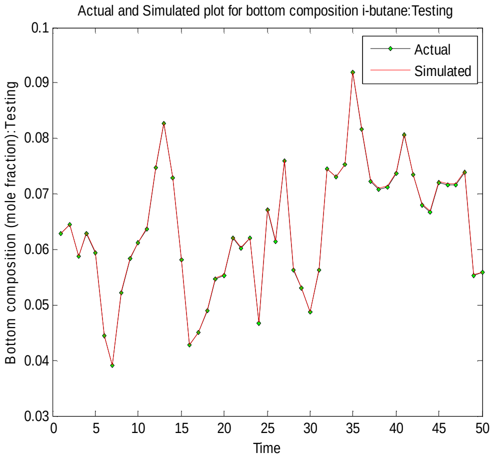


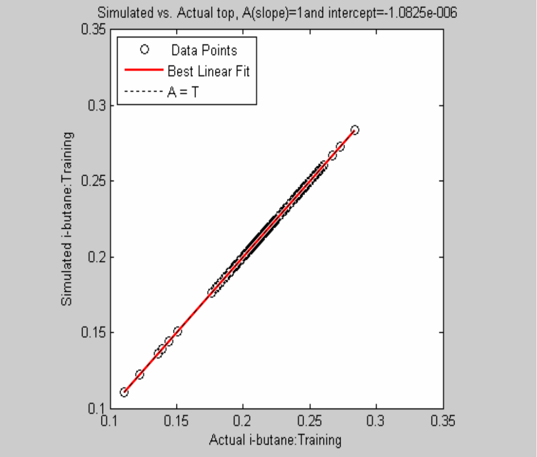
Figure A15 Actual and simulated i-butane top composition line plot testing

Figure A16 Actual and simulated i-butane bottom composition line plot testing

Table A3 Statistical analysis for i-butane with partition into 3

Parameter	Open loop	Extract
rmse_bottom_training	1.30E-04	0.00013
rmse_top_training	6.20E-05	6.2E-05
CDC_bottom_training	100.00	100
CDC_top_training	30.46	30.4569
R_bottom_training	1	1
R_top_training	1	1
AIC_bottom_training	1068.67	1068.7
AIC_top_training	1312.14	1312.1
BIC_bottom_training	1055.52	1055.5
BIC_top_training	1298.99	1299
MAPE_bottom_training	2.01	2.00831
MAPE_top_training	0.03	0.02761
Cp_bottom_training	1	1
Cp_top_training	1	1
rmse_bottom_validation	1.32E-04	9.9E-05
rmse_top_validation	6.32E-05	4.8E-05
CDC_bottom_validation	100.00	98.0769
CDC_top_validation	17.31	17.3077
R_bottom_validation	1	1
R_top_validation	1	1
AIC_bottom_validation	319.58	319.58
AIC_top_validation	490.14	490.18
BIC_bottom_validation	311.69	311.7
BIC_top_validation	482.26	482.3
MAPE_bottom_validation	0.22	0.16285
MAPE_top_validation	0.03	0.02293
Cp_bottom_validation	1	1
Cp_top_validation	1	1
rmse_bottom_testing	1.30E-04	9.8E-05
rmse_top_testing	6.22E-05	4.7E-05
CDC_bottom_testing	100.00	100
CDC_top_testing	16.33	16.3265
R_bottom_testing	1	1
R_top_testing	1	1
AIC_bottom_testing	300.76	300.76
AIC_top_testing	405.88	405.91
BIC_bottom_testing	293.11	293.12
BIC_top_testing	398.23	398.26
MAPE_bottom_testing	0.21	0.15727
MAPE_top_testing	0.03	0.02324
Cp_bottom_testing	1	1
Cp_top_testing	1	1

Validate based on close loop data for n-butane



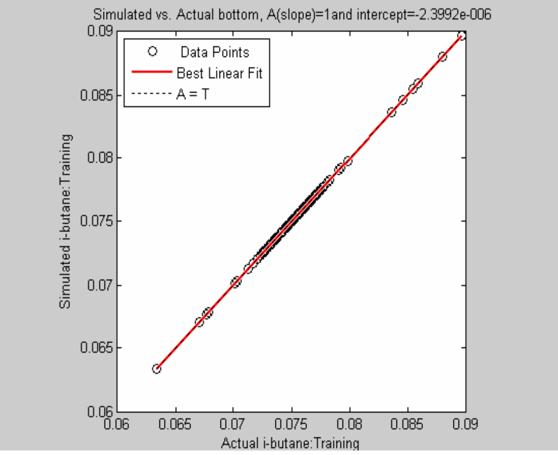
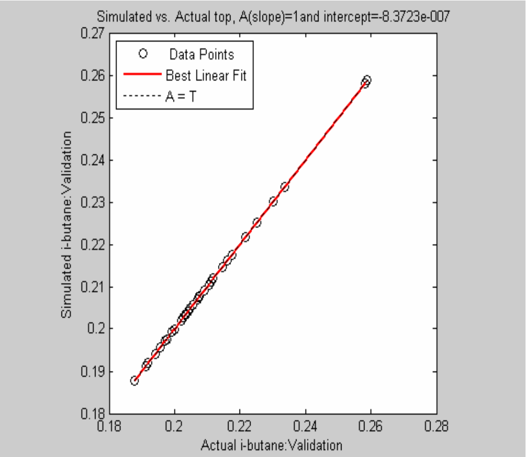


Figure A17 Actual and simulated i-butane top online composition training

Figure A18 Actual and simulated i-butane bottom online composition training



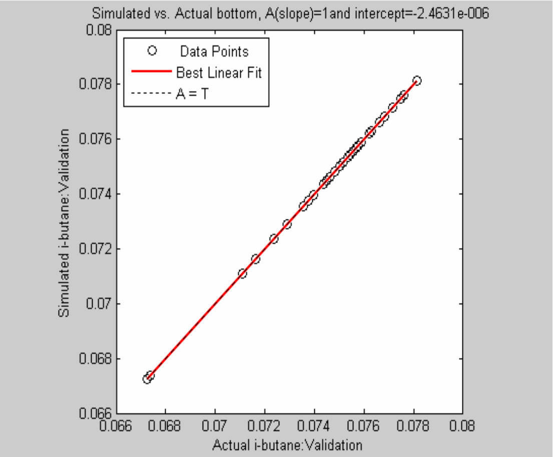
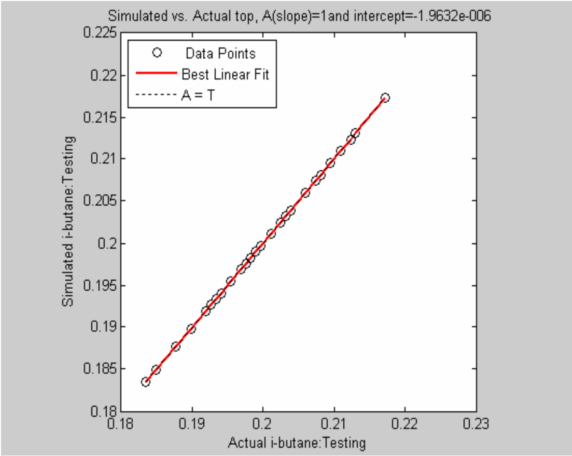


Figure A19 Actual and simulated i-butane top online composition validation

Figure A20 Actual and simulated i-butane bottom online composition validation



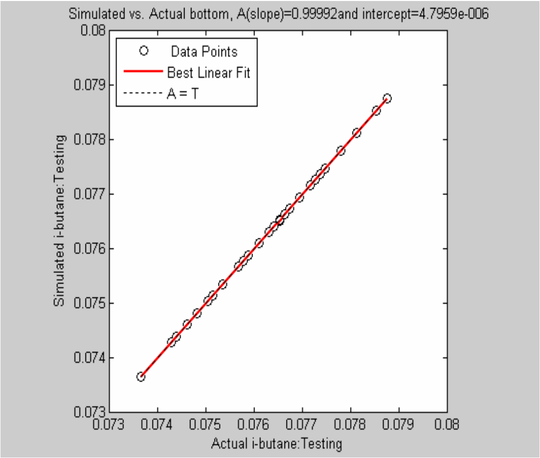
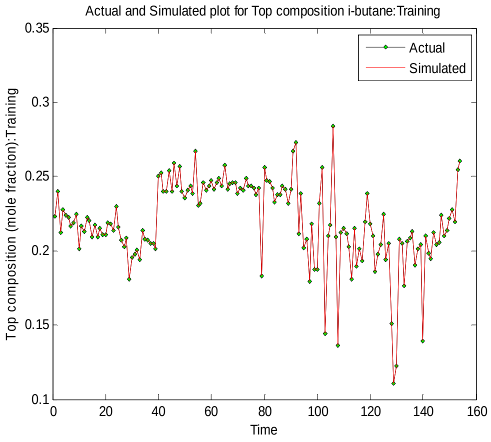


Figure A21 Actual and simulated i-butane top online composition testing

Figure A22 Actual and simulated i-butane bottom online composition testing



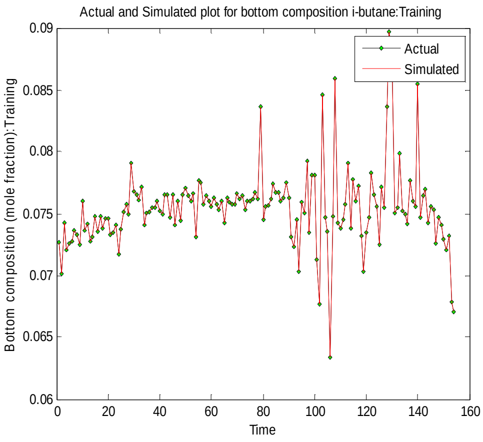
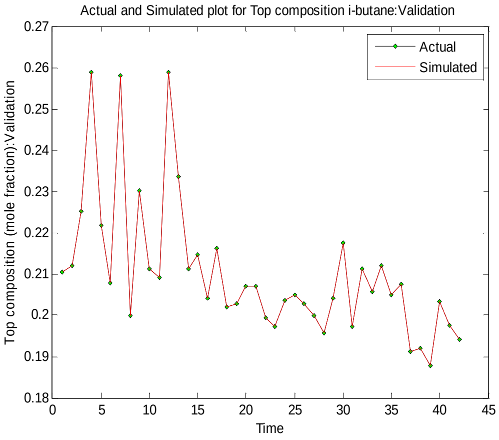


Figure A23 Actual and simulated i-butane top online composition line plot training

Figure A24 Actual and simulated i-butane bottom online composition line plot training



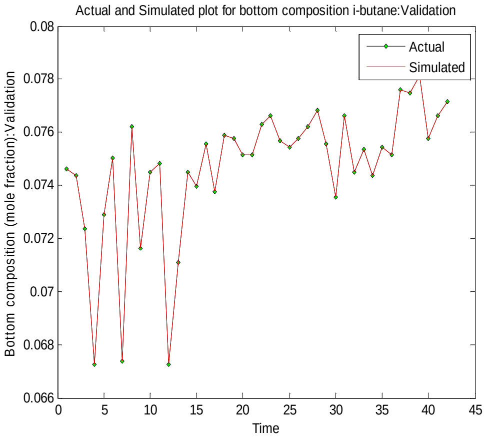
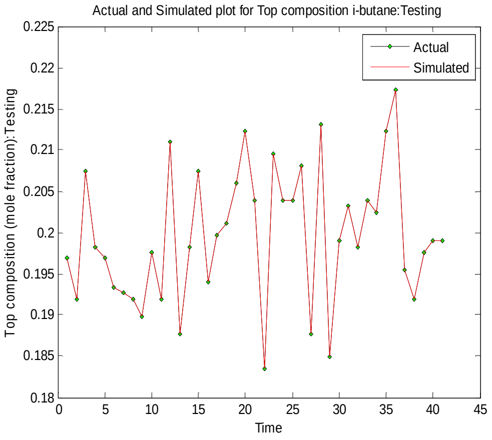


Figure A25 Actual and simulated i-butane top online composition line plot validation

Figure A26 Actual and simulated i-butane bottom online composition line plot validation



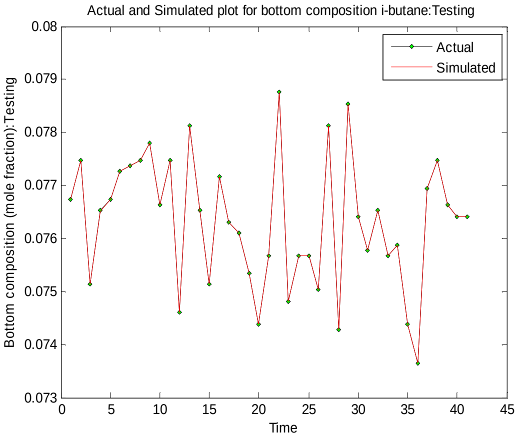


Figure A27 Actual and simulated i-butane top online composition line plot testing

Figure A28 Actual and simulated i-butane bottom online composition line plot testing

Table A4 Statistical analysis for online i-butane with partition into 3

Parameter	Online
rmse_bottom_training	2.16E-06
rmse_top_training	1.05E-06
CDC_bottom_training	95.42
CDC_top_training	95.42
R_bottom_training	0.99
R_top_training	1
AIC_bottom_training	1374.29
AIC_top_training	689.69
BIC_bottom_training	1362.14
BIC_top_training	677.54
MAPE_bottom_training	-0.0028
MAPE_top_training	-0.00048
Cp_bottom_training	0.99
Cp_top_training	1
rmse_bottom_validation	1.97E-06
rmse_top_validation	9.42E-07
CDC_bottom_validation	97.56
CDC_top_validation	97.56
R_bottom_validation	0.99
R_top_validation	1
AIC_bottom_validation	387.96
AIC_top_validation	230.22
BIC_bottom_validation	381.00
BIC_top_validation	223.27
MAPE_bottom_validation	-0.0026
MAPE_top_validation	-0.00044
Cp_bottom_validation	0.99
Cp_top_validation	1
rmse_bottom_testing	1.59E-06
rmse_top_testing	7.50E-07
CDC_bottom_testing	95
CDC_top_testing	95
R_bottom_testing	1
R_top_testing	0.99
AIC_bottom_testing	426.30
AIC_top_testing	273.56
BIC_bottom_testing	419.44
BIC_top_testing	266.71
MAPE_bottom_testing	-0.0019
MAPE_top_testing	-0.00033
Cp_bottom_testing	1
Cp_top_testing	0.99

Neural network i-butane equation based model

p is the inputs to the neural network and for this case study is given by the vector

$$\left[mv2(k)\ mv2(k-1)\ mv3(k)\ mv3(k-1)\ e(k)\ e(k-1)\ o_{top}(k)\ o_{top}(k-1)\ o_{bot}(k)\ o_{bot}(k-1)\right]^{T} \qquad (1)$$

Training, validation and testing

$$y = \begin{bmatrix} y1 \\ y2 \end{bmatrix} \begin{bmatrix} 0.36\ 0.23\ 0.41 - 0.28\ 0.78\ 0.004 - 0.48\ 0.70\ 0.18 - 0.76 \\ -0.42 - 0.24 - 0.31\ 0.36 - 0.21 - 0.11\ 0.06 - 0.19 - 0.10\ 0.45 \end{bmatrix} p + \begin{bmatrix} -1.24 \\ 0.69 \end{bmatrix} \qquad (2)$$

The equation of PLS for prediction of i-butane at top composition is given as

$$Y_{1,PLS} = 0.21996 + \begin{bmatrix} mv2(k) \\ mv2(k-1) \\ mv3(k) \\ mv3(k-1) \\ e(k) \\ e(k-1) \\ o\_top\ (k) \\ o\_top(k-1) \\ o\_bot(k) \\ o\_bot(k-1) \end{bmatrix} \begin{bmatrix} -0.00055 \\ -0.00074 \\ -0.00029 \\ -0.00061 \\ 4.76e-05 \\ 5.81e-05 \\ 0.98 \\ 0.93 \\ -0.018 \\ -0.10 \end{bmatrix} + \begin{bmatrix} -0.005 \\ 0.007 \\ 0.0031 \\ -0.001 \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ -0.00032 \end{bmatrix} \qquad (3)$$

The equation of PLS for predictions of i-butane at the bottom prediction is given as;

$$Y_{2,PLS} = 6.03e - 05 + \begin{bmatrix} mv2(k) \\ mv2(k-1) \\ mv3(k) \\ mv3(k-1) \\ e(k) \\ e(k-1) \\ o\_top(k) \\ o\_top(k-1) \\ o\_bot(k) \\ o\_bot(k-1) \end{bmatrix} \begin{bmatrix} 0.001 \\ -0.001 \\ -0.0016 \\ -0.0011 \\ 6e-055 \\ .4e-05 \\ 0.0136 \\ -0.013 \\ -1.55 \\ 0.22 \end{bmatrix} + \begin{bmatrix} -0.005 \\ -0.005 \\ -0.005 \\ -0.005 \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ -0.001 \end{bmatrix} \tag{4}$$

For the regression model, the equations for the top and bottom prediction i-butane are described below;

$$\begin{aligned} Y_{1,RA}\_top &= 0.0011\ mv2(k) - 0.0012\mathrm{mv2(k-1)} + 0.00079\mathrm{mv3(k)} \\ &- 0.001\mathrm{mv3(k-1)} + 4.17\mathrm{e}-05\,\mathrm{e(k)} - 9.073\mathrm{e}-06\ \mathrm{e(k-1)} + 0.96\,\mathrm{o\_top(k)} \\ &+ 0.08\,\mathrm{o\_top(k-1)} - 0.014\mathrm{o\_bot(k)} - 0.02\mathrm{o\_bot(k-1)} - 0.024 \end{aligned} \tag{5}$$

$$\begin{aligned} Y_{2,RA} &= 0.003\ mv2(k) - 0.003\ \mathrm{mv2\,(k-1)} - 0.001\ \mathrm{mv3\,(k)} + \\ &0.0019\mathrm{mv3\ (k-1)} + 0.0002\ \mathrm{e\,(k)} - 0.0004\ \mathrm{e\,(k-1)} + 0.04\ \mathrm{o\_top\ (k)} \\ &- 0.039\ \ \mathrm{o\_top\ (k-1)} + 0.87\mathrm{o\_bot\ \ (k)} - 0.02\mathrm{o\_bot\ \ (k-1)} - 0.013 \end{aligned} \tag{6}$$

Analysis of variance (ANOVA) i-butane

Table A5. ANOVA of the i-butane top composition

Regression Statistics	
Multiple R	0.983
R Square	1.00
Adjusted R Square	0.965
Standard Error	0.005051
Observations	301

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	10	0.215	0.0215	844	6.66E-208
Residual	290	0.0073	2.55E-05		
Total	300	0.222			

Table A6 ANOVA of i-butane bottom composition

Regression Statistics	
Multiple R	0.918
R Square	1.00
Adjusted R Square	0.937
Standard Error	0.0088
Observations	301

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	10	0.120	0.0120	155	2.50E-110
Residual	290	0.0224	7.75E-05		
Total	300	0.143			

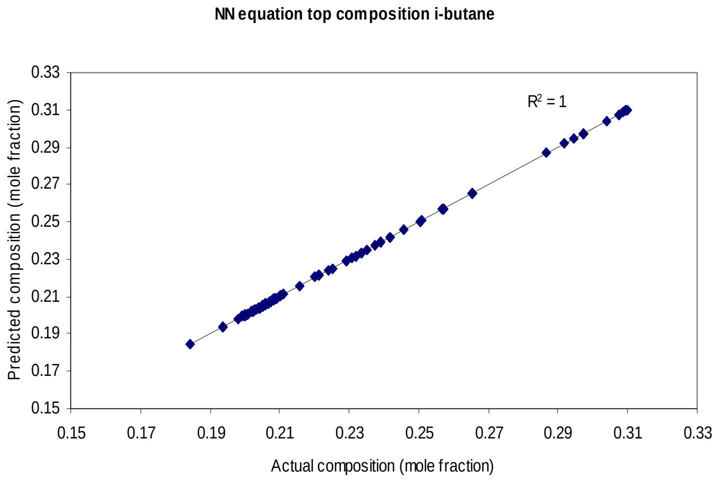


Figure. A29 Prediction versus actual value neural network equation top composition i-butane

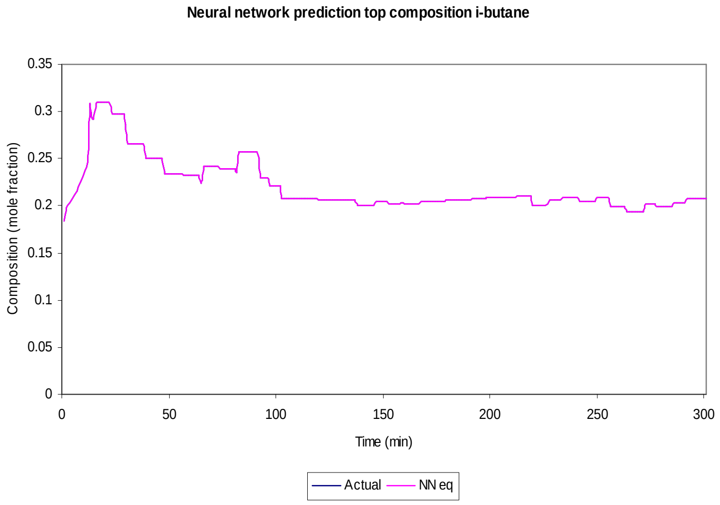


Figure. A30 Prediction and actual value for top composition i-butane line plot

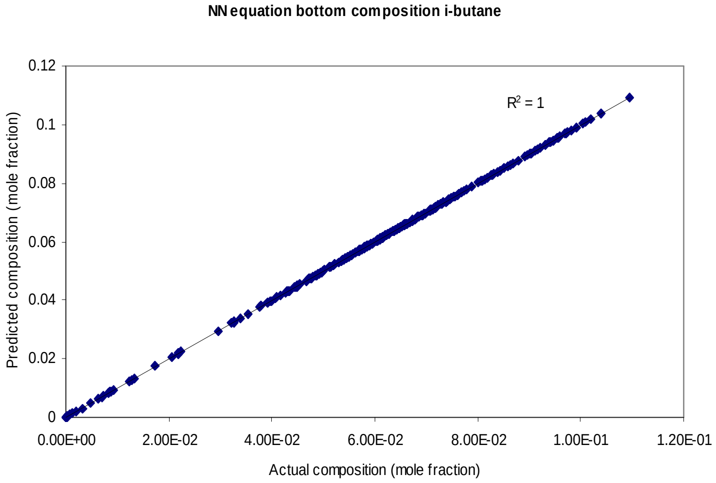


Figure. A31 Prediction versus actual value neural network equation bottom composition i-butane

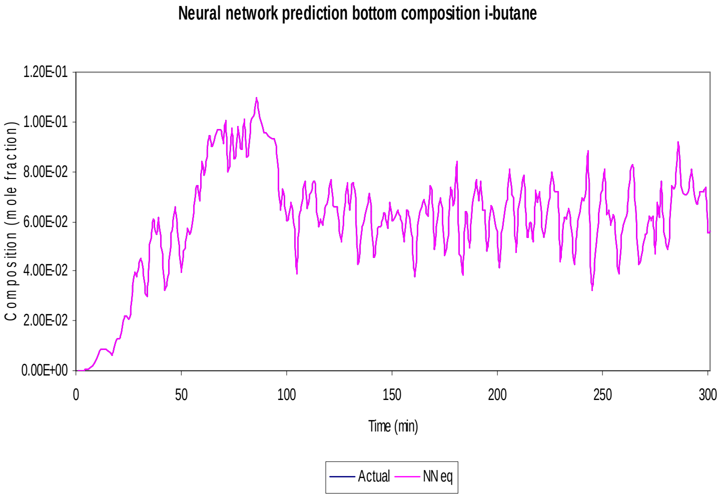


Figure. A32. Prediction and actual value for bottom composition i-butane line plot

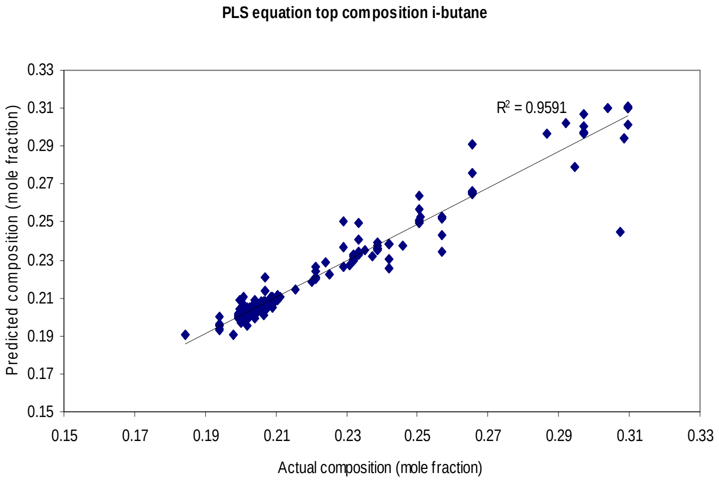


Figure. A33. Prediction versus actual value PLS equation top composition i-butane

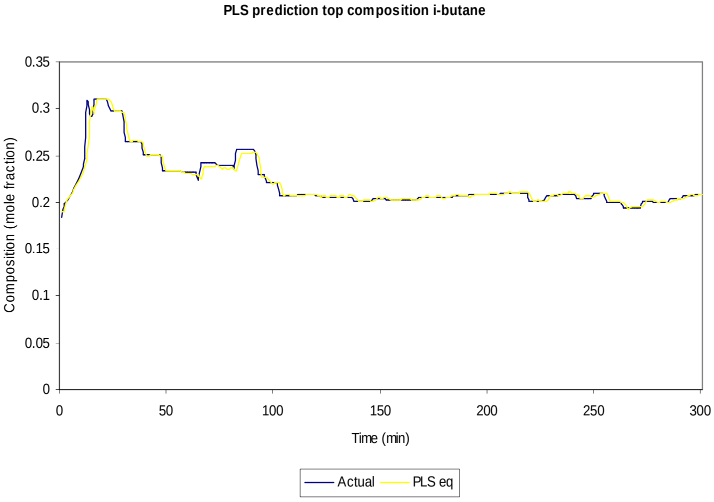


Figure. A34 Prediction and actual value for top composition i-butane line plot

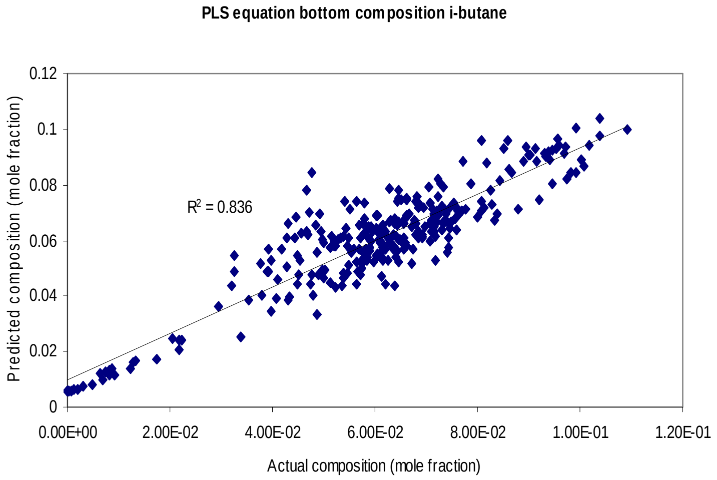


Figure. A35 Prediction versus actual value PLS equation bottom position i-butane

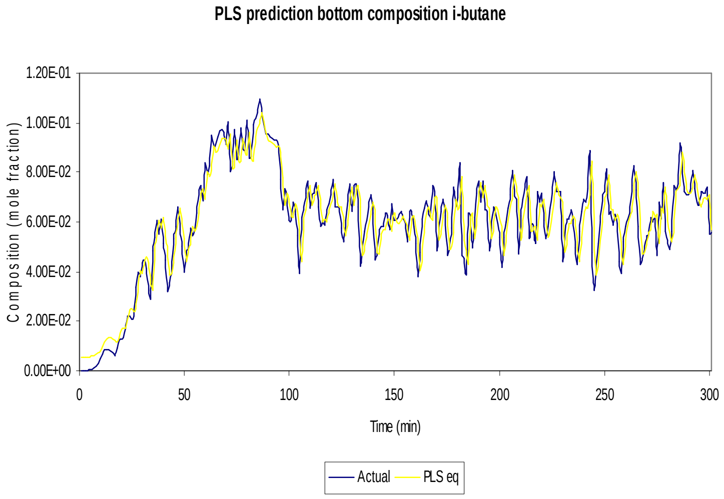


Figure. A36. Prediction and actual value for bottom composition i-butane line plot top composition.

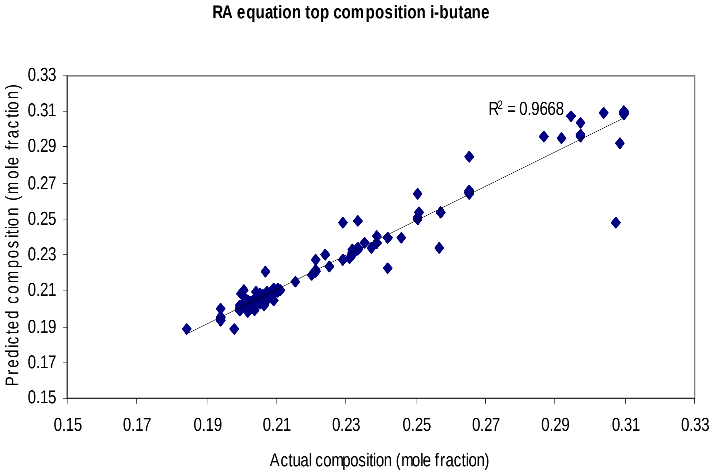


Figure. A37. Prediction versus actual value RA equation top composition i-butane

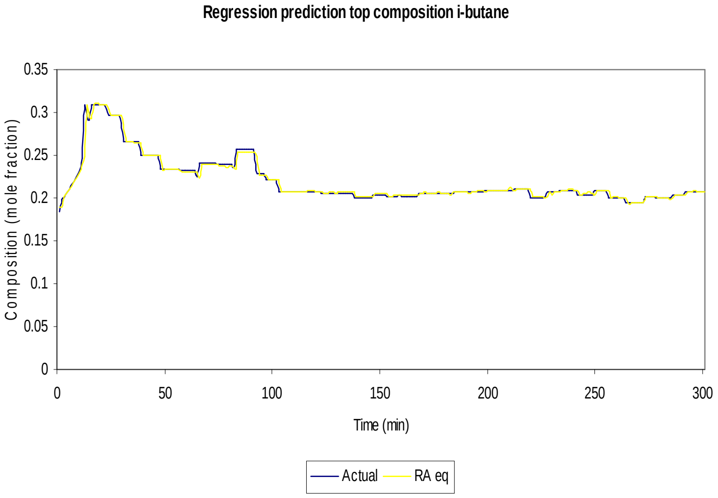


Figure. A38 Prediction and actual value for top composition i-butane line plot

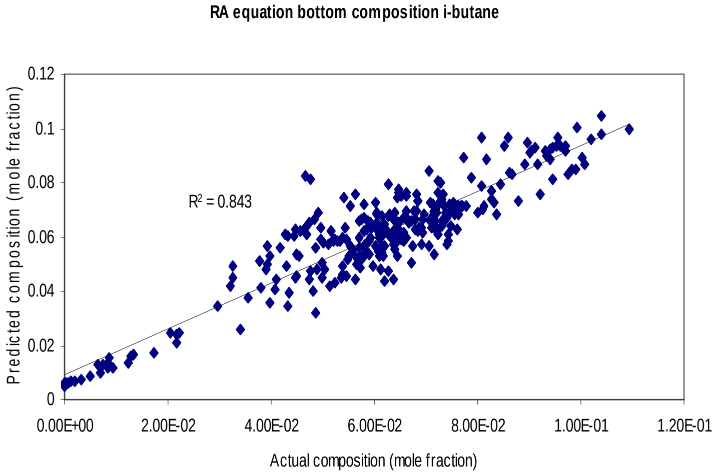


Figure. A39 Prediction versus actual value RA equation bottom composition i-butane

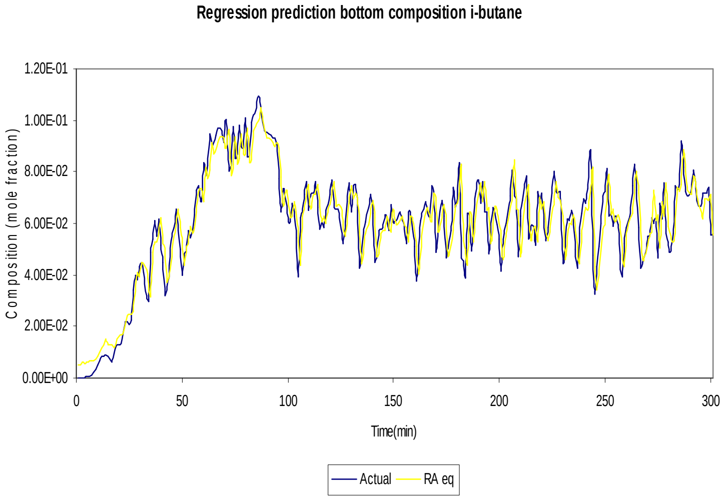


Figure A40. Prediction and actual value for bottom composition i-butane line plot

Table A7. i-butane statistical analysis of NN equation, PLS equation and RA equation

Parameter	NN eq	PLS eq	RA eq
rmse_bottom	1.87E-07	0.0088	0.0091
rmse_top	1.13E-07	0.0055	0.0051
CDC_bottom	100	55	57.33
CDC_top	26	16.66	17.33
R_bottom	1	0.91	0.91
R_top	1	0.97	0.98
AIC_bottom	-1699.72	-1788.55	-1767.6
AIC_top	-2178.67	-2221.56	-2180.5
BIC_bottom	-1684.89	-1773.72	-1752.77
BIC_top	-2163.84	-2206.73	-2165.67
MAPE_bottom	0.0064	2.79	-4.91
MAPE_top	-5.1E-05	0.0056	-0.087
Cp_bottom	1	0.91	0.91
Cp_top	1	0.97	0.98

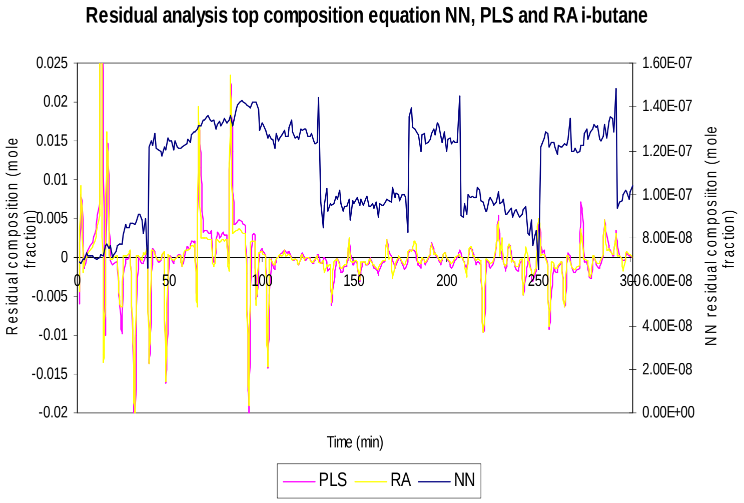


Figure A41 Residual analysis for neural network equation, PLS equation and regression analysis equation top composition i-butane

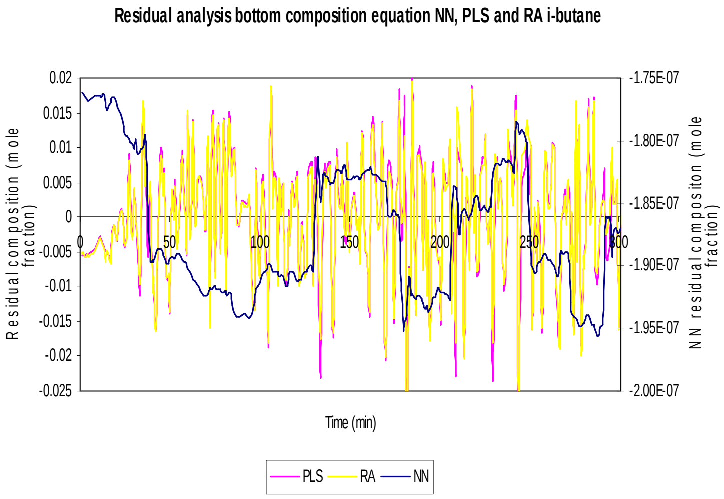


Figure A42 Residual analysis for neural network equation, PLS equation and regression analysis equation bottom composition i-butane

Residual neural network i-butane

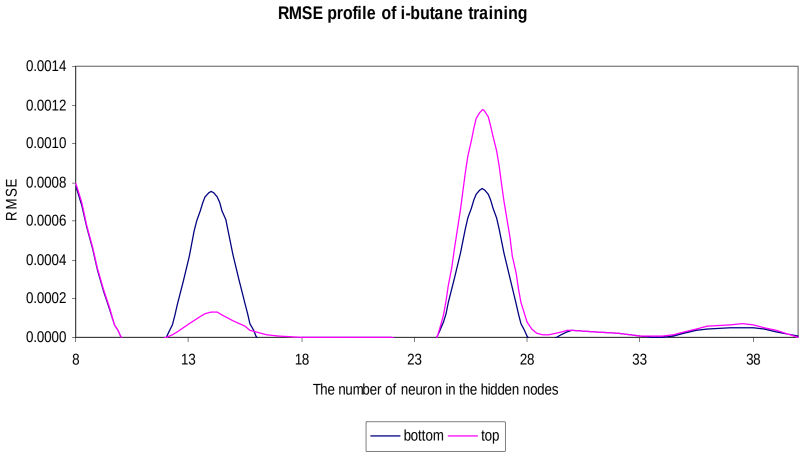


Figure A43 Profile of the RMSE of i-butane training

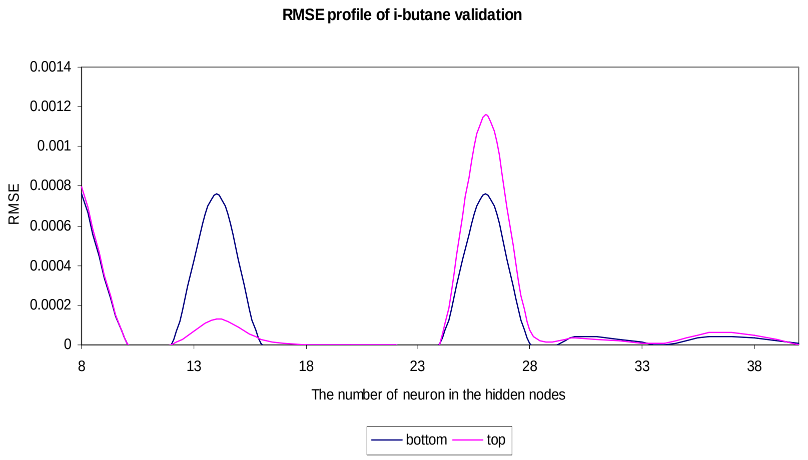


Figure A44 Profile of the RMSE of i-butane validation

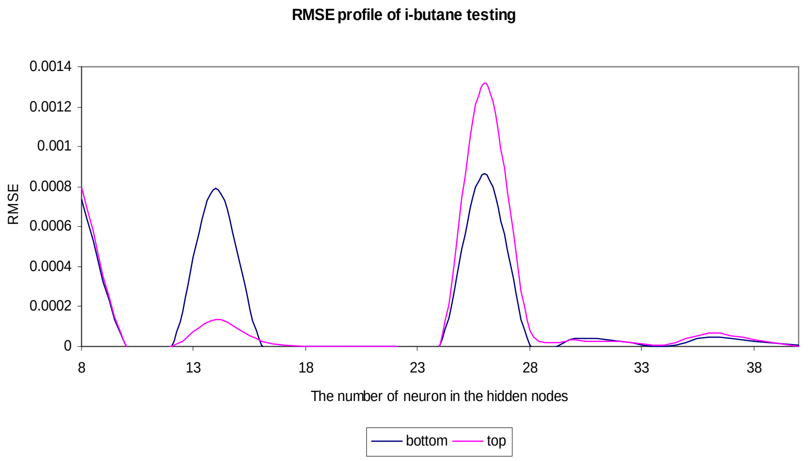
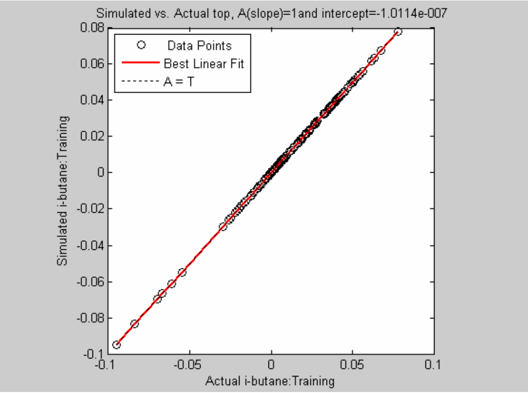


Figure A45 Profile of the RMSE of i-butane testing



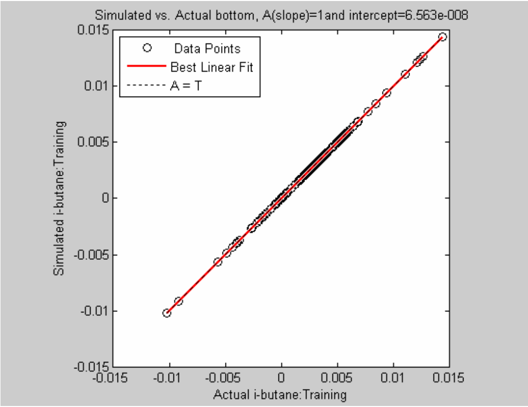
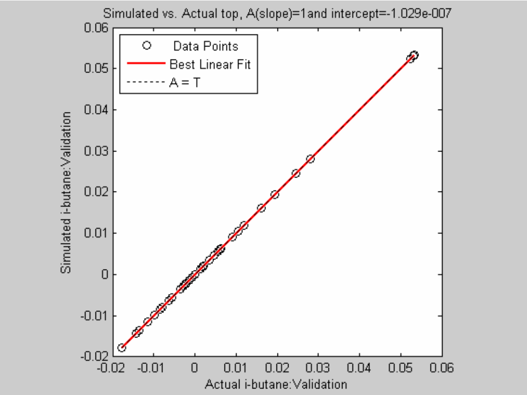


Figure A46 Actual and simulated i-butane top residual composition training

Figure A47 Actual and simulated i-butane bottom residual composition training



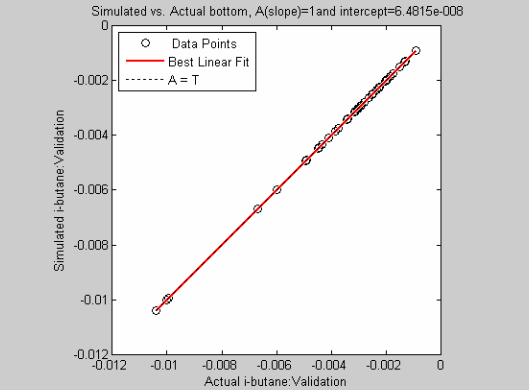
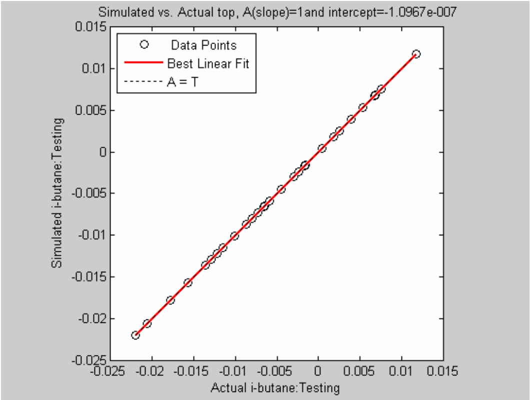


Figure A48 Actual and simulated i-butane top residual composition validation

Figure A49 Actual and simulated i-butane bottom residual composition validation



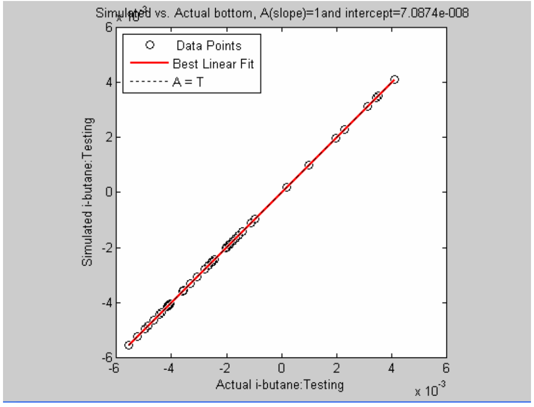
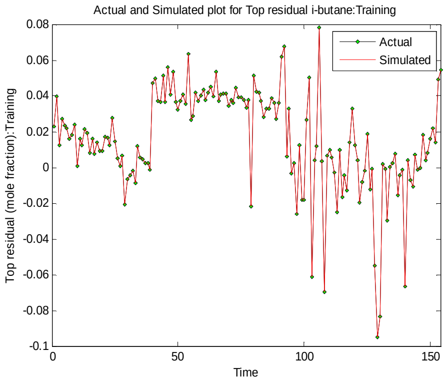


Figure A50 Actual and simulated i-butane top residual composition testing

Figure A51 Actual and simulated i-butane bottom residual composition testing



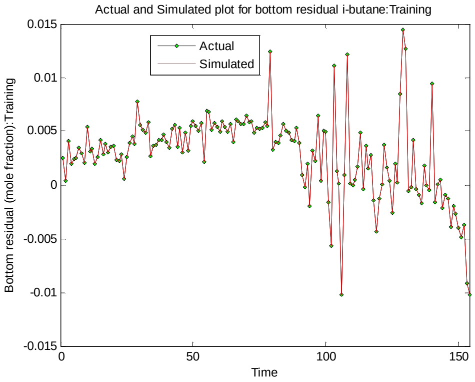


Figure A52 Actual and simulated i-butane top residual composition line plot training

Figure A53 Actual and simulated i-butane bottom residual composition line plot training

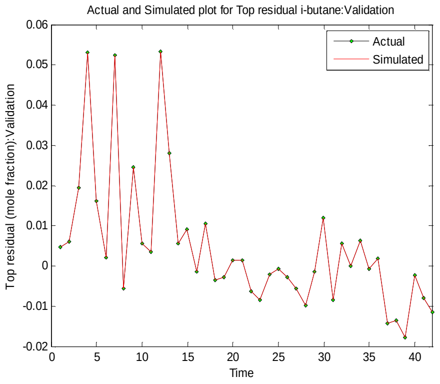


Figure A54 Actual and simulated i-butane top residual composition line plot validation

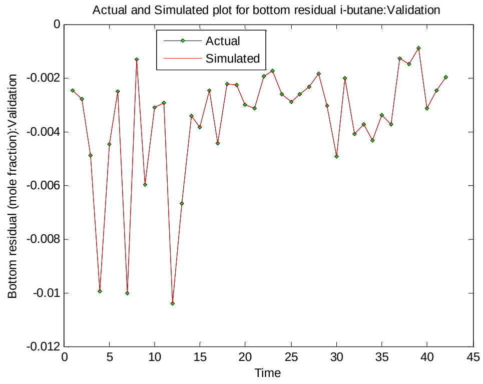


Figure A55 Actual and simulated i-butane bottom residual composition line plot validation

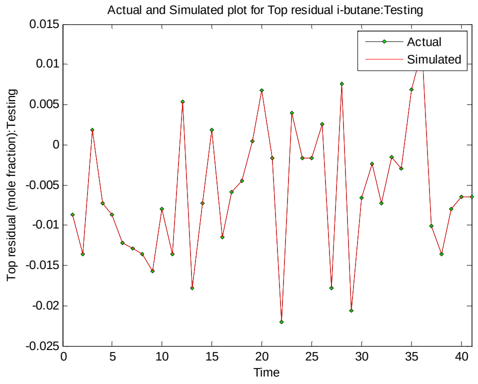


Figure A56 Actual and simulated i-butane top residual composition line plot testing

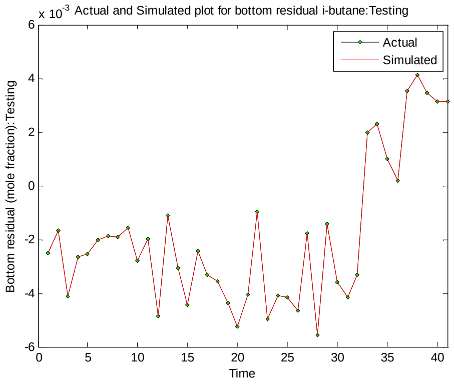


Figure A57 Actual and simulated i-butane bottom residual composition line plot Testing

Table A8 Statistical analysis for residual i-butane with partition into 3

Parameter	Open loop
rmse_bottom_training	6.51-08
rmse_top_training	1.00E-07
CDC_bottom_training	100
CDC_top_training	100
R_bottom_training	1
R_top_training	1
AIC_bottom_training	1353.45
AIC_top_training	692.02
BIC_bottom_training	1341.30
BIC_top_training	679.87
MAPE_bottom_training	0.013
MAPE_top_training	6.361E-05
Cp_bottom_training	1
Cp_top_training	1
rmse_bottom_validation	6.71E-08
rmse_top_validation	1.041E-07
CDC_bottom_validation	100
CDC_top_validation	100
R_bottom_validation	1
R_top_validation	1
AIC_bottom_validation	393.47
AIC_top_validation	230.26
BIC_bottom_validation	386.52
BIC_top_validation	223.31
MAPE_bottom_validation	0.0024
MAPE_top_validation	0.0060
Cp_bottom_validation	1
Cp_top_validation	1
rmse_bottom_testing	6.98E-08
rmse_top_testing	1.084E-07
CDC_bottom_testing	97.5
CDC_top_testing	97.5
R_bottom_testing	1
R_top_testing	1
AIC_bottom_testing	381.32
AIC_top_testing	273.51
BIC_bottom_testing	374.46
BIC_top_testing	266.66
MAPE_bottom_testing	0.00048
MAPE_top_testing	0.00042
Cp_bottom_testing	1
Cp_top_testing	1

Hybrid modeling of i-butane

The equation i-butane for reflux drum;

$$M_{RD}\frac{dx_D}{dt} = 1.68y_{35} - 0.51x_D \tag{7}$$

The equation i-butane column base equation;

$$M_B\frac{dx_B}{dt} = 160\,x_1 - 82\,y_B - 75\,x_B \tag{8}$$

i-butane

In this case, p is the inputs to the neural network residual composition given by the vector

$$\left[mv2(k)\ mv2(k-1)\ mv3(k)\ mv3(k-1)\ e(k)\ e(k-1)\ o_{top}(k)\ o_{top}(k-1)\ o_{bot}(k)\ o_{bot}(k-1)\right]^T$$

After pruning the neural network structure (simplifying the weights and biases values) the equation above can further be simplified to give the residual composition equation below;

$$\begin{bmatrix}\Delta y1\\ \Delta y2\end{bmatrix} = \begin{bmatrix}-1.52\ -0.2\ -0.16\ 1.32\ -0.88\ 0.47\ 2.00\ 0.16\ -0.04\ 0.038\\ 0.32\ 0.47\ -0.83\ -0.64\ 0.012\ -1.13\ -0.42\ -0.19\ 0.64\ 0.45\end{bmatrix}p + \begin{bmatrix}2.03\\ -1.38\end{bmatrix} \tag{9}$$

The equation below is the combination of first principle model composition and the residual top and bottom composition

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \begin{bmatrix} \int 1.68 y_{35} - 0.51 x_D \, dt \\ \int 160 x_1 - 82 y_B - 75 x_B \, dt \end{bmatrix} + \begin{bmatrix} -1.52 & -0.2 & -0.16 & 1.32 & -0.88 & 0.47 & 2.00 & 0.16 & -0.04 & 0.038 \\ 0.32 & 0.47 & -0.83 & -0.64 & 0.012 & -1.13 & -0.42 & -0.19 & 0.64 & 0.45 \end{bmatrix} p + \begin{bmatrix} 2.03 \\ -1.38 \end{bmatrix}$$

(10)

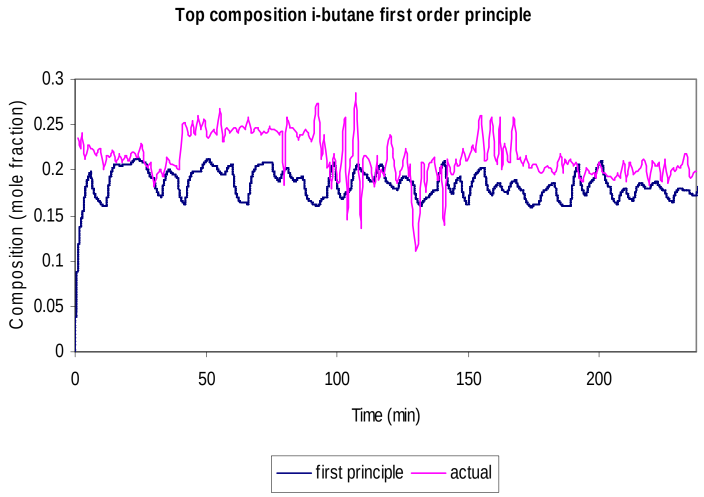


Figure A58 Top composition i-butane first principle model

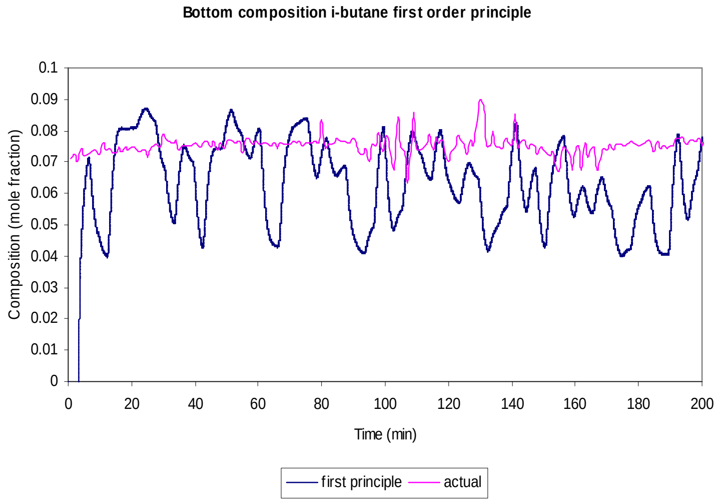


Figure A59 Bottom composition i-butane first principle model

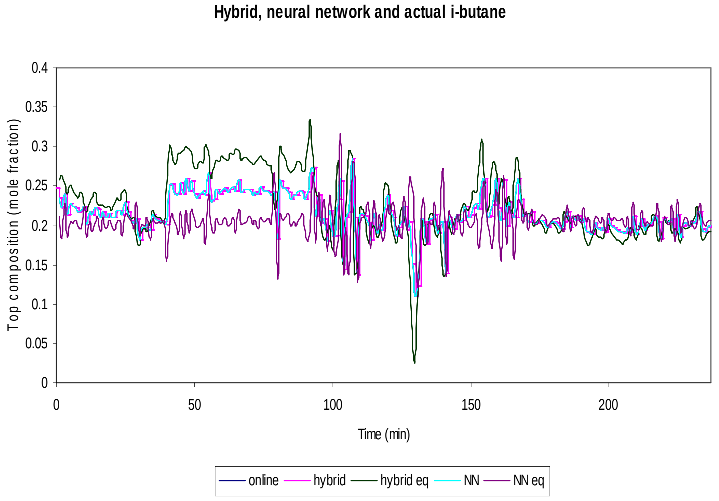


Figure A60 Hybrid model, neural network and actual top composition i-butane

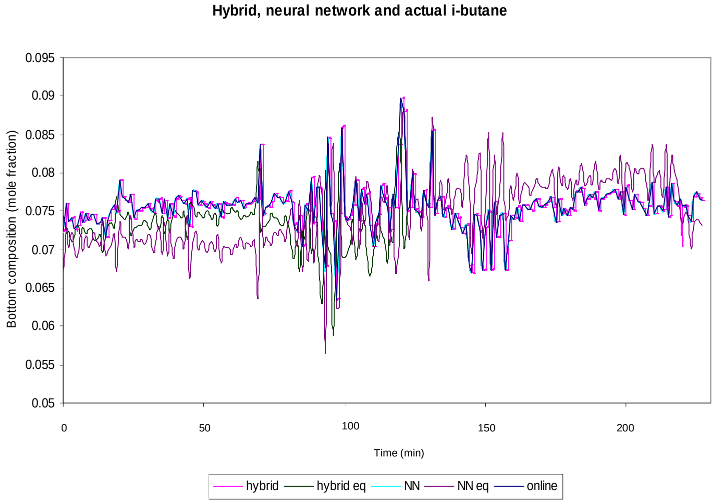


Figure A61 Hybrid model, neural network and actual bottom composition i-butane

Table A9 Statistical analysis for composition prediction i-butane.

		Black box model	
	Hybrid eq	Hybrid	NN
rmse_bottom	3.27E-05	1.31E-04	0.002258
rmse_top	2.36E-05	6.27E-05	0.027047
R_bottom	1	0.96	0.75
R_top	1	0.98	0.45
MAPE_bottom	-0.086	0.21	-0.52
MAPE_top	-0.027	0.030	-5.56

Advanced process control

Neural network estimator i-butane

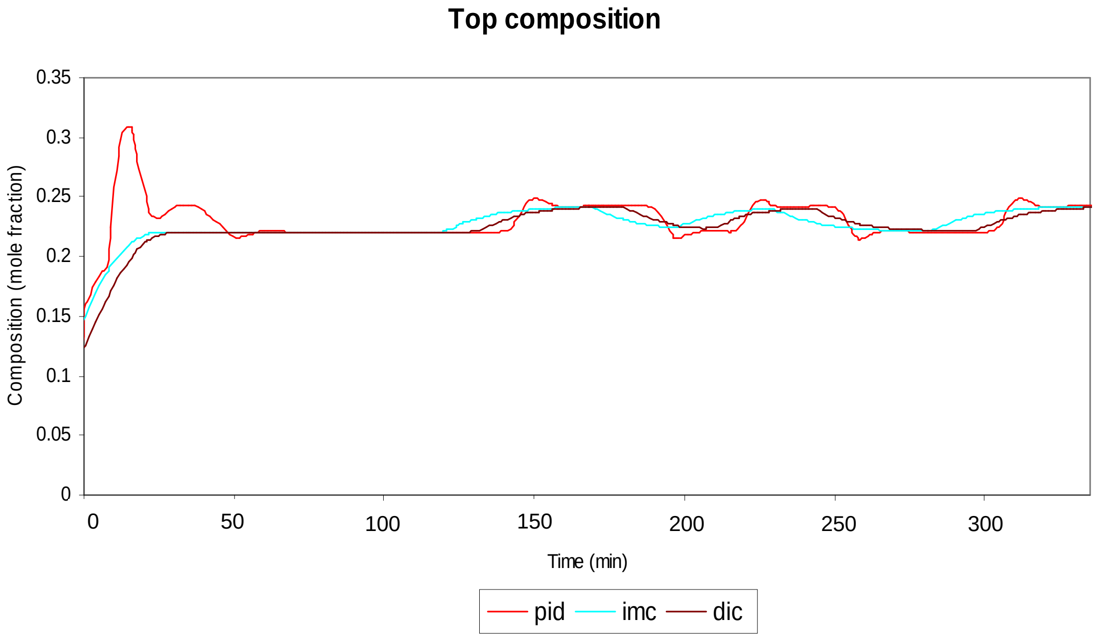


Figure A62 Neural network estimator top composition

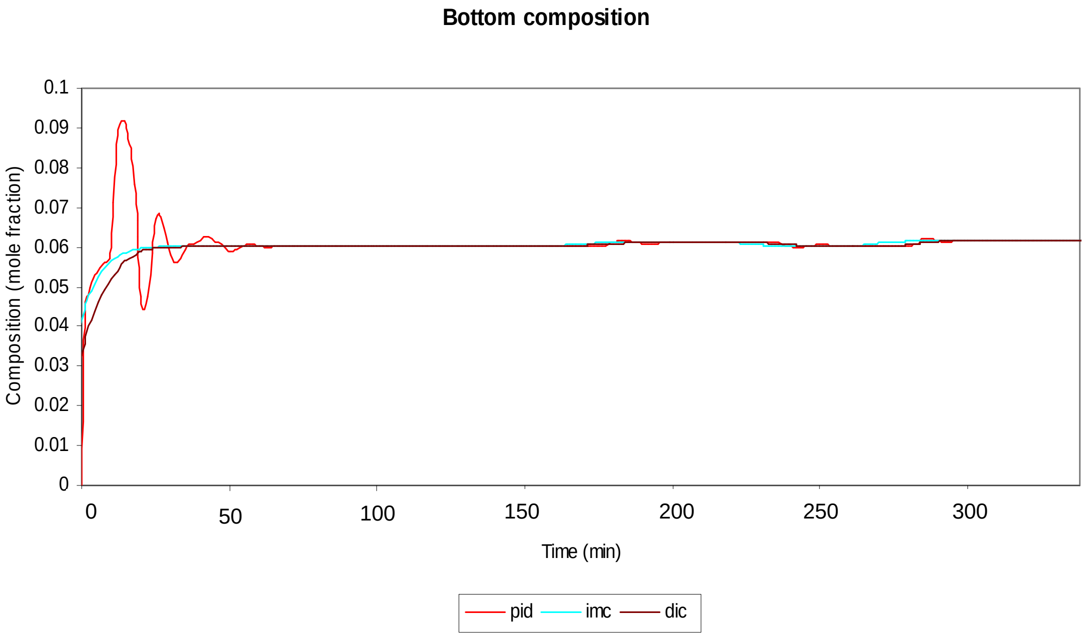


Figure A63 Neural network estimator bottom composition

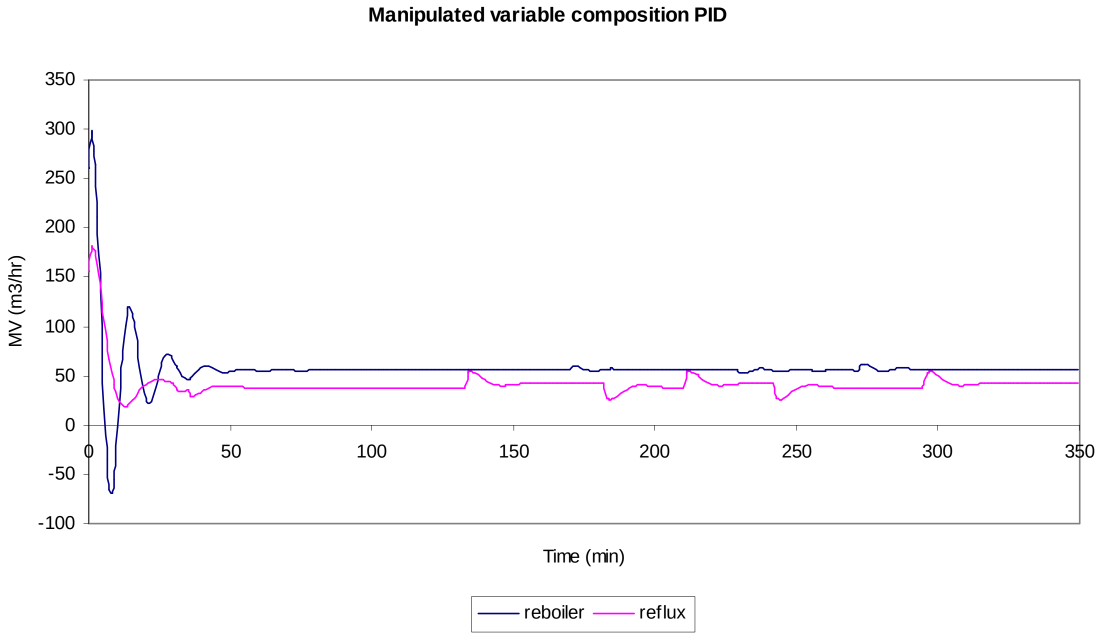


Figure A64 Manipulated variable composition PID

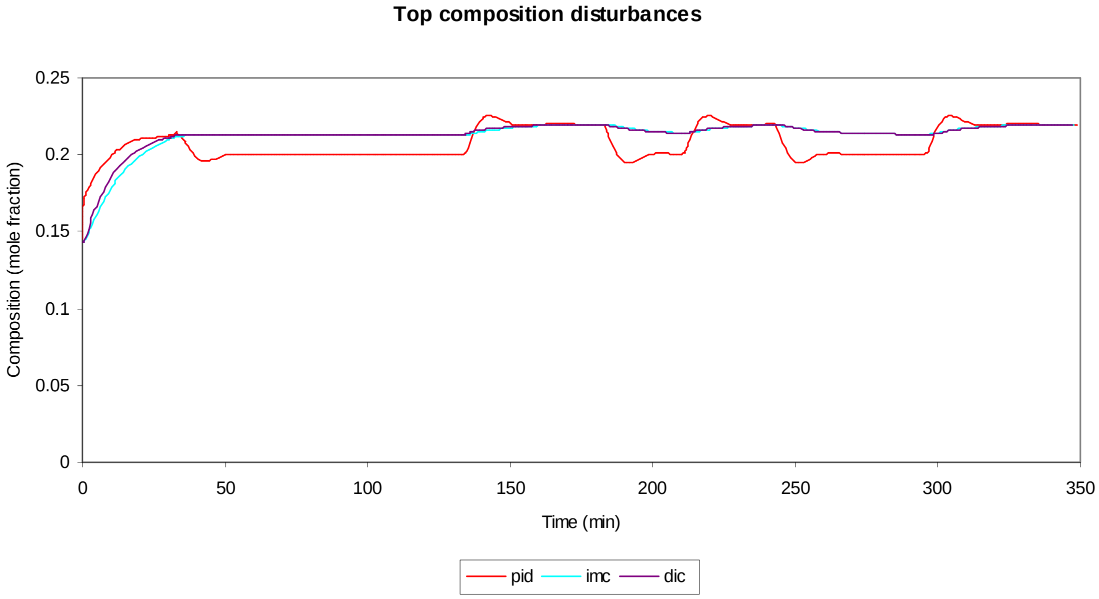


Figure A65 Disturbances of the top composition

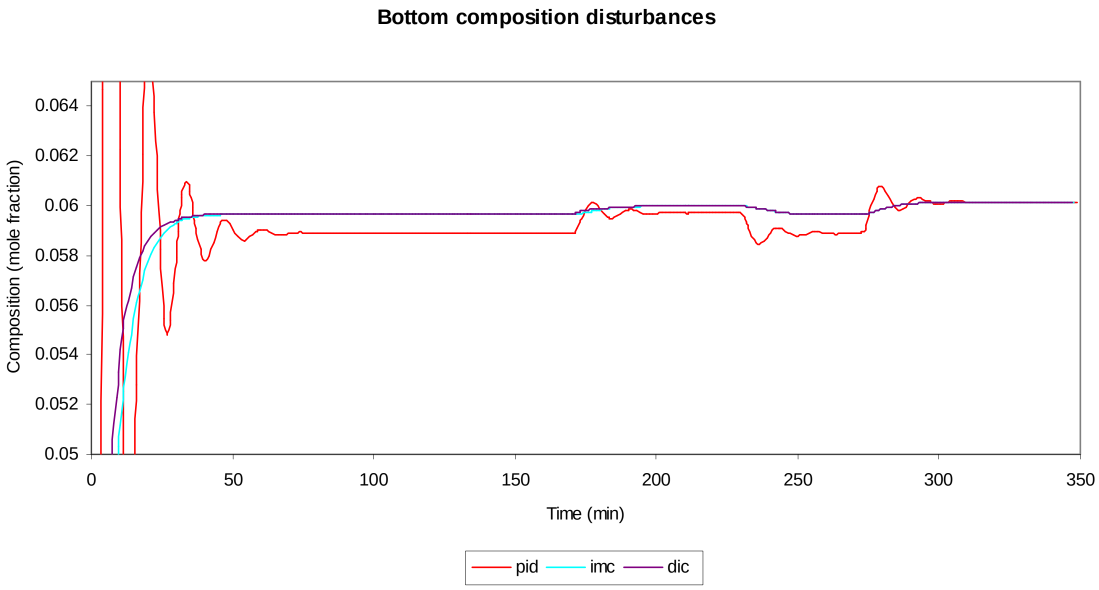


Figure A66 Disturbances of the bottom composition

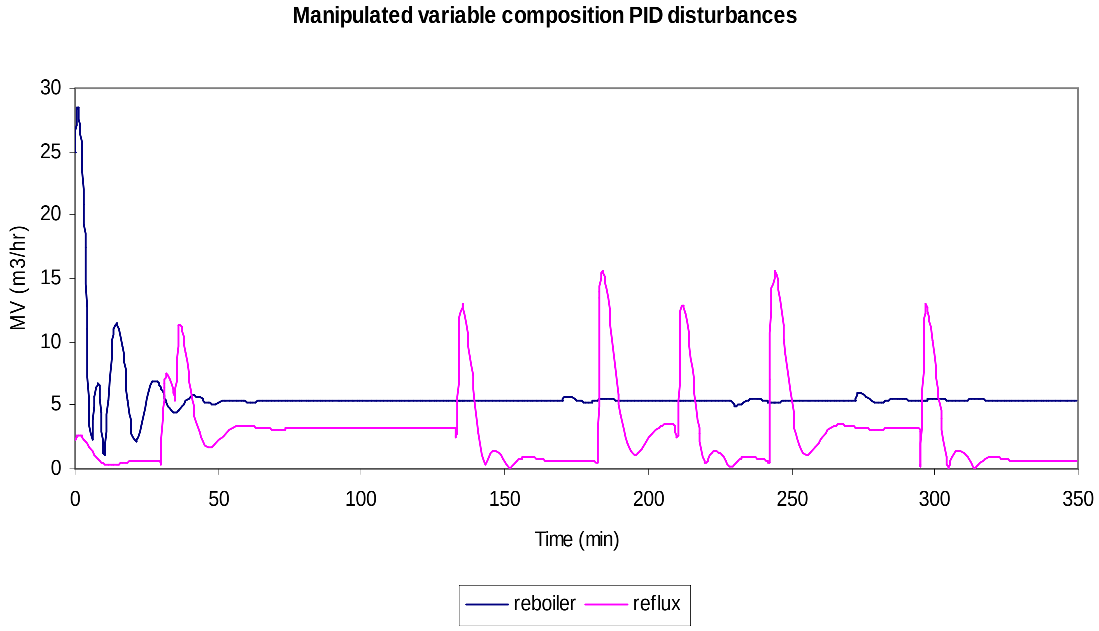


Figure A67 Manipulated variable of the composition PID due to disturbances

APPENDIX 1.2

Neural network programming

Composition n-butane

```
clc;
clear all;
close all;

%Data
A = xlsread('Reboiler reflux n-butane test remove mv','T');
B = xlsread('Reboiler reflux n-butane test remove mv','V');
C = xlsread('Reboiler reflux n-butane test remove mv','Te');


%Partitioning 40% Tr, 40% V, 20% Te
P_tr=A(:,1:10)';
P_v=B(:,1:10)';
P_te=C(:,1:10)';
T_tr=A(:,11:12)';
T_v=B(:,11:12)';
T_te=C(:,11:12)';


P_tr=con2seq(P_tr);
T_tr=con2seq(T_tr);
p=[P_tr;T_tr];
t=T_tr;

P_v=con2seq(P_v);
T_v=con2seq(T_v);
p1=[P_v;T_v];
t1=T_v;

P_te=con2seq(P_te);
T_te=con2seq(T_te);
p2=[P_te;T_te];
t2=T_te;


%set up network
d1=0;
d2=0;

net = newnarxsp(minmax(p),d1,d2,[10 34
2],{'purelin','purelin','purelin'},'trainlm','learngdm','mse');
net.trainParam.show = 10;
net.trainParam.epochs = 1000;
net.trainParam.goal   = 1e-6;
```

```
% Train netwrk with early stopping
rand('seed',417000);
net = init(net);

%Set up the validation and testing sets in a structure form
val.P=p1; val.T=t1;
test.P=p2; test.T=t2;
net = train(net,p,t,[],[],val,test);


%simulate network
at =sim(net,p2); %Testing set
at=cell2mat(at);
T_te=cell2mat(T_te);

at1 =sim(net,p1); %validation set
at1=cell2mat(at1);
T_v=cell2mat(T_v);

at2 =sim(net,p); %training set
at2=cell2mat(at2);
T_tr=cell2mat(T_tr);

%------------------graphs----------------------------

figure(1)
[slope1,intercept1,R1] = postreg(at(1,:),T_te(1,:));%Top
ylabel('Simulated n-butane:Testing'), xlabel('Actual n-butane:Testing'),...
title([ ' Simulated vs. Actual top, A(slope)=', num2str(slope1),'and
intercept=',num2str(intercept1)]);

figure(2)
[slope2,intercept2,R2] = postreg(at(2,:),T_te(2,:));%Bottom
ylabel('Simulated n-butane:Testing'), xlabel('Actual n-butane:Testing'),...
title([ ' Simulated vs. Actual bottom, A(slope)=', num2str(slope2),'and
intercept=',num2str(intercept2)]);

figure(3)
[slope3,intercept3,R3] = postreg(at1(1,:),T_v(1,:));%Top
ylabel('Simulated n-butane:Validation'), xlabel('Actual n-butane:Validation'),...
title([ ' Simulated vs. Actual top, A(slope)=', num2str(slope3),'and
intercept=',num2str(intercept3)]);

figure(4)
[slope4,intercept4,R4] = postreg(at1(2,:),T_v(2,:));%Bottom
ylabel('Simulated n-butane:Validation'), xlabel('Actual n-butane:Validation'),...
title([ ' Simulated vs. Actual bottom, A(slope)=', num2str(slope4),'and
intercept=',num2str(intercept4)]);

figure(5)
[slope5,intercept5,R5] = postreg(at2(1,:),T_tr(1,:)); %Top
ylabel('Simulated n-butane:Training'),xlabel('Actual n-butane:Training'),...
```

```
title([ ' Simulated vs. Actual top, A(slope)=', num2str(slope5),'and
intercept=',num2str(intercept5)]);

figure(6)
[slope6,intercept6,R6] = postreg(at2(2,:),T_tr(2,:));%Bottom
ylabel('Simulated n-butane:Training'), xlabel('Actual n-butane:Training'),...
title([ ' Simulated vs. Actual bottom, A(slope)=', num2str(slope6),'and
intercept=',num2str(intercept6)]);

figure(7) %Testing
time = 1:length(T_te(1,:));
plot(time,T_te(1,:),'kd-', time,at(1,:),'r-','LineWidth',0.5,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',4);
             xlabel('Time'), ylabel('Top composition (mole fraction):Testing'),...
                legend('Actual','Simulated')
Title('Actual and Simulated plot for Top composition n-butane:Testing'),

figure(8) %Testing
time = 1:length(T_te(2,:));
     plot(time,T_te(2,:),'kd-', time,at(2,:),'r-','LineWidth',0.5,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',4);
     xlabel('Time'), ylabel('Bottom composition (mole fraction):Testing'),...
     legend('Actual','Simulated')
     Title('Actual and Simulated plot for bottom composition n-butane:Testing'),

figure(9) %Validation
time = 1:length(T_v(1,:));
 plot(time,T_v(1,:),'kd-', time,at1(1,:),'r-','LineWidth',0.5,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',4);
      xlabel('Time'), ylabel('Top composition (mole fraction):Validation'),...
      legend('Actual','Simulated')
      Title('Actual and Simulated plot for Top composition n-butane:Validation'),

figure(10) %Validation
time = 1:length(T_v(2,:));
plot(time,T_v(2,:),'kd-', time,at1(2,:),'r-','LineWidth',0.5,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',4);
     xlabel('Time'), ylabel('Bottom composition (mole fraction):Validation'),...
     legend('Actual','Simulated')
     Title('Actual and Simulated plot for bottom composition n-butane:Validation'),

figure(11) %Training
time = 1:length(T_tr(1,:));
      plot(time,T_tr(1,:),'kd-', time,at2(1,:),'r-','LineWidth',0.5,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',4);
    xlabel('Time'), ylabel('Top composition (mole fraction):Training'),...
    legend('Actual','Simulated')
    Title('Actual and Simulated plot for Top composition n-butane:Training'),

figure(12) %Training
time = 1:length(T_tr(2,:));
    plot(time,T_tr(2,:),'kd-', time,at2(2,:),'r-','LineWidth',0.5,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',4);
    xlabel('Time'), ylabel('Bottom composition (mole fraction):Training'),...
    legend('Actual','Simulated')
    Title('Actual and Simulated plot for bottom composition n-butane:Training'),

%-----------------Training set performance measurement---------------------------

% rmse calculation Training
[row1,col1] = size(T_tr);
error_col = zeros(row1,col1);
for i = 1:1:row1,
for j = 1:1:col1,
   error_col(i,j) = (at2(i,j) - T_tr(i,j))^2;

end
end
sum_error = sum(error_col(1,:));
rmse_top_training = sqrt(sum_error/col1);
sum_error = sum(error_col(2,:));
rmse_bottom_training = sqrt(sum_error/col1);

% CDC calculation for training
d1=zeros(1,col1-1);
w1=zeros(1,col1-1);
i=2;
for n=1:1:col1-1
   a=T_tr(1,i) - T_tr(1,i-1);
   b=at2(1,i) - at2(1,i-1);
   c=a*b;
   d1(:,i-1)=c;
   g=T_tr(2,i) - T_tr(2,i-1);
   h=at2(2,i) - at2(2,i-1);
   x=g*h;
   w1(:,i-1)=x;
   i=i+1;

end

D_top=zeros(1,col1-1);
D_botm=zeros(1,col1-1);
```

```
m=1;
for q=1:1:col1-1
    if d1(:,m)>0
         D_top(:,m)=1;
    else
        D_top(:,m)=0;
    end

     if w1(:,m)>0
         D_botm(:,m)=1;
    else
        D_botm(:,m)=0;
    end
    m=m+1;
end

[row2,col2] = size(D_top);
[row3,col3] = size(D_botm);
CDC_top_training = (sum(D_top))*(100/(col2));
CDC_bottom_training = (sum(D_botm))*(100/(col3));


%AIC and BIC calculation for training
[Coeff_top,Errors_top,LLF_top,Innovations_top,Sigmas_top,Summary_top]=garchfit(at2(1,:));
[Coeff_bottom,Errors_bottom,LLF_bottom,Innovations_bottom,Sigmas_bottom,Summary_bottom]=garchfit(at2(2,:));
[AICt_trainng BICt_trainng]=aicbic(LLF_top,4,col1);
[AICT_trainng BICT_trainng]=aicbic(LLF_bottom,4,col1);

% MAPE calculation Training
[row1,col1] = size(T_tr);
Percentage_error = zeros(row1,col1);
for i = 1:1:row1,
    for j = 1:1:col1,
        Percentage_error(i,j) = ((at2(i,j) - T_tr(i,j))/at2(i,j))*100;
    end
end
sum_Percentage_error = sum(Percentage_error(1,:));
MAPE_top_training = (sum_Percentage_error/col1);
sum_Percentage_error = sum(Percentage_error(2,:));
MAPE_bottom_training = (sum_Percentage_error/col1);

%pearson correlation coeff  calculation Training
[row1,col1] = size(T_tr);
Predicted = zeros(row1,col1);
Actual = zeros(row1,col1);
val1 = zeros(row1,col1);
val2 = zeros(row1,col1);
val3 = zeros(row1,col1);
for i = 1:1:row1,
    Ep=mean(at2(i,:));
```

```
Ea=mean(T_tr(i,:));
   for j = 1:1:col1,
  Predicted(i,j) = at2(i,j);
  Actual(i,j)=T_tr(i,j);
  val1(i,j)=(Predicted(i,j)-Ep)*(Actual(i,j)-Ea);
  val2(i,j)=(Predicted(i,j)-Ep)^2;
  val3(i,j)=(Actual(i,j)-Ea)^2;
   end
end

Cp_top_training=(sum(val1(1,:)))/(sqrt(sum(val2(1,:))*sum(val3(1,:))));
Cp_bottom_training=(sum(val1(2,:)))/(sqrt(sum(val2(2,:))*sum(val3(2,:))));

%----------------validation set performance measurement---------------------------

% rmse calculation validation
[row1,col1] = size(T_v);
error_col = zeros(row1,col1);
for i = 1:1:row1,
for j = 1:1:col1,
   error_col(i,j) = (at1(i,j) - T_v(i,j))^2;

end
end
sum_error = sum(error_col(1,:));
rmse_top_validation = sqrt(sum_error/col1);
sum_error = sum(error_col(2,:));
rmse_bottom_validation = sqrt(sum_error/col1);

% CDC calculation for training
d1=zeros(1,col1-1);
w1=zeros(1,col1-1);
i=2;
for n=1:1:col1-1
   a=T_v(1,i) - T_v(1,i-1);
   b=at1(1,i) - at1(1,i-1);
   c=a*b;
   d1(:,i-1)=c;
   g=T_v(2,i) - T_v(2,i-1);
   h=at1(2,i) - at1(2,i-1);
   x=g*h;
   w1(:,i-1)=x;
   i=i+1;
end

D_top=zeros(1,col1-1);
D_botm=zeros(1,col1-1);
m=1;
for q=1:1:col1-1
   if d1(:,m)>0
       D_top(:,m)=1;
   else
```

```
        D_top(:,m)=0;
    end

     if w1(:,m)>0
         D_botm(:,m)=1;
    else
        D_botm(:,m)=0;
    end
    m=m+1;
end

[row2,col2] = size(D_top);
[row3,col3] = size(D_botm);
CDC_top_validation = (sum(D_top))*(100/(col2));
CDC_bottom_validation = (sum(D_botm))*(100/(col3));

%AIC and BIC calculation for validation
[Coeff_top,Errors_top,LLF_top,Innovations_top,Sigmas_top,Summary_top]=garchfit(at
1(1,:));
[Coeff_bottom,Errors_bottom,LLF_bottom,Innovations_bottom,Sigmas_bottom,Summ
ary_bottom]=garchfit(at1(2,:));
[AICt_validation BICt_validation]=aicbic(LLF_top,4,col1);
[AICT_validation BICT_validation]=aicbic(LLF_bottom,4,col1);

% MAPE calculation validation
[row1,col1] = size(T_v);
Percentage_error = zeros(row1,col1);
for i = 1:1:row1,
    for j = 1:1:col1,
        Percentage_error(i,j) = ((at1(i,j) - T_v(i,j))/at1(i,j))*100;
    end
end
sum_Percentage_error = sum(Percentage_error(1,:));
MAPE_top_validation = (sum_Percentage_error/col1);
sum_Percentage_error = sum(Percentage_error(2,:));
MAPE_bottom_validation = (sum_Percentage_error/col1);

%pearson correlation coeff  calculation validation
[row1,col1] = size(T_v);
Predicted = zeros(row1,col1);
Actual = zeros(row1,col1);
val1 = zeros(row1,col1);
val2 = zeros(row1,col1);
val3 = zeros(row1,col1);
for i = 1:1:row1,
    Ep=mean(at1(i,:));
    Ea=mean(T_v(i,:));
    for j = 1:1:col1,
   Predicted(i,j) = at1(i,j);
   Actual(i,j)=T_v(i,j);
   val1(i,j)=(Predicted(i,j)-Ep)*(Actual(i,j)-Ea);
   val2(i,j)=(Predicted(i,j)-Ep)^2;
   val3(i,j)=(Actual(i,j)-Ea)^2;
    end
end

Cp_top_validation=(sum(val1(1,:)))/(sqrt(sum(val2(1,:))*sum(val3(1,:))));
Cp_bottom_validation=(sum(val1(2,:)))/(sqrt(sum(val2(2,:))*sum(val3(2,:))));

%-----------------testing set performance measurement---------------------------

% rmse calculation testing
[row1,col1] = size(T_te);
error_col = zeros(row1,col1);
for i = 1:1:row1,
for j = 1:1:col1,
    error_col(i,j) = (at(i,j) - T_te(i,j))^2;

end
end
sum_error = sum(error_col(1,:));
rmse_top_testing = sqrt(sum_error/col1);
sum_error = sum(error_col(2,:));
rmse_bottom_testing = sqrt(sum_error/col1);

% CDC calculation for training
d1=zeros(1,col1-1);
w1=zeros(1,col1-1);
i=2;
for n=1:1:col1-1
    a=T_te(1,i) - T_te(1,i-1);
    b=at(1,i) - at(1,i-1);
    c=a*b;
    d1(:,i-1)=c;
    g=T_te(2,i) - T_te(2,i-1);
    h=at(2,i) - at(2,i-1);
    x=g*h;
    w1(:,i-1)=x;
    i=i+1;

end

D_top=zeros(1,col1-1);
D_botm=zeros(1,col1-1);
m=1;
for q=1:1:col1-1
    if d1(:,m)>0
         D_top(:,m)=1;
    else
        D_top(:,m)=0;
    end

     if w1(:,m)>0
         D_botm(:,m)=1;
    else
        D_botm(:,m)=0;
    end
    m=m+1;
 end

[row2,col2] = size(D_top);
[row3,col3] = size(D_botm);
CDC_top_testing = (sum(D_top))*(100/(col2));
CDC_bottom_testing = (sum(D_botm))*(100/(col3));


%AIC and BIC calculation for testing
[Coeff_top,Errors_top,LLF_top,Innovations_top,Sigmas_top,Summary_top]=garchfit(at(1,:));
[Coeff_bottom,Errors_bottom,LLF_bottom,Innovations_bottom,Sigmas_bottom,Summary_bottom]=garchfit(at(2,:));
[AICt_testing BICt_testing]=aicbic(LLF_top,4,col1);
[AICT_testing BICT_testing]=aicbic(LLF_bottom,4,col1);

% MAPE calculation testing
[row1,col1] = size(T_te);
Percentage_error = zeros(row1,col1);
for i = 1:1:row1,
    for j = 1:1:col1,
        Percentage_error(i,j) = ((at(i,j) - T_te(i,j))/at(i,j))*100;
    end
end
sum_Percentage_error = sum(Percentage_error(1,:));
MAPE_top_testing = (sum_Percentage_error/col1);
sum_Percentage_error = sum(Percentage_error(2,:));
MAPE_bottom_testing = (sum_Percentage_error/col1);

%pearson correlation coeff  calculation testing
[row1,col1] = size(T_te);
Predicted = zeros(row1,col1);
Actual = zeros(row1,col1);
val1 = zeros(row1,col1);
val2 = zeros(row1,col1);
val3 = zeros(row1,col1);
for i = 1:1:row1,
    Ep=mean(at(i,:));
Ea=mean(T_te(i,:));
    for j = 1:1:col1,
   Predicted(i,j) = at(i,j);
   Actual(i,j)=T_te(i,j);
   val1(i,j)=(Predicted(i,j)-Ep)*(Actual(i,j)-Ea);
   val2(i,j)=(Predicted(i,j)-Ep)^2;
   val3(i,j)=(Actual(i,j)-Ea)^2;
    end
end
```

```
Cp_top_testing=(sum(val1(1,:)))/(sqrt(sum(val2(1,:))*sum(val3(1,:))));
Cp_bottom_testing=(sum(val1(2,:)))/(sqrt(sum(val2(2,:))*sum(val3(2,:))));

%------------------display output----------------------------
clc;
output.rmse_top_training=rmse_top_training;
output.rmse_bottom_training=rmse_bottom_training;
output.CDC_top_training=CDC_top_training;
output.CDC_bottom_training=CDC_bottom_training;
output.R_top_training=R5;
output.R_bottom_training=R6;
output.AIC_top_training=AICt_trainng;
output.AIC_bottom_training=AICT_trainng;
output.BIC_top_training=BICt_trainng;
output.BIC_bottom_training=BICT_trainng;
output.MAPE_top_training=MAPE_top_training;
output.MAPE_bottom_training=MAPE_bottom_training;
output.Cp_top_training=Cp_top_training;
output.Cp_bottom_training=Cp_bottom_training;
output.rmse_top_validation=rmse_top_validation;
output.rmse_bottom_validation=rmse_bottom_validation;
output.CDC_top_validation=CDC_top_validation;
output.CDC_bottom_validation=CDC_bottom_validation;
output.R_top_validation=R3;
output.R_bottom_validation=R4;
output.AIC_top_validation=AICt_validation;
output.AIC_bottom_validation=AICT_validation;
output.BIC_top_validation=BICt_validation;
output.BIC_bottom_validation=BICT_validation;
output.MAPE_top_validation=MAPE_top_validation;
output.MAPE_bottom_validation=MAPE_bottom_validation;
output.Cp_top_validation=Cp_top_validation;
output.Cp_bottom_validation=Cp_bottom_validation;
output.rmse_top_testing=rmse_top_testing;
output.rmse_bottom_testing=rmse_bottom_testing;
output.CDC_top_testing=CDC_top_testing;
output.CDC_bottom_testing=CDC_bottom_testing;
output.R_top_testing=R1;
output.R_bottom_testing=R2;
output.AIC_top_testing=AICt_testing;
output.AIC_bottom_testing=AICT_testing;
output.BIC_top_testing=BICt_testing;
output.BIC_bottom_testing=BICT_testing;
output.MAPE_top_testing=MAPE_top_testing;
output.MAPE_bottom_testing=MAPE_bottom_testing;
output.Cp_top_testing=Cp_top_testing;
output.Cp_bottom_testing=Cp_bottom_testing;
disp(output),

o(1,1)=output.rmse_bottom_training;
o(2,1)=output.rmse_top_training;
o(3,1)=output.CDC_bottom_training;
```

```
o(4,1)=output.CDC_top_training;
o(5,1)=output.R_bottom_training;
o(6,1)=output.R_top_training;
o(7,1)=output.AIC_bottom_training;
o(8,1)=output.AIC_top_training;
o(9,1)=output.BIC_bottom_training;
o(10,1)=output.BIC_top_training;
o(11,1)=output.MAPE_bottom_training;
o(12,1)=output.MAPE_top_training;
o(13,1)=output.Cp_bottom_training;
o(14,1)=output.Cp_top_training;
o(15,1)=output.rmse_bottom_validation;
o(16,1)=output.rmse_top_validation;
o(17,1)=output.CDC_bottom_validation;
o(18,1)=output.CDC_top_validation;
o(19,1)=output.R_bottom_validation;
o(20,1)=output.R_top_validation;
o(21,1)=output.AIC_bottom_validation;
o(22,1)=output.AIC_top_validation;
o(23,1)=output.BIC_bottom_validation;
o(24,1)=output.BIC_top_validation;
o(25,1)=output.MAPE_bottom_validation;
o(26,1)=output.MAPE_top_validation;
o(27,1)=output.Cp_bottom_validation;
o(28,1)=output.Cp_top_validation;
o(29,1)=output.rmse_bottom_testing;
o(30,1)=output.rmse_top_testing;
o(31,1)=output.CDC_bottom_testing;
o(32,1)=output.CDC_top_testing;
o(33,1)=output.R_bottom_testing;
o(34,1)=output.R_top_testing;
o(35,1)=output.AIC_bottom_testing;
o(36,1)=output.AIC_top_testing;
o(37,1)=output.BIC_bottom_testing;
o(38,1)=output.BIC_top_testing;
o(39,1)=output.MAPE_bottom_testing;
o(40,1)=output.MAPE_top_testing;
o(41,1)=output.Cp_bottom_testing;
o(42,1)=output.Cp_top_testing;

a=net.IW{1,1};
a1=net.IW{1,2};
b=net.LW{2,1};
c=net.LW{3,2};
d=net.b{1};
e=net.b{2};
f=net.b{3};

sprintf('\n%15.7f',o) ;

save newnarxsp_with_partition_test
```

Temperature

```
clc;
clear all;
close all;

%Data
A = xlsread('Reboiler reflux n-butane test remove mv','T');
B = xlsread('Reboiler reflux n-butane test remove mv','V');
C = xlsread('Reboiler reflux n-butane test remove mv','Te');


%Partitioning 40% Tr, 40% V, 20% Te
P_tr=A(:,1:10)';
P_v=B(:,1:10)';
P_te=C(:,1:10)';
T_tr=A(:,11:12)';
T_v=B(:,11:12)';
T_te=C(:,11:12)';


P_tr=con2seq(P_tr);
T_tr=con2seq(T_tr);
p=[P_tr;T_tr];
t=T_tr;

P_v=con2seq(P_v);
T_v=con2seq(T_v);
p1=[P_v;T_v];
t1=T_v;

P_te=con2seq(P_te);
T_te=con2seq(T_te);
p2=[P_te;T_te];
t2=T_te;


%set up network
d1=0;
d2=0;

net = newnarxsp(minmax(p),d1,d2,[10 10
2],{'purelin','purelin','purelin'},'trainlm','learngdm','mse');
net.trainParam.show = 10;
net.trainParam.epochs = 1000;
net.trainParam.goal   = 1e-6;


% Train netwrk with early stopping
rand('seed',417000);
net = init(net);
```

```
%Set up the validation and testing sets in a structure form
val.P=p1; val.T=t1;
test.P=p2; test.T=t2;
net = train(net,p,t,[],[],val,test);


%simulate network
at =sim(net,p2); %Testing set
at=cell2mat(at);
T_te=cell2mat(T_te);

at1 =sim(net,p1); %validation set
at1=cell2mat(at1);
T_v=cell2mat(T_v);

at2 =sim(net,p); %training set
at2=cell2mat(at2);
T_tr=cell2mat(T_tr);

%------------------graphs----------------------------

figure(1)
[slope1,intercept1,R1] = postreg(at(1,:),T_te(1,:));%Top
ylabel('Simulated temperature:Testing'), xlabel('Actual temperature:Testing'),...
title([ ' Simulated vs. Actual top, A(slope)=', num2str(slope1),'and
intercept=',num2str(intercept1)]);

figure(2)
[slope2,intercept2,R2] = postreg(at(2,:),T_te(2,:));%Bottom
ylabel('Simulated temperature:Testing'), xlabel('Actual temperature:Testing'),...
title([ ' Simulated vs. Actual bottom, A(slope)=', num2str(slope2),'and
intercept=',num2str(intercept2)]);

figure(3)
[slope3,intercept3,R3] = postreg(at1(1,:),T_v(1,:));%Top
ylabel('Simulated temperature:Validation'), xlabel('Actual temperature:Validation'),...
title([ ' Simulated vs. Actual top, A(slope)=', num2str(slope3),'and
intercept=',num2str(intercept3)]);

figure(4)
[slope4,intercept4,R4] = postreg(at1(2,:),T_v(2,:));%Bottom
ylabel('Simulated temperature:Validation'), xlabel('Actual temperature:Validation'),...
title([ ' Simulated vs. Actual bottom, A(slope)=', num2str(slope4),'and
intercept=',num2str(intercept4)]);

figure(5)
[slope5,intercept5,R5] = postreg(at2(1,:),T_tr(1,:)); %Top
ylabel('Simulated temperature:Training'),xlabel('Actual temperature:Training'),...
title([ ' Simulated vs. Actual top, A(slope)=', num2str(slope5),'and
intercept=',num2str(intercept5)]);

figure(6)
```

```
[slope6,intercept6,R6] = postreg(at2(2,:),T_tr(2,:));%Bottom
ylabel('Simulated temperature:Training'), xlabel('Actual temperature:Training'),...
title([ ' Simulated vs. Actual bottom, A(slope)=', num2str(slope6),'and
intercept=',num2str(intercept6)]);

figure(7) %Testing
time = 1:length(T_te(1,:));
plot(time,T_te(1,:),'kd-', time,at(1,:),'r-','LineWidth',0.5,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','g',...
            'MarkerSize',4);
          xlabel('Time'), ylabel('Top temperature (deg C):Testing'),...
            legend('Actual','Simulated')
Title('Actual and Simulated plot for Top temperature:Testing'),

figure(8) %Testing
time = 1:length(T_te(2,:));
    plot(time,T_te(2,:),'kd-', time,at(2,:),'r-','LineWidth',0.5,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','g',...
            'MarkerSize',4);
    xlabel('Time'), ylabel('Bottom temperature (deg C):Testing'),...
    legend('Actual','Simulated')
    Title('Actual and Simulated plot for bottom temperature:Testing'),

figure(9) %Validation
time = 1:length(T_v(1,:));
 plot(time,T_v(1,:),'kd-', time,at1(1,:),'r-','LineWidth',0.5,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','g',...
            'MarkerSize',4);
     xlabel('Time'), ylabel('Top temperature (deg C):Validation'),...
     legend('Actual','Simulated')
     Title('Actual and Simulated plot for Top temperature:Validation'),

figure(10) %Validation
time = 1:length(T_v(2,:));
plot(time,T_v(2,:),'kd-', time,at1(2,:),'r-','LineWidth',0.5,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','g',...
            'MarkerSize',4);
    xlabel('Time'), ylabel('Bottom temperature (deg C):Validation'),...
    legend('Actual','Simulated')
    Title('Actual and Simulated plot for bottom temperature:Validation'),

figure(11) %Training
time = 1:length(T_tr(1,:));
     plot(time,T_tr(1,:),'kd-', time,at2(1,:),'r-','LineWidth',0.5,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','g',...
            'MarkerSize',4);
     xlabel('Time'), ylabel('Top temperature (deg C):Training'),...
```

```
        legend('Actual','Simulated')
        Title('Actual and Simulated plot for Top temperature:Training'),

figure(12) %Training
time = 1:length(T_tr(2,:));
       plot(time,T_tr(2,:),'kd-', time,at2(2,:),'r-','LineWidth',0.5,...
                       'MarkerEdgeColor','k',...
                       'MarkerFaceColor','g',...
                       'MarkerSize',4);
       xlabel('Time'), ylabel('Bottom temperature (deg C):Training'),...
       legend('Actual','Simulated')
       Title('Actual and Simulated plot for bottom temperature:Training'),

%-----------------Training set performance measurement---------------------------

% rmse calculation Training
[row1,col1] = size(T_tr);
error_col = zeros(row1,col1);
for i = 1:1:row1,
for j = 1:1:col1,
    error_col(i,j) = (at2(i,j) - T_tr(i,j))^2;

end
end
sum_error = sum(error_col(1,:));
rmse_top_training = sqrt(sum_error/col1);
sum_error = sum(error_col(2,:));
rmse_bottom_training = sqrt(sum_error/col1);

% CDC calculation for training
d1=zeros(1,col1-1);
w1=zeros(1,col1-1);
i=2;
for n=1:1:col1-1
    a=T_tr(1,i) - T_tr(1,i-1);
    b=at2(1,i) - at2(1,i-1);
    c=a*b;
    d1(:,i-1)=c;
    g=T_tr(2,i) - T_tr(2,i-1);
    h=at2(2,i) - at2(2,i-1);
    x=g*h;
    w1(:,i-1)=x;
    i=i+1;

end

D_top=zeros(1,col1-1);
D_botm=zeros(1,col1-1);
m=1;
for q=1:1:col1-1
    if d1(:,m)>0
         D_top(:,m)=1;
    else
        D_top(:,m)=0;
    end

     if w1(:,m)>0
         D_botm(:,m)=1;
    else
        D_botm(:,m)=0;
    end
    m=m+1;
end

[row2,col2] = size(D_top);
[row3,col3] = size(D_botm);
CDC_top_training = (sum(D_top))*(100/(col2));
CDC_bottom_training = (sum(D_botm))*(100/(col3));


%AIC and BIC calculation for training
[Coeff_top,Errors_top,LLF_top,Innovations_top,Sigmas_top,Summary_top]=garchfit(at2(1,:));
[Coeff_bottom,Errors_bottom,LLF_bottom,Innovations_bottom,Sigmas_bottom,Summary_bottom]=garchfit(at2(2,:));
[AICt_trainng BICt_trainng]=aicbic(LLF_top,4,col1);
[AICT_trainng BICT_trainng]=aicbic(LLF_bottom,4,col1);

% MAPE calculation Training
[row1,col1] = size(T_tr);
Percentage_error = zeros(row1,col1);
for i = 1:1:row1,
    for j = 1:1:col1,
        Percentage_error(i,j) = ((at2(i,j) - T_tr(i,j))/at2(i,j))*100;
    end
end
sum_Percentage_error = sum(Percentage_error(1,:));
MAPE_top_training = (sum_Percentage_error/col1);
sum_Percentage_error = sum(Percentage_error(2,:));
MAPE_bottom_training = (sum_Percentage_error/col1);

%pearson correlation coeff  calculation Training
[row1,col1] = size(T_tr);
Predicted = zeros(row1,col1);
Actual = zeros(row1,col1);
val1 = zeros(row1,col1);
val2 = zeros(row1,col1);
val3 = zeros(row1,col1);
for i = 1:1:row1,
    Ep=mean(at2(i,:));
Ea=mean(T_tr(i,:));
    for j = 1:1:col1,
   Predicted(i,j) = at2(i,j);
   Actual(i,j)=T_tr(i,j);
   val1(i,j)=(Predicted(i,j)-Ep)*(Actual(i,j)-Ea);
   val2(i,j)=(Predicted(i,j)-Ep)^2;
   val3(i,j)=(Actual(i,j)-Ea)^2;
    end
end

Cp_top_training=(sum(val1(1,:)))/(sqrt(sum(val2(1,:))*sum(val3(1,:))));
Cp_bottom_training=(sum(val1(2,:)))/(sqrt(sum(val2(2,:))*sum(val3(2,:))));

%-----------------validation set performance measurement----------------------------

% rmse calculation validation
[row1,col1] = size(T_v);
error_col = zeros(row1,col1);
for i = 1:1:row1,
for j = 1:1:col1,
    error_col(i,j) = (at1(i,j) - T_v(i,j))^2;

end
end
sum_error = sum(error_col(1,:));
rmse_top_validation = sqrt(sum_error/col1);
sum_error = sum(error_col(2,:));
rmse_bottom_validation = sqrt(sum_error/col1);

% CDC calculation for training
d1=zeros(1,col1-1);
w1=zeros(1,col1-1);
i=2;
for n=1:1:col1-1
    a=T_v(1,i) - T_v(1,i-1);
    b=at1(1,i) - at1(1,i-1);
    c=a*b;
    d1(:,i-1)=c;
    g=T_v(2,i) - T_v(2,i-1);
    h=at1(2,i) - at1(2,i-1);
    x=g*h;
    w1(:,i-1)=x;
    i=i+1;
end

D_top=zeros(1,col1-1);
D_botm=zeros(1,col1-1);
m=1;
for q=1:1:col1-1
    if d1(:,m)>0
         D_top(:,m)=1;
    else
        D_top(:,m)=0;
    end

     if w1(:,m)>0
```

```
         D_botm(:,m)=1;
    else
        D_botm(:,m)=0;
    end
    m=m+1;
end

[row2,col2] = size(D_top);
[row3,col3] = size(D_botm);
CDC_top_validation = (sum(D_top))*(100/(col2));
CDC_bottom_validation = (sum(D_botm))*(100/(col3));

%AIC and BIC calculation for validation
[Coeff_top,Errors_top,LLF_top,Innovations_top,Sigmas_top,Summary_top]=garchfit(at
1(1,:));
[Coeff_bottom,Errors_bottom,LLF_bottom,Innovations_bottom,Sigmas_bottom,Summ
ary_bottom]=garchfit(at1(2,:));
[AICt_validation BICt_validation]=aicbic(LLF_top,4,col1);
[AICT_validation BICT_validation]=aicbic(LLF_bottom,4,col1);

% MAPE calculation validation
[row1,col1] = size(T_v);
Percentage_error = zeros(row1,col1);
for i = 1:1:row1,
    for j = 1:1:col1,
        Percentage_error(i,j) = ((at1(i,j) - T_v(i,j))/at1(i,j))*100;
    end
end
sum_Percentage_error = sum(Percentage_error(1,:));
MAPE_top_validation = (sum_Percentage_error/col1);
sum_Percentage_error = sum(Percentage_error(2,:));
MAPE_bottom_validation = (sum_Percentage_error/col1);

%pearson correlation coeff  calculation validation
[row1,col1] = size(T_v);
Predicted = zeros(row1,col1);
Actual = zeros(row1,col1);
val1 = zeros(row1,col1);
val2 = zeros(row1,col1);
val3 = zeros(row1,col1);
for i = 1:1:row1,
    Ep=mean(at1(i,:));
    Ea=mean(T_v(i,:));
    for j = 1:1:col1,
   Predicted(i,j) = at1(i,j);
   Actual(i,j)=T_v(i,j);
   val1(i,j)=(Predicted(i,j)-Ep)*(Actual(i,j)-Ea);
   val2(i,j)=(Predicted(i,j)-Ep)^2;
   val3(i,j)=(Actual(i,j)-Ea)^2;
    end
end
```

```
Cp_top_validation=(sum(val1(1,:)))/(sqrt(sum(val2(1,:))*sum(val3(1,:))));
Cp_bottom_validation=(sum(val1(2,:)))/(sqrt(sum(val2(2,:))*sum(val3(2,:))));

%-----------------testing set performance measurement---------------------------

% rmse calculation testing
[row1,col1] = size(T_te);
error_col = zeros(row1,col1);
for i = 1:1:row1,
for j = 1:1:col1,
    error_col(i,j) = (at(i,j) - T_te(i,j))^2;

end
end
sum_error = sum(error_col(1,:));
rmse_top_testing = sqrt(sum_error/col1);
sum_error = sum(error_col(2,:));
rmse_bottom_testing = sqrt(sum_error/col1);

% CDC calculation for training
d1=zeros(1,col1-1);
w1=zeros(1,col1-1);
i=2;
for n=1:1:col1-1
    a=T_te(1,i) - T_te(1,i-1);
    b=at(1,i) - at(1,i-1);
    c=a*b;
    d1(:,i-1)=c;
    g=T_te(2,i) - T_te(2,i-1);
    h=at(2,i) - at(2,i-1);
    x=g*h;
    w1(:,i-1)=x;
    i=i+1;

end

D_top=zeros(1,col1-1);
D_botm=zeros(1,col1-1);
m=1;
for q=1:1:col1-1
    if d1(:,m)>0
         D_top(:,m)=1;
    else
        D_top(:,m)=0;
    end

     if w1(:,m)>0
         D_botm(:,m)=1;
    else
        D_botm(:,m)=0;
    end
    m=m+1;
```

```
 end

[row2,col2] = size(D_top);
[row3,col3] = size(D_botm);
CDC_top_testing = (sum(D_top))*(100/(col2));
CDC_bottom_testing = (sum(D_botm))*(100/(col3));


%AIC and BIC calculation for testing
[Coeff_top,Errors_top,LLF_top,Innovations_top,Sigmas_top,Summary_top]=garchfit(at(1,:));
[Coeff_bottom,Errors_bottom,LLF_bottom,Innovations_bottom,Sigmas_bottom,Summary_bottom]=garchfit(at(2,:));
[AICt_testing BICt_testing]=aicbic(LLF_top,4,col1);
[AICT_testing BICT_testing]=aicbic(LLF_bottom,4,col1);

% MAPE calculation testing
[row1,col1] = size(T_te);
Percentage_error = zeros(row1,col1);
for i = 1:1:row1,
    for j = 1:1:col1,
        Percentage_error(i,j) = ((at(i,j) - T_te(i,j))/at(i,j))*100;
    end
end
sum_Percentage_error = sum(Percentage_error(1,:));
MAPE_top_testing = (sum_Percentage_error/col1);
sum_Percentage_error = sum(Percentage_error(2,:));
MAPE_bottom_testing = (sum_Percentage_error/col1);

%pearson correlation coeff  calculation testing
[row1,col1] = size(T_te);
Predicted = zeros(row1,col1);
Actual = zeros(row1,col1);
val1 = zeros(row1,col1);
val2 = zeros(row1,col1);
val3 = zeros(row1,col1);
for i = 1:1:row1,
    Ep=mean(at(i,:));
Ea=mean(T_te(i,:));
    for j = 1:1:col1,
   Predicted(i,j) = at(i,j);
   Actual(i,j)=T_te(i,j);
   val1(i,j)=(Predicted(i,j)-Ep)*(Actual(i,j)-Ea);
   val2(i,j)=(Predicted(i,j)-Ep)^2;
   val3(i,j)=(Actual(i,j)-Ea)^2;
    end
end

Cp_top_testing=(sum(val1(1,:)))/(sqrt(sum(val2(1,:))*sum(val3(1,:))));
Cp_bottom_testing=(sum(val1(2,:)))/(sqrt(sum(val2(2,:))*sum(val3(2,:))));

%------------------display output----------------------------
```

```
clc;
output.rmse_top_training=rmse_top_training;
output.rmse_bottom_training=rmse_bottom_training;
output.CDC_top_training=CDC_top_training;
output.CDC_bottom_training=CDC_bottom_training;
output.R_top_training=R5;
output.R_bottom_training=R6;
output.AIC_top_training=AICt_trainng;
output.AIC_bottom_training=AICT_trainng;
output.BIC_top_training=BICt_trainng;
output.BIC_bottom_training=BICT_trainng;
output.MAPE_top_training=MAPE_top_training;
output.MAPE_bottom_training=MAPE_bottom_training;
output.Cp_top_training=Cp_top_training;
output.Cp_bottom_training=Cp_bottom_training;
output.rmse_top_validation=rmse_top_validation;
output.rmse_bottom_validation=rmse_bottom_validation;
output.CDC_top_validation=CDC_top_validation;
output.CDC_bottom_validation=CDC_bottom_validation;
output.R_top_validation=R3;
output.R_bottom_validation=R4;
output.AIC_top_validation=AICt_validation;
output.AIC_bottom_validation=AICT_validation;
output.BIC_top_validation=BICt_validation;
output.BIC_bottom_validation=BICT_validation;
output.MAPE_top_validation=MAPE_top_validation;
output.MAPE_bottom_validation=MAPE_bottom_validation;
output.Cp_top_validation=Cp_top_validation;
output.Cp_bottom_validation=Cp_bottom_validation;
output.rmse_top_testing=rmse_top_testing;
output.rmse_bottom_testing=rmse_bottom_testing;
output.CDC_top_testing=CDC_top_testing;
output.CDC_bottom_testing=CDC_bottom_testing;
output.R_top_testing=R1;
output.R_bottom_testing=R2;
output.AIC_top_testing=AICt_testing;
output.AIC_bottom_testing=AICT_testing;
output.BIC_top_testing=BICt_testing;
output.BIC_bottom_testing=BICT_testing;
output.MAPE_top_testing=MAPE_top_testing;
output.MAPE_bottom_testing=MAPE_bottom_testing;
output.Cp_top_testing=Cp_top_testing;
output.Cp_bottom_testing=Cp_bottom_testing;
disp(output),

o(1,1)=output.rmse_bottom_training;
o(2,1)=output.rmse_top_training;
o(3,1)=output.CDC_bottom_training;
o(4,1)=output.CDC_top_training;
o(5,1)=output.R_bottom_training;
o(6,1)=output.R_top_training;
o(7,1)=output.AIC_bottom_training;
```

```
o(8,1)=output.AIC_top_training;
o(9,1)=output.BIC_bottom_training;
o(10,1)=output.BIC_top_training;
o(11,1)=output.MAPE_bottom_training;
o(12,1)=output.MAPE_top_training;
o(13,1)=output.Cp_bottom_training;
o(14,1)=output.Cp_top_training;
o(15,1)=output.rmse_bottom_validation;
o(16,1)=output.rmse_top_validation;
o(17,1)=output.CDC_bottom_validation;
o(18,1)=output.CDC_top_validation;
o(19,1)=output.R_bottom_validation;
o(20,1)=output.R_top_validation;
o(21,1)=output.AIC_bottom_validation;
o(22,1)=output.AIC_top_validation;
o(23,1)=output.BIC_bottom_validation;
o(24,1)=output.BIC_top_validation;
o(25,1)=output.MAPE_bottom_validation;
o(26,1)=output.MAPE_top_validation;
o(27,1)=output.Cp_bottom_validation;
o(28,1)=output.Cp_top_validation;
o(29,1)=output.rmse_bottom_testing;
o(30,1)=output.rmse_top_testing;
o(31,1)=output.CDC_bottom_testing;
o(32,1)=output.CDC_top_testing;
o(33,1)=output.R_bottom_testing;
o(34,1)=output.R_top_testing;
o(35,1)=output.AIC_bottom_testing;
o(36,1)=output.AIC_top_testing;
o(37,1)=output.BIC_bottom_testing;
o(38,1)=output.BIC_top_testing;
o(39,1)=output.MAPE_bottom_testing;
o(40,1)=output.MAPE_top_testing;
o(41,1)=output.Cp_bottom_testing;
o(42,1)=output.Cp_top_testing;

a=net.IW{1,1};
a1=net.IW{1,2};
b=net.LW{2,1};
c=net.LW{3,2};
d=net.b{1};
e=net.b{2};
f=net.b{3};

sprintf('\n%15.7f',o) ;

save newnarxsp_with_partition_test
```